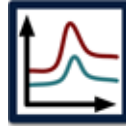


Plot



version 0.97

Michael Wesemann (c) 2005-2006
plot.micw.eu

Table of Contents

Plot Introduction	1
Data Inspector	3
Inspector	5
General Inspector	7
Data Style Inspector	9
Axis Inspector	12
Axis Format Inspector	14
Axis Labels Inspector	16
Error Bars Inspector	17
Legend Inspector	18
Calculations Inspector	19
Function Generator	19
Buffer Calculations	19
Normalize Inspector	21
Normalize Functions	21
Calculations	22
Linear Background Subtraction	22
Fit Inspector	23
Smooth & Data reduction	23
Spline Interpolation	23
Differentiate & Integrate	24
FFT (Fast Fourier Transform)	24
Text Inspector	25
Graphic Inspector	26
Advanced Inspector	28
Data Update Settings	28
Automatic Macro Execution	28
Mouse Modes	29
Measure	29
Zoom	29
Normalize X	29
Normalize Y	29
Downscale	29
Upscale	30

Table of Contents

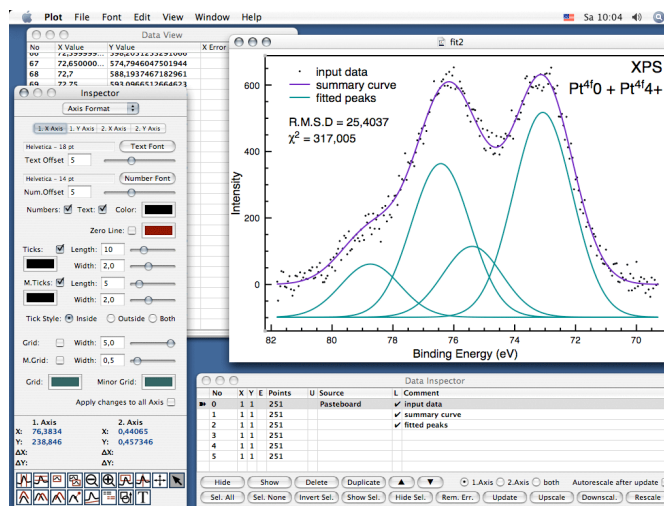
<u>Mouse Modes</u>	
<u>Range</u>	30
<u>Select Tool</u>	30
<u>Move Y</u>	30
<u>Move X</u>	31
<u>Move XY</u>	31
<u>Move Points</u>	31
<u>Linear Background</u>	31
<u>Legend Moving</u>	31
<u>Graphic</u>	31
<u>Text</u>	32
<u>Subview</u>	32
<u>Change Subview</u>	32
<u>Curve Fit Inspector</u>	33
<u>Fit result</u>	34
<u>Function definition</u>	34
<u>Special Functions</u>	35
<u>DataView</u>	37
<u>Calculator</u>	38
<u>Expressions</u>	39
<u>Time Format</u>	41
<u>Time format tokens</u>	41
<u>Preferences</u>	42
<u>General</u>	42
<u>Colors</u>	43
<u>Import Filter</u>	43
<u>Menu Item</u>	44
<u>Import Filter</u>	46
<u>Import</u>	46
<u>Custom Import Filter</u>	47
<u>Binary Import Filter</u>	48
<u>MySQL Import</u>	49
<u>Macro Inspector</u>	50
<u>Macro Language</u>	51
<u>Introduction</u>	51
<u>Command Reference</u>	51
<u>Variables</u>	51
<u>Arguments</u>	52

Table of Contents

Macro Language: Range Commands.....	55
Macro Language: Buffer Commands.....	56
Macro Language: Control Commands.....	58
Macro Language: Interactive Commands.....	59
Macro Language: Calculation Commands.....	60
Macro Language: Style Commands.....	64
Macro Language: Data Style Commands.....	65
Macro Language: Axis Style Commands.....	68
Macro Language: Legend Commands.....	73
Macro Language: Text & Graphic Commands.....	74
Macro Language: Document Commands.....	80
Macro Language: Miscellaneous Commands.....	82
Macro Quick Reference.....	83
About Plot.....	86
History.....	86
Sources.....	86
Spline Interpolation.....	86
Smooth.....	86
FFT (Fast Fourier Transform).....	86
FFT smooth & Differentiation.....	86
Curve Fit.....	86
About the author.....	87
Warranty.....	88

Plot Introduction

This documentation is valid for *Plot* version 0.97



Screenshot

Plot is a scientific 2D plotting program for Mac OS X. *Plot* is a completely rewritten descendant of *SciPlot* which was a plotting program for NeXTStep.

Plot is designed for everyday plotting, it is easy to use, it creates high quality plots, and it allows easy and powerful manipulations and calculations of data.

Basically *Plot* allows to work with multiple documents where every document consists of one or more XY data buffer. Unlike some other programs each XY data buffer is independent (without are shared X axis or so). In addition it is also possible to save individual X and Y error values for each data point.

There is also the possibility to add subviews (plots inside plots) to your document. A subview is completely independent with its own data buffers, graphic attributes etc.

Data can be imported from ASCII or binary files, directly from a MySQL database, with a custom import filter or from old *SciPlot* files. It is also possible to insert data buffers by using copy and paste from other programs (e.g. Excel)

Plot supports many plotting styles like lines, symbols, grids, bars, filled areas, sticks, dots, histograms, errorbars, automatic legend, text and graphic objects, etc.

Plot has many data manipulation methods integrated: curve fitting, fft smooth, least square smooth, spline interpolation, differentiation, integration, FFT, least square regression, normalizing, moving, data calculations, function generator, etc.

The data buffers are managed by the Data Inspector.

To make plotting easy many attributes (e.g. axis ranges, axis labels) will be generated automatically but it is possible to change most graphic attribute by hand.

The main tool in *Plot* is the Inspector which controls all attributes of your document, show measured values and allow the mouse mode selection. The Inspector also controls the graphic attributes of the data buffers.

There is also a tool called Data View which allows direct manipulation of data points in a spreadsheet like table.

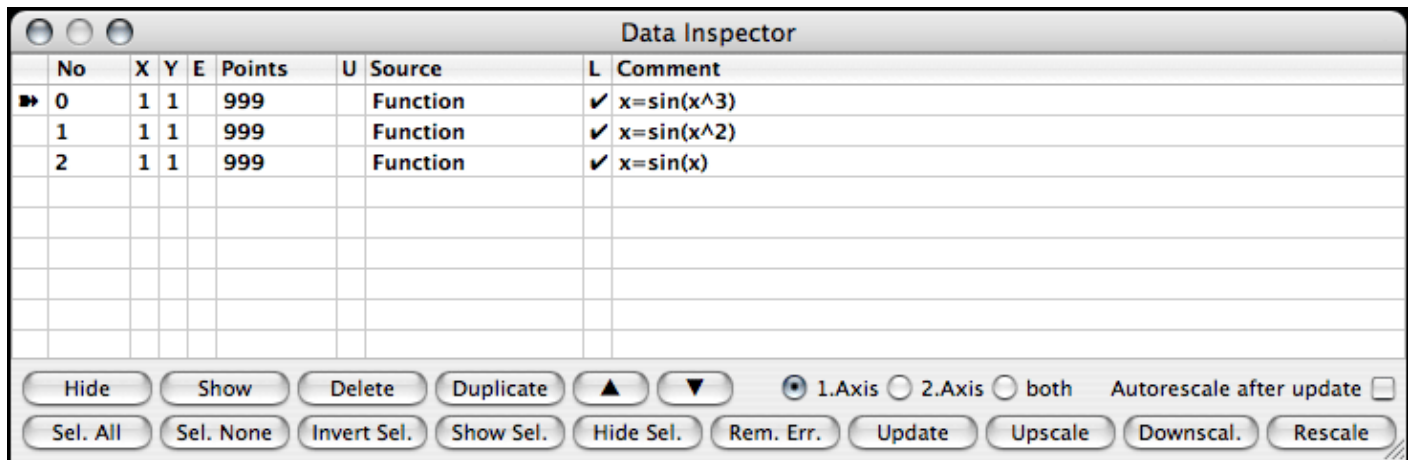
Plot has also a macro language with ~ 180 builtin commands. The macro language allow to automate complex tasks.

Last not least there is a Calculator for instant calculations.

Of course there are also several menu items which allow many operations.

Data Inspector

The *Data inspector* mainly controls the data buffer in your document. Each line represents a set of XY data. It is possible to select some buffers and transfer them via copy and paste to another *Plot* document.



Data Inspector

The columns in the *Data Inspector* show all important data about the buffer and allow some changes:

1. column: In this column appears an arrow which indicates the *working buffer*, e.g. if you select the *Data Style* Tab in the *Inspector* it show the attributes of the buffer with the arrow in front. Changing the working buffer is done by double clicking in this column.

No column: The number of the buffer. The buffer number can be useful for macro programming.

X column: Shows if the buffer belongs to the first (left) X axis or to the second (right) X axis of the plot. Can be changed by double clicking the column.

Y column: Shows if the buffer belongs to the first (bottom) Y axis or to the second (top) Y axis of the plot. Can be changed by double clicking the column.

E column: This column indicates that the buffer contains error values.

Points column: Number of data points in the buffer.

U column: Indicates if the buffer should automatically updated from dynamic data sources like MySQL. You can change the state by double clicking the column.

Source column: The source of the data. This column is editable.

L column: If checked the buffers comment will be shown in the automatically generated legend. You can change the state by double clicking the column.

Comment column: As the name says. This column is editable.

At the bottom of this inspector appears a gallery of buttons:

Hide and **Show** hides or shows all selected buffers. Hidden buffers will not be displayed but the data are still available for later use. The hide/show state of a buffer can also be changed by double clicking the *Points* or *No* column.

Delete: delete all selected buffers.

Duplicate: duplicates all selected buffers.

Up, Down: with these buttons you can move the current working buffer up and down. The order in the *Data inspector* is also the order in which the buffer will be plotted and changing the order may also change the appearance of your document.

The **1.Axis, 2.Axis, both:** radio button defines if operations like rescale take effect on both axis or only on one of them.

If **Autorescale after update** is selected the document will be rescaled after an data update from dynamic sources like MySQL.

Sel. All: Select all buffers.

Sel. None: Unselect all buffers.

Invert Sel.: Inverts the buffer selection

Show Sel.: Shows only selected buffers

Hide Sel.: Hide only selected buffers.

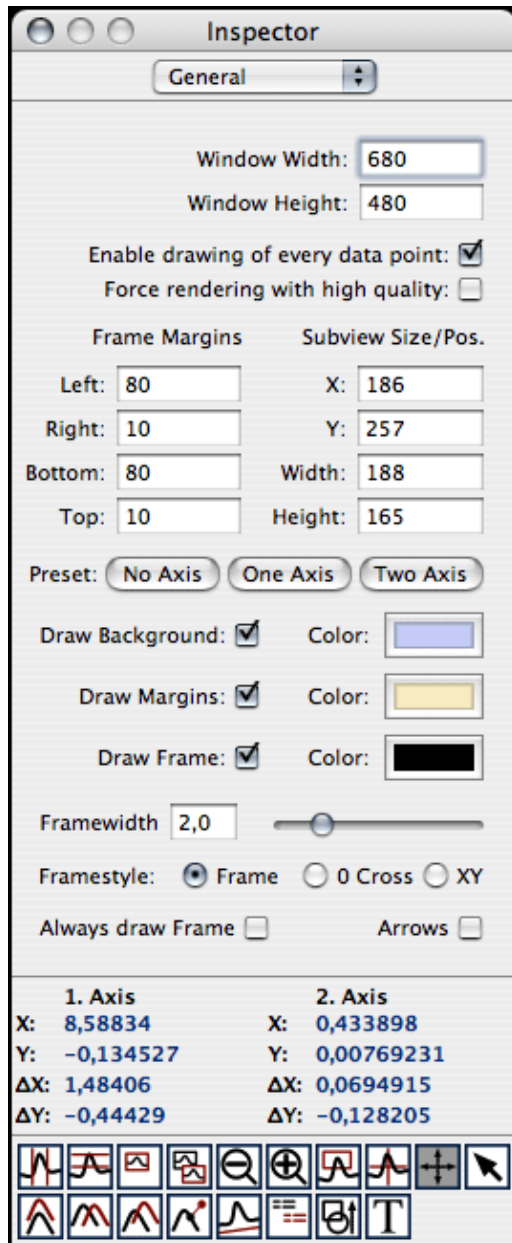
Rem. Err.: This function removes error values from all selected buffer. For large buffers this can reduce the file size.

Update: Buffers with dynamic data sources like MySQL will be updated.

Upscale and **Downscal.:** Increase or lowers the range of the plot.

Rescale: Rescales the plot so that every datapoint is just visible.

Inspector



Inspector

with the *Inspector* you can control every attribute of your document. The *Inspector* is divided into three parts. The upper area is the main part with different tabs. The middle part show values e.g measured with the crosshair mouse mode. In the lower part of the *Inspector* you can select the mouse mode which controls the behavior of the mouse in your document.

The following *Inspectors* are available:

General Inspector

Data Style Inspector

Axis Inspector

Axis Format Inspector

Error Bars Inspector

Legend Inspector

Calculations Inspector

Normalize Inspector
Fit Inspector
Text Inspector
Graphic Inspector
Advanced Inspector

General Inspector

Window Width: 680
Window Height: 480

Enable drawing of every data point: ☒
Force rendering with high quality: ☐

Frame Margins		Subview Size/Pos.	
Left:	10	X:	70
Right:	10	Y:	210
Bottom:	28	Width:	300
Top:	10	Height:	245

Preset: ☐ No Axis ☐ One Axis ☐ Two Axis

Draw Background: ☒ Color:

Draw Margins: ☒ Color:

Draw Frame: ☒ Color:

Framewidth: 2,0

Framestyle: ☒ Frame ☐ O Cross ☐ XY

Always draw Frame ☐ Arrows ☐

General Inspector

The *General Inspector* controls some general attributes of your document.

Window Width, Window Height: The size of the window in pixel. Useful to generate two documents with the same size.

Enable drawing of every data point: If checked *Plot* draws always every data point. Typically *Plot* decides how many data points will be plotted and the result is reasonable. In some cases enabling this option can enhance your plot.

Force rendering with high quality: *Plot* automatically decides how to render your plot. Checking this option forces to render always in highest quality (which can dramatically slow down *Plot* on huge data sets).

Frame Margins: The margins between window and plot frame. You can use the mouse to control the framesize if you choose the select tool (see [mouse modes](#)).

Subview Size/Pos.: The size and location of the currently selected subview.

Preset: Allows easy selection of margin preset for plot with one or two axis.

Draw Background: Defines the background color inside the plot frame and if the background should be drawn or not.

Draw Margins: Defines the background color on the margins and if the margin background should be drawn or not.

Draw Frame Defines the color of the plot frame and if the frame should be drawn or not.

Framewidth: Defines the width of the frame.

Framestyle: Allows to select different framestyles. If *Frame* is selected a full frame will be plotted, *XY* style plots only a line on the X and Y axis, *0 Cross* draws a XY cross at the null position.

Always draw frame: Forces to plot a full frame even if your framestyle is *XY* or *0 cross*.

