

Easy Graphing in QuickCalc BASIC

About this Documentation.

QuickCalc BASIC is designed to let you **produce graphs** of data which you calculate and analyze in your BASIC program, and to do so **easily** and **quickly**.

These goals are in conflict with “powerful” and “flexible”. If you are just starting your first graph with QuickCalc, presenting you with all the options and capabilities of QuickCalc graphics could be overwhelming.

The documentation is therefore divided into **Beginning**, **Intermediate**, and **Advanced** Graphics.

Beginning Graphics (Read this one first)

Introduces you to the concepts and basic functions of QuickCalc graphics.
Allows you to produce a graph of your data with just a few statements.
Presents a **subset** of the features available, to keep it simple.

Intermediate Graphics

Shows how to tailor your graph to make it look exactly like you want it.
Describes the **SHAPE** statement to draw figures on your graph.
Describes the various ways to put **TEXT** on your graph.
Gives complete descriptions of the **GRAPH**, **SHAPE**, **PLOT** and **TEXT** statements.
Describes how to create and use Logarithmic and **DATETIME** scales.
Tells you how to specify colors for your graphic objects.

Advanced Graphics

Includes more advanced functions you do not normally need:

- Polylines
- Bezier Curves
- Smoothing and Averaging Plots, Polylines, and Polygons
- Complex Connected Shapes
- Outline Text Characters
- Selecting Shapes with the Mouse
- Deleting Shapes
- Changing Previously-drawn Shapes

So it is best to read these documents in order. First find out what you need to know to do the job at hand, then later read the more advanced (and more interesting) things that QuickCalc Basic Graphics can do.

Introduction

The graphing functions in QuickCalc BASIC are intended to produce graphs of data.

Older versions of BASIC used graphic functions to create computer “games” like “Pong” and “Space Invaders”. That is not what QuickCalc graphics is about.

The goal here is to produce graphs of data which you calculate and analyze in your BASIC program, and to do so **easily** and **quickly**.

You may require a graph of your data to include in a lab report, term paper, or presentation. The graph can be printed in a professional, publishable format. It can also be written to a **JPEG file** or Windows enhanced **metafile** for inclusion into another document.

Features:

QuickCalc graphs feature **automatic scaling** (which you can override). This automatically scales the graph so that all your data fits on the graph. Automatic scaling will dynamically re-scale the graph as new points are added.

The **default size** of a graph is 7.5 inches wide and 10 inches high (“portrait”), which fits well on a standard 8.5 x 11” page and works on most printers. You can resize the graph to whatever size you wish, provided it fits on your printer.

QuickCalc plots the graph **on the screen**, in a window that you can move and re-size, maximize or minimize. The on-screen graph shows all the data plotted in real time. The graph remains on the screen after the BASIC program terminates, so you can adjust its parameters and/or print it.

You can plot data on **logarithmic scales**, if desired (See “Intermediate Graphics”)

You can also **put text on your graph** and **label the axes** (See below).

There is now a way to **draw shapes** (filled or open) on your graph. Using this feature, you can draw pie charts, bar graphs, polygons, etc. Shapes may be **rotated**. (see “Intermediate Graphics”).

Note: Graphing is not supported in Console (DOS) mode.

Quick and Easy Graphs.

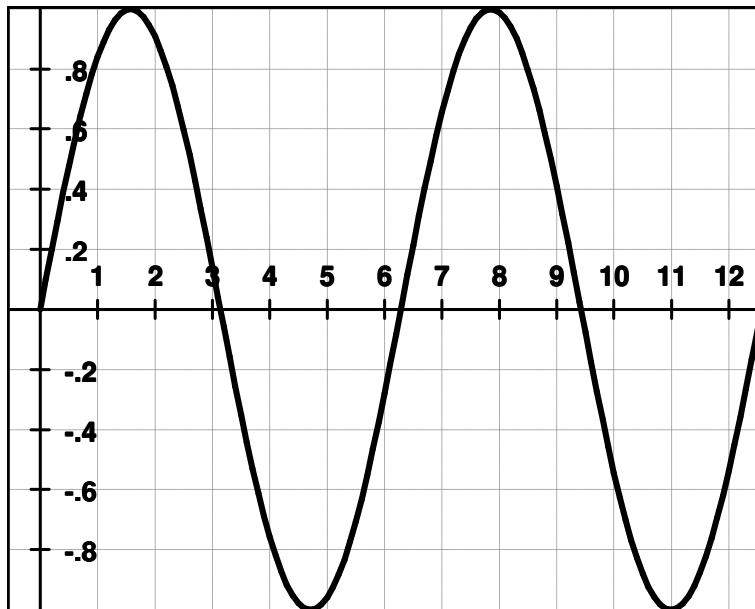
You don't have to learn a lot of fancy commands to produce a graph. Most options default to a reasonable value so that you can get your graph with just a few statements. You can see your graph immediately and then fine-tune it later, if desired.

