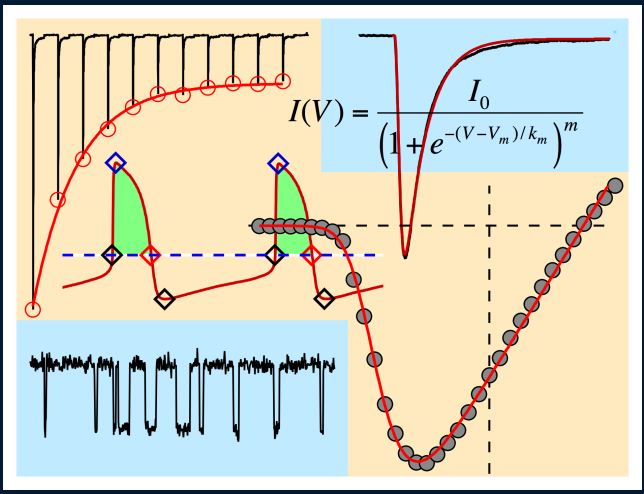
Manual 2.69

FITMASTER

Analysis software



HEKA

HEKA Elektronik
Dr. Schulze GmbH
Wiesenstrasse 71
D-67466 Lambrecht/Pfalz
Germany

Phone +49 (0) 6325 / 95 53-0
Fax +49 (0) 6325 / 95 53-50
Web Site www.heka.com
Email sales@heka.com
support@heka.com

HEKA Electronics Inc.
643 Highway #14
R.R. #2
Chester, NS B0J 1J0
Canada

Phone +1 902 624 0606
Fax +1 902 624 0310
Web Site www.heka.com
Email nasales@heka.com
support@heka.com

HEKA Instruments Inc.
2128 Bellmore Avenue
Bellmore, New York 11710-5606
USA

Phone +1 516 882-1155
Fax +1 516 467-3125
Web Site www.heka.com
Email ussales@heka.com
support@heka.com

COFIMA/7

Contents

Chapter 1

Introduction

1.1 Disclaimer

This `Product` relies on the tools of Microsoft Windows 2000, Windows XP, Windows Vista or Mac OS. HEKA is not responsible for the contents of these third-party products, any links contained in these third-party products and changes or upgrades to these third-party products for any consequential damages resulting from the use of these products.

This `Product` may not be reverse engineered, decompiled or disassembled without the express written consent of HEKA.

In no event shall HEKA be responsible for any incidental, punitive, indirect, or consequential damages whatsoever, (including but not limited to loss of data, privacy of data or other pecuniary loss), arising from or relating to the use, or the inability to use, this `Product` or the provision, or lack of provision, of support services.

In all cases where HEKA is liable, the extend of HEKA's liability shall be limited to the actual cost of the `Product` or to the provision of a replacement version of the `Product`.

1.2 Scope of the Program

The FITMASTER program provides tools for the analysis of electrophysiological data acquired with PATCHMASTER. It operates on two levels: on the first level, it analyzes raw data traces by fitting specific functions to them. This level is called `TraceFit`. On the second level, parameters obtained

with a `TraceFit` procedure or the `Online Analysis` are then further analyzed. This level is called `SeriesFit`.

Other programs expanding the possibilities of FITMASTER are available from HEKA. Using an export filter in FITMASTER creates data that can be interpreted by the following programs:

- **PULSETOOLS** allows editing of data, special-purpose leak subtraction or non-stationary noise analysis.
- **PULSESIM** provides setup, solution and optimization of kinetic schemes.

1.3 FITMASTER for PULSEFIT Users

This section is intended for those users familiar with the PULSEFIT data analysis software. We will quickly summarize the most relevant differences between PULSEFIT and FITMASTER. For detailed description of all functions please refer to the FITMASTER reference manual.

1.3.1 Major Changes

Although in some respects FITMASTER 'looks' similar to PULSEFIT, it is a completely new program, following an improved strategy for process handling and programming. Therefore, sometimes the PULSEFIT user may miss some typical PULSEFIT behavior of the program until it becomes clear what the benefits of the new features are. Obviously, a substantially increased realm of functions and flexibility comes at some price. In most cases, this 'price' is to set some definitions before using FITMASTER in order to customize the program according to the individual needs.

Fitmaster uses `OnlineAnalysis` results.

Unlike PULSEFIT, FITMASTER supports a flexible data format for the analyses on the level of a Sweep. As a consequence, the user has to decide which parameters to be stored in the analysis data structure. Among the available parameters are the results of the selected fit function and all parameters from the `OnlineAnalysis` for which Notebook=on was selected.

Let's suppose the user wants to determine a time constant for Sweeps of a Series with varying voltage, the selected `OnlineAnalysis` method should determine the 'amplitude', i.e. voltage of the relevant X-segment. Parameters not stored at the point of Sweep analysis cannot be retrieved later on. The advantage of this feature is that the user can now select parameters from the long list of functions provided by the `OnlineAnalysis`, e.g. also including complex expressions.

`TraceFit`: Multiple analyses supported.

FITMASTER supports analysis (and storage) with multiple functions for a given Series. I.e. one can now analyze current in different segments without deleting the previous results.

Fit functions.

Mathematical functions provided by `TraceFit` and `SeriesFit` are now extended. The functions can now be displayed, providing information about the underlying math and the status of the fit parameters, i.e. whether a parameter is constant or variable. The variable parameters are identified according to whether they are linear or non-linear parameters. This is helpful, because for non-linear parameters good starting values are needed, while for linear parameters no starting values are necessary. Additional features are: selection of parameters, in particular constraining parameters is improved, Boltzmann function has two components, there is a new function called 'General Exponential'. This is the sum of three exponential of the form $(1 - exp)^n or (exp)^n$ with variable time zero. Gaussian functions are now included in `SeriesFit`. In addition, arbitrary functions can be defined by the user; they are interpreted with a built-in parser.

Cursors.

Cursors are directly set from the `TraceFit` window. For the purpose of semi-automatic analysis, the cursor bounds can also be retrieved from a selected `OnlineAnalysis` method.

Zoom for fit.

When executing data fits in `TraceFit`, data sections of interest can be zoomed automatically to provide an optimal view necessary to decide upon the quality of the fit. This feature is very helpful if some segments of a Series do have variable durations or if multiple analyses in different segments are

to the performed using the cyclic analysis feature (see below).

Cyclic analysis.

In several cases the same type of analysis is to be executed for one data trace, but offset by a fixed number of pulse segments. FITMASTER now provides such cyclic analyses. I.e. multiple analyses are performed on a data trace. The resulting parameters are stored as 'Events' in the data file. This feature is very helpful for analyzing data obtained as response to a tetanus stimulation or cyclic photometric stimulation (e.g. for Fura-2 measurements).

`SeriesFit`: data selection.

`SeriesFit` has changed considerably. Data are retrieved from the analysis file; all parameters can be used as X- or as Y-data. Available data are the parameters selected from the fit results provided by `TraceFit` and the results taken from the `OnlineAnalysis`. Such results can either be single results per Trace or an arbitrary list of 'events' associated to such results. I.e., `SeriesFit` can analyze conventional Sweep-oriented `TraceFit` data as well as event tables. These selected data are then copied into a working buffer. This working buffer has no direct logical link to the Series anymore. Therefore, analysis results from multiple Series can be copied into the same working buffer before Series analysis. This method of data handling provides two important features:

Append results:

TraceFit results from one Series can be appended to the results of another Series. This allows merging of multiple Series into one data structure to be analyzed. Example: the user recorded a Series 'A' consisting of a number of identical Sweeps. This Series was then interrupted because, for example, the amplifier gain had to be adjusted. Then 'A' is started again to continue the experiment. For analysis the TraceFit results are to be analyzed as a function of the real-time. In PULSEFIT this was not possible because the data came from two independent Series. In FITMASTER we append the results from both Series into the same work buffer. Another example is the recording of current-voltage relationships with non-evenly distributed step voltages. As illustrated in the figure below, current-voltage relationships can be generated such that in the region of steepest voltage dependence there is a higher density of pulses. Figure 1.3.1 shows two ways of gener-

ating such IVs: A) Append IVs with different increment intervals of the varying voltage segment. B) Append two IVs that are partially interleaved.

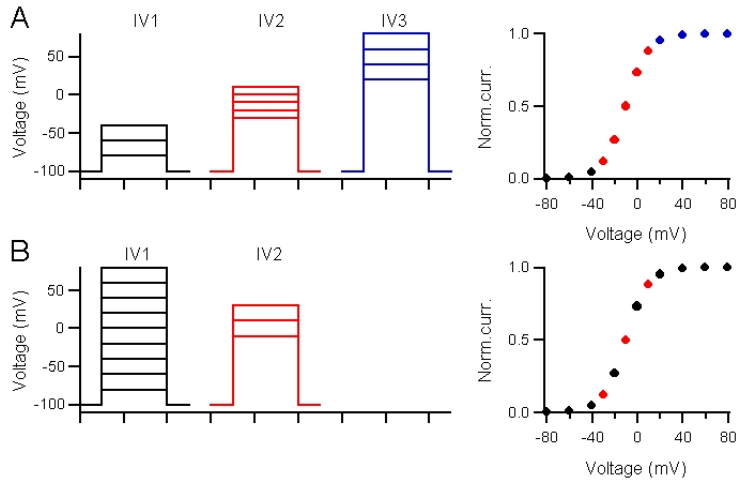


Figure 1.1: Generation and analysis of current-voltage relationships with non-equal voltage intervals. A) Three IV series (IV1 through IV3) are recorded in succession. IV2 has a smaller voltage increment interval than IV1 and IV3. The results are then analyzed using a `TraceFit` function (e.g. peak determination of `OnlineAnalysis`). Then, inside `SeriesFit`, the results, i.e. PeakCurrent versus Voltage, are appended to the work buffer, yielding a current-voltage relationship shown on the right. This can now be analyzed using functions provided by `SeriesFit`, e.g. 'Current-Voltage'. The results of this analysis are then stored at the level of the last active Series. In our case this would mean that the results are associated to IV3. B) Here is another solution to the same problem. Two Series are recorded where IV2 'fills the gaps' as it is partially interleaving to IV1. Upon analysis, this looks like the plot on the right, where the black data points come from IV1 and the red ones from IV2. Inside the work buffer the data can be sorted according to the X-argument by pressing the 'X-sort' button. In our case this will sort the data in the buffer in an increasing order according to the voltage. This feature is also helpful for data from Series with alternating segment increments.

Average results:

In addition to appending `TraceFit` results to the analysis work buffer, FITMASTER also allows to process data with identical X-references. With this feature, called 'Y-math', one can form the average (accumulate) of multiple results. Such averaged data are then shown with or without error bars and a function can be fit to the mean data. Error bars can be one of the following: Min/Max, SD, SEM. In addition, 'Y-math' allows to perform some basic mathematical operations '+', '-', '*', '/' between the incoming data and the data in the work buffer.

1.3.2 Additional features

Work buffer: The work buffer in `SeriesFit` can be loaded with data provided by the TraceFit/`OnlineAnalysis` procedure. This selection can be a simple Copy operation to the buffer. In addition, data can be appended as discussed above. When the number of data points in a Series are identical to the one already stored in the work buffer, standard mathematical function can be applied: +, -, *, /. The contents of the work buffer can be subjected to further math operations ('exp', 'ln', 'square', 'sqrt') and constants, provided by the user can be included (+, -, *, /).