Arrows: Draw arrows and the line ends for framestyle *0 Cross* and *XY*.

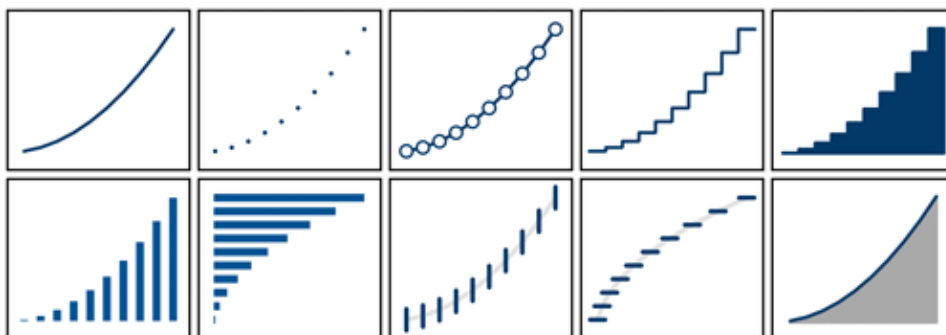
Data Style Inspector

Y Bar: ☐ Y Stick: ☒ Symbol: ☐ Line: ☒
X Bar: ☐ X Stick: ☐ Fill: ☐ Dot: ☐
Histogram: ☐ Filled Histogram: ☐
Data Fill Frame Stick
Linewidth: 1,0
Line Pattern:
Patternwidth: 0
Symbol:
Symbolsize 10,0
Sticksiz 11,6
Sticksiz is absolute: ☒
Dynamic Size (Symbols & Sticks)
abs(sin(rad(x))*1)
Apply Default Style
Apply changes to selected buffers ☐
Apply changes to all buffers ☐
Automatically apply different styles:
Symbols Dash Color

Data Style Inspector

The *Data Style Inspector* controls the graphic attributes of each data buffer.

Y Bar, Y Stick, Fill, Symbol, Line, X Bar, X Stick, Fill, Dot, Histogram, Filled Histogram: Defines in which style a buffer should be drawn. All Styles can be combined.



Lines, Dots, Symbols, Histogram, Filled Histogram, Y Bars, X Bars, Y Sticks, X Sticks, Filled Area

Data: Defines the color for the data points, symbols, lines etc.

Fill: Defines the color for filled areas and filled histograms.

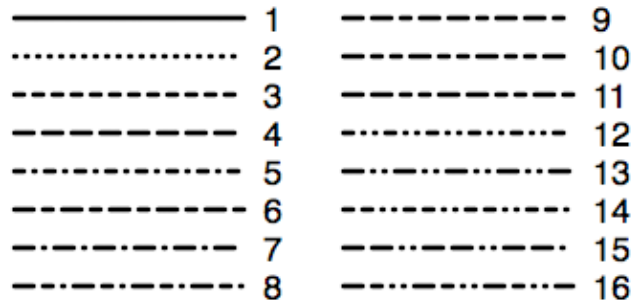
Frame: Defines the color for the frame of bars.

Stick: Defines the color for sticks.

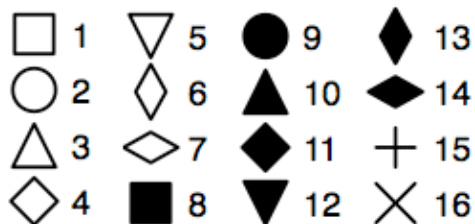
Linewidth: Defines the width of lines.

Line Pattern: Defines the dash pattern for the line style. 16 pattern are available.

Patternwidth: Allows to stretch the dash patterns.



Symbol: Defines the symbol which will be used for symbol style. 16 symbols are available.



Symbolsize: Defines the size of the selected symbol.

Sticksize: Defines the size (= length) for the sticks in the stick style.

Sticksize is absolute: If checked the length of sticks is defined in the coordinate system of the plot. Together with the dynamic size field this allows very flexible stick length.

Dynamic Size: Allows to enter an expression which defines the size of symbols or the length of sticks. The following special variables are available:

x	x value
y	y value
ex	x error value
ey	x error value
bn	buffer number
np	number of buffers
np	number of datapoints
dp	the current datapoint

Apply Default Style: When clicked the default data style will be applied.

Apply changes to selected buffers: If checked changes will provided to all selected buffers and not only to the current working buffer (the buffer with the arrow in the first column of the [*Data Inspector*](#)).

Apply changes to all buffers: If checked changes will be provided to every data buffer and not only to the current working buffer (the buffer with the arrow in the first column of the *Data Inspector*).

Automatically apply different styles: This is a comfortable way to assign different styles automatically to all buffers, available for symbols, dash pattern and colors.

Axis Inspector

1. X Axis 1. Y Axis 2. X Axis 2. Y Axis

Min: 0

Max: 10

Scaling Start: 0,1

Scaling End: 9,9

Tick Distance: 3,33

Minor Steps: 3

Time Format:

Trigger:

Scaling Expr.:

Automatically set decimal places ☒

Dec.Places: 2

Exp.Offset: 5

Axis Format: ☒ Linear ☐ Log ☐ Time

Treat smallest log value as 0 ☐

Plot axis reverse ☐

Axis Text: Flip axis text ☒

2Y AXIS TEXT

Enable: 1.X: ☒ 1.Y: ☒ 2.X: ☐ 2.Y: ☐

Axis Inspector

The *Axis Inspector* allow settings depending to the 4 available axis. The button at the top of the inspector selects on which axis the attributes below apply to.

Min, Max: Controls the range of the axis. The range can also be changed with different mouse modes. Dependent on the number format you have to enter a number or a time value.

Scaling Start, Scaling End: Defines values where the scaling of the axis should start and stop. Dependent on the number format you have to enter a number or a time value.

Tick Distance: Defines the distance between two major ticks. If your axis is a time axis you have to enter *days:hours:minutes:seconds* or *w* (1 week) or *m* (1 month) or *y* (1 year).

Minor Steps: Defines the number of minor ticks between two major ticks.

Scaling Start, Scaling End, Tick Width, Minor Steps are optional values. If you leave the fields empty *Plot* calculates an optimal value automatically.

Time Format: Defines the format used for the displaying time values on time axis (time format tokens).

Trigger: This allow to define a *trigger*, a number where a tick should appear and all other ticks should depend on. for example: if you have an axis from 0 to 10 and you want ticks at 3.33 and 6.66 enter the values shown in the image.

Scaling Expr.: This allows scaling of axis numbers without changing the data itself. For example to display seconds instead of milliseconds enter $v/1000$ in this field. v is the variable which represents the axis number.

Automatically set decimal places: If checked the number of decimal places for axis numbers will be determined automatically.

Dec.Places: The number of decimal places for axis numbers.

Exp.Offset: Allows you to define the limit, above which numbers are plotted in exponential notation. The number you define with the slider is the exponent of the limit.

Axis Format: Here you can select between linear axis, logarithmic axis and time axis.

Treat smallest log value as 0: If checked the smallest value on a logarithmic axis will be replaced with 0 (even if this is not correct).

Plot axis reverse: If checked the axis will be plotted in reverse direction.

Axis Text: The major label for the axis. This field has limited formatting possibilities. If you want a more sophisticated axis label use a text object together with the auto positioning option.

Flip axis text: If checked the axis text orientation will be flipped.

Enable: Allows to enable or disable an axis completely.

Axis Format Inspector

1. X Axis 1. Y Axis 2. X Axis 2. Y Axis

Helvetica - 18 pt Text Font

Text Offset 5

Helvetica - 14 pt Number Font

Num.Offset 5

Numbers: ☒ Text: ☒ Color:

Zero Line: ☐

Ticks: ☒ Length: 10 Width: 2,0

M.Ticks: ☒ Length: 5 Width: 0,9

Tick Style: ☒ Inside ☐ Outside ☐ Both

Grid: ☐ Width: 0,5

M.Grid: ☐ Width: 0,5

Grid: Minor Grid:

Apply changes to all Axis ☒

Axis Format Inspector

The *Axis Format Inspector* controls the graphic attributes of the 4 available axis. The button at the top of the inspector selects on which axis the attributes below apply to.

If *Apply changes to all Buffer* is selected the changes will provided to all 4 axis.

Text Font: The font for the axis label.

Text Offset: The distance between text label and frame.

Number Font: The font for the axis numbers.

Text Offset: The distance between numbers and frame.

Text, Numbers: Enables or disables axis text and numbers. The color field defines the color for axis text and numbers.

Zero Line: If this option is check a line at the value zero in the given color will be drawn.

Ticks: If checked major ticks will be drawn. The color field defines the color of the major ticks.

Length: Defines the length of the axis major ticks.

Width: Defines the width of the axis major ticks.

M.Ticks: If checked minor ticks will be drawn. The color field defines the color of the minor ticks.

Length: Defines the length of the axis minor ticks.

Width: Defines the width of the axis minor ticks.

Tick Style: Allows to control on which side of the frame the ticks will be drawn.

Grid: If checked the major grid lines will be drawn.

Width: The width of the major grid lines.

M.Grid: If checked the minor grid lines will be drawn.

Width: The width of the minor grid lines.

Grid: This color field defines the color of the major grid lines.

Minor Grid: This color field defines the color of the minor grid lines.

Axis Labels Inspector

[illegible]

Axis Label Inspector

The *Axis Labels Inspector* allows to assign arbitrary labels instead of the automatically generated numbers. The button at the top of the inspector selects on which axis the attributes below apply to.

The table with two columns shows on the left the numbers which are generated by *Plot*. In the right column you can enter a replacement for the numbers. It is a good idea to do this after you have defined the desired range for the axis because changing the range can make your labels invisible.

Clear: deletes all labels for the currently selected axis.

Error Bars Inspector

X Error Value: 0,2

☐ Data ☒ Absolut ☐ Relative ☐ Off

Y Error Value: 20

☐ Data ☒ Absolut ☐ Relative ☐ Off

Err.Lin.Wid. 2,0

Errorbar Color: [Blue]

Apply changes to all buffers ☐

Apply changes to selected buffers ☐

Error Bars Inspector

The *Error Bars Inspector* controls the error bars for your data points.

X Error Value, Y Error Value: Defines a value for errorbars. The radio button allows to switch between absolute, relative and individual data based error bars.

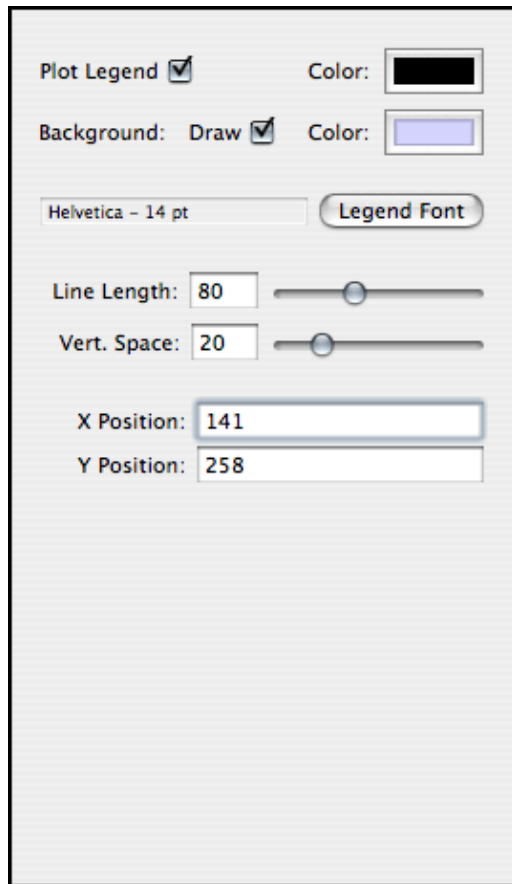
Err.Lin.Wid. Defines the linewidth for error bars.

Errorbar Color: The color for the error bars.

Apply changes to selected buffers: If checked changes will provided to all selected buffers and not only to the current working buffer (the buffer with the arrow in the first column of the *Data Inspector*).

Apply changes to all buffers: If checked changes will provided to every data buffer and not only to the current working buffer (the buffer with the arrow in the first column of the *Data Inspector*).

Legend Inspector



The Legend Inspector dialog box contains the following controls:

- Plot Legend:** A checked checkbox.
- Color:** A color selection box showing black.
- Background:** A checked checkbox.
- Draw:** A checked checkbox.
- Color:** A color selection box showing light blue.
- Font:** A text field showing "Helvetica - 14 pt" and a "Legend Font" button.
- Line Length:** A text field showing "80" and a slider.
- Vert. Space:** A text field showing "20" and a slider.
- X Position:** A text field showing "141".
- Y Position:** A text field showing "258".

Legend Inspector

The *Legend Inspector* sets the attributes for the automatically generated legend. Only buffers with a checked *L* column in the *Data Inspector* appears in the legend.

Plot Legend: Enables or disables the legend. The color field defines the color of the legend text.

Background: Enables or disables drawing of legend background. The color field defines the background color.

Legend Font: The font for the legend.

Line Length: The length of lines in the legend.

Vert. Space: The vertical space between two legend entries.

X Position, Y Position: The position of the legend. It is also possible to control the legend position with the mouse mode *Move Legend*.

Calculations Inspector

The screenshot shows the 'Calculations Inspector' window. It is divided into two main sections: 'Function Generator' and 'Buffer Calculations'.

Function Generator:

- Target selection: Radio buttons for 'Y' (selected) and 'X'.
- Expression field: Contains 'sin(rad(x))'.
- Min: Input field with '0'.
- Steps: Input field with '1000'.
- Max: Input field with '3'.
- Increment: Input field with '0,0030030'.
- Generate Buffer button.

Buffer Calculations:

- Target selection: Radio buttons for 'X Value', 'Y Value' (selected), 'X Error', and 'Y Error'.
- Expression field: Contains 'y*2'.
- Perform Calculation button.
- Buffer 1: Input field with '0'.
- Buffer 2: Input field with '1'.
- Arithmetic operators: '+', '-', 'x', '÷'.
- Perform Calculation button.
- Swap XY: Label.
- Swap Data button.

Calculations Inspector

This inspector contains the function generator and the calculator which allows buffer calculations.

Function Generator

As the name implies the *Function Generator* allows to generate function plots from arbitrary expressions.

X, Y: Select the target. Enter the expression in the field below.

Min, Max: The range in which the expression should be calculated.

Steps, Increment: The number of steps which should be calculated. These two fields depends on each other; if you enter one the other one will be calculated automatically. The number of steps is limited to 1000000.

Generate Buffer: Runs the function generator and generate a new buffer.

Buffer Calculations

The *Buffer Calculation* function allows to perform an arbitrary calculation on all selected buffers.

X Value, Y Value, X Error, Y Error: Select the target. Enter the expression in the field below.

Perform Calculation: Execute the calculation.

Buffer Calculations allow the use of some special variables:

x	x value
y	y value
ex	x error value
ey	x error value
bn	buffer number
nb	number of buffers
cb	number of the current buffers
np	number of datapoints
dp	the current datapoint

Buffer 1, Buffer 2: The lower part of the calculation area allows to perform basic calculations between two buffers. This work also for buffers with different X axis and different numbers of data points. Only the overlapping area will be processed and missing points will be interpolated.