A Simple Graph program

The following program will plot the sine of an angle over the range 0-720 degrees (0 to 4π radians).

```
GRAPH WIDTH=5, HEIGHT=4
FOR angle = 0 TO 4 * PI + .1 STEP .1
  PLOT X=angle, Y=SIN (angle)
NEXT
```

This will plot the function on a graph 5" wide by 4" high, automatically scaled, using a 3-point black line, as shown below:



The graph appears in its own window on the screen.

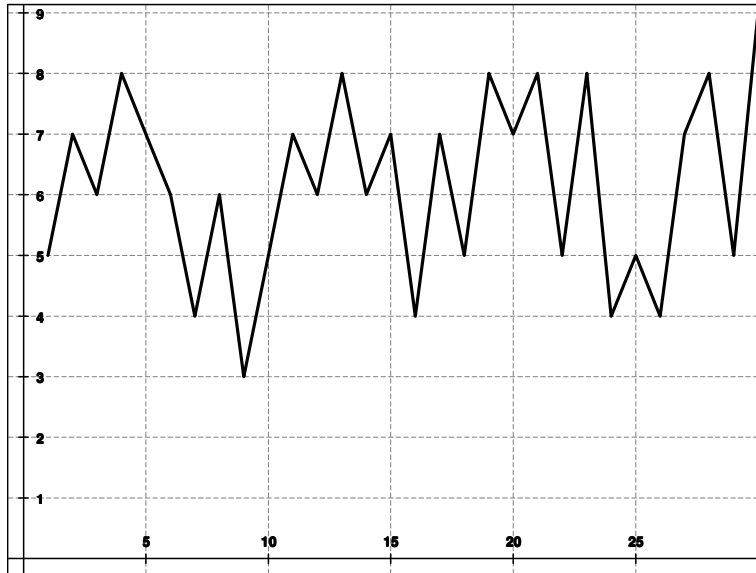
Plotting Your Own Data.

Suppose you are plotting a stock price over a period of a month. You can provide the stock prices in some DATA statements and then graph them.

```
DATA 5, 7, 6, 8, 7, 6, 4, 6, 3, 5, 7, 6, 8, 6, 7
DATA 4, 7, 5, 8, 7, 8, 5, 8, 4, 5, 4, 7, 8, 5, 9
```

```
GRAPH LANDSCAPE
FOR I = 1 TO 30
READ price
PLOT X = i, Y = price
NEXT
```

This produces the following graph:



Note: The numbers and lines appear smaller here because the printed size (“landscape” or 10 in by 7.5 in) is larger than the previous graph, so the graph was reduced for inclusion into this document.

The data plotted on the graph may come from any source. You can read it in from a file, type it in, or generate it from within your program.

Printing Your Graph.

The easiest way to print your graph is to select **FILE / PRINT** from the Graph window.

Note: The graph window will disappear from the screen until printing is complete or cancelled.

Note: You may experience some delay if you are on a network and your default printer is not currently on-line. Turning off your network adaptor when you are not connected will avoid this problem.

The graph will be printed using the default printer orientation (portrait or landscape) unless you override it in the printer properties dialog. If your graph will not fit using the selected orientation, the program will attempt to print it in

the other orientation. (A “landscape” (10 x 7.5) graph will not fit on the printer in portrait mode, so the program will switch the printer to landscape mode where the graph *will* fit, then print it.) If the graph will not fit in either orientation, an error message will appear, and you can re-size your graph or select a different printer (or page orientation). There is no support for multiple page graphs, although some printer drivers offer a scale-to-page feature.

You can change the attributes of your graph and re-print it, if desired, as long as the graph is still active.