At any time the mean and SD of the entire work buffer are shown at the bottom of the columns.

Sort: Data can be sorted according to their X- or Y-values.

Skipped data points: Data points from the work buffer can be omitted from analysis. This is done by deselecting individual data points in the buffer. The 'skip' flag is preset by the `TraceFit` analysis; it is only on when the corresponding parameter was successfully determined beforehand. Toggling of the skip status can be performed by clicking on the data points in the graph. In addition, their values optionally can be shown in the notebook.

Overlay, background traces: The content of the working buffer can be sent into a background buffer. Up to 16 data sets can be stored to be shown in the background as reference. For the background traces it is not necessary to have the same number of data points.

Export: Results of the `SeriesFit` can be exported into ASCII and IgorPro format. The export mode can now be configured in FITMASTER as to optimize output. Features are: Export Skipped Data, Export Background Data, Export Error Bars, Export Fit, Export Selected Fit Results.

Clipboard: Data in `TraceFit` as well as in `SeriesFit` can now be sent to the clipboard with a single mouse click or a key. In this way FITMASTER data can be transferred to other programs easily.

Logbook: All user operations performed in `TraceFit` and `SeriesFit` can be output to the notebook as to keep track of the analysis history.

Fit status: The status of the fitting procedure is now displayed: 'fitting', 'converged', 'max. iterations', 'break'.

Simplex fit: Unlike in PULSEFIT, in FITMASTER the settings for the simplex fit routine can be set separately for `TraceFit` and `SeriesFit`. After completion of a fit the residual is stored to the file. The residual can be used as parameter for the next level and can be exported as indicator for the faithfulness of the fit.

`SeriesFit` graph: This graph can now be scaled and modified in many ways such as changing the colors, tics, scaling of the axes etc. These features are accessed via 'Scale Graph'. In addition, the axis range can now be defined as 'fixed' or 'auto' for minimum and maximum as well as X- and Y-axis separately. The scale range in the 'auto' mode can be based either on all data points or on the active, i.e. non-skipped, data points only. The fit function is shown in the graph; the bounds of the fit function (min/max X-values) can be specified as either the min/max data points or the full range of the graph, e.g. when fixed X-scaling is used. The data in the graph can be offset or normalized easily. The main data can be shown as histograms instead of markers.

1.4 Supported System Software

FITMASTER is supported on all recent Windows Versions: Windows 98, Windows ME, Windows NT 4.0, Windows 2000 and Windows XP.

FITMASTER is supported on Mac OS Versions between 8.0 and X.

1.5 Naming Conventions

1.5.1 Windows versions

Throughout this manual we will address all the supported Windows versions as "Windows". We will explicitly mention the particular Windows versions when required.

1.5.2 Syntax

- FITMASTER: Capital italic letters denote a HEKA program.
- F9: Small Capitals denote keys on the keyboard.
- `Tree`: Denotes menu entries, options and buttons.
- *Italic*: Window names, Chapter names and Emphasis.
- Bullet list: Lists points.
- Numbered list: Lists actions to be performed sequentially.

 &  or (Windows) & (Mac) denotes system-specific keys and actions.

1.6 Windows and Mac Key Conventions

In FITMASTER, all key commands are saved in the file `FitMaster.key` and will be read at program start. Currently there are no FITMASTER specific keys predefined. For a description how keys can be defined by the user please refer to the section "Modifying Dialogs and Controls" in the chapter "User Interface" of the PATCHMASTER Reference Manual.

Note that all key commands specified in this manual refer to the default setting in `FitMaster.key`. Since all commands can be customized by the user, the settings in your working version of FITMASTER might differ from these default settings.

> ***Note:*** *Please be aware that in case the file* ***`FitMaster.key`*** *is not available at program start, no key commands are available!*

The basic conventions for the system keys on Windows and Mac are as follows:

CTRL stands for the CONTROL key.
ALT stands for the ALT key.

CTRL stands for the CONTROL key.
CMD stands for the COMMAND key (apple or cloverleaf symbol).
ALT stands for the OPTION key.

1.7 User Interface

The user interface of FITMASTER is essentially the same as in PATCHMASTER. For details please refer to the PATCHMASTER reference manual. Windows and functions that are unique for FITMASTER will be described in this manual.

1.8 Installing FITMASTER

Please install the hardware and the software according to the `Installation Guide` that will be delivered with the amplifier and that is be obtained from HEKA.

1.9 Starting FITMASTER

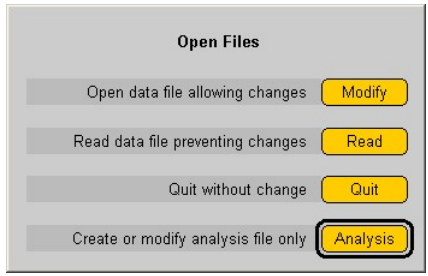
Upon clicking on FITMASTER the software starts and various controls become available:

- The drop-down menus (`File, Edit, Windows, Replay, Display, Buffer, Notebook, Protocols, Help`).
- In addition to the PATCHMASTER windows (Control Window, Configuration, Oscilloscope, Replay, Protocol Editor, Pulse Generator, Online Analysis, Online Windows 1 and 2, Parameters, I/O Control, LockIn, Spectroscopy, Solution Base, Markers, Calculator) the FITMASTER `FitConfiguration`, `TraceFit`, and `SeriesFit` window will come up.
- A scrolling text window (`Notebook`).

Most of these windows can be iconized; they can be re-opened by clicking on the icon or by selecting them in the drop-down menu `Windows`.

All open windows are updated whenever the program encounters some changes; therefore, program execution can be accelerated by keeping open the needed windows only.

All program settings are stored in a file called `.set`. If such a file is not available, FITMASTER may ask for hardware specifications upon startup. Even if no hardware is connected, it is important to specify the hardware used to generate the data file to be analyzed. Only if the hardware was specified correctly, the program can access all hardware-specific information in the data file properly. Thus, if you plan to analyze data from various sources, such as EPC-10 and iTEV 90, you are advised to generate separate `.set` files.



Upon startup, an `Open Files` dialog will appear to ask for the mode of file opening. The options are:

- Modify: Open data file and allow modifying the data.
- Read: Open data file in a read-only mode. Neither the data file nor the analysis file are modified.
- Quit: Exit the program.
- Analysis: Open the data file in a read-only mode, but allow generation or changes of the analysis file.

1.10 Get Online Help on Keys

The option `Show Keys` in the drop-down menu `Help` displays the shortcut keys that are assigned to various controls of the active windows.

By selecting `List Keys` in the drop-down menu `Help` you can list all keys in the `Notebook` window. This is useful, e.g., if you want to print the complete list.

For a description of how keys can be defined by the user, please refer to the section "Modifying Dialogs and Controls" in the chapter "User Interface" of the PATCHMASTER Reference Manual.

1.11 Closing FITMASTER

To exit from FITMASTER, do this:

 Choose `Quit` from the menu `File` or press CTRL + Q.

 press CMD + Q.

You have three options:

- **Save + Exit:** Saves configuration and data files and quits program.
- **Exit:** Saves data files and quits program.
- **Cancel:** Exit process will be aborted, you return to the program.

Note: *If you changed the Protocol Editor, the Pulse Generator pool, or the Online Analysis, you will be asked independently if you want to save them.*

Note: *Data files are always saved.*

1.12 Support Hotline

If you have any question, suggestion, or improvement, please contact HEKA. The best way is to sent us an e-mail or a fax specifying:

- **Your postal and e-mail address (or fax number)**
- **The program name:** FITMASTER
- **The program version number:** v1.12, v2.66, etc.
- **Your operating system and its version:** Mac OS X (10.6), Windows XP, etc.
- **Your type of computer:** Mac G5, Pentium IV, etc.

- **Your acquisition hardware, if applicable:** ITC–16, ITC–18, LIH 1600
- **Your amplifier, if applicable:** EPC 10 USB, EPC 10 Double, Axon 200B, iTEV 90, etc.
- **The serial number and version of your amplifier, if applicable:** EPC 9 single, version "920552 D"
- **The questions, problems, or suggestions you have.**
- **Under which conditions and how often the problem occurs.**

We will address the problem as soon as possible.

HEKA Elektronik	phone:	+49 (0) 6325 9553 0
Wiesenstrasse 71	fax:	+49 (0) 6325 9553 50
D-67466 Lambrecht/Pfalz	e-mail:	support@heka.com
Germany	web:	http://www.heka.com

Chapter 2

Menus

The following section describes the various drop-down menus in FITMASTER.

2.1 File Menu

The `File` menu is slightly different to the menu in PATCHMASTER. Instead of `New` there is the menu entry `Modify Analysis`. With this option one opens a data file in the read-only mode, but allows for the generation and/or alteration of an analysis file. This is the typical way how one opens files when performing data analysis with FITMASTER.

2.2 Edit Menu

The `Edit` menu is the same as in PATCHMASTER.

2.3 Windows Menu

The `Windows` menu applies to the windows in FITMASTER. Windows of PATCHMASTER accessing the hardware are disabled (e.g., Amplifier, Spectroscopy). Additional windows are described here:

Fit Configuration: Opens the `FitConfiguration` window or brings it to the front (see chapter *Fit Configuration*, page 19).

Trace Fit: Opens the `TraceFit` window or brings it to the front (see chapter *Trace Fit*, page 31).

Series Fit: Opens the `SeriesFit` window or brings it to the front (see chapter *Series Fit*, page 43).

2.4 Replay Menu

The drop-down menu `Replay` is the same as in PATCHMASTER.

2.5 Display Menu

This menu is the same as in PATCHMASTER.

2.6 Buffer Menu

This menu is the same as in PATCHMASTER.

2.7 Notebook Menu

The `Notebook` is the same as in PATCHMASTER.

2.8 Protocols Menu

The drop-down menu `Protocols` is the same as in PATCHMASTER.

2.9 Help Menu

This menu is the same as in PATCHMASTER.

Chapter 3

Configuration Window

Settings like sources for external parameters, default values, display settings, colors, fonts, default files, etc. can be edited in the `Configuration` window. To access the `Configuration` window select the drop-down menu Windows Configuration. These and other settings can be stored in `.set` files; by default this file is `FitMaster.set`. By means of different `*.set` files every user can define her/his individual program layout to meet the specific requirements. The window operates in the same way as in PATCHMASTER.

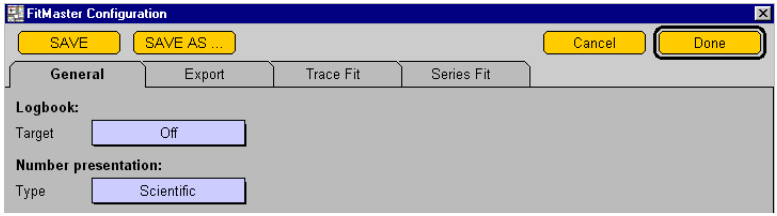
Additional program settings (see FitConfiguration) used by FITMASTER are also stored in the `FitMaster.set` file.

Chapter 4

Fit Configuration Window

Settings used by FITMASTER are defined in this dialog. The settings are stored together with the PATCHMASTER settings in the `FitMaster.set` file. By means of different `*.set` files every user can define her/his individual program layout to meet the specific requirements.