Swap XY: Clicking this button swaps the X and Y values in all selected buffers. Error values will also be swapped if available.

Normalize Inspector

Normalize & Cut Data

1. Ref. Value: 691,032
2. Ref. Value: 410,403
1. New Value: 0
2. New Value: 1

Axis: ☒ X ☐ Y

Normalize:

Cut data:

Calculations

Integrate:

Regression:

Linear Background Subtraction

A: 290 B: -300

Normalize Inspector

With this inspector some manipulation of data like moving can be done.

Normalize Functions

1. Ref. Value, 2. Ref. Value: Defines two values from either the X or the Y axis. These values can also be defined with the normalize [mouse modes](#).

1. New Value, 2. New Value: These values will only be used for the *Normalize* function as new reference values.

Move: This function moves the selected buffers in the specified direction (selected with the *Axis* radio button) from the *1. reference* to the *1. new* value.

Normalize: This function manipulates the buffers by stretching or shrinking along the specified axis. The specified reference points in the data are changed in such a way that the *1. Ref. Value* becomes the *1. New Value* and the *2. Ref. Value* becomes the *2. New Value*.

Cut Outside: These function deletes all data points in all selected buffers which are not in the range between *1. Ref. Value* and *2. Ref. Value*.

Cut Inside: These function deletes all data points in all selected buffers which are in the range between *1. Ref. Value* and *2. Ref. Value*.

Calculations

Integrate: This function calculates the integral between X min (*1. Ref. Value*) and X max (*2. Ref Value*) and writes the result to the comment fields in the Data Inspector.

Regression: This function calculates the linear, logarithmic, or exponential regression between the X min (*1. Ref. Value*) and X max (*2. Ref. Value*) and generates a new buffer with the result. The following functions will be used:

Linear

$$A = \frac{\sum y - B * \sum x}{n}$$

$$B = \frac{n * \sum xy - \sum x - \sum y}{n * \sum x^2 - (\sum x)^2}$$

$$r = \frac{n * \sum xy - \sum x * \sum y}{\sqrt{(n * \sum x^2 - (\sum x)^2) * (n * \sum y^2 - (\sum y)^2)}}$$

$$y = A + Bx$$

Logarithmic

$$y = A + B * \ln(x)$$

For A, B, and r replace x with ln(x) in the equations above.

Exponential

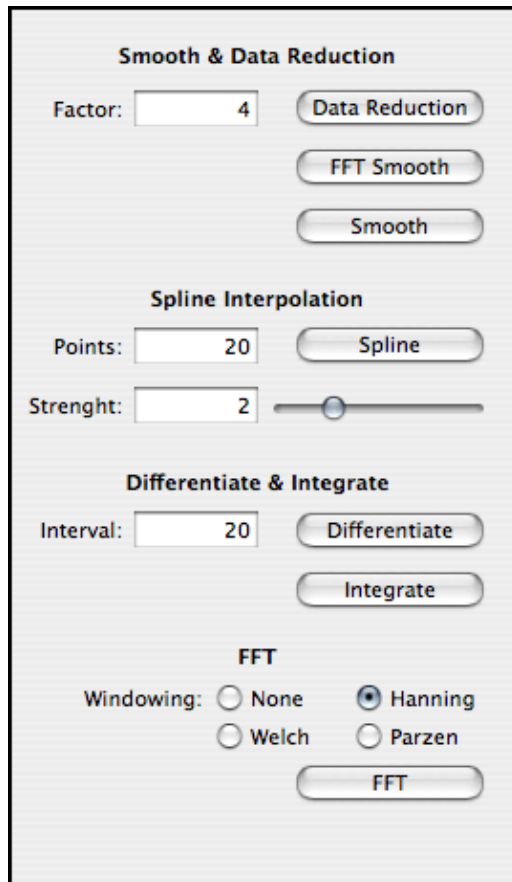
$$y = A * e^{Bx}$$

For A, B, and r replace y with ln(y) in the equations above.

Linear Background Subtraction

This function subtract a linear background from your data ($y=y-a-bx$). The reference values *a* and *b* can be set with the mouse mode *Linear Background*.

Fit Inspector



The image shows a software control panel titled "Fit Inspector". It is organized into several sections. The first section, "Smooth & Data Reduction", contains a "Factor:" input field with the value "4" and three buttons: "Data Reduction", "FFT Smooth", and "Smooth". The second section, "Spline Interpolation", contains a "Points:" input field with the value "20", a "Strenght:" input field with the value "2", and a slider control, along with a "Spline" button. The third section, "Differentiate & Integrate", contains an "Interval:" input field with the value "20" and two buttons: "Differentiate" and "Integrate". The final section, "FFT", contains a "Windowing:" label followed by four radio button options: "None", "Hanning" (which is selected), "Welch", and "Parzen", and an "FFT" button.

Fit Inspector

With this inspector some manipulation of data can be done.

Smooth & Data reduction

Data Reduction: Sometimes it is useful to reduce a large number of data points which consist mainly of noise to a more practical amount of points. This function takes n points (n is specified in the *Factor* field), builds the average, and replaces the n points with the new one.

FFT Smooth: This functions uses as effective smooth width the value from the *Factor* field and performs a FFT based smooth to reduce the noise on the buffer (see [sources](#)).

Smooth: This functions uses as effective smooth width the value from the *Factor* field and performs a least square smooth to reduce the noise on the buffer.

Spline Interpolation

This function calculates a nonparametric cubic spline interpolation (see [sources](#)) for all selected buffers.

Points: Defines the number of points which should be calculated for each interval.

Strength: Defines the *strength* of the interpolation. 2 is in most cases a god choice.

Differentiate & Integrate

This function allow to differentiate and integrate buffers (see [sources](#)).

Interval: This the effective interval which will be used to calculate the differentiation.

Differentiate: Differentiate all selected buffer and generate new buffers with the result data.

Integrate: Integrates all selected buffer and generate new buffers with the result data. The integrate function does not need the interval.

FFT (Fast Fourier Transform)

This function makes a Fast Fourier Transform for all selected buffers.

Windowing: This radio button defines the data windowing mode. For more information about FFT and data windowing read the [sources](#).

Text Inspector

Background: Draw ☒ Color:

Alignment: ☒ Left ☐ Center ☐ Right

Angle:

X Position:

Y Position:

Textbox Alignment:

Coord.Sys.: ☐ Pixel ☒ 1.Axis ☐ 2.Axis

Auto Positioning ☐

Angle: ☒ 0 ☐ 180 ☐ 90 ☐ 270

Offset:

Text Inspector

This inspector controls the graphical attributes of text objects.

Background: Enables or disables background drawing for text objects.

Color: The color the the text backgrounds.

Alignment: Defines the text alignment of a text label.

Angle: The angle of text objects. The four buttons allow easy selection of the usual angles.

X Position, Y Position: The position of the text.

Textbox Alignment: Defines the alignment of the enclosing frame of a text object.

Coord.Sys.: Defines the coordinate system to which the text object belongs. This allows text with a fixed position in the window and also text which floats with the data.

Auto Positioning: This option allow to give a text a fixed position inside your plot. The button matrix on the right selects the position relative to the plot frame.

Auto Positioning Angle: The angle of auto positioning text objects.

Auto Positioning Offset: The offset between plot frame and auto positioning text.

Graphic Inspector

The screenshot shows the 'Graphic Inspector' panel with the following settings:

- Type:** ☐ Rectangle ☒ Circle ☐ Line
- 1. X Position:** 116
- 1. Y Position:** 202
- 2. X Position:** 267
- 2. Y Position:** 104
- Width:** 151
- Height:** -98
- Coord.Sys.:** ☒ Pixel ☐ 1.Axis ☐ 2.Axis
- Linewidth:** 2,0 (with a slider)
- Line Pattern:** (with a slider)
- Display:** ☐ On Background ☒ In Front
- Fill:** ☒ (with a pink color swatch)
- Color:** (with a black color swatch)
- Arrow at Linestart:** ☐ **Arrow at Lineend:** ☐
- Arrow Size:** 10 (with a slider)
- Arrow Type:** ☒ Open ☐ Closed ☐ Triangle ☐ Double

Graphic Inspector

This inspector controls the graphical attributes of graphic objects. Currently lines, arrows, rectangle, and circles are supported.

Type: Selects the graphic type.

Background: Enables or disables background drawing for rectangles and circles.

1. X Position, 1. Y Position, 2. X Position, 2. Y Position: The position of the graphic object.

Width, Height: Size of the graphic object.

Coord.Sys.: Defines the coordinate system to which the graphic object belongs. This allows graphics with a fixed position in the window and also graphics which floats with the data.

Linewidth: Defines the width of lines.

Line Pattern: Defines the dash pattern for the line style. 16 pattern are available.

Display: Display the graphic object on the background or in front.

Fill: Enables filling of rectangles and circles. The color field defines the fill color.

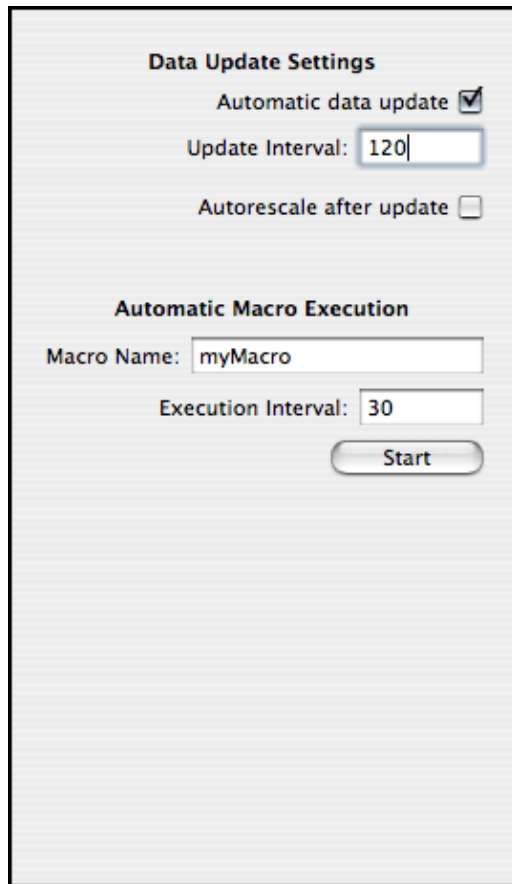
Color: The line color of the graphic object.

Arrow at linestart, Arrow at lineend: Switch arrows on or off.

Arrow Size: The size of arrows.

Arrow Type: The style of the arrows.

Advanced Inspector



The screenshot shows a dialog box titled "Advanced Inspector" with two main sections. The first section, "Data Update Settings", contains three options: "Automatic data update" which is checked, "Update Interval" set to 120, and "Autoscale after update" which is unchecked. The second section, "Automatic Macro Execution", contains two options: "Macro Name" set to "myMacro" and "Execution Interval" set to 30. At the bottom of the second section is a "Start" button.

Advanced Inspector

The *Advanced Inspector* controls some very special options which may be uninteresting for most of the users.

Data Update Settings

Controls the behavior of data buffer with dynamic data sources (e.g MySQL).

Automatic data update: If checked the automatic data update is active.

Update Interval: The interval in seconds for automatic data update.

Autoscale after update: If checked the document will be rescaled after data update.

Automatic Macro Execution

Allow to execute a macro with an specified interval.

Macro Name: The name of the macro which should be executed.

Execution Interval: The interval in seconds for automatic macro execution.

Start/Stop: Start or stop automatic macro execution.

Mouse Modes

The lower part of the *Inspector* allows to select different mouse modes. Mouse modes defines the behavior of the mouse in your document.

Measure



The crosshair allow to measure points and the distance between two points. The result will be shown in the middle part of the *Inspector*. The values will be displayed for both axis and depending on the axis mode as time value or as number.

Keyboard shortcut: m

Zoom



Zooming, the plot will be rescaled to the selected rectangle.

Keyboard shortcut: z

Normalize X



With this mouse mode two reference values on the X axis can be selected. The values will be used by the *Calculations Inspector*.

Keyboard shortcut: n

Normalize Y



With this mouse mode two reference values on the Y axis can be selected. The values will be used by the *Calculations Inspector*.

Keyboard shortcut: b

Downscale



If you click with this mouse mode inside your plot the clicked point will be centered and the range of the plot will be increased.

Keyboard shortcut: d

Upscale



If you click with this mouse mode inside your plot the clicked point will be centered and the range of the plot will be lowered.

Keyboard shortcut: u

Range



This is for easy navigating through your data. Clicking and moving in the plot changes the range accordingly. If you have a mouse wheel you can also zoom in and out. If you hold the option keys while zooming only the X axis changes. Holding the command key does the same for the Y axis.

Keyboard shortcut: r

Select Tool



This mouse mode allows different manipulations on your plot. You can resize the frame of your plot; move and edit texts; select data points which will then highlighted in the [data view](#); change the current working buffer; and select axis. After selection the appropriate inspector appears. If you select a text or graphic object you can use this keys:

left, right, up, down	move the object
control + left, right, up, down	move the 1. graphic object handle
option + left, right, up, down	move the 2. graphic object handle
backspace	delete the object
command+right, command+left	select the next or previous object

Keyboard shortcut: s

Move Y



This mouse mode allows moving of data from all selected buffers in Y direction. This operation changes the data itself and not just the range of the plot.

Keyboard shortcut: y

Move X



This mouse mode allows moving of data from all selected buffers in X direction. This operation changes the data itself and not just the range of the plot.

Keyboard shortcut: x

Move XY



This mouse mode allows moving of data from all selected buffers in X and Y direction. This operation changes the data itself and not just the range of the plot.

Keyboard shortcut: q

Move Points



With this mouse mode you can select and move a single data point. This operation changes the data itself and not just the range of the plot.

Keyboard shortcut: p

Linear Background



Allows to define the reference values for linear background subtraction with the mouse.

Keyboard shortcut: w

Legend Moving



With this mouse mode the automatically generated legend can be moved.

Keyboard shortcut: l

Graphic



With this mouse mode graphic objects can be created, sized and moved.

Keyboard shortcut: g

Text



This mouse mode allows to add and edit texts in your plot.

Keyboard shortcut: t

Subview



With this mouse mode new empty subviews can be generated.