Creating a File Image (JPEG or Metafile) of Your Graph.

(See “Intermediate Graphics”)

Page Size.

The page size is given in inches of height and width. If you are going to print the graph, the page size must not be greater than the printable area on the page.

The graph shown on the screen is not necessarily the size it will be when printed, however the height and width are kept proportional as you re-size the screen graph window.

(See “Controlling the Size and Shape of Your Graph” in “Intermediate Graphics”)

Automatic Scaling.

Automatic scaling insures that all your data will fit on the graph. The range, on each axis, is from the lowest to the highest values plotted. The x- and y-axes are included in the plot, unless you override that (see “Intermediate Graphics”). The scaling is not necessarily the same for x and y coordinates, and will vary if the graph size changes.

Note: If you wish to make the horizontal and vertical scales the same, you can include the parameter **EQUALSCALES** in the **GRAPH** statement. This will cause automatic scaling to take the larger of the two automatic scales and use it for both axes.

You can specify the scale factors and offsets for your graph manually, if desired (see “Intermediate Graphics”).

The Graph Window.

Graphs appear in a separate window. The window shows a representation of what your graph will look like when it is printed. You can re-size the window, however the height and width are kept proportional. Maximizing the window will

not necessarily fill the screen; that depends on the height/width proportions. Maximized windows may still be moved around on the screen.

The graph in the window reflects all the commands that have been processed up to that time; if you step through the program you can watch your graph being plotted.

The **menu bar** on the graph window allows you to quickly do some of the graphics commands. If the option is already checked, selecting it again will turn it off.

FILE / PRINT	Prints the graph plotted so far.
FILE / METAFILE	Generates a metafile from your graph.
FILE / JPEG	Generates a JPEG file from your graph.
FILE / END	Ends the graph and closes the graph window.
FILE / PAUSE	Interrupts the BASIC program.
OPTIONS / GRID	Shows the grid lines.
OPTIONS / BORDER	Puts a border around your graph.
OPTIONS / AXES and NUMBERS	Displays x-y axes and numbers.
OPTIONS / MOREGRIDLINES	Increases the density of grid lines.
OPTIONS / FEWERGRIDLINES	Decreases the density of grid lines.
OPTIONS / SHOW COORDINATES	Displays the x-y coordinates of the spot on the graph to which you are pointing.
OPTIONS / CROSSHAIRS	Puts red crosshairs at the center of the physical page. Useful for centering items. Crosshairs do not print or appear in a metafile or JPEG file.
ZOOM / IN or OUT	Changes the scale & offset by zooming in and out.
ZOOM / AUTO SCALE	Resets SCALE and OFFSET parameters to automatic scaling.

You can also re-position your graph (change its offsets) by clicking on the graph itself and **dragging** it. This, of course, changes the offsets and turns off auto-scaling.

Note: Dragging the graph will not happen if the **SHIFT** or **CONTROL** key is held down. That changes the mouse function and is used for other things (See “*Advanced Graphics*”).

Zooming and dragging the graph is limited to the range of valid DOUBLE floating-point numbers (or their logarithms).

Note: Zooming the graph changes the **scale** of the graph. It does *not* change the size of the **printed** graph. If you zoom in for a closer look, the new view is what will get printed.

One way to use the “zoom” function is to let auto-scaling scale the graph, and then zoom in or out to fine-tune the view.

Note: There are **shortcut keys** for the “zoom” functions. **Up-Arrow** zooms in 50%, **Down-Arrow** zooms out 50%. If you want to change the horizontal scale while leaving the vertical scale alone, you can use **Right-Arrow** to stretch the graph horizontally and **Left-Arrow** to compress it horizontally. These keys only work when the graph window has the focus.

Note: You can also use the **scroll wheel** on the mouse to zoom in and out.

Note: When the graph window has the focus, you may return to the QuickCalc BASIC window by pressing the **TAB** key.

Controlling the Size and Shape of Your Graph.

(See “Intermediate Graphics”)

The GRAPH statement.

The **GRAPH** statement controls the size, shape, and scaling of the graph. It also starts the creation of the graph and can terminate the graph and print it.

GRAPH [*parameter* [, *parameter* ...]]

If a parameter requires a value, *e.g.*, **HEIGHT**=*num-expression*, any valid numeric expression may be supplied, as long as it evaluates to a number with the correct range.