The `FitConfiguration` window provides 4 different panes containing `General`, `Export`, `Trace Fit`, and `Series Fit` configurations.



Save: Saves the configuration to the `*.set` file. Note that this also saves all settings provided in the `Configuration` window.

Save As: Saves the configuration under a user-defined name.

Cancel: Exits the dialog without change.

Done: Sets the new settings and exits the dialog.

4.1 General Settings

Logbook: With this setting one can activate a logbook output to the notebook, a file, or to both for all operations the user performs in the `TraceFit` and `SeriesFit` dialogs.

The logbook file has the same name as the data file and the extension `*.log`. For every data file a separate logbook file will be created. Each time a log file is opened and closed, an identifier with time is written to indicate when the analysis was started or ended. Log output is always appended to the log file.

Logbook entries can be used to keep track of the analysis performed. The syntax is very simple. It starts with an identifier for the dialog window: "T" for `TraceFit` and "S" for `SeriesFit`. It follows an identifier of the operation, separated by a colon. If the operation has one or more arguments, these are shown in brackets, separated by commas. The following examples show the command for an exponential fit in `TraceFit` to the 3rd Sweep in the first Series of the second Group and the subtraction of the constant value 1.5 from the waves buffer using the 'MathY' function in `SeriesFit`:

```
T:Fit[Exponential,2_1_3]
S:MathYWave[Sub, 1.50000E+000]
```

Number Presentation: Numbers in FitMaster can be represented in either `Engineering` or `Scientific` notation. Having this change to become effective requires restart of FitMaster.

4.2 Export Settings

These settings determine how data from FitMaster are exported.

4.2.1 File Export

Export Skipped Data: Data that are omitted from the fit by deselecting them in the waves buffer will be exported as separate arrays.

Export Error: Error bars are exported when selected.

Export Error Counts: The number of data points contributed to the error are exported when selected.

Export Background: Background traces are exported when selected.

Export Fit: Two arrays, one for the abscissa (X) and one for the fit results (FitFunction(X)) are exported. The resolution of these arrays is specified under `SeriesFit` (see below).

Export Marked Results: Export two arrays: one holds ASCII descriptors of selected parameters of the fit function, the other the numerical values of the parameters. Parameters are selected for export by activating the checkbox on the right side of the parameter fields.

Wave Names: With this list it is determined what kind of export array names (or wave names in IgorPro) are generated. Options are:

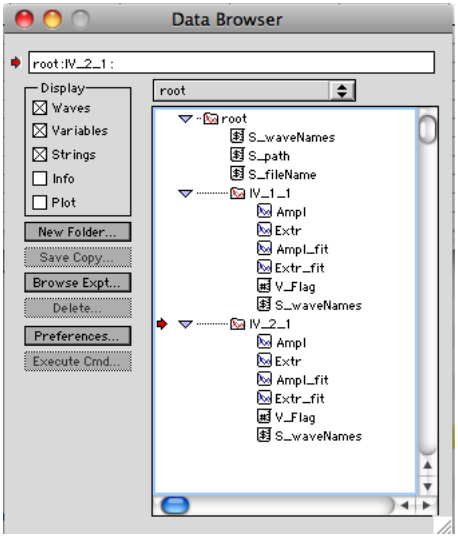
- Constant: In this case the bases of the wave names are the names of the X and Y arrays as shown on top of the waves buffer.
- Tree structure: In this case the bases of the wave names are the names of the X and Y array as shown on top of the waves buffer plus an identifier of the current Series. E.g., the parameter array “Voltage” is

to be output for Series number 6 from Group 3, then the wave name will be “Voltage_3_6”.

4.2.2 IgorPro

In addition to the setting found in the `Replay` menu, these settings specify the way in which data are exported to IgorPro.

Make Data Folders: Generate data folder in IgorPro and put all data associated with one export function into such folders. This option helps to organize data in IgorPro. In particular, it allows exporting items of identical names such that macros in IgorPro can easily access such items in a repetitive manner. The following example shows the result of two exports from `SeriesFit`, containing IV data and the corresponding fits, loaded into the IgorPro data browser. The folders are named according to the tree structure, i.e. “IV_2_1” refers to the `Series` 1 with name “IV” from `Group` 2.



4.2.3 Clipboard

These settings provide additional control over the clipboard functions for `TraceFit` and `SeriesFit` output.

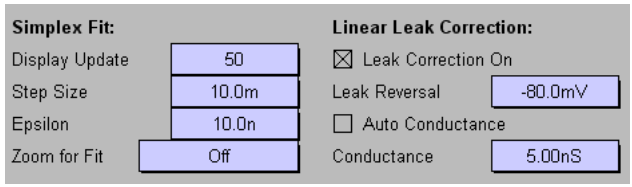
Include Equation: A string holding the name of the fit equation is output. This is particularly important when `Parsed Equations` are used.

Include Titles: If this flag is on, the table columns written to the clipboard will start with an ASCII identifier, i.e. a name of the column.

Include Tree Identifier: A tree identifier is exported with each data set.

Include Units: Units will be appended to the identifier names (column titles) in the form of "`Name[Unit]`".

4.3 Trace Fit Settings



4.3.1 Simplex Fit Settings - TraceFit

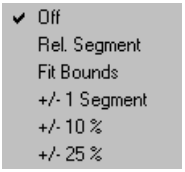
These settings regard the adjustment of the Simplex fit algorithm used in the `TraceFit` routines.

Display Update: Number of fit iterations before next display update is performed. Display updates includes redrawing of the fit function and update of the parameters and residuals in the fit function fields.

Step Size: Initial relative deviation of the free fit parameters for the first iteration. "10.0m" means that all parameters are altered by 1% for finding the direction to a new minimum. This setting proved quite useful. Making this figure much smaller may result in very slow convergence; making it too big may cause divergence, i.e. unreasonable results.

Epsilon: This parameter determines the convergence level of the fit. "10.0n" means that the fit is terminated and considered to have converged when the residual has not changed by more than 10^{-6}%.

Zoom for Fit: With this list of selections one defines how the data traces and the `TraceFit` results are displayed in the Oscilloscope window during the fit procedure. The X-resolution of the display can be zoomed in the following manner:



- Off: Data traces are not zoomed. The display mode selected in the Oscilloscope window is used.
- Rel. Segment: Only the relevant Y-segment is shown.
- Fit Bounds: Only the data within the limits of the Fit Bounds are shown (cursor settings defined in `TraceFit`).
- ±1 Segment: The relevant Y-segment and the segments preceding and following it are shown.
- ±10%: The relevant Y-segment and ±10% before and after the segment are shown.
- ±25%: The relevant Y-segment and ±25% before and after the segment are shown.

4.3.2 Linear Leak Correction

Data traces can be subjected to a linear leak correction procedure prior to a `TraceFit` operation. This feature should be used with caution because it usually is required to inspect the leak correction results for all traces individually. Thus, it is recommended not to use the `AutoFit` option in this case. In addition, linear leak correction is only meaningful if there are no p/n pulses subtracted from the data and if the `ZeroOffset` subtraction in the `Display Menu` is turned off.

Whenever a data trace is loaded from the tree, it is shown without leak correction; subsequently it is shown with leak correction applied and the fit function superimposed.

The linear leak correction assumes that the data are offset by an ohmic leak current. This is characterized by a `Conductance` and a `Leak Reversal` potential (the latter may be different from zero if, for example, the leak is mainly attributed to chloride currents with a negative reversal potential).

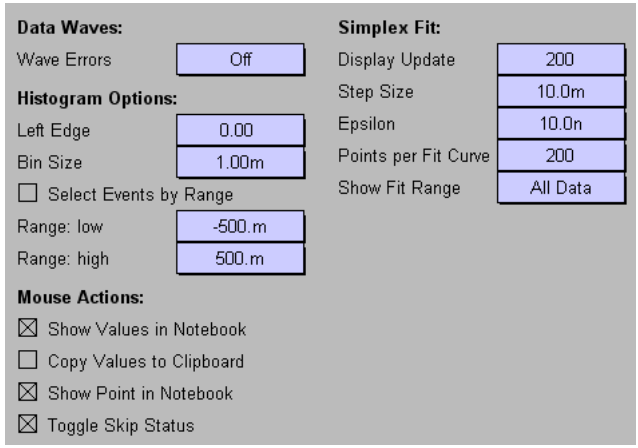
Leak Correction On: This flag turns the leak correction on. It will be called upon each trace fit.

Leak Reversal: The reversal potential for the leak current, i.e., the potential where the leak currents are zero.

Auto Conductance: If this flag is on, the programs tries to estimate the `Leak Conductance` from the data traces. It assumes that the zero current (i.e. mean data in the data segment specified for zero subtraction) is representing the leak only. Thus, one has to make sure that in the zero segment no ion channels are open.

Conductance: A leak conductance can be specified here (in Siemens). This is used whenever the `Auto Conductance` flag is off. Note that manual setting of the leak conductances requires close inspection of the results. In most cases several iterations will be necessary.

4.4 Series Fit Settings



4.4.1 Data Waves

These settings regard the handling of analysis parameters (or results from the Online Analysis) once copied into the work buffer of the `SeriesFit Window`.

Wave errors: When data in the waves buffer are averaged, statistical errors can be calculated and are used for display. The options are:

- Off: No errors are compiled or displayed.
- Min/Max: The minimal and maximal data points are considered as error.
- S.D.: Standard deviation.
- S.E.M.: Standard error of the mean, i.e. S.D./sqrt(n); n = number of accumulated waves.

The waves buffer is cleared whenever the type of error is changed in the configuration.

4.4.2 Histogram Options

The `Histogram Options` define the histogram generated with the `histo preview` function of the `Waves Operation` in `SeriesFit`.

The resolution of the histogram is defined by a starting value (left edge on the X-axis of the histogram) and the size of an individual histogram bin. The number of bins is automatically given by the largest data point included in the histogram.

Left Edge: Defines the starting point of the histogram (left point of the X-axis of the histogram).

Bin Size: Is the individual size of the histogram bin.

By default all events are included in the histogram. Optionally, the selec-

tion of data can be restricted by defining a data range.

Select Events by Range: If checked, the events included in the histogram are selected from a given data range of the X-values of the `Preview Wave`.

Range: low: Defines the minimum value of the data range.

Range: high: Defines the maximum value of the data range.

A typical example is to select single-channel events with a certain level only. When the waves buffer contains the `Level` in the X-array and `Amplitude` in the y-array, $LowRange = 0.5$ and $HighRange = 1.5$ will result in the compilation of an amplitude histogram for all events with $Level = 1$.

4.4.3 Mouse Actions

The following settings determine how mouse clicks inside the graph in the `SeriesFit Window` are handled:

Show Values in Notebook: The coordinates of the cursor are output to the Notebook. The output is formatted according to the settings in the `Notebook Menu` (Scientific of Engineering convention) and the `Configuration Window` (Notebook Digits). In addition to the starting coordinates, i.e. those where the mouse button was pressed, the translations (dX and dY) are shown. While dragging the mouse, such translation is shown as broken line in the graph. The coordinates and dragging translations are internally stored. They can be retrieved for setting parameters of the `FitFunctions` (e.g., the X-coordinate is used to set a mean value of a Gaussian function, the dX value defines the standard deviation of a Gaussian).