Keyboard shortcut: h

Change Subview

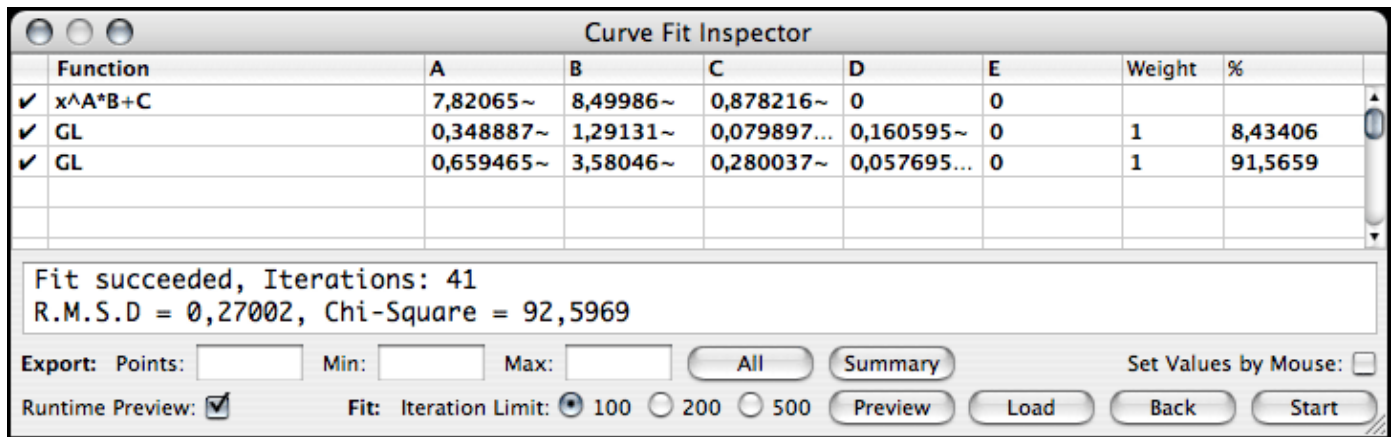


This mouse mode allow to change the size and position of a subview.

Keyboard shortcut: j

Curve Fit Inspector

This function allows you to fit data with some predefined functions or free defined functions. Fits are possible with up to 24 functions each with up to 5 free parameters (see [sources](#)).



Curve Fit Inspector

The *Curve Fit Inspector* consists of a few controls and a table for the function and fit parameter definition.

Active functions are displayed with a check mark in the first column. Double clicking the first column changes the state.

Function: In this column you have to define the function you want to fit (see below).

A, B, C, D, E: This are the free parameters for the fit function.

Weight: The weight of a function for the integration routine (only used by the special functions).

%: The result of the integration will be displayed in this column (only calculated for the special functions).

Start: Starts the fit for the current working buffer.

Preview: Displays all the fit functions and the summary curve in the current document.

Back: Every time you start a fit the state will be saved in a history buffer. With this button you can go back trough the fit steps.

Load: Loads fit data from another document.

All: Exports all the fit functions and the summary curve as new buffer to your document.

Summary: Exports the summary curve as new buffer to your document.

Points, Min, Max: Defines the range and the number of points for curve export. If this fields are empty for each point in the current working buffer a new value will be calculated.

Iteration Limit: Defines a cycle limit after which it is possible to stop the fit.

Runtime Preview: If this button is checked a preview will be displayed after each fit cycle.

Set Values by Mouse: If this button is checked the values from the mouse mode *Measure* will be sent to the currently selected line in the parameter table (A = X position and B = Y position). After this the next line in the parameter table will be selected automatically. This makes it easy to predefine a bunch of peaks.

Fit result

The text field in the middle of the inspector shows the status and the result of the fit. The first line displays the fit state and the second the fit result:

R.M.S.D: Root Mean Square Deviation of the last cycle:

$$R.M.S.D = \sqrt{\frac{\sum_{i=1}^n [y_i - y(x_i)]^2}{n}}$$

Chi-Square:

$$\chi^2 = \sum_{i=1}^n \left(\frac{y_i - y(x_i)}{\sigma_i} \right)^2$$

sigma is the error of your data. If your data set doesn't contain error values *Plot* makes a statistically approximation of the standard deviation.

Function definition

In the *Function* column you can enter normal mathematical expression as functions for fit (there are some special functions, see below).

Available variables:

x	x value
A	A parameter
B	B parameter
C	C parameter
D	D parameter
E	E parameter

Example: (x+A)^2+B

For the 5 fit parameter (A,B,C,D,E) the following syntax is possible:

<value>	a fixed value
<value>~	a value which should be fitted
<value>~<dvalue>	a value which should be fitted in the range from <i>value-dvalue</i> to <i>value+dvalue</i>
<row>[+,-,*,/]<rel>	Sets value to the entry of same column in a previous row, plus/minus/times/divided by rel.

Examples: 284 1234~ 10.7~0.2 1-2.3 5*0.12

Special Functions

The fit function supports some special functions. These functions are easy to use and a little bit faster than free defined functions. The special functions can not be used together with other expressions in one row.

GL (Gauss-Lorentz mix curve)

$$l = \frac{l_0 * y^2}{y^2 + 4(1 - M)(x - x_0)} * e^{-4 \frac{M(x-x_0)^2}{y^2}}$$

A = position (X0)

B = height (I0)

C = width (*gamma*, FWHM)

D = Gauss-Lorentz ratio (M ,1.0=pure Gauss, 0.0 = pure Lorentz)

E = unused

DS (Doniach-Sunjic curve)

$$l = \frac{l_0(1 - n\alpha) * \Gamma(1 - \alpha)}{\left(\left(\frac{2(x-x_0)}{y} + k * \cot\left(\frac{\pi}{2-\alpha}\right) \right)^2 + 1 \right)^{\frac{1-\alpha}{2}}} * \cos(\chi)$$

$$\chi = \frac{\pi\alpha}{2} + (1 - \alpha) \operatorname{atan}\left(\frac{2k}{\gamma}(x - x_0) + \cot\left(\frac{\pi}{2 - \alpha} \right) \right)$$

n=0.588468, k=1 (right skew) or -1 (left skew)

A = position (X0)

B = height (I0)

C = width (*gamma*, Lorentzian FWHM)

D = Anderson's exponent (*alpha*, -0.5 ... 0.5)

E = unused

ET (Gauss-Lorentz mix curve with exponential Tail)

$$l_e = l + (l_0 - l)e^{\frac{0.1|x-x_0|}{\alpha}}$$

A = position (X0)

B = height (I0)

C = width (*gamma*, Lorentzian FWHM)

D = Gauss-Lorentz ratio (M ,1.0=pure Gauss, 0.0 = pure Lorentz)

E = tail exponent factor (*alpha*, -infinity - +infinity)

The numerical Gauss convolution of the following two functions (Lorentz and Doniach-Sunjic, F(Y0, x0, *gamma* [,*alpha*])) in the interval x1 to xn in steps of dx is performed with the following formula:

$$i = 1 \dots \text{integer} \left(\frac{x_N - x_1}{d_x} + 1 \right)$$

$$l_i = Q_{y_0} d_x \sum_{j=-m}^m \left(F(x_1 + \Delta_{x_0} + d_x(i+j)) e^{-\frac{j^2 d_x^2}{G^2}} \right)$$

$$m = \text{integer} \left(\frac{2G}{d_x} \right)$$

G is the FWHM of the convoluting Gauss function; Y0, x0, *gamma* [,alpha] are the parameters of the function to be convoluted, as in the earlier expressions for Lorentz and Doniach-Sunjic functions. The normalizing factor QY0 and shift dx0 are defined as QY0=y0 / y'0 and dx0=x'0 - x0, where y'0 and x'0 are the amplitude and position of the maximum of the not normalized convoluted function.

GL* (Gauss convoluted Lorentz curve)

A = position

B = height

C = width (FWHM)

D = Gauss-Lorentz ratio (must be set to 0.0)

E = Gauss FWHM (0 - +infinity)

DS* (Gauss convoluted Doniach-Sunjic curve)

A = position

B = height

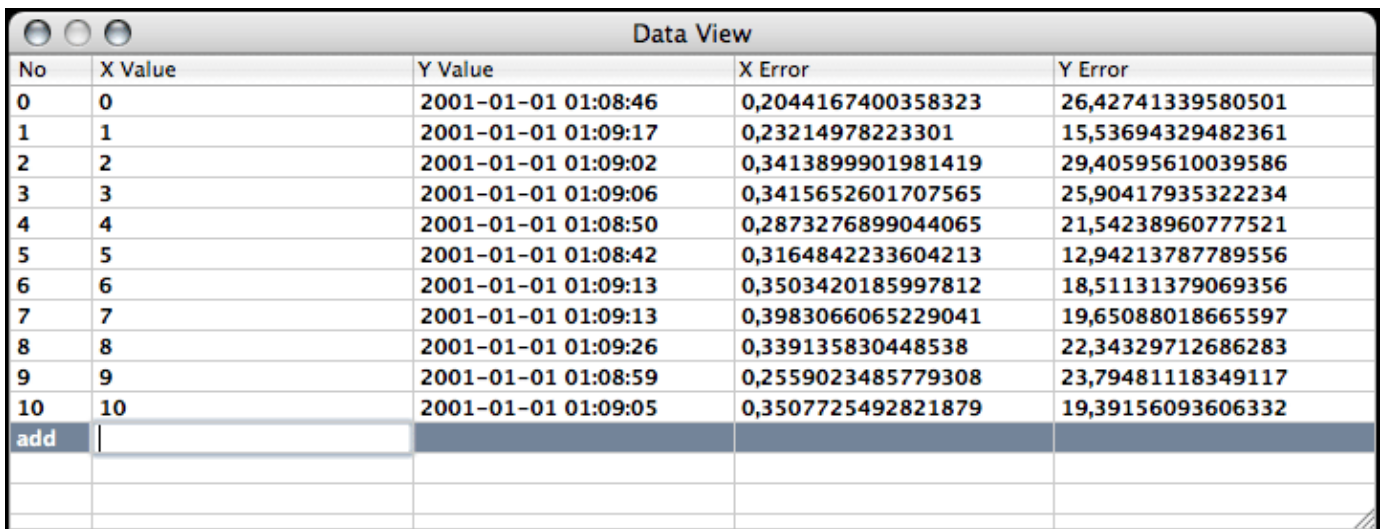
C = width (Lorentzian FWHM)

D = Anderson's exponent (-0.5 ... 0.5)

E = Gauss FWHM (0 - +infinity)

DataView

The *Data View* is a spreadsheet like editor for your data. It show the data of the working buffer.



No	X Value	Y Value	X Error	Y Error
0	0	2001-01-01 01:08:46	0,2044167400358323	26,42741339580501
1	1	2001-01-01 01:09:17	0,23214978223301	15,53694329482361
2	2	2001-01-01 01:09:02	0,3413899901981419	29,40595610039586
3	3	2001-01-01 01:09:06	0,3415652601707565	25,90417935322234
4	4	2001-01-01 01:08:50	0,2873276899044065	21,54238960777521
5	5	2001-01-01 01:08:42	0,3164842233604213	12,94213787789556
6	6	2001-01-01 01:09:13	0,3503420185997812	18,51131379069356
7	7	2001-01-01 01:09:13	0,3983066065229041	19,65088018665597
8	8	2001-01-01 01:09:26	0,339135830448538	22,34329712686283
9	9	2001-01-01 01:08:59	0,2559023485779308	23,79481118349117
10	10	2001-01-01 01:09:05	0,3507725492821879	19,39156093606332
add				

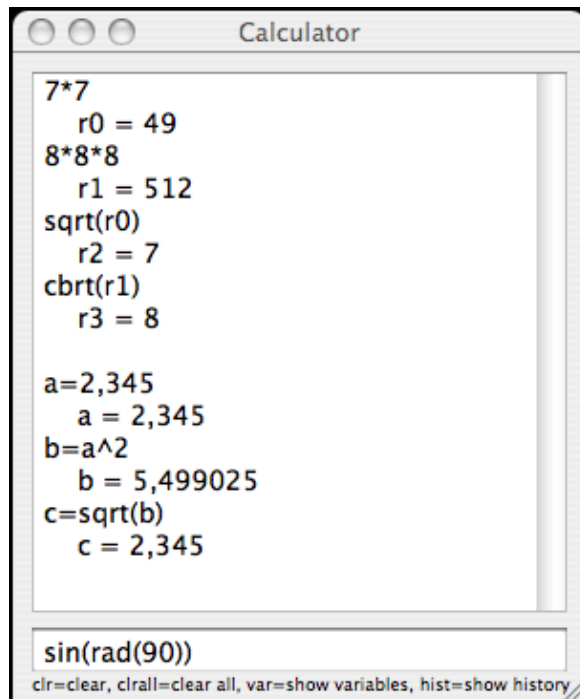
Data View

Double clicking on the *X Value* or *Y Value* column header switches the display format between number and time values.

The *X Error* and *Y Error* column stores individual error values for each data point.

If you select rows in the *Data View* the points will be highlighted in your document.

Calculator



Calculator

This is a simple builtin calculator for instant calculations. The *Calculator* supports the common math expressions of *Plot*. Just enter your calculation in the lower text field and hit return to get the result in the upper textfield. The result of every calculation is stored in a variable (*r0* ... *r<n>*). These variables can be used in later calculation and also in all other functions where *Plot* allow expressions (e.g. in the function generator).

There are three special commands in the calculator:

<i>clr</i>	reset the calculator and deletes all variables and the history.
<i>clrall</i>	reset the calculator and deletes all variables, user variables and the history.
<i>vars</i>	show a list of available variables
<i>hist</i>	show the calculator history

It is also possible to define user specific variables. Just enter:

<name>=<value>

User variables can also be used in all other functions where *Plot* allow expressions.

Expressions

Plot functions which allow entering of mathematical expressions supports the following functions:

+, -, *, /	arithmetic operations
%, mod	modulo
()	grouping
^, **	power
rad(x), deg(x)	conversion between radians and degrees
sin(x), cos(x), tan(x)	trigonometric functions
asin(x), acos(x), atan(x)	inverse trigonometric functions
sinh(x), cosh(x), tanh(x)	hyperbolic functions
rnd(h)	random number (h = height)
ln(x), log(x)	natural and logarithm to base 10
sqrt(x)	square root
cbrt(x)	cubic root
frac(x)	returns the fraction of x
int(x)	returns the integer of x
round(x;n)	round up and down to the nth place on the right of the decimal point
gau(x;x0;a;w)	Gauss (x0 = position, a = amplitude, w = width)
lor(x;x0;a;w)	Lorentz (x0 = position, a = amplitude, w = width)
galo(x;x0;a;w;r)	Gauss-Lorentz (x0 = position, a = amplitude, w = width, r=Gauss-Lorentz ratio (1.0=pure Gauss, 0.0 = pure Lorentz))
tail(x;x0;a;w;r;t)	Gauss-Lorentz with exponential Tail (x0 = position, a = amplitude, w = width, r = Gauss-Lorentz ratio (1.0=pure Gauss, 0.0 = pure Lorentz)), t = tail exponent factor
j0(x), j1(x), jn(x,n)	bessel functions
y0(x), y1(x), yn(x,n)	bessel functions
pi	3.14159265359
e	2.71828182846
xval(b;i)	x value of point <i>i</i> in buffer <i>b</i>
yval(b;i)	y value of point <i>i</i> in buffer <i>b</i>
xerr(b;i)	x error value of point <i>i</i> in buffer <i>b</i>
yerr(b;i)	y error value of point <i>i</i> in buffer <i>b</i>
xmin(b)	minimum x value in buffer <i>b</i>
xmax(b)	maximum x value in buffer <i>b</i>
ymin(b)	minimum y value in buffer <i>b</i>
ymax(b)	maximum y value in buffer <i>b</i>
points(b)	number of points in buffer <i>b</i>
xpoint(v;a)	x screen coordinate of v (a can be 1 for the first or 2 for the second axis).
xvalue(v;a)	x value of the screen coordinate v (a can be 1 for the first or 2 for the second axis).
ypoint(v;a)	y screen coordinate of v (a can be 1 for the first or 2 for the second axis).
yvalue(v;a)	y value of the screen coordinate v (a can be 1 for the first or 2 for the second axis).
findx(v;b)	x value which depends to y in buffer b. If there are more than one it returns the first it found.

findy(v;b)

y value which depends to x in buffer b. If there are more than one it returns the first it found.