The **GRAPH** statement with no parameters **starts a graph** with all the defaults, if a graph is not already active. If a graph is currently active, this statement will generate an error.

The **GRAPH** statement with parameters may appear at any time, and will affect and over-ride the settings of the current graph without changing the data you have plotted so far.

Ending a Graph.

A graph remains active until

- you issue the statement **GRAPH END**,
- another BASIC program begins executing,
- you close the graph window, or
- QuickCalc is terminated.

Note: If you close the graph window before the graph is complete, you will get errors the next time you try to plot a point.

GRAPH Statement parameters.

Note: Here we are showing only a **small subset** of the **GRAPH** statement parameters, the ones we have referenced so far. For the complete list of parameters and their descriptions, see “Intermediate Graphics”.

GRAPH statement parameters override the default parameters and settings from previous **GRAPH** statements.

The parameters you need for simple graphs are:

PORTRAIT	Sets WIDTH =7.5 in, HEIGHT = 10 in.
LANDSCAPE	Sets WIDTH =10 in, HEIGHT = 7.5 in.
WIDTH=num-expression	Sets printed graph width, in inches.
HEIGHT=num-expression	Sets printed graph height, in inches.
EQUALSCALES	Keeps H and V scales the same.

Plotting Data.

Data plotted on the graph is controlled by the **PLOT** statement (see below).

A series of (x,y) values (“points”) are by default sorted by their x -value, then plotted with line segments connecting them (see “Sorted Points” in “Intermediate Graphics”).

You may have more than one (up to 50) plots on the same graph, each with a different color or line weight, if desired.

Data plotted may be `DOUBLE` or `LONGMATH`, however, the values must be convertible to `DOUBLE`. Values plotted may be specified as numeric expressions.

To begin a new plot, simply specify the **PLOT** statement with no parameters. If desired, the **PLOT** statement can specify other parameters such as line weight and color (see “Intermediate Graphics”).

A **GRAPH** must be active before you issue a **PLOT** statement. Typically, the sequence is as follows:

```
GRAPH      [optional parameters]
...
PLOT      [optional parameters]
[controlling loop]
...
PLOT X=numeric-expression, Y=numeric-expression
...
[end controlling loop]
```

(see the examples above.)

The points do not need to be sorted, as the program will sort them by their x -values before plotting them (see “Sorting Data”, in “Intermediate Graphics”).

The PLOT Statement.

A **PLOT** statement begins a plot (connected points) or adds a point to the series. A **PLOT** statement which does not specify a point ($X=...,Y=.$) will begin a new plot.

The “**simplified**” **PLOT** parameters are:

```
X=num-expression,
Y=num-expression.
```

The x - and y -values must be specified together on the same **PLOT** statement. Nothing else should be on that statement, unless you wish to start a new plot.

Note: the x - and y -values must be **DOUBLE** or convert correctly to **DOUBLE**. **LONGMATH** numbers will be truncated and converted, if possible. Exponent underflow will result in zero.

(See “Intermediate Graphics” for the full list of **PLOT** parameters.)

Multiple Plots on the Same Graph.

(See “Intermediate Graphics”)

Logarithmic Scales.

(See “Intermediate Graphics”)

Date-Time Scales

Note: See the Document “*Working With Dates and Times*” for a description of the **DATETIME** variables and how to create and use them. If you are going to use **DATETIME** scales in your graph, you should first read that document.

DATETIME scales are described in “Intermediate Graphics”.

Putting Text On Your Graph.

Although this is not, strictly speaking, *simple* graphing, there is often a need to put text on a graph to

- label the x and y axes,
- identify objects and plots,
- title the graph,
- put date, author, copyright information on the graph, etc.

QuickCalc BASIC has implemented an easy-to-use way to accomplish this.

Text is placed on the graph using the **TEXT** statement. The format of the **TEXT** statement depends on how you want the text displayed.

Note: Only a few simple **TEXT** functions are described here. For the complete description of the **TEXT** statement, refer to “Intermediate Graphics”.

A graph must be active (**GRAPH** statement) before the **TEXT** statement is executed.

(Nearly) all forms of the **TEXT** statement require the parameter **STRING**=[*string-variable* | "*string-constant*"]. The string variable or constant contains the text you want to display. For example:

```
TEXT STRING = "My Graph Title", ...(other parameters)
```

Labeling the Axes.