Copy Values to Clipboard: The coordinates of the mouse and, as available, of the selected rectangle are output to the clipboard (X0, Y0, X1, Y1). Completion of this operation is signaled with a beep. With this feature one can quickly "measure" things in the graph window and transfer the data to other programs.

Toggle Skip Status: When this flag is set, the status of the data point clicked at is toggled, i.e. turned from "on" to "off" and vice versa.

Show Point in Notebook: The data point clicked at is identified in the

Notebook by showing its index and its status ("on" or "off").

4.4.4 Simplex Fit Settings - SeriesFit

These settings regard the adjustment of the Simplex fit algorithm used for the `SeriesFit` routines.

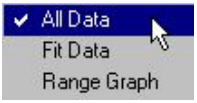
Display Update: Number of fit iterations before the next display update is performed. Display updates includes redrawing of the fit function and update of the parameters and residuals in the fit function fields.

Step Size: Initial relative deviation of the free fit parameters for the first iteration. "10.0m" means that all parameters are altered by 1% for finding the direction to a new minimum. This setting proved quite useful. Making this figure much smaller may result in very slow convergence; making it too big, may cause divergence, i.e. unreasonable results.

Epsilon: This parameter determines convergence level of the fit. "10.0n" means that the fit is terminated and considered to have converged when the residual has not changed by more than 10^{-6}%.

Points per Fit Curve: This parameter determines how many data points are generated for a fit function to be shown in the graph. This can be between 3 and 200. The default setting is 200, but low values can be used if update of the fit function becomes too slow on a slow computer when using manual fit, for example. In addition, in some cases one may want to have exported fit function data with less than 200 points.

Show Fit Range: Here it is specified what range the display of the fit functions should cover. Options are:



- All Data: The fit is displayed between the minimum and maximum of all X-data points (including the skipped data).
- Fit Data: The fit is displayed between the minimum and maximum of only those X-data points considered for the fit (not including the skipped data).
- Range Graph: The fit is displayed in the full X-range of the graph. This may deviate from the above when at least one of the X-axis

scalings is set to “fixed” or to “Round to 0/1/2/5”.

Chapter 5

Trace Fit Window

This window is used to analyze raw data traces.

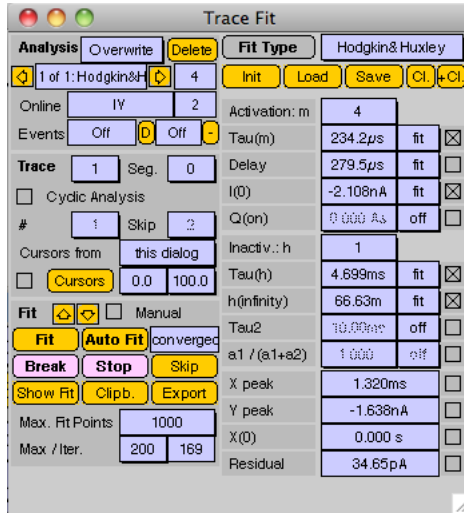


Figure 5.1: The TraceFit Window

5.1 Overview

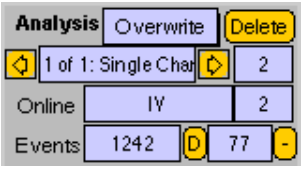
With this dialog mathematical functions (here called fit functions) are fit to selected areas of data traces. All or a subset of the parameters that determine the fit function can be selected to be stored to file. These, and

only these, are at later stages available for further analysis with `SeriesFit`. In addition to the results of a `TraceFit`, analysis parameters provided by the `Online Analysis` can be stored to file. All results from the `Online Analysis` for which `Notebook=true` will be stored to the analysis file as well. The user has to make sure to provide all information via the Online Analysis that is later needed for further analysis. In particular these parameters will be used as X-reference in the SeriesFit procedure. E.g., if in `SeriesFit` a current-voltage relationship is to be analyzed, the user must provide a voltage of a variable segment to be used as X-reference.

Analysis results cannot be mixed within one Series. When the user analyzed several Sweeps of a Series with a specific fit function and then selects another fit function, alters the parameters to be stored, or alters the `Online Analysis` method to be applied, then upon the next trace analysis operation the results previously stored on the level of the same Series and Analysis will be initialized. When another analysis for the same data set (or a different trace of the present Series) is to be performed, a new Analysis has to be generated. This is done by selecting in the Analysis mode `Append` instead of `Overwrite` (see below).

5.2 Data Selection

This section of the dialog provides information on the content of the analysis data tree on the level of the "Analysis".



Analysis storage mode: This selection list determines how new analyses are stored to the analysis file. The options are:

- Overwrite: The presently selected analysis is overwritten.
- Append: The new analysis is appended to the tree, i.e. it will become the last analysis.

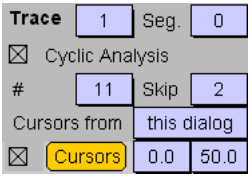
Delete analysis: This button deletes the presently selected analysis. Deletion is performed without any warning. Thus, be sure you really want to delete the analysis results.

Analysis tree selector: These icons show the name and number of the presently selected `Analysis`. With the arrows one can navigate through the analysis data tree from the first to the last Analysis associated to the presently selected data Series (see Replay Tree). The number at the right identifies the number of Results stored in the present `Analysis`.

Online identifier: This icon shows the name of the `Online Analysis` associated with the selected `Analysis`. The number at the right identifies the number of `Online Analysis` results stored to file.

Event counter: This field shows the total number of `Events` stored in the selected `Analysis` for the entire `Series`. The counter to the right shows all `Events` of the active `Sweep`. All `Events` per `Sweep` can be deleted with the "D" icon, while "-" only deletes the last event.

Trace selector: The data trace to be analyzed.



Segment: This is a segment offset applied to the relevant Y-segment of the trace to be analyzed.

Cyclic Analysis: When this checkbox is selected, the chosen analysis is performed several times (specified by the # field). Each time the relevant segments are offset by the number specified in the "Skip" field. Cyclic analysis creates `Events` rather than `Results` in the data analysis file. Its use is explained in a later chapter.

Cursor source: With this list one selects the mode by which the cursor positions for the data fit are to be specified. The options are:

- this dialog: The cursor settings are used from this dialog (see below).
- first Online: The cursor settings are used from the first Online Analysis Function of the selected Online Analysis Method.
- last Online: The cursor settings are used from the last Online Analysis Function of the selected Online Analysis Method.

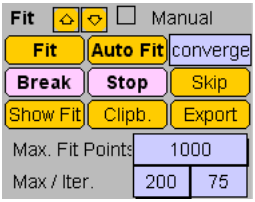
The latter two options can be used to predefine cursor bounds for the fit procedure and to associate them with an `Online Analysis Method`. This

is a very useful option when one analyzes different types of data (resulting from different Series) that all require different cursor settings. Once associated to the appropriate `Online Analysis Method`, the cursor settings for the `TraceFit` are then automatically set when `Automatic Stimulus Control` in the `Online Analysis` is activated.

> ***Note:*** *The cursor settings to be used in the Online Analysis always have to be set in the Online Analysis Window. The cursor settings specified in the TraceFit window only apply to TraceFit.*

Cursors: When checked, the cursor settings specified on the right (left, right cursors in % of the relevant Y-segment ± segment offset) are shown in the Oscilloscope window when operations in the `TraceFit` window are performed. During dragging of the cursor bounds the cursors are shown even though the checkbox is off. The cursors in the `Oscilloscope Window` are activated with the yellow button. The new cursor positions are then sent to `TraceFit`.

5.3 Fit Operations



Upward arrow: Move backward in the data tree by one Sweep.

Downward arrow: Move forward in the data tree by one Sweep.

Manual: When this checkbox is selected, the fit function is superimposed to the data whenever a parameter in the fit function is changed.

Fit: This button starts the fit operation applied to the selected Sweep/Trace.

Auto Fit: This button starts an automatic fit operation. It is applied to the selected Sweep/Trace and continued until the end of the present Series.

Fit Status: This text field provides information on the status of the fit. The options are:

- blank: Initialized status; no fit was performed thus far.
- converged: The fit has converged successfully according to the convergence criterion (see Epsilon in FitConfiguration).
- max. iter.: The maximal number of iterations has been reached without convergence of the fit. The field turns red.
- break: The fit has been interrupted by the user. The field turns red.

Break: Interrupt the fitting of the currently selected trace. When this button is pressed during an `AutoFit` procedure, the present trace fit is terminated and the next is started.

Stop: Interrupt the fitting. The currently running fit is terminated. When this button is pressed during an `AutoFit` procedure, the present trace fit is terminated; no further fits are started.

Skip: Skip this Sweep for fitting. Consequently, an analysis entry for this Sweep is generated specifying it as "not fitted".

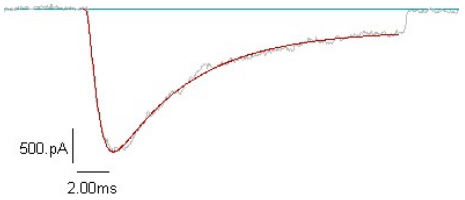
Show Fit: Superimpose the fit function to the present data trace according to the selected settings.

Clipb.: Copies the selected trace and the active fit function into the clipboard.

Export: Exports the selected trace and the active fit function according to the settings of the `Replay Menu`. Currently, the export mode `MatLab` is not supported.

Max. Fit Points: In this field the maximal number of data points to be considered for the fit is specified. This feature is only used for Parsed Equations, Exponentials and for Hodgkin&Huxley fits. The data are binned into "Max. Fit Points" slots with exponentially increasing intervals. In this way the actual data density is high in the beginning of the fit range, and sparse at the end. This feature substantially speeds up the data fits and yields good results for the named functions. If a Parsed Equation is incompatible with such an exponentially distributed data set, the user is requested to set the maximal fit points high enough to accommodate all data points.

Max./Iter.: In the left field the maximal number of iterations can be specified. The right field displays the iterations counter during the fitting.

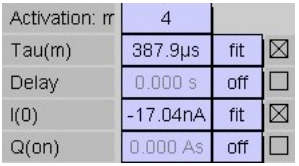


In this figure an example of a data fit to a current trace from voltage-gated sodium channels is shown. The raw data are shown in gray. The superimposed fit curve (red) was optimized according to a Hodgkin&Huxley formalism with $m = 3$, $\tau_m = 532$ μs, $I(0) = -3.06$ nA, $h = 1$, $\tau_h = 4.94$ ms, $h_\infty = 0.01$.

5.4 Fit Functions

The right column of the `TraceFit Window` displays the fit functions. They are selected with the list at the top.

All fit functions will be described in more detail below. Here we just describe some general aspects. There are different fields specifying a fit function.



Name: The identifiers on gray background are the names of either function parameters or of derived parameters.

Parameter: This field holds the parameter. Fit parameters can be edited, derived parameters are just displayed.

Parameter Status: This list specifies how the respective parameter is treated during the fit. The status is stored, so at later stages during the analysis one can retrieve this information. The options are:

- off: Parameter is not considered. Usually set to zero.
- fit: Parameter is fitted.
- hold: Parameter is kept constant during the fit.

Store Flag: This checkbox determines whether or not the respective parameter is to be stored in the analysis file. In the example above there are

four derived parameters:

- X(0): The X-value where the fit function is zero (first zero crossing).
- Y(0): The Y-value at the start of the fit data range.
- Integral: The integral of the fit function in the selected data range.
- Residual: This parameter is available for all fit functions. It specifies the quality of the fit - it is a mean squared deviation.