Time Format

Plot stores time data as the number of seconds relative to an absolute reference time: the first instant of 1 January, 2001, Greenwich Mean Time (GMT). Dates before then are stored as negative numbers; dates after then are stored as positive numbers.

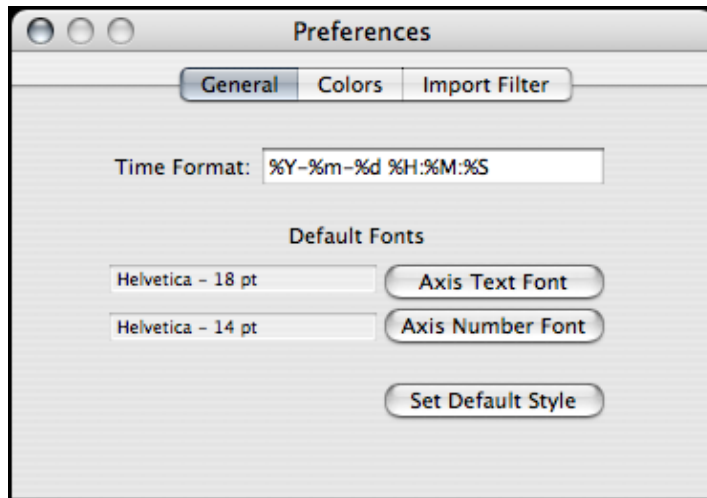
To convert a UNIX time to a *Plot* time simply subtract 978307200.0 from the UNIX time.

Time format tokens

- `%a` Abbreviated weekday name
- `%A` Full weekday name
- `%b` Abbreviated month name
- `%B` Full month name
- `%c` Shorthand for `%x`, the locale format for date and time
- `%d` Day of the month as a decimal number (01-31)
- `%e` Same as `%d` but does not print the leading 0 for days 1 through 9 (unlike `strftime()`, does not print a leading space)
- `%F` Milliseconds as a decimal number (000-999)
- `%H` Hour based on a 24-hour clock as a decimal number (00-23)
- `%I` Hour based on a 12-hour clock as a decimal number (01-12)
- `%j` Day of the year as a decimal number (001-366)
- `%m` Month as a decimal number (01-12)
- `%M` Minute as a decimal number (00-59)
- `%p` AM/PM designation for the locale
- `%S` Second as a decimal number (00-59)
- `%w` Weekday as a decimal number (0-6), where Sunday is 0
- `%x` Date using the date representation for the locale, including the time zone (produces different results from `strftime()`)
- `%X` Time using the time representation for the locale (produces different results from `strftime()`)
- `%y` Year without century (00-99)
- `%Y` Year with century (such as 1990)
- `%Z` Time zone name (such as Pacific Daylight Time; produces different results from `strftime()`)
- `%z` Time zone offset in hours and minutes from GMT (HHMM)

Preferences

The *Preferences Panel* allow to change some global setting in *Plot*



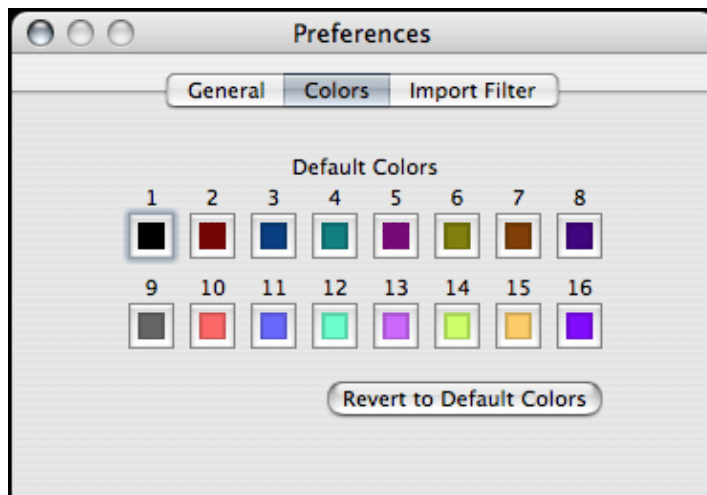
General

General

Time Format: Defines the format for time values used in *Plot*.

Default Font: Allow to define default fonts for axis labels and axis numbers.

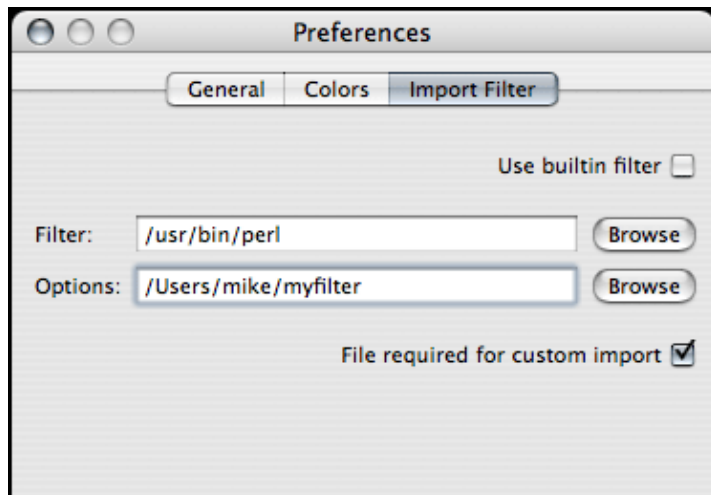
Set Default Style: If clicked the style of the current working buffer in the current document will be the default style for newly generated data buffers.



Colors

Colors

Here you can define your own default colors used by *Plot*. It is also possible to restore the default colors provided by *Plot*.



Import Filter

Import Filter

Besides the builtin import filter for ASCII data you can use your own custom filter to import data.

Use builtin filter: Enables or disables the builtin import filter.

Filter: Defines the custom filter. This has to be an executable file.

Options: Defines additional option for the import filter.

File required for custom import filter: If enabled you will be asked for a file to import. If disabled your filter needs to know by itself where to find the data.

Information how to make an import filter can be found in the [import filter section](#).

Menu Item

Apart from the usual Apple menu items *Plot* has some more which will be described here:

File Import ASCII (<Shift>+<Command>+A)

Using this item brings up a open panel in which you can select a file for ASCII import.

File Import Binary (<Shift>+<Command>+B)

Using this item brings up a open panel in which you can select a file for binary import.

File Import MySQL (<Shift>+<Command>+L)

This items brings up the MySQL Import dialog which allows direct data import from a MySQL database.

File Import SciPlot (<Shift>+<Command>+O)

Allow to import data from old *SciPlot* files. This option allow to recover all data and some graphic attributes from SciPlot files. The result may look different as you know it from *SciPlot*.

File Load Partial File Layout (<Control>+<Command>+L)

File Load Partial File Data (<Control>+<Command>+V)

File Load Partial File Fit (<Control>+<Command>+X)

File Load Partial File Range (<Control>+<Command>+R)

File Load Partial File Colors (<Control>+<Command>+C)

File Load Partial File Fonts (<Control>+<Command>+F)

File Load Partial File Text & Graphic (<Control>+<Command>+T)

These item allow partial load of *Plot* files. This is handy to give different documents the same graphical attributes.

File Export Image PDF (<Option>+<Command>+P)

File Export Image JPG (<Option>+<Command>+J)

File Export Image EPS (<Option>+<Command>+E)

File Export Image PDF (<Option>+<Command>+F)

Saves your document in one of the given formats as graphic file.

File Export ASCII

This exports all selected buffers of the current document as ASCII data. A new file for every buffer will be generated with the given name and the appended buffer number.

File New Empty Buffer (<Shift>+<Command>+N)

Generates a new empty buffer in the current document. You can fill it with data by using the Data View.

File Delete Subview (<Command>+D)

Deletes the currently selected subview.

View Inspector (<Shift>+<Command>+I)

Brings up the Inspector

View Data View (<Shift>+<Command>+D)

Brings up the Data View

View Data Inspector (<Shift>+<Command>+V)

Brings up the Data Inspector

View Curve Fit Inspector (<Shift>+<Command>+X)

Brings up the Curve Fit Inspector

View Macro Inspector (<Shift>+<Command>+M)

Brings up the Macro Inspector

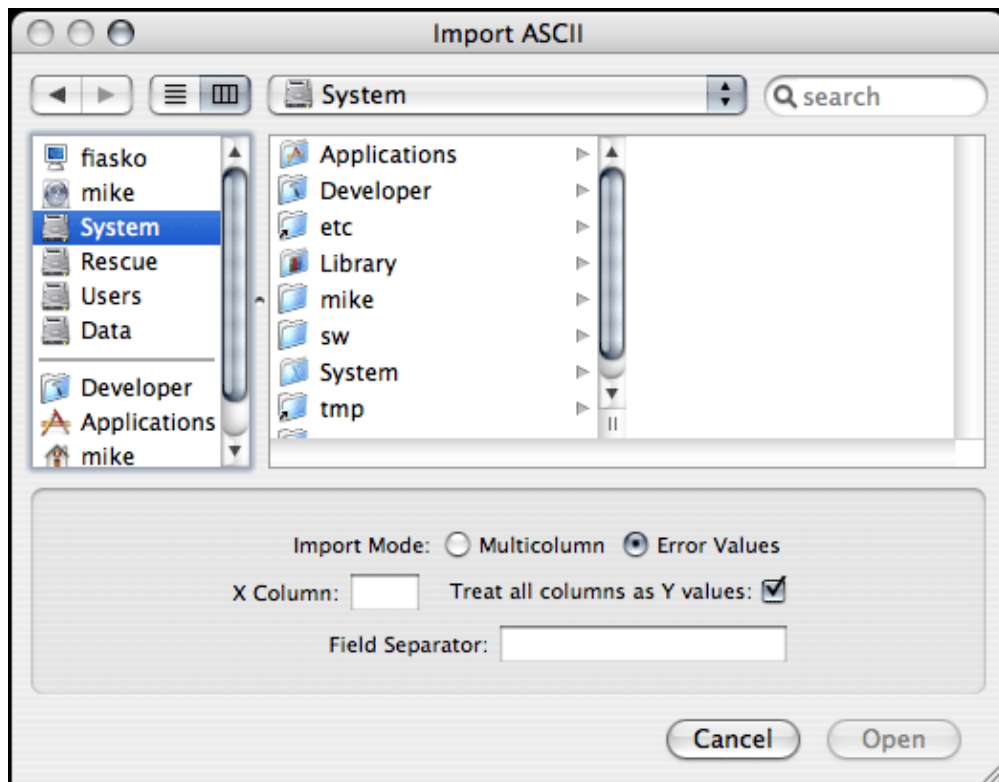
View Calculator (<Command>+R)

Brings up the Calculator

Import Filter

Import

General the import filter accepts file with UNIX, Mac and Windows line end characters. Each line should contain at least one number. Numbers in the line will be separated by any character which could not be part of a number.



ASCII Import Panel

The import dialog allows to select between two modes of importing ASCII data.

Multicolumn: The builtin import filter expects an ASCII file with one or more columns.

If your file has only one column the values will be treated as Y values, the X values will be generated as sequence number.

If you have more than one column the first will be treated as X value and every other column as Y value.

Error Values: This mode expects up to 4 columns of data. The 1. and 2. will be treated as X and Y values. The 3. and 4. as X error values and Y error values. If only three column are available the 3. will used as Y error values.

X Column: Defines which column should be the X column starting with 1 for the first column.

Treat all columns as Y values: If checked all columns will be interpreted a Y values and the X values will be generated as sequence number.

Defines which column should be the X column starting with 1 for the first column.

Field Separator: Defines the field separator for ASCII import. Explicit definition of the separator allows to import files with empty fields.

Custom Import Filter

For using a custom filter you need to know what *Plot* expects from the filter. Right now this is really easy to implement.

Plot calls the filter with some arguments where the last is always the filename.

If you want to use a perl script to import some data you have to set the *Filter* field in the preferences panel with `/usr/bin/perl` and in the *Options* field with the path to your script (e.g. `/Users/mike/myscript`). *Plot* will now run the following command line to call the filter:

```
/usr/bin/perl /Users/mike/myscript <file>
```

Now your script has to parse the file and write to standard output a stream that *Plot* understands. The stream *Plot* expects is just a sequence of numbers separated by colons (:). The first number in the sequence must be the number of columns in your data set.

e.g. if you want to import this data:

```
1 1
2 4
3 9
4 16
```

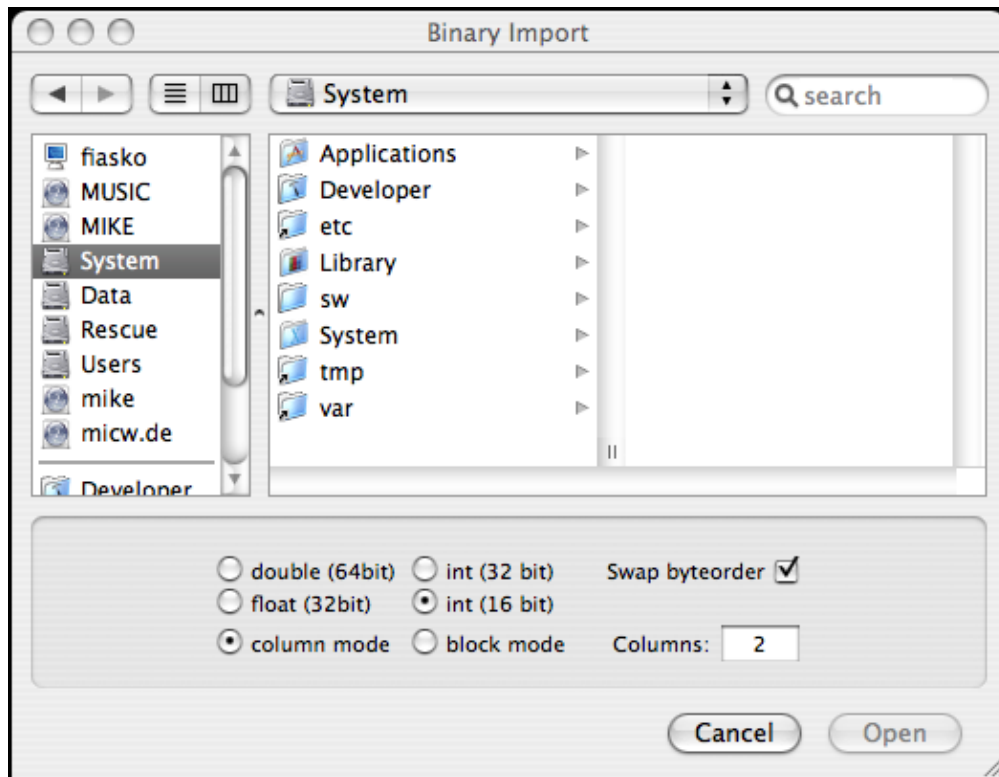
your script have to write this data stream:

```
2:1:1:2:4:3:9:4:16
```

You have to provide at least 2 columns of data.