This is easy. Just use the statement:

```
TEXT X AXIS, STRING = "title-to-display-on-x-axis"  
and/or TEXT Y AXIS, STRING = "title-to-display-on-y-axis"
```

See “Intermediate Graphics” for more information.

Two Types of Text.

Text on your graph may be tied to a point in “data space”, in which case it will be positioned on the graph at that x - y position, which is dependent on the scale and offsets of the graph. If you scroll the graph, the text will move with it.

Alternately, text may be fixed to a location on the page. This is useful for titles, legends, etc.

How to use each of these types of text is described in “Intermediate Graphics”. For now, I will just give a couple of examples.

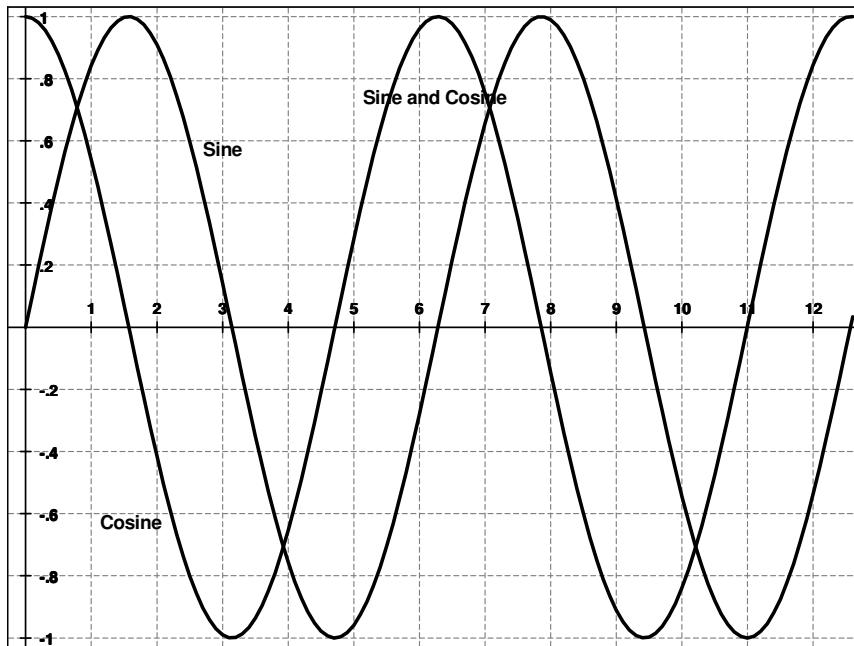
Example.

The following program illustrates how text may be applied to a graph. In this program, we will plot the sine and cosine with text added:

```
GRAPH LANDSCAPE  
// Plot the sine  
PLOT  
FOR angle = 0 TO 4 * PI + .1 STEP .1  
    PLOT X=angle, Y=SIN(angle)  
NEXT  
// Plot the cosine  
PLOT // starts a new plot  
FOR angle = 0 TO 4 * PI + .1 STEP .1  
    PLOT X=angle, Y=COS(angle)  
NEXT  
TEXT STRING="Sine", X=3, Y=.55
```

```
TEXT STRING="Cosine", X=1.6, Y=-.65
// The above text is anchored to data points on the graph (X and Y)
TEXT STRING="Sine and Cosine", H=50, V=85
// Title, anchored to the top center of the page (H and V).
```

This produces the following graph (shown reduced):



See “Intermediate Graphics” for a more interesting example of this same graph with colors, labeling the axes, and text effects.

Drawing Shapes

QuickCalc now supports **drawing shapes** (filled or open) on your graph. This is done using the **SHAPE** statement.

You can draw the following shapes:

Rectangle, Square, Ellipse, Circle, Arc, Chord of an arc, Pie segment,
Line, Polygon, PolyLine, Bezier curve, Complex Connected Shape,
Text Outlines.

(see “Intermediate Graphics” and “Advanced Graphics” for descriptions of these functions.)

Conclusion

Now try it out on some programs of your own. Once you have a graph of your data, you can go on to “Intermediate Graphics”, where most of the good stuff is described, and tailor the graph the way you like it.

Also, the **Sample Programs** page on the website has many examples of things you can do with graphic functions.