Fit Type: When this button is pressed, the mathematical fit function is shown in the `Oscilloscope Window`.

- All operations, variables and constant parameters are shown in black.
- Linear fit parameters are shown in blue.
- Non-linear parameters are shown in red.

This information is important because linear parameters are directly calculated via a least-squares method. Thus, no initial guesses have to be provided. Non-linear parameters, however, are optimized with a Simplex fit algorithm. The fit success chiefly depends on how well the initial conditions were selected. This is particularly true for functions with very strongly non-linear components (e.g. Boltzmann functions).

> ***Note:*** *The formula only shows those parts of the fit function that are set to* `fit` *or* `hold`*.*

Below the fit function name there are four more buttons controlling the content of the fit function.

Init: This button resets all fit function parameters to the default settings provided by FITMASTER.

Load: This button resets the selected fit function parameters to the values previously saved by the user.

Save: This button saves the parameters of the selected fit function to be used for a later "Load" operation.

These three functions are useful when fits are to be started always with the same initial conditions. They facilitate restoring the parameters to realistic values once the fit algorithm has diverged, and hence has generated unreasonable fit parameters.

Cl.: With this button the selected parameters of the present fit function are *written* to the clipboard. Depending on the settings in `Fit Configuation`, the names and units of the parameters and a tree identifier are included or not. In other programs the content of the clipboard can be retrieved by 'CTRL' V or 'CMD' V.

+Cl.: With this button the selected parameters of the present fit function are *appended* to the clipboard. Depending on the settings in `Fit Configuation`, the names and units of the parameters and a tree identifier are included or not. In other programs the content of the clipboard can be retrieved by 'CTRL' V or 'CMD' V. Note that the clipboard is not initialized with this function. Thus, to clear the clipboard and to put in the first item from FITMASTER one has to use "Cl."; only the subsequent information is put into the clipboard with "+Cl.".

5.5 Cyclic Analysis

The normal fit functions are restricted to one fit per trace and analysis function. In several cases, however, one may want to repeat an analysis several times for the same trace, but offset by a specified number of pulse segments. Consider the example where a train of identical pulses was generated to elicit ion currents that then change their properties in a use-dependent manner. In the example shown below, voltage-gated sodium channels are activated by a train of 11 pulses in the presence of a local anaesthetic that blocks the channels use-dependently.

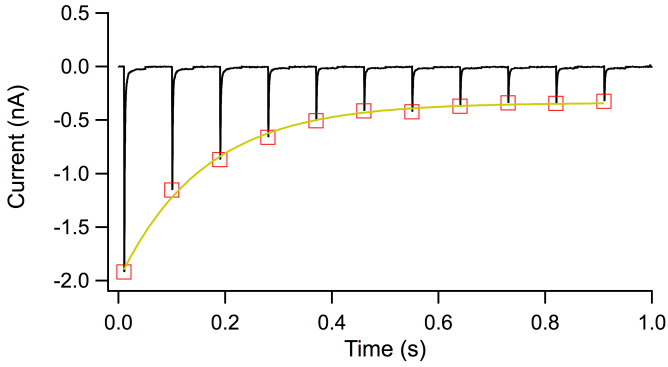
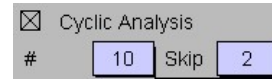


Figure 5.2: Example for Cyclic Analysis. A train of 11 pulses was used to elicit sodium current. Using Cyclic Analysis, for each pulse the peak current and the segment time were determined with the Online Analysis. These data were then further analyzed in `SeriesFit` to fit a single exponential function. Both, the Online Analysis results and the `SeriesFit` result are superimposed to the raw data trace. They were left-shifted by the length of the relevant Y-segment in order to match the peak current.

The aim of an analysis could thus be to determine the peak currents in each of the pulses and to further analyze them in `SeriesFit` as a function of the time of the segment within the trace.

Cyclic Analysis Settings: The above specified task can be achieved by using `Cyclic Analysis` in the following way.

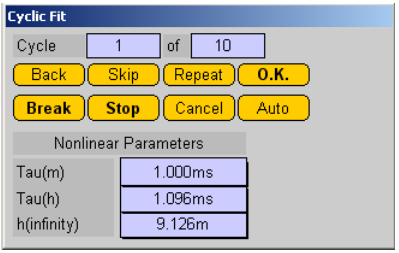


Activate the `Cyclic Analysis` checkbox. Then the items below will be enabled. Here we specify the number of analyses to be performed (e.g. 10) and the segment interval (here "Skip" = 2, i.e. for each iteration the relevant segment is incremented by 2). Upon execution of a "Fit" operation, this fit will be performed 10 times. All produced results will be appended to the data tree as `Events`. The last analysis result is always additionally stored in the `Results` section. For our example above, the job can be done by just using "OnlineOnly" as the fit function type. In

the Online Analysis one would need to generate two Analysis Functions: "Segment Time" and "Minimum". Thus, the fit operation will generate 10 `Events` consisting of the entries "Segment Time" and "Minimum". In `SeriesFit` these events can be retrieved to be analyzed in the data buffer (see below).

5.5.1 Cyclic Analysis Dialog

Cyclic analysis can also be used for fit functions that require a Simplex algorithm (e.g. "Hodgkin & Huxlex"). In this case, after each fit operation a `Cyclic Fit Dialog` appears providing a user interface.



The dialog consists of three sections: A cycle identifier, a set of control buttons, and, if applicable, a list of nonlinear fit parameters.

Cycle: The present cycle number and the maximal number of cycles is displayed. The maximal number of cycles is either the number specified in the `TraceFit Dialog` or the largest possible number for the given `Trace`. The largest possible number is taken when either of the two possibilities apply: The specified number of cycles is either zero or greater than the number possible given the number of segments per `Trace`.

Back: Go one cycle back and redo analysis from there.

Skip: Consider the analysis of the present cycle as "not fitted". Thus, upon loading these `Events` into `SeriesFit`, the corresponding data points in the wave buffer will be set to "skip".

Repeat: Redo the present analysis. This is useful if the fit has not converged properly.

O.K.: Accept the present fit and proceed with the next cycle.

Break: Interrupt analysis for the selected trace.

Stop: Interrupt analysis for the selected trace and also interrupt analysis of a Series in case Cyclic Analysis was called via the `Auto Fit` function.

Cancel: Interrupt the fit and leave the dialog without changes.

Auto: Accept present result and continue fit of remaining cycles without further notice.

Nonlinear Parameters: When the selected fit function contains nonlinear fit parameters, these are displayed here. They can be edited in order to provide optimized start values. Typically this is done before one hits the `Repeat` button.

Chapter 6

Series Fit Window

This window is used to analyze the results from the `TraceFit` and from the `Online Analysis` on the hierarchical level of a `Series`.

6.1 Different Types of Analysis Results

`TraceFit` analysis operations store their results in the analysis tree file on different hierarchical levels. Per `Sweep` and `Analysis Function` there are a list of `Results`, i.e. one `Result` value per `Sweep` and `Result Type`.

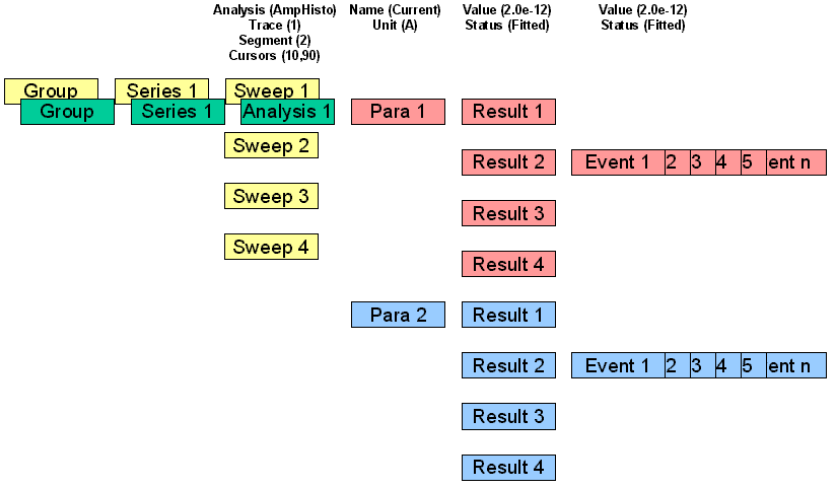


Figure 6.1: Analysis Tree. Illustration of the link between a data tree (*.pul, yellow) and an analysis tree (*.ana, foreground).

In some cases there are several results obtained for a single `Sweep`. Such cases are cyclic analysis results (see Section 5.5 on page 38) or results from `Measure`, `Trace Copy`, `Amplitude Histogram`, `Power Spectra` or `Event Detection` analyses. These multiple values per `Sweep` are called `Events`, and they are appended to the `Results` tree structure. The structure of this analysis tree is depicted in Fig. 6.1.

For each `Series` there can be multiple analyses specifying the type of `TraceFit` analysis. An analysis has a number of `Parameters` (e.g. time constant, amplitude, or whatever was selected in `TraceFit` and `Online Analysis`). For each `Sweep` in the data file and for each `Parameter` there are `Result` branches. These hold the result values and a status flag. If there are multiple results for a given `Sweep` and `Parameter`, then they are stored as `Event`, again holding a value and a status flag. There can be an arbitrary number of events per `Sweep` and `Analysis`. Note that there are always the same number of `Results` and `Sweeps`. In addition, there are always the same number of `Events` attached to the `Results` branches corresponding to the same `Sweep` number.

Thus, the first task in SeriesFit is to extract the data to be analyzed from this tree. The scrolling bar in SeriesFit presents the Results stored in the tree for the given Series and Analysis. To define which types of data to be extracted, Parameters from: has to be set to either Results or to an Event extraction. In the first case the selected X- and Y-results are extracted. In the latter, the events are used. These can be extracted from the selected Sweep or, instead, all Events of a Series can be used. In the latter case, one has to specify whether all Events have to be appended to the preview buffer or whether they are to be sorted according to the corresponding Sweep. This can be done by adding the Y-values of all Events or by averaging them. The latter two options are useful only in cases where all Sweeps have the same number of Events (e.g. for Amplitude Histograms or Power Spectra).

6.2 The Waves Buffer Concept

The data to be extracted from the analysis file, i.e. Results or Events, are transferred to a preview wave by clicking on Preview. Now the data are shown as numbers in the table and they are plotted in the graph. More data can be appended to this table by selecting other data and clicking on Append. With these data in the preview wave all SeriesFit operations, such as scaling and sorting of the data, and, of course, fitting of functions can be performed.

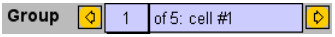
One may, however, want to perform operations with waves (e.g. add, subtract, multiply, divide waves). In this case the preview wave has to be transferred to a buffer wave by hitting Copy. Subsequently, new data can be loaded into the preview wave by the icon Preview. The preview wave can then be sent to the buffer by a Wave Operation. Alternatively, the preview wave can be accumulated in the buffer wave by the icon Accum.. In this case the buffer wave holds the average of all waves accumulated in the buffer. In addition, the table shows the number of averaged points for all individual data points (note that skipped points from the preview wave are not considered for averaging). The buffer wave also holds all information for computing error bars (as selected in the SeriesFit pane of the FitConfiguration).