Binary Import Filter

Plot allows importing of binary data. This can be done by selecting *Binary Import* from the *File* menu. In the open panel you have to select your binary file and to provide some extra informations.



Binary Import Panel

Data type: Specifies the type of data in your file. Valid types are double (64 bit), float (32 bit), integer (32 bit) or integer (16 bit).

Import mode: Defines how the data are ordered in your file. *Column mode* expects data ordered in columns and *Block mode* expects a block for each column.

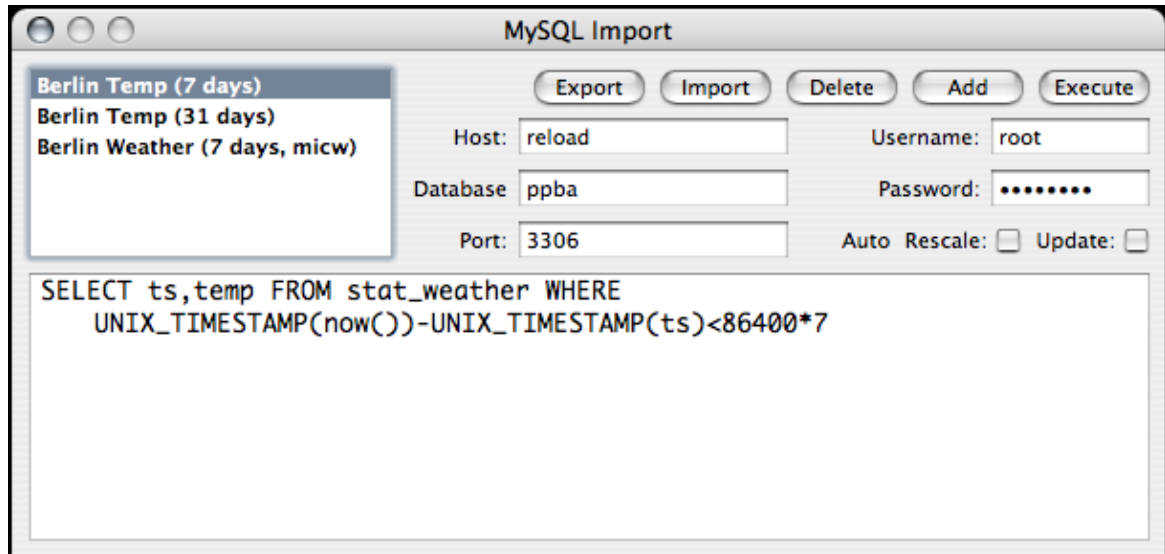
Swap byteorder: Allows to swap data which are generated on a platform with different byte order.

Columns: The number of columns in your file (has to be at least 1). If your file has only one column the values will be treated as Y values, the X value will be generated as sequence number. If you have more than one column the first will be treated as X value and every other column as Y value.

MySQL Import

This function allow directly to import data from a MySQL database. In the upper field of the panel you have to define the database parameters. The list on the left contains a list of all your MySQL queries. In the lower part you can enter an SQL query; typically in the form:

```
SELECT <x column>,<y column> FROM <table> WHERE ...
```



The screenshot shows a window titled "MySQL Import". On the left is a list box containing three items: "Berlin Temp (7 days)", "Berlin Temp (31 days)", and "Berlin Weather (7 days, micw)". To the right of the list box are five buttons: "Export", "Import", "Delete", "Add", and "Execute". Below these buttons are input fields for "Host:" (containing "reload"), "Database:" (containing "ppba"), "Port:" (containing "3306"), "Username:" (containing "root"), and "Password:" (containing seven dots). There are also checkboxes for "Auto Rescale:" and "Update:". At the bottom is a large text area containing the SQL query:

```
SELECT ts,temp FROM stat_weather WHERE
  UNIX_TIMESTAMP(now())-UNIX_TIMESTAMP(ts)<86400*7
```

MySQL Import Panel

Import, Export: Allows to import or export the whole query set and store it in a file for later use.

Delete: Delete the selected MySQL query.

Add: Add a new empty MySQL query.

Execute: Performs the MySQL query and generate buffers as needed.

Host: The host where your database resides.

Database: The database name.

Port: The TCP port of your database (usually 3306).

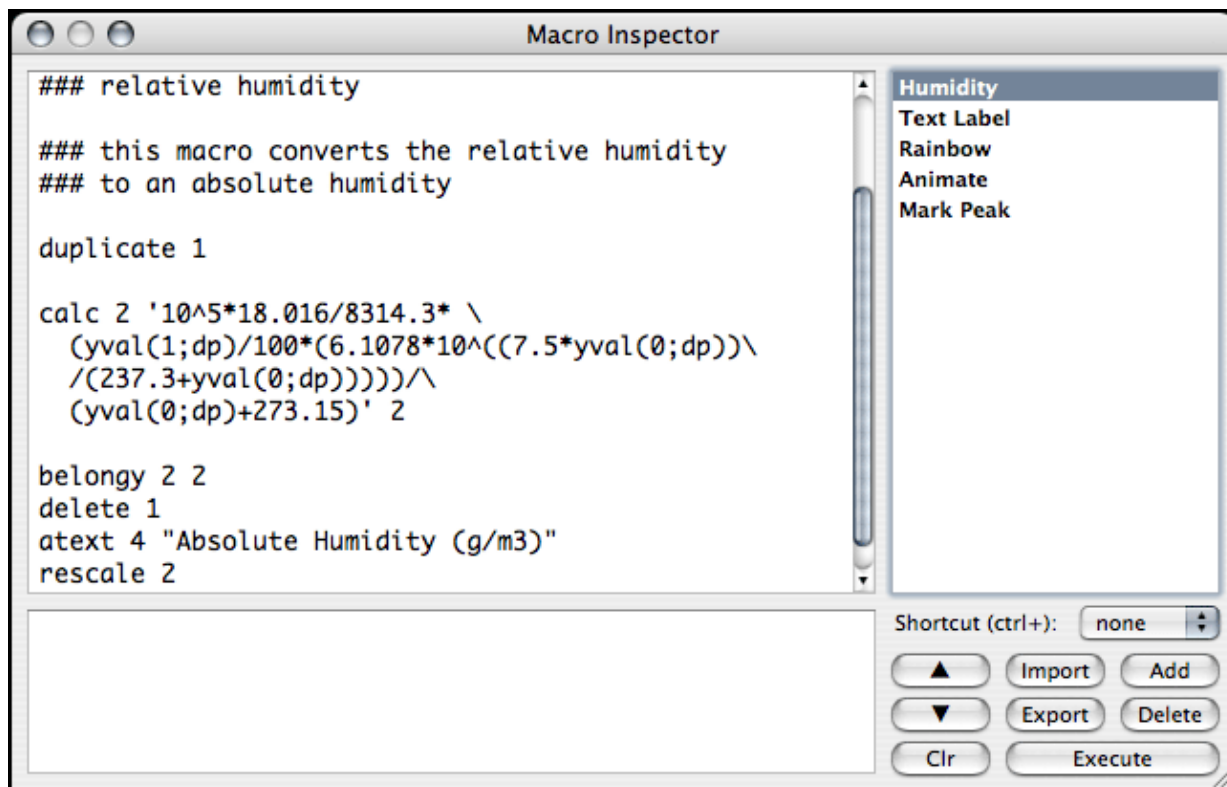
Username, Password: MySQL login informations.

Auto Update: If checked the data will automatically updated e.g. if you open the document.

Auto Rescale: If checked your document will be rescaled after an data update.

Macro Inspector

Plot has a build in macro language which allows to automate complex task which can be called with a single shortcut in your document.



Macro Inspector

The big text field contain the macro itself. The small text field is the output console where messages or debug info can be printed. The list on the right contains a list of all your macros. You can create an unlimited number of macros but only 26 keyboard shortcuts for macros.

Up, Down: Moves the selected macro in the list up or down.

Import, Export: Allows to import or export the whole macro set and store it in a file for later use.

Add: Create a new empty macro.

Delete: Delete the currently selected macro.

Clr: This clears the console output.

Execute: Executes the macro on the current document.

Macro Language

Introduction

There are ~ 180 commands available in the *Plot* macro language. Most of the commands needs one or more argument. Simply enter the command followed by the arguments separated by spaces. If an argument contains spaces you have to quote the argument with ' or ". Comments can be escaped with #. For example a command may look like this:

```
atext 1 'Axis Text'
```

This commands set the axis text label for axis 1 (the first X axis).

Command Reference

[Range Commands](#)

[Buffer Commands](#)

[Control Commands](#)

[Interactive Commands](#)

[Calculation Commands](#)

[Style Commands](#)

[Data Style Commands](#)

[Axis Style Commands](#)

[Legend Commands](#)

[Text & Graphic Commands](#)

[Document Commands](#)

[Miscellaneous Commands](#)

Variables

During macro execution several variables are available. String variable starts with a \$.

cb	number of the current working buffer in the document
\$date	the current date
\$document	the filename of the current document
dx1	the distance between the last two measures with the mouse (1. X axis)
dx2	the distance between the last two measures with the mouse (2. X axis)
dy1	the distance between the last two measures with the mouse (1. Y axis)
dy2	the distance between the last two measures with the mouse (2. Y axis)
\$file	the filename from the browse command
framebottom	the position of bottom frame line
frameleft	the position of left frame line
framerright	the position of right frame line
frametop	the position of top frame line
framewidth	the width of the frame
\$home	the users home directory
input	the result of the last input command
\$input	the result of the last input command as string

<code>integral</code>	the result from the last <code>calcint</code> command
<code>l</code>	the run variable for loops
<code>lastbuffer</code>	the number of the last buffer generated with one of the calculation commands
<code>marginbottom</code>	the bottom margin
<code>marginleft</code>	the left margin
<code>marginright</code>	the right margin
<code>margintop</code>	the top margin
<code>nb</code>	number of data buffer in the document
<code>option</code>	the result of the <code>askoption</code> command
<code>ref1</code>	the 1. reference value
<code>ref2</code>	the 2. reference value
<code>linbga, linbgb</code>	the reference values for linear background subtraction
<code>rega, regb, regr</code>	the result of the last regresion
<code>textheight</code>	the height of the last added text
<code>textwidth</code>	the width of the last added text
<code>\$time</code>	the current time
<code>\$user</code>	the users name
<code>windowheight</code>	the window height
<code>windowwidth</code>	the window width
<code>xpos1</code>	the last result of the measure with the mouse (1. X axis)
<code>xpos2</code>	the last result of the measure with the mouse (2. X axis)
<code>ypos1</code>	the last result of the measure with the mouse (1. Y axis)
<code>ypos2</code>	the last result of the measure with the mouse (2. Y axis)
<code>xmin1</code>	min value of the 1. (bottom) X axis
<code>xmax1</code>	max value of the 1. (bottom) X axis
<code>ymin1</code>	min value of the 1. (left) Y axis
<code>ymax1</code>	max value of the 1. (left) Y axis
<code>xmin2</code>	min value of the 2. (top) X axis
<code>xmax2</code>	max value of the 2. (top) X axis
<code>ymin2</code>	min value of the 2. (right) Y axis
<code>ymax2</code>	max value of the 2. (right) Y axis

Arguments

In the command descriptions on the following pages optional argument are written with surrounded square brackets (`[argument]`) and required arguments with angle brackets (`<argument>`).

There are also some special arguments:

`<BUFFER>`

a list of one or more data buffers. Possible values are:

`all` all buffers in the document

Variables

selected	selected buffers in the document
unselected	unselected buffers in the document
visible	visible buffers in the document
hidden	hidden buffers in the document
none	no buffer
$b_1, b_2, b_3, \dots, b_n$	a list of buffers where the arguments may be math expressions.
$bs..be$	buffers from bs to be

<AXIS>

a definition for the four axis. Possible values are:

all	all four axis
x	both X axis
y	both Y axis
1	1. X axis
2	1. Y axis
3	2. X axis
4	2. Y axis

<AXISGROUP>

the coordinate system:

0	both axis
1	first axis (left and bottom)
2	second axis (right and top)

<COOR>

the coordinate system:

0	screen coordinates
1	first axis (left and bottom)
2	second axis (right and top)

<RANGE>

a list of numbers (e.g. used for the `loop` command):

n	defines a range from 0 to n
$from:to:step$	defines a range where the arguments may be math expressions.
$b_1, b_2, b_3, \dots, b_n$	a list of numbers where the arguments may be math expressions.

`bs..be`

numbers from *bs* to *be*

Macro Language: Range Commands

`range <AXIS> <min> <max>`

Sets the axis min and max.

`rescale [AXISGROUP]`

Rescales the plot so that every datapoint is just visible.

`upscale <AXISGROUP> [factor]`

Lowers the range of the plot by the given factor (default=10).

`downscale <AXISGROUP> [factor]`

Increase the range of the plot by the given factor (default=10).

Macro Language: Buffer Commands

`select <BUFFER>`

Select buffers in the [Data Inspector](#).

`setcurrent <number>`

Defines the current working buffer.

`invertselections`

Inverts the selection in the [Data Inspector](#).

`hide <BUFFER>`

Hide buffers.

`show <BUFFER>`

Show buffers.

`delete <BUFFER>`

Delete buffers.

`duplicate <BUFFER>`

Duplicate buffers.

`setcomment <BUFFER> <text>`

set comment field of buffers.

`setsource <BUFFER> <text>`

set source field of buffers.

`addcomment <BUFFER> <text>`

add text to the comment field of buffers.

`addsource <BUFFER> <text>`

add text to the source field of buffers.

`belongx <BUFFER> <axis>`

defines if the buffers belongs to the 1. (bottom) or the 2. (top) X axis (possible values: 1, 2).

`belongy <BUFFER> <axis>`

defines if the buffers belongs to the 1. (left) or the 2. (right) Y axis (possible values: 1, 2).

`update`

update buffers which has a dynamic data source (e.g. MySQL).

`addvalue <buffer> <x value> <y value> <x error> <y error>`

add value pair to data buffer.

`setxvalue <buffer> <index> <value>`

set X value in data buffer.

`setxerror <buffer> <index> <value>`

set X error in data buffer.

`setyvalue <buffer> <index> <value>`

set Y value in data buffer.

`setyerror <buffer> <index> <value>`

set Y error in data buffer.

`rmvalue <buffer> <index>`

delete value from data buffer.

Macro Language: Control Commands

```
if <expr> <operator> <expr>
else
endif
```

Executes the commands between `if` ... and `else` if the condition is true. If the condition is false it executes the commands between `else` and `endif`. Possible operators are `==`, `!=`, `>`, `<`, `>=`, `<=` for numerical comparison and `eq`, `ne` for string comparison. It is not possible to define `if`, `else`, `endif` statements inside another `if`, `else`, `endif` statements.

```
loop <RANGE>
```

Starts a loop for the given range. The running variable for the loop command is `l`. It is not possible to define loops inside another loop.

```
bufferloop <BUFFER>
```

Starts a loop for the given buffer numbers. The running variable for the loop command is `l`. It is not possible to define buffer loops inside another loop.

```
endloop
```

The endpoint of a `loop` or `bufferloop` command.

Macro Language: Interactive Commands

`ask <text>`

This command stops the macro, shows an alert box, displays the `text`, and let you choose to stop or continue the macro.

`askoption <text> <button1> <button0>`

This command stops the macro, shows an alert box, displays the `text`, and provide two buttons. The result will be stored in the variable `option` (1=`button1` clicked, 0=`button0` clicked)

`browse`

Opens a panel for file selection. The selected file will be stored in `$file`.

`input`

This command brings up an input panel and let you enter a value. The value will be stored in variables `input` and `$input`.

`plot`

Normally your document will be updated at the end of the macro execution. This command allow updates the document when called.

`print <argument> ...`

Print messages to the console window. Argument will be treated as expression or string.

`sleep <seconds>`

Stops the macro execution for the given time in seconds.

Macro Language: Calculation Commands

`fg <axis> <min> <max> <steps> <expr>`

Generate a function with the given parameters.

<axis>: The target axis (1 = X axis, 2 = Y axis)

<min>, **<max>**: The range for the new generated function.

<steps>: Number of steps for the the function.

<expr>: Expression which defines the function.

`bcalc <buffer1> <operator> <buffer2>`

Performs basic calculations between two buffers. This work also for buffers with different X axis and different numbers of data points. Only the overlapping area will be processed and missing points will be interpolated.

<operator>: + - * /

`calc <data> <expr> <BUFFER>`

Performs a calculation on the data in the buffers.

<data>: The target axis (1 = X values, 2 = Y values, 3 = X error, 4 = Y error)

<expr>: Expression which defines the calculation.

`swapxy <BUFFER>`

This function swaps the X and Y values. Error values will also be swapped if available.

`normx <ref1> <ref2> <new1> <new2> <BUFFER>`

This function manipulates the buffers by stretching or shrinking along the X axis. The specified reference points in the data are changed in such a way that the `ref1` value becomes the `new1` value and the `ref2` value becomes the `new2` value.

`normy <ref1> <ref2> <new1> <new2> <BUFFER>`

This function manipulates the buffers by stretching or shrinking along the Y axis. The specified reference points in the data are changed in such a way that the `ref1` value becomes the `new1` value and the `ref2` value becomes the `new2` value.

`movex <from> <to> <BUFFER>`

This function moves the buffers on the X axis.

`movey <from> <to> <BUFFER>`

This function moves the buffers on the Y axis.

`cutoutsidex <min> <max> <BUFFER>`

These function deletes all data points in the buffers which are not in the X range between `min` and `max`

`cutinsidex <min> <max> <BUFFER>`

These function deletes all data points in the buffers which are in the X range between `min` and `max`

`calcint <min> <max> <BUFFER>`

Integrates the buffers and add the result to the comment field. The result of the last integration is stored in the variable `integral`.

`linreg <min> <max> <BUFFER>`

Perform a linear regression between `min` and `max`. The result of the last regression will be stored in the variables `rega`, `regb` and `regr`.

`logreg <min> <max> <BUFFER>`

Perform a logarithmic regression between `min` and `max`. The result of the last regression will be stored in the variables `rega`, `regb` and `regr`.

`expreg <min> <max> <BUFFER>`

Perform a exponential regression between `min` and `max`. The result of the last regression will be stored in the variables `rega`, `regb` and `regr`.

`linbg <a> <BUFFER>`

Perform a linear background subtraction ($y=y+a+bx$)

`smooth <factor> <BUFFER>`

This functions uses **factor** as the effective smooth width and performs a least square smooth to reduce the noise on the buffer.

fftsmooth <factor> <BUFFER>

This functions uses **factor** as the effective smooth width and performs a FFT based smooth to reduce the noise on the buffer (see [sources](#)).

datareduction <n> <BUFFER>

Sometimes it is useful to reduce a large number of data points which consist mainly of noise to a more practical amount of points. This function takes n points, builds the average, and replaces the n points with the new one.

spline <points> <strength> <BUFFER>

This function calculates a nonparametric cubic spline interpolation (see [sources](#)).

points: Defines the number of points which should be calculated for each interval.

strength: Defines the strength of the interpolation. 2 is in most cases a god choice.

fft <windowing> <BUFFER>

Performs a Fast Fourier Transform and generates a new buffer with the result.

windowing: 1=none, 2=Hanning, 3=Welch, 4=Parzen

diff <interval> <BUFFER>

Differentiate the buffers and generate new buffers with the result data (see [sources](#)).

integrate <BUFFER>

Integrates buffers and generate new buffers with the result data.

setxval <BUFFER> <index> <value>

Set X value at the given index of the buffers.

setyval <BUFFER> <index> <value>

Set Y value at the given index of the buffers.

`setxeval <BUFFER> <index> <value>`

Set X error value at the given index of the buffers.

`setyeval <BUFFER> <index> <value>`

Set Y error value at the given index of the buffers.

Macro Language: Style Commands

`margins <left> <right> <bottom> <top>`

Defines the margins between window and frame.

`window size <width> <height>`

Defines the window size.

`bgcolor <c> <c> <c>`

Defines the background color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`margincolor <c> <c> <c>`

Defines the margin color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`framecolor <c> <c> <c>`

Defines the frame color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`framewidth <width>`

Defines the frame width.

`framestyle <style>`

Defines the frame style. (0=full frame, 1=line at X and Y axis, 2=draw a cross at 0).

`forceframe <bool>`

If enabled a full frame will always be drawn (possible values: yes/no).

`framearrows <bool>`

If enabled arrow will be drawn for frame style 1 and 2.

Macro Language: Data Style Commands

`bstyle <lines> <symblos> <fill> <ybars> <xbar> <ystick> <xstick> <dots>
<hist> <fhist> <BUFFER>`

Enable or disable styles for data buffers. All styles can be combined (possible values: yes/no).

`bstyleclear <BUFFER>`

removes all styles from buffer

`bstylelines <bool> <BUFFER>`

add or remove line style (possible values: yes/no).

`bstylesymbols <bool> <BUFFER>`

add or remove symbol style (possible values: yes/no).

`bstylefill <bool> <BUFFER>`

add or remove filled area style (possible values: yes/no).

`bstyleybars <bool> <BUFFER>`

add or remove Y bar style (possible values: yes/no).

`bstylexbars <bool> <BUFFER>`

add or remove X bars style (possible values: yes/no).

`bstylesticks <bool> <BUFFER>`

add or remove Y stick style (possible values: yes/no).

`bstylexsticks <bool> <BUFFER>`

add or remove X sticks style (possible values: yes/no).

`bstyledots <bool> <BUFFER>`

add or remove dot style (possible values: yes/no).

bstylehist <bool> <BUFFER>

add or remove histogram style (possible values: yes/no).

bstylefhist <bool> <BUFFER>

add or remove filled histogram style (possible values: yes/no).

bcolor <c> <c> <c> <BUFFER>

Defines the color for buffers. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

bfillcolor <c> <c> <c> <BUFFER>

Defines the color for the fill style. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

bframecolor <c> <c> <c> <BUFFER>

Defines the frame color for bars. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

bstickcolor <c> <c> <c> <BUFFER>

Defines the stick color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

blinewidth <linewidth> <BUFFER>

Defines the line width for buffers.

blinepattern <pattern> <BUFFER>

Defines the dash pattern for buffers (0-15).

bpatternwidth <patternwidth> <BUFFER>

Defines the width of dash pattern.

bsymbol <symbol> <BUFFER>

Defines the symbols for buffers (0-15).

bsymbolsize <symbolsize> <BUFFER>

defines the symbol size for buffers.

bsticksiz <sticksize> <BUFFER>

defines the sticksize size for buffers.

bstickabs <bool> <BUFFER>

defines if the sticksize should be an absolute value or not (possible values: yes/no).

bdynsize <BUFFER> [expr]

sets an expression which defines the size of symbols or the length of sticks.

bxerrorval <error> <BUFFER>

Defines the value for X error bars.

bxerror <type> <BUFFER>

Defines the X error bar type (0=off, 1=absolute, 2=relative, 3=form data).

byerrorval <error> <BUFFER>

Defines the value for Y error bars.

byerror <type> <BUFFER>

Defines the Y error bar type (0=off, 1=absolute, 2=relative, 3=form data).

berrorlinewidth <linewidth> <BUFFER>

Defines the line width for error bars.

Macro Language: Axis Style Commands

setaxis <x axis 1> <y axis 1> <x axis 2> <y axis 2>

Enable or disable axis drawing (possible values: yes/no)

axisscalstart <AXIS> <start>

Defines where axis scaling should start.

ascalend <AXIS> <end>

Defines where axis scaling should end.

atickdist <AXIS> <dist>

Defines the distance between two major ticks. If your axis is a time axis you have to enter days:hours:minutes:seconds or w (1 week) or m (1 month) or y (1 year).

aminsteps <AXIS> <steps>

Defines the number of minor ticks between two major ticks.

atimeformat <AXIS> <format>

Defines the format used for the displaying time values on time axis (time format tokens).

atrigger <AXIS> <trigger>

This allow to define a trigger, a number where a tick should appear and all other ticks should depend on.

ascalexpr <AXIS> <expr>

This allows scaling of axis numbers without changing the data itself. For example to display seconds instead of milliseconds enter $v/1000$ in this field. v is the variable which represents the axis number.

aexpoffset <AXIS> <offset>

Allows you to define the limit, above which numbers are plotted in exponential notation. The number you define with the slider is the exponent of the limit.

adecplaces <AXIS> <n>

Defines the number of decimal places for axis numbers.

`adecplaces <AXIS> <bool>`

If enabled the number of decimal places for axis numbers will be determined automatically (possible values: yes/no).

`aformat <AXIS> <format>`

Defines the axis format (0=linear, 1=logarithmic, 2=time).

`asetsmalllogzero <AXIS> <bool>`

If enabled the smallest value on a logarithmic axis will be replaced with 0 (even if this is not correct) (possible values: yes/no).

`atext <AXIS> <text>`

Defines the axis text label.

`anumfont <AXIS> <size>`

Defines the font for axis numbers.

`anumoffset <AXIS> <offset>`

The distance between numbers and frame.

`atextfont <AXIS> <size>`

Defines the font for axis text labels.

`atextoffset <AXIS> <offset>`

The distance between text label and frame.

`asetnum <AXIS> <bool>`

Enable or disable axis numbers (possible values: yes/no).

`asettext <AXIS> <bool>`

Enable or disable axis text labels (possible values: yes/no).

`asetfliptext <AXIS> <bool>`

Enable or disable flipped axis text labels (possible values: yes/no).

`atextcolor <AXIS> <c> <c> <c>`

Defines the color for axis numbers and text. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`asetzeroline <AXIS> <bool>`

Enable or disable a line at the value zero (possible values: yes/no).

`azerolinecolor <AXIS> <c> <c> <c>`

Defines the color for zero lines. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`aticklength <AXIS> <length>`

Defines the tick length.

`aminticklength <AXIS> <length>`

Defines the minor tick length.

`atickwidth <AXIS> <width>`

Defines the tick width.

`amintickwidth <AXIS> <width>`

Defines the minor tick width.

`asetticks <AXIS> <bool>`

Enable or disable axis ticks (possible values: yes/no).

`asetminticks <AXIS> <bool>`

Enable or disable axis minor ticks (possible values: yes/no).

`atickcolor <AXIS> <c> <c> <c>`

Defines the tick color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`amintickcolor <AXIS> <c> <c> <c>`

Defines the minor tick color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`atickstyle <AXIS> <style>`

Defines the tick style (0=inside, 1=outside, 2=both)

`asetgrid <AXIS> <bool>`

Enable or disable the grid (possible values: yes/no).

`asetmingrid <AXIS> <bool>`

Enable or disable the minor grid (possible values: yes/no).

`agridcolor <AXIS> <c> <c> <c>`

Defines the grid color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`amingridcolor <AXIS> <c> <c> <c>`

Defines the minor grid color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`agridwidth <AXIS> <width>`

Defines the grid width.

`amingridwidth <AXIS> <width>`

Defines the minor grid width.

`alabel <AXIS> <number> <replacement>`

Defines a axis number replacement

Macro Language: Legend Commands

`setlegend <bool>`

Enable or disable the legend (possible values: yes/no).

`legendpos <x> <y>`

Defines the legend position.

`setlegendbg <bool>`

Enable or disable legend background (possible values: yes/no).

`legendlength <length>`

Defines the length for legend lines.

`legendspace <space>`

Defines the vertical space between two legend entries.

`legendcolor <c> <c> <c>`

Defines the color for legend text. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`legendbgcolor <c> <c> <c>`

Defines the legend background color. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command colormode (0=RGB, 1=HSB).

`legendfont `

Defines the legend font.

Macro Language: Text & Graphic Commands

`text <tag> <x> <y> <text> [c] [c] [c] [angle] [align]
[pos] [offset] [cb] [cb] [cb]`

Adds a text label to the document.

<tag>: The `tag` argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<x>, <y>: the position in screen coordinates of the text.

****: The font for the text (e.g. *Helvetica-Bold*)

****: The font size of the text.

[c] [c] [c]: The color of the text. Dependent on the `colormode` the three values are RGB or HSB values from 0.0 to 1.0. The `colormode` can be set with the command `colormode` (0=RGB, 1=HSB).

[angle]: The text angle.

[align]: The text alignment (0=left, 1=center, 2=right).

[pos]: The position for automatically positioned text label (0=off, 1-21 fixed text positions).

	10	11	12	
21	1	2	3	13
20	4	5	6	14
19	7	8	9	15
	18	17	16	

[offset]: Offset for automatically positioned text label.

[cb] [cb] [cb]: The color of the text background. Dependent on the `colormode` the three values are RGB or HSB values from 0.0 to 1.0. The `colormode` can be set with the command `colormode` (0=RGB, 1=HSB).

`ftext <tag> <C00R> <x> <y> <text> [c] [c] [c] [angle]
[align] [tbalign] [cb] [cb] [cb]`

Adds a floating text label to the document.

<tag>: The `tag` argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the rectangle belongs.

<x>, <y>: the position in screen coordinates of the text.

``: The font for the text (e.g. *Helvetica-Bold*)

``: The font size of the text.

`[c] [c] [c]`: The color of the text. Dependent on the `colormode` the three values are RGB or HSB values from 0.0 to 1.0. The `colormode` can be set with the command `colormode` (0=RGB, 1=HSB).

`[angle]`: The text angle.

`[align]`: The text alignment (0=left, 1=center, 2=right).

`[tbalign]`: The textbox alignment.

1	2	3
4	5	6
7	8	9

`[pos]`: The position for automatically positioned text label (0=off, 1-21 fixed text positions).

	10	11	12	
21	1	2	3	13
20	4	5	6	14
19	7	8	9	15
	18	17	16	

`[offset]`: Offset for automatically positioned text label.

`[cb] [cb] [cb]`: The color of the text background. Dependent on the `colormode` the three values are RGB or HSB values from 0.0 to 1.0. The `colormode` can be set with the command `colormode` (0=RGB, 1=HSB).

`qtext <C00R> <x> <y> <text> [angle] [align] [tbalign]`

Adds a text label to the document. This command uses the default color and the default font. The generated text label has the tag 999.

`<C00R>`: The coordinate system to which the rectangle belongs.

`<x>`, `<y>`: the position in screen coordinates of the text.

`[angle]`: The text angle.

`[align]`: The text alignment (0=left, 1=center, 2=right).