The `Preview Wave` and the `Buffer Wave` are both displayed in the table and the graph, but not simultaneously. The type of wave selected can be set by the selector list. Note that `Preview Wave` is shown with a different color than the `Buffer Wave` (per default preview=black, buffer=red).

All further operations, such as data fits, export etc. are directed to the *active* wave.

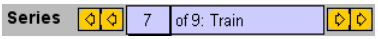
The results of fits performed to the active `Preview Wave` or `Buffer Wave` are always stored in the analysis file under the presently activated tree entry, i.e. the combination of `Series` and `Analysis`. Since the `Buffer Wave` can contain complex data sets that are not directly associated to such tree entries, there is also the option to store the `Buffer Wave` and the associated information on the fits in separate files. For this feature refer to Section 6.5.

6.3 Access to the Tree Structure



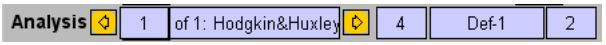
Group: This scrolling bar provides access to the data tree structure. By clicking on the left or right arrow one can move to the next or previous `Group`, respectively. Alternatively, a `Group` can be accessed by entering the group number.

The text field shows the total number of `Groups` and the group label.



Series: This scrolling bar provides access to the data tree structure. By clicking on the *inner* left or right arrow one can move to the next or previous `Series`, respectively. By clicking on the *outer* left or right arrow one can move directly to the next or previous `Series` with the same name. Alternatively, a `Series` can be accessed by entering the series number.

The text field shows the total number of `Series` in the active `Group` as well as the name of the selected `Series`.

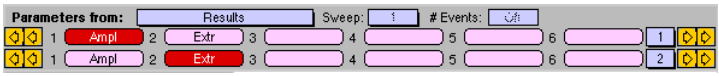


Analysis: This scrolling bar provides access to the stored analysis results of the selected `Series`. It shows the number of the currently selected `Analysis`, the total number of Analyses stored as well as the name of the `Analysis`, i.e. the `FitType` name used in `TraceFit`. The following fields on the right identify the number of parameters associated with the `TraceFit` operation, the name of the `Online Analysis Method` applied as well as the number of results extracted from the `Online Analysis`.

Events: The total number of `Events` associated with the selected `Analysis`.

The `Delete` button is used to remove the current `Analysis`. Note that with the action of this button you inevitably remove the analysis results from the data file.

The `Export` button is used to export the content of the analysis tree to the clipboard or to a file (see Section 6.4.1).



Parameters from: Two scrolling bars provide access to the results stored in the selected `Analysis`, i.e. both the results from `TraceFit` and from the associated `Online Analysis`. The parameters are highlighted by clicking on them. The upper bar is used to select parameters for the `X-Wave`, the lower bar for the `Y-Wave`. We use the term `Y-Wave` to identify data arrays used for further analysis in `SeriesFit`. The set of `X-Wave` plus `Y-Wave` is referred to as "waves buffer". Parameters can also be selected by typing in the number in the index fields on the right.

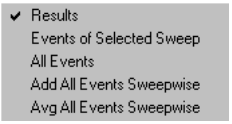
The source of the data for the waves buffer is specified in the list with the following entries:

Results: The results of all `Sweeps` of the selected `Series` are taken.

Events of Selected Sweep: Here we load all `Events` associated with the selected results. We do this for the specified Sweep number. The number

of events available is indicated.

All Events: In this case `Events` associated with the selected results are taken from all `Sweeps` of the selected `Series`. Note that these can be many, such that the maximum number of data points in the waves buffer may be reached.



Add All Events Sweepwise: Starting from the selected Sweep number, all `Events` are added on a `Sweep`-by-`Sweep` basis. A typical application is the accumulation of amplitude histograms.

Avg All Events Sweepwise: Starting from the selected Sweep number, all `Events` are averaged `Sweep`-by-`Sweep`. A typical application is the averaging of power spectra.

6.4 Processing of Analysis Results

6.4.1 Export Analysis:

The Export Analysis dialog provides a list of the active parameters which can be selected by mouse click. The results stored in these selected parameters are then either exported to the clipboard or to a data file. The mode of export is specified in the `Replay Menu` and `FitConfiguration`. The checkbox `Export Status` determines whether a status identifier is exported for each result. These identifiers are: skip=0, fit=1, and hold=2. In addition, it is specified what kind of data to be exported:

- Results: Export `Results` only.
- Events: Export `Events` only.
- Results + Events: Export `Results` and `Events`.

6.4.2 Previewing Analysis Results

Once X- and Y-results are selected from the parameter list, they can be (pre-)viewed in the `SeriesFit` graph. Once the data are displayed they can be immediately fitted with a `SeriesFit` function (refer to Sections 6.9 and 6.10) or further processed in the `Buffer Wave`.

Preview: Previews the selected analysis results in the `SeriesFit` graph.

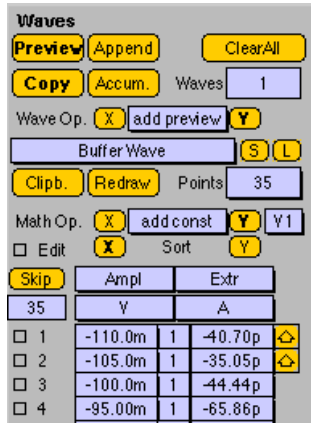
Append: With this button selected analysis results can be appended to the `Preview Wave`, i.e. the existing data in the `Preview Wave` are not deleted. With this function, results from different `Series` can be merged in a single `Preview Wave` for later analysis. This feature is quite important when data of multiple `Series`, which logically belong together, have to be merged.

> **Example 1:** Data were recorded based on a family of repetitive pulses. During that recording process, the current signal may have risen such that `Series` execution had to be stopped for gain adjustments. Then the experiment was resumed by executing the same `Series` again. For analysis, these two `Series` are to be concatenated to yield, for example, a graph with current as a function of time.

> **Example 2:** To record current-voltage relationships with non-equally distributed data points on the voltage axis one may generate several `Series`. Some may have sparse voltage intervals, others dense intervals. For analysis, the data from these different `Series` are to be combined.

Clear All: With this button all waves buffers (`Preview Wave` and `Buffer Wave`) are cleared.

6.4.3 Copying Data into the Waves Buffer



The `Preview Wave` can be further processed in the `Buffer Wave`.

Copy: Writes the `Preview Wave` into the `Buffer Wave`. I.e., the previous content of the `Buffer Wave` is replaced by the `Preview Wave`.

Accum.: When `Accum` is pressed, the data of the `Preview Wave` are averaged with the content of the `Buffer Wave`. The number of waves averaged is indicated by the `Waves` counter. Depending on the setting in `FitConfiguration`, errors are calculated and displayed in the graph. The number of data that contributed to an individual error estimate is displayed in the middle column of the waves buffer. Note that the sizes of the arrays to be processed in this way must be compatible. Averaging is only performed for Y-data.

Waves: Counter for the number of waves that are accumulated in the `Buffer Wave`.

Wave Operation: When transferring data from the `Preview Wave` to the `Buffer Wave` you can apply the following operations that can be selected from the list between the `X`- and the `Y`-button:

- add preview: Add incoming data to the `Buffer Wave`.

- sub preview: Subtract incoming data from the `Buffer Wave`.
- mul preview: Multiply incoming data with the `Buffer Wave`.
- div preview: Divide incoming data by the `Buffer Wave`.
- histo preview: A histogram will be generated from the `Preview Wave` with respect to the settings defined on the `SeriesFit` pane of the `FitConfiguration` (see page 25).

X: When the `X` button is pressed, only the X-values of the `Preview Wave` are processed with the X-values of the `Buffer Wave`.

Y: When the `Y` button is pressed, only the Y-values of the `Preview Wave` are processed with the Y-values of the `Buffer Wave`.

> ***Note:*** *Skipped data points of the* `Preview Wave` *are processed as zero.*

Wave Indicator - Buffer Selection: This bar shows the type of wave (`Preview Wave` or `Buffer Wave`) that is currently displayed in the `SeriesFit` graph and in the wave table below. This is also the *active* wave on which e.g. `Math` operations and fits are performed.

> ***Note:*** *All export functions, including clipboard, always apply to the active wave. Also, the active wave will be put into the background when the* `Background` *button is pressed (see below).*

The `Preview Wave` and `Buffer Wave` are displayed in different colors in the `SeriesFit` graph.

Save (S) and Load (L): These buttons are used to save and load the content of the `Buffer Wave`, hence, they are only active when the `Buffer Wave` was selected. In addition, these buttons store/load the `Background Waves` as well as the active `SeriesFit`. Details are described in Section 6.5.

Clipb.: With this button the content of the `Buffer Wave` is copied into the clipboard of the computer. A table of at least two columns is generated.

Depending on the settings specified in `FitConfiguration`, the columns will have titles with units. In addition, it can be specified whether or not the skipped data and errors for averaged data (see below) are to be exported to the clipboard. On Windows computers the content of the clipboard can then be retrieved in other programs (e.g., spreadsheet programs) by "CTRL V", on Macintosh computers by "CMD V".

Redraw: Display the content of the `Buffer Wave` in the graph according to the selected graph settings (see below).

Points: Counter for the number of data points in the `Buffer Wave`.

Math Operation: The following operations that can be selected from the list between the `X-` and the `Y-`button are applied to the active wave:

- abs: Form absolute value of the data in the waves buffer.
- inverse: Form the inverse of the data in the waves buffer.
- square: Form the square of the data in the waves buffer.
- sqrt: Form the square root of the data in the waves buffer.
- ln: Form the natural logarithm of the data in the waves buffer.
- log 10: Form the logarithm with base 10 of the data in the waves buffer.
- exp: Form an exponential of the data in the waves buffer.
- add const: Add a constant to the waves buffer.
- sub const: Subtract a constant from the waves buffer.
- mul const: Multiply a constant with the waves buffer.
- div const: Divide the waves buffer by a constant.

The constants are to be input via a dialog that appears upon executing the `Math Op.` function. There are different constants for the X- and the Y-waves.

The remaining operations use the built-in `Value` as selected from the list on the right. (For the use of `Values` please refer to the PATCHMASTER manual.)

- add value: Add the selected value to the waves buffer.
- sub value: Subtract the selected value from the waves buffer.
- mul value: Multiply the selected value with the waves buffer.
- div value: Divide the waves buffer by the selected value.

X: When the `X` button is pressed, only the X-values of the `Buffer Wave` are processed.

Y: When the `Y` button is pressed, only the Y-values of the `Buffer Wave` are processed.

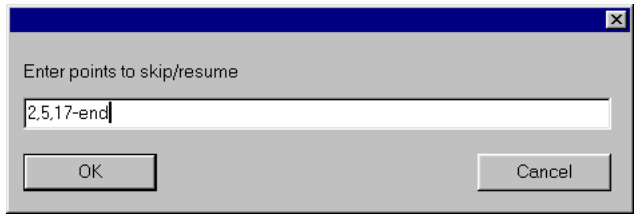
The mathematical operations are normally performed on Y-data (use button `Y`). However, they can also be applied to the X-data (button `X`). Note that the sizes of the arrays to be processed in this way must be compatible. Please be aware that mathematical operations on the data may results in unreasonable results if not properly applied (e.g., logarithm of a negative number results in "NAN" (not a number)).

- event corr: This function is used to correct single-channel event tables for skipped events.

Edit: When this checkbox is selected, the user has access to the data in the `Waves Buffer`. The individual values, the names, and the units can be edited. When the `Preview Wave` is active, in addition, the number of data points stored in the `Waves Buffer` can be specified. This is how the user can remove data from the list or can generate an hand-edited list of data points; these are then displayed in the graph and are used for further analysis.

> ***Note:*** *Manually editing data in the* `Waves Buffer` *has no consequence for data in the analysis data file. All edited values are overwritten upon the next copy action.*

Sort: Sort the data in the `Waves Buffer` in an ascending manner according to the X-wave (`X`) or the Y-wave (`Y`). This functions is, for example, very useful when data were generated using a stimulation protocol with alternating `Segment` values.



Skip: With this button one can change the "skip" state of individual data points. When a range of the graph was selected by mouse (i.e. a dashed rectangle is still present in the graph window), a click on the `Skip` button will toggle the state of all data points inside this rectangle. If no data range is selected, clicking on the button will activate an input dialog. Here, data points for which the state is to be toggled can be specified in several ways.

- 2,5,6: Identify data points by comma separation.
- 2-8: Provide ranges of data points.
- 17-end: The keyword "end" marks the last data point.

In addition, the keywords "on" and "none" set all data points to *active*, while the keyword "off" sets all data points to *skipped*.

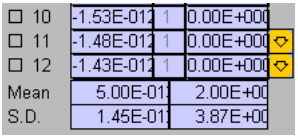
Label/Unit: Label and unit of the active wave. The `Buffer Wave` inherits the label and unit from the `Preview Wave` during the first buffer operation. Further processing does not overwrite the label and unit.

Wave Table: The `Waves Buffer` is displayed as two columns, holding the `X-Wave` and the `Y-Wave`, separated by a third column containing the number of data points contributing to the calculation of the error. In the example shown, the `X-Wave` contains the voltages of a variable segment and the `Y-Wave` the corresponding minimal currents. The list of checkboxes at the left indicates whether or not the respective result is to be included for

the data fit. If the checkbox is selected, data are shown in the graph as a different symbol. These data are referred to as "skipped data". Note that checkboxes can be activated immediately after copying of the results into the `Waves Buffer`. This happens when no successful fit was obtained for the respective `Trace` in `TraceFit` or when the user chose to skip the analysis of that `Trace`.

The `Waves Buffer` is a scrolling table. Use the arrows on the right to access the data points outside the display range. The uppermost/lowermost arrow moves to the start/end of the list; the other arrows forward/backward the list by one page. At the top of the columns there are identifiers for the names of the results as well as for their units. The counter above the `Skip` button shows the number of "non-skipped" data points, i.e. those data points considered for the fits and the statistics (see below).

Statistics: Mean and standard deviation of the data in the `Waves Buffer` are shown here. "Skipped data", i.e. those for which the skip flag is on, are not considered for statistics.



6.5 Export and Import of Wave Data

With the `S` (save) and `L` (load) buttons next the to Wave Identifier one can save and load the `Buffer Wave` to/from an ASCII file. This file contains the `Wave Buffer` plus all individual accumulated waves (as present). Thus, upon reload, the errors of the wave are automatically recalculated. In addition, this file contains all `Background Waves` as present and the type of `SeriesFit` with all of its parameters. Upon reload of such a `Waves File`, the `Buffer Wave`, the `Background Waves`, and the fit is completely restored.

This feature is meant to be used to store complex data that are not necessarily logically linked to the presently selected position of the data tree. In addition, it can be used to process FitMaster results in other programs or to generate or edit import files that are then further processed on the `SeriesFit` level.

Before the file is stored, the user can enter a comment to identify the content of the file. The file extension is `"wav"`.

The structure of the file is as follows:

Line 1: `FitMasterWave V1`
Keyword and version number of the Waves File.

Line 2: `HD:Data:DemoDat_1_2_2`
Original data file name and presently active tree identifier: Group_Series_Analysis.

Line 3: `Averaged IV data after substance X application`
User-defined comment to describe the data stored.

Line 4: `12, BufferPoints`
5-digit INTEGER specifying the number of points stored in the `Buffer Wave`. The identifier after the comma is written to the file, but is not necessary when the file is loaded.

Line 5: `3, Accumulated Waves`
5-digit INTEGER specifying the number of accumulated waves.

Line 6: `2, Background Waves`
5-digit INTEGER specifying the number of background waves.

Line 7: `6, Boltzmann`
5-digit INTEGER specifying the fit function. The identifier after the comma names the fit function, the number is the list entry. Only the latter is required for reading the file.

Line 8: `Ampl, V, Extr, A`
Comma-separated string containing labels and units of the X- and Y-waves.

Block of BufferPoints lines: `1, -3.34000E-08, -6.00000E-02`
The first number is the skip flag: `0=off, 1=on`. The following REAL numbers are the X- and Y-values.

In the next lines there are the Accumulated Waves (only Y-values) with their skip status. Each Accumulated Wave block starts with the identifier `WaveN` with `N` being the number of the wave.

Blocks of Background Waves consisting of the X- and Y-values. Each block starts with an identifier line `BackgroundWaveN` with `N` being the wave num-

ber. This is followed by a 5-digit INTEGER specifying the number of points per wave.

At the end of the file there is a block of data specifying the fit parameters. These are comma-separated strings of one of the following forms:

```
2, Exponent1
```

This line specifies the first exponent of a Boltzmann function.

```
2.00000E-05, fit, 1, Amplitude1
```

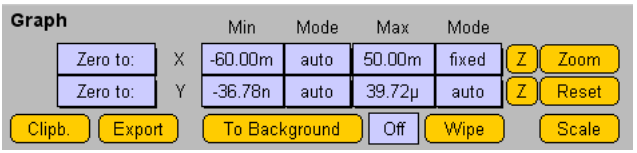
This line defines a result, i.e. values, fit status, store flag, identifier text. The latter is only output for the user's information; it is not necessary for reading the file. Note that all parameters are output in the order they appear in the `FitFunction` dialog. Also output are derived values, such as `XPeak` and the fit residual.

When `Parsed Equation` was used as the `FitFunction`, the first line holds the equation string. Then the parameter lines follow; in this case the parameter identifiers are used. Make sure to provide a list of parameters at least as long as the highest coefficient used and not more than the maximal number of nine. If, for example, `c[1]` and `c[3]` are used, also the second parameter line for `c[2]` has to be provided.

6.6 Graphing of Data

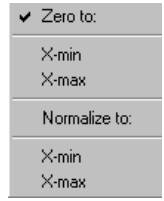
These settings are used to specify the axis settings of the display graph as well as to perform operations on the graph and the data shown in the background.

6.6.1 Axes and Scaling



X/Y, Min/Max, Mode, Zoom: Minimum and maximum of the X- and Y-axis are shown here. The mode determines whether the axis scaling is calculated from the data in the waves buffer (`auto`) or is taken from the fields shown here (`fixed`). The zoom buttons ("Z") are used to adjust the X- and Y-scaling according to a mouse-selected rectangle.

Zero offset / Normalization: For both, the X-Wave and the Y-Wave, an automatic zero offset subtraction as well as a normalization can be performed. Normalization will remove the units of the corresponding wave.



- Zero to X-min: Subtract the minimum value of the wave, i.e. set the minimum to zero.
- Zero to X-max: Subtract the maximum value of the wave, i.e. set the maximum to zero.
- Normalize to X-min: Divide wave by minimum value.
- Normalize to X-max: Divide wave by maximum value.

Zoom: Adjust both axes to the mouse-selected data range (dashed rectangle) and set the axes mode to `fixed`.

Reset: Reset the entire graph scaling to default settings.

Scale: Switch to a dialog for scaling of the graph (see below).

To Background / Counter / Wipe: The data in the `Waves Buffer` can be put into a background buffer by clicking on `To Background`. They are displayed in the graph with a separate set of markers and colors. The background buffer can be cleared by `Wipe`. The counter shows the number of currently stored background waves. There is a limit of 16 waves that can be stored. Note that no fit operation is applied to the background data.

Clipb.: Copy the picture of the `SeriesFit` graph into the clipboard of the computer.

Export: Export the content of the graph according to the settings specified in the `Replay` menu and the `FitConfiguration` dialog. Note that the export applies to the active `Waves Buffer`.

If exported in `IgorPro` format, some caution regarding the generated wave names has to be taken. In `IgorPro`, wave names cannot include mathematical symbols such as +, /, etc. During export these are automatically replaced by underscores. I.e., when the name of the X-wave is "a1/a4", a parameter from the `Online Analysis`, it will be exported to `IgorPro` as "a1_a4". In addition, `IgorPro` will not accept wave names that are identical to names of `IgorPro` functions or commands. Since there are many of them and since every user may use different sets of commands etc., these cannot be taken care of automatically. A typical problem is that the name of a wave to be exported is "Mean". Problems are prevented by either re-naming the "Mean" in the `Online Analysis` to something else. Alternatively, in the `SeriesFit Window` the names of the X- and Y-wave can be edited manually after activating the `Edit` checkbox in the `Waves` section.

6.7 Graph Scaling

This dialog appears upon clicking on `Scale`. The yellow dialog controls have the following meaning:

Init: Reset all settings formating the graph to default values.

Cancel: Leave the dialog without any changes of the scaling parameters.

Done: Leave the dialog and accept all changes made.

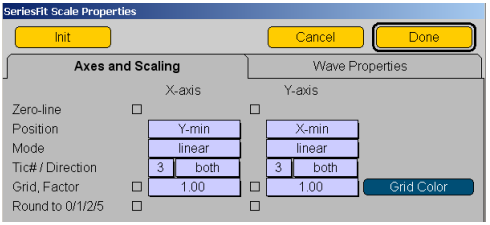


Figure 6.2: The Axes and Scaling Pane

For both, the X- and the Y-axis, the following settings can be entered:

- Zero-line: Draw zero line in the graph.

- Position: Defines the positioning of the axis. The options are: minimum, zero, and maximum of the other axis.
- Mode: Mode of axis scaling. The options are: linear, log, inverse, square root, and square. Be aware that unreasonable results may occur when taking the logarithm or square root of negative numbers. Whenever possible such cases are circumvented by setting $log(x) = 0$ and $\sqrt{x} = 0$ for $x \leq 0$ and $1/x = 0$ for $x = 0$.
- Tics / Direction: Number of tics and the tic direction. The options for the latter are: up, down, and both for the X-axis and left, right, and both for the Y-axis.
- Grid / Factor: Checkbox determining whether or not a grid is shown. The factor specifies the density of grid lines with respect to the tics, i.e. factor=1 means one grid line per tic, factor=2 means a grid line for every other tic.
- Grid Color: Color of grid lines.
- Round to 0/1/2/5: When this checkbox is selected, the min/max values of the axis ranges are determined such that they yield round values according to the $0 - 1 - 2 - 5$ system. For this feature to become effective, the axis scaling mode has to be set to "auto".