`[tbalign]`: The textbox alignment.

1	2	3
4	5	6
7	8	9

```
rect <tag> <C00R> <x> <y> <w> <h> [linewidth] [dash] [c] [c] [c]
```

Draw a rectangle with the given attributes.

<tag>: The **tag** argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the rectangle belongs.

<x>, <y>: The position of the rectangle in the defined coordinates system.

<w>, <h>: The size of the rectangle in the defined coordinates system.

[linewidth]: The rectangles line width.

[dash]: The dash pattern for the rectangle (0-15).

[c] [c] [c]: The color of the rectangle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command **colormode** (0=RGB, 1=HSB).

```
frext <tag> <C00R> <x> <y> <w> <h> [linewidth] [dash] [cf] [cf] [cf]
```

Draw a filled rectangle with the given attributes.

<tag>: The **tag** argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the rectangle belongs.

<x>, <y>: The position of the rectangle in the defined coordinates system.

<w>, <h>: The size of the rectangle in the defined coordinates system.

[linewidth]: The rectangles line width.

[dash]: The dash pattern for the rectangle (0-15).

[c] [c] [c]: The color of the rectangle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command **colormode** (0=RGB, 1=HSB).

[cf] [cf] [cf]: The fill color of the rectangle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command **colormode** (0=RGB, 1=HSB).

`circle <tag> <C00R> <x> <y> <w> <h> [linewidth] [dash] [c] [c] [c]`

Draw a circle with the given attributes.

<tag>: The **tag** argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the circle belongs.

<x>, <y>: The position of the circle in the defined coordinates system.

<w>, <h>: The size of the circle in the defined coordinates system.

[linewidth]: The circles line width.

[dash]: The dash pattern for the circle (0-15).

[c] [c] [c]: The color of the circle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command **colormode** (0=RGB, 1=HSB).

`fcircle <tag> <C00R> <x> <y> <w> <h> [linewidth] [dash] [cf] [cf] [cf]`

Draw a filled circle with the given attributes.

<tag>: The **tag** argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the circle belongs.

<x>, <y>: The position of the circle in the defined coordinates system.

<w>, <h>: The size of the circle in the defined coordinates system.

[linewidth]: The circles line width.

[dash]: The dash pattern for the circle (0-15).

[c] [c] [c]: The color of the circle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command **colormode** (0=RGB, 1=HSB).

[cf] [cf] [cf]: The fill color of the circle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command **colormode** (0=RGB, 1=HSB).

`arrow <tag> <C00R> <x1> <y1> <x2> <y2> <a1> <a2> <at> <as> [linewidth]
[dash] [c] [c] [c]`

Draw an arrow with the given attributes.

<tag>: The **tag** argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the arrow belongs.

<x1>, <y1>, <x2>, <y2>: The start and end position of the arrow in the defined coordinates system.

<a1>, <a2>: Draw arrow at line start and/or end (possible values: yes/no)

<at>: The arrow type (0-3).

<as>: The arrow size.

[linewidth]: The arrow line width.

[dash]: The dash pattern for the arrow (0-15).

[c] [c] [c]: The color of the arrow. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command `colormode` (0=RGB, 1=HSB).

`line <tag> <C00R> <x1> <y1> <x2> <y2> [linewidth] [dash] [c] [c] [c]`

Draw a line with the given attributes.

<tag>: The `tag` argument should be an integer value which can be used later to address a graphic or text object. Interactively created text or graphic objects always have the tag 0.

<C00R>: The coordinate system to which the line belongs.

<x1>, <y1>, <x2>, <y2>: The start and end position of the line in the defined coordinates system.

[linewidth]: The line width.

[dash]: The dash pattern for the line (0-15).

[c] [c] [c]: The color of the line. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command `colormode` (0=RGB, 1=HSB).

`rmtext <tag>`

Remove text object with the tag `tag`.

`rmgraphic <tag>`

Remove graphic object with the tag `tag`.

`defaultfont `

Defines a default font for all text and graphic commands.

`defaultcolor <c> <c> <c>`

Defines a default color for all text and graphic commands.

`<c> <c> <c>`: The color of the circle. Dependent on the colormode the three values are RGB or HSB values from 0.0 to 1.0. The colormode can be set with the command `colormode` (0=RGB, 1=HSB).

Macro Language: Document Commands

new

Opens a new document.

open <filename>

Opens the document with the given filename.

save

Save the current document.

printdoc

Print the current document.

import <filename> <mode> [separator] [xcol] [noxcol] [comment]

Import an ASCII file

<mode>: Import mode (0=multicolumn, 1=error values).

[separator]: Defines the field separator for ASCII import.

[xcol]: Defines the X column starting with 1 as the first column.

[noxcol]: If true all columns will be interpreted a Y values and the X values will be generated as sequence number.

[comment]: Overwrites the default comment for the imported buffers.

binaryimport <filename> <type> <mode> <swap> <columns> [comment]

Performs a binary import of data.

<type>: Defines the data type (0=double (64bit), 1=float (32bit), 2=integer (32 bit), 3=integer (16 bit)).

<mode>: Defines the import mode (0=column mode, 1=block mode).

<swap>: Enables or disables byte order swapping.

<columns>: The number of columns to import.

`mysqlimport <host> <db> <user> <password> <port> <sql>`

Import data from mysql database.

`export <file> <BUFFER>`

Export buffers as ASCII file.

Macro Language: Miscellaneous Commands

`colormode <mode>`

With this command you can define if colors will be defined as RGB or HSB (0=RGB, 1=HSB).

`clr`

Clears the console output.

`setvar <var> <expr>`
`<var>=<expr>`

Defines a variable.

`setstring <var> <value>`
`$<var>=<expr>`

Defines a string variable.

`eval <name> <expr>`

Evaluates the expression and defines the result as string variable.

`format <name> <format> <expr> ... <expr>`

Returns a string created with the given format string. The format can be defined in the usual `printf` notation.

Macro Quick Reference

Range Commands

```
range <AXIS> <min> <max>
rescale [AXISGROUP]
upscale <AXISGROUP> [factor]
downscale <AXISGROUP> [factor]
```

Buffer Commands

```
select <BUFFER>
setcurrent <number>
invertselections
hide <BUFFER>
show <BUFFER>
delete <BUFFER>
duplicate <BUFFER>
setcomment <BUFFER> <text>
setsource <BUFFER> <text>
addcomment <BUFFER> <text>
addsource <BUFFER> <text>
belongx <BUFFER> <axis>
belongy <BUFFER> <axis>
update
addvalue <buffer> <x value> <y
value> <x error> <y error>
setxvalue <buffer> <index> <value>
setxerror <buffer> <index> <value>
setyvalue <buffer> <index> <value>
setyerror <buffer> <index> <value>
rmvalue <buffer> <index>
```

Control Commands

```
if <expr> <operator> <expr>
else
endif
loop <RANGE>
bufferloop <BUFFER>
endloop
```

Interactive Commands

```
ask <text>
askoption <text> <button1> <button0>
browse
input
plot
print <argument> ...
sleep <seconds>
```

Data Style Commands

```
bstyle <lines> <symblos> <fill>
<ybars> <xbar> <ystick> <xstick>
<dots> <hist> <fhist> <BUFFER>
bstyleclear <BUFFER>
bstylelines <bool> <BUFFER>
bstylesymbols <bool> <BUFFER>
bstylefill <bool> <BUFFER>
bstyleybars <bool> <BUFFER>
bstylexbars <bool> <BUFFER>
bstyleysticks <bool> <BUFFER>
bstylexsticks <bool> <BUFFER>
bstyledots <bool> <BUFFER>
bstylehist <bool> <BUFFER>
bstylefhist <bool> <BUFFER>
bcolor <c> <c> <c> <BUFFER>
bfillcolor <c> <c> <c> <BUFFER>
bframecolor <c> <c> <c> <BUFFER>
bstickcolor <c> <c> <c> <BUFFER>
blinewidth <linewidth> <BUFFER>
blinepattern <pattern> <BUFFER>
bpatternwidth <patternwidth>
<BUFFER>
bsymbol <symbol> <BUFFER>
bsymbolsize <symbolsize> <BUFFER>
bsticksizesize <sticksizesize> <BUFFER>
bstickabs <bool> <BUFFER>
bdynsize <BUFFER> [expr]
bxerrorval <error> <BUFFER>
bxerror <type> <BUFFER>
byerrorval <error> <BUFFER>
byerror <type> <BUFFER>
berrorlinewidth <linewidth> <BUFFER>
```

Axis Style Commands

```
setaxis <x axis 1> <y axis 1> <x
axis 2> <y axis 2>
axisscalstart <AXIS> <start>
ascalend <AXIS> <end>
atickdist <AXIS> <dist>
aminsteps <AXIS> <steps>
atimeformat <AXIS> <format>
atrigger <AXIS> <trigger>
ascalexpr <AXIS> <expr>
aexpoffset <AXIS> <offset>
adecplaces <AXIS> <n>
adecplaces <AXIS> <bool>
```

Calculation Commands

fg <axis> <min> <max> <steps> <expr>
bcalc <buffer1> <operator> <buffer2>
calc <data> <expr> <BUFFER>
swapxy <BUFFER>
normx <ref1> <ref2> <new1> <new2>
<BUFFER>
normy <ref1> <ref2> <new1> <new2>
<BUFFER>
movex <from> <to> <BUFFER>
movey <from> <to> <BUFFER>
cutoutsidex <min> <max> <BUFFER>
cutinsidex <min> <max> <BUFFER>
calcint <min> <max> <BUFFER>
linreg <min> <max> <BUFFER>
logreg <min> <max> <BUFFER>
expreg <min> <max> <BUFFER>
linbg <a> <BUFFER>
smooth <factor> <BUFFER>
fftsmooth <factor> <BUFFER>
datareduction <n> <BUFFER>
spline <points> <strength> <BUFFER>
fft <>windowing> <BUFFER>
diff <interval> <BUFFER>
integrate <BUFFER>
setxval <BUFFER> <index> <value>
setyval <BUFFER> <index> <value>
setxval <BUFFER> <index> <value>
setyval <BUFFER> <index> <value>

Documents Commands

new
open <filename>
save
printdoc
import <filename> <mode> [separator]
[xcol] [noxcol] [comment]
binaryimport <filename> <type>
<mode> <swap> <columns> [comment]
mysqlimport <host> <db> <user>
<password> <port> <sql>
export <file> <BUFFER>

Miscellaneous Commands

colormode <mode>
clr
setvar <var> <expr>
<var>=<expr>
setstring <var> <value>

aformat <AXIS> <format>
asetsmalllogzero <AXIS> <bool>
atext <AXIS> <text>
anumfont <AXIS> <size>
anumoffset <AXIS> <offset>
atextfont <AXIS> <size>
atextoffset <AXIS> <offset>
asetnum <AXIS> <bool>
asettext <AXIS> <bool>
asetfliptext <AXIS> <bool>
atextcolor <AXIS> <c> <c> <c>
asetzeroline <AXIS> <bool>
azerolinecolor <AXIS> <c> <c> <c>
aticklength <AXIS> <length>
aminticklength <AXIS> <length>
atickwidth <AXIS> <width>
amintickwidth <AXIS> <width>
asetticks <AXIS> <bool>
asetminticks <AXIS> <bool>
atickcolor <AXIS> <c> <c> <c>
amintickcolor <AXIS> <c> <c> <c>
atickstyle <AXIS> <style>
asetgrid <AXIS> <bool>
asetmingrid <AXIS> <bool>
agridcolor <AXIS> <c> <c> <c>
amingridcolor <AXIS> <c> <c> <c>
agridwidth <AXIS> <width>
amingridwidth <AXIS> <width>
alabel <AXIS> <number> <replacement>

Legend Commands

setlegend <bool>
legendpos <x> <y>
setlegendbg <bool>
legendlength <length>
legendspace <space>
legendcolor <c> <c> <c>
legendbgcolor <c> <c> <c>
legendfont

Text & Graphic Commands

text <tag> <x> <y> <font
size> <text> [c] [c] [c] [angle]
[align] [pos] [offset] [cb] [cb]
[cb]
ftext <tag> <COORD> <x> <y>
 <text> [c] [c] [c]
[angle] [align] [tbaligned] [cb] [cb]
[cb]
qtext <COORD> <x> <y> <text> [angle]


```

$<var>=<expr>
eval <name> <expr>
format <name> <format> <expr> ...
<expr>

```

Style Commands

```

margins <left> <right> <bottom>
<top>
windowsize <width> <height>
bgcolor <c> <c> <c>
margincolor <c> <c> <c>
framecolor <c> <c> <c>
framewidth <width>
framestyle <style>
forceframe <bool>
framearrows <bool>

```

```

[align] [tbalign]
rect <tag> <C00R> <x> <y> <w> <h>
[linewidth] [dash] [c] [c] [c]
frect <tag> <C00R> <x> <y> <w> <h>
[linewidth] [dash] [cf] [cf] [cf]
circle <tag> <C00R> <x> <y> <w> <h>
[linewidth] [dash] [c] [c] [c]
fcircle <tag> <C00R> <x> <y> <w> <h>
[linewidth] [dash] [cf] [cf] [cf]
arrow <tag> <C00R> <x1> <y1> <x2>
<y2> <a1> <a2> <at> <as> [linewidth]
[dash] [c] [c] [c]
line <tag> <C00R> <x1> <y1> <x2>
<y2> [linewidth] [dash] [c] [c] [c]
rmtext <tag>
rmgraphic <tag>
defaultfont <font> <font size>
defaultcolor <c> <c> <c>

```

About Plot

History

A long time ago in the last millennium I wrote a program called *SciPlot* for the fantastic NeXT computers (the older ones of you maybe remember).

After the NeXT period I had the idea to make a similar program for windows computers. After some work on this I did not find any fun in programming for Windows computers and so I stopped the development of *SciPlot* for Windows.

Now we have Mac OS X and I started the *Plot* project which takes some of the ideas of *SciPlot* but is a completely rewritten program with a different focus than *SciPlot*.

Sources

Spline Interpolation

Formelsammlung zur Numerischen Mathematik mit C-Programmen
G.Engeln-Muellges and F.Reuter

Smooth

based on least square fit
A.Savitzky & M.J.E. Golay
Anal. Chem. 1964, 36,1627-1639

FFT (Fast Fourier Transform)

Numerical Recipes in C
W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling
Cambridge University Press

FFT smooth & Differentiation

Numerical Recipes in C
W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling
Cambridge University Press

Curve Fit

Numerical Recipes in C
W.H.Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling
Cambridge University Press
(implemented by *W.Mahdi*)

Gauss-Lorentz R.O. Ansell, T. Dickinson, A.F. Povey, and P.M.A. Sherwood
J.Electroanal.Chem. 98:79-89 (1979)

Doniach-Sunjic
S. Doniach, and M. Sunjic,
J.Phys. [C] 3:285-291 (1970)

Exponential Tail
D.Briggs and M.P.Seah
Practical Surface Analysis by Auger and X-Ray Photoelectron Spectroscopy

About the author

Nothing to say about me but you are invited to send [email](#) and also to visit also my [pets page](#).

Warranty

Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, wie es ist, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich aber nicht begrenzt auf Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich aber nicht beschränkt auf Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.