6.7.1 Waves Properties

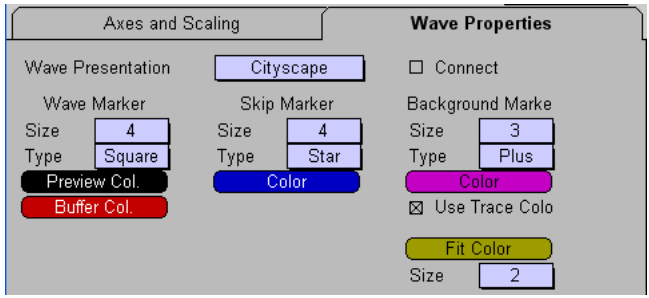


Figure 6.3: The Waves Properties Pane

At the top of the right column there is a list termed `Wave Presentation`. The selection in this list determines how the main waves are presented (background waves are always presented as markers). The options are:

- **Marker:** Data points are presented as markers.
- **Sticks & Markers:** Markers connected to zero by vertical lines.
- **Histogram:** Data points are presented as histogram bars. These bars are truncated to the minimum Y-value of the graph. The bars are drawn such that the center of the bar represents the data X-value, i.e. the bars extend from the centers between the present to the preceding and the following data point, respectively. Thus, a histogram will only look reasonable when the data waves are sorted according to the X-wave. When a histogram is exported to IgorPro, data waves are shown as "bars", offset by half a bin size to the left. The error bars are appended to a separate, non-shifted data wave, shown as dots. This way it is ensured that the error bars appear at the center of the data bars.
- **Cityscape:** Shows the outline of the histogram representation.

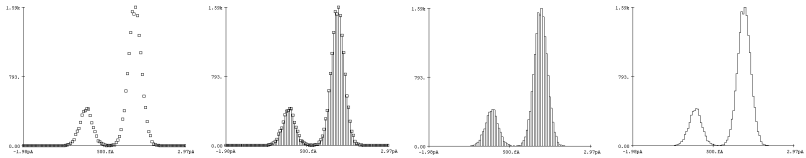


Figure 6.4: Examples for different wave presentation: Markers, Sticks&Markers, Histogram, and Cityscape.

The `Connect` flag determines whether or not data points are to be connected by straight lines. They are connected according to the appearance in the `Waves Buffer`. Thus, for non-ordered data, it may be necessary to execute a `Sort` command before.

In addition, size, type, and color of the following markers can be specified:

- Wave Marker: Main data points from the `Waves Buffer`.

- Skip Marker: Those data points from the `Waves Buffer` for which the "skip" flag was checked.
- Background Marker: Data points that are held in the background buffer.

The `Use Trace Colors` flag is valid for the background data. If selected, the individual waves in the background buffer are shown in the trace colors as specified in the `Configuration Window`. When it is off, the background color as specified above is used for all background waves.

`Fit Color` is the color of the fit function. The resolution of the fit function is determined in the `FitConfiguration` dialog.

`Size` specifies the thickness of the fit function curve.

6.8 Mouse Operations in the Graph

When waves are displayed in the graph area, coordinates and mouse drag values (dX and dY) can be output to the `Notebook` of the clipboard by mouse click. In addition, individual data points can be identified and/or their skip status can be toggled. These functions are activated in the `Fit Configuration Window`.

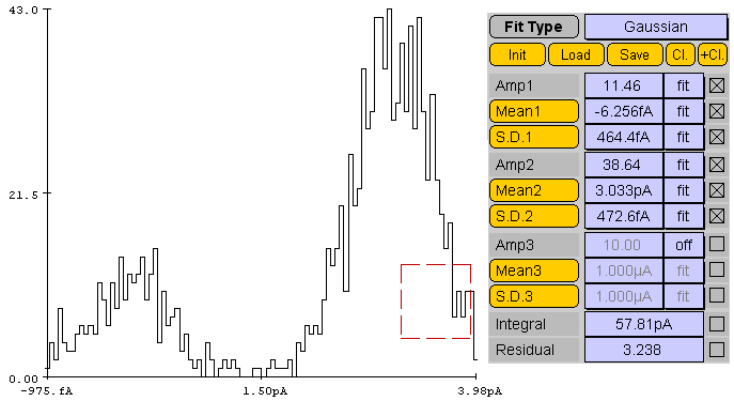
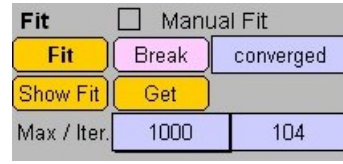


Figure 6.5: Left: Data in CityScape presentation with mouse-selected rectangle. Right: Dialog of a Gaussian fit. Clicking on "Mean" will copy the left X-coordinate of the selection rectangle into the parameter field, clicking on "S.D." will take the X-width of the rectangle as standard deviation of the Gaussian distribution.

The coordinates and drag values are stored internally. They can be used to set starting parameters of the individual fit functions. These are indicated by buttons in the `FitFunction` parameters lists. Depending on the nature of this parameter, the coordinate (X-value) or the drag value (dX-value) is used. In the example shown, a Gaussian fit, X-values are retrieved to set the mean values of the distribution, dX-values defined their standard deviations. Furthermore, X-values are used to set time constants and offsets, dX-values are used for slope values of Boltzmann distributions.

6.9 Fit Operations

This part of the dialog holds information and tools for the Simplex fits applied to the data in the `Waves Buffer`.



Manual Fit: When this flag is set, the fit curve is shown upon every change in the parameters of the fit function.

Fit: Start fitting the function to the selected data using a Simplex algorithm. Several properties of the fit including convergence criteria are set in the `FitConfiguration` dialog.

Break: Stop fitting procedure.

Fit status: In this field the status of the fitting procedure is shown. The possibilities are: blank, fitting, converged, break, max. iterations.

Show Fit: Draw fit curve into the graph according to the settings of the fit function.

Get: Retrieve fit function with all parameters from the analysis file.

Max. / Iter.: Maximum fit iterations to be executed and iterations counter.

6.10 Fit Functions

For a description of the fit functions that are available for `SeriesFit` please refer to *SeriesFit Functions* of Section 7.2 on page 88.

Like in `TraceFit`, the fit function is selected from a list at the top. The parameters have a name, value, and status. Unlike in `TraceFit`, the results of the fits are always stored to file, i.e. overwriting the old fit results. The flag at the right of the parameter list is only used for output purposes. Only parameters with the checkbox on will be exported to the clipboard (see below) or to any other export file.

Fit Type: When this button is pressed, the mathematical fit function is shown in the graph area. All operations, variables and constant parameters are shown in black. Linear fit parameters are shown in blue, non-linear parameters are shown in red. This information is important of the user because linear parameters are directly calculated via a least-squares method. Thus, no initial guesses have to be provided. Non-linear parameters, however, are optimized with a Simplex fit algorithm. The fit success chiefly depends on how well the initial conditions were selected. This is particularly true for functions with very strongly non-linear components (e.g. Boltzmann functions).

$$y(x) = \frac{Ampl}{(1 + \exp(-(x-Xhalf1)/Slope1))^{n}} + Offset$$

> ***Note:*** *The formula shows only the parts of the fit function that are set to* `fit` *or* `hold`*.*

Below the fit function name there are four more buttons controlling the content of the fit function.

Init: This button resets all fit function parameters to the default settings provided by FITMASTER.

Load: This button resets the selected fit function parameters to the values previously saved by the user.

Save: This button saves the parameters of the selected fit function to be used for a later "Load" operation.

These three functions are useful when fits are to be started always with the same initial conditions. They facilitate restoring the parameters to realistic values once the fit algorithm has diverged, and hence has generated unreasonable fit parameters.

Cl.: With this button the selected parameters of the present fit function are *written* to the clipboard. Depending on the settings in `Fit Configuation`, the equation name as well as the names and units of the parameters are included or not. In other programs the content of the clipboard can be retrieved by 'CTRL' V or 'CMD' V.

+Cl.: With this button the selected parameters of the present fit function are *appended* to the clipboard. Depending on the settings in `Fit Configuation`, the equation name as well as the names and units of the parameters are included or not. In other programs the content of the clipboard can be retrieved by 'CTRL' V or 'CMD' V. Note that the clipboard is not initialized with this function. Thus, to clear the clipboard and to put in the first item from FITMASTER one has to use "Cl."; only the subsequent information is put into the clipboard with "+Cl.".

Chapter 7

Fit Functions

In the following we describe all functions that can be fit either to data traces (`TraceFit`) or to data in the `Waves Buffer` of the `SeriesFit` panel. Some of the functions are identical; in these cases they are described under `TraceFit` only.

In most cases the functions are expressed as $y(x)$. Thus, x is the x-variable and y is the result. When applied to current traces in `TraceFit`, x will be the time and y the current.

The functions can be displayed in the program by clicking on `FitType` in the `TraceFit` and the `SeriesFit` window. When displayed, three colors are used. Black denotes mathematical symbols, variables, and constant parameters. Linear parameters to be fit are shown in blue, non-linear parameters are shown in red. It is important to note that only the non-linear parameters are subject to an iterative fit routine (Simplex algorithm in this case). The linear parameters are directly calculated according to a least-squares algorithm for a linear system. Therefore, the user only has to provide good initial guesses for the non-linear parameters before starting the fit. The initial values for the linear parameters are irrelevant.

An exception to this rule applies when using the `Parsed Equation` function. In this case all fit parameters are treated as non-linear parameters, i.e. they are subject to a Simplex optimization (see below).

7.1 TraceFit Functions

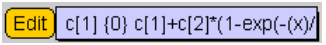
Available `TraceFit` functions are:

- OnlineOnly
- Parsed Equation
- Polynomial
- Exponential
- 1-Exponential
- Hodgkin & Huxley
- Gaussian
- Risetime
- Measure
- Trace Copy
- Ampl. Histogram
- Power Spectrum
- Single Channel
- Action Potential

7.1.1 Online Only

When this `FitType` is selected, no explicit fitting is performed. Only the results provided by the `OnlineAnalysis` (i.e. those results marked "Notebook = TRUE") are stored in the analysis file.

7.1.2 Parsed Equation



When this `FitType` is selected, an equation as provided in a string variable will be parsed and evaluated. This option provides full flexibility as to generating arbitrary functions, also sectioned into different definition ranges if desired.

However, since the equation is parsed, there are several obvious disadvantages: (1) Execution will be slower than the built-in functions as described below, (2) the fit parameters are treated as non-linear parameters. This means that initial guesses are required for all fit parameters and convergence problems can arise much more easily compared with built-in functions.

The first line of the dialog shows an `Edit` button and a short version of the equation string. An equation can be directly keyed into this text field. However, for more complex equations it is recommended to use the `Edit` dialog or to read an equation from a text file (see below).

It follows a list of up to 9 fit parameters. `Integral` is the integral of the fit function within the cursor range.

For parsed functions in `TraceFit` the time base can be selected. The options are:

- Segment Start – Time starts at the left edge of the relevant segment.
- Cursor Position – Time starts at the left cursor.
- Given Time 0 – Time starts at a value (counting from the beginning of the `Trace`) specified in the `Time0` field.

7.1.2.1 Parser Syntax

The parser evaluates all standard numbers (e.g. `32.8`, `2.0E-3`, but not `.3`, i.e. there must be a decimal point and at least one digit preceding it) and operations (`+`, `-`, `*`, `/`, `^`) using standard priority and bracketing convention (e.g. "`3.0 * (1.0-2.0)`", do not use "`[`" or "`]`", see below). In addition, it recognizes the constants `pi` ($\pi = 3.141592...$) and `e` ($e = 2.718281...$). As in all further operations, the entries are not case sensitive, e.g. `pi` and `PI` will yield the same result. The x-variable is specified by `x`.

The following single-argument operations are supported:

- `exp` – exponential
- `ln` – natural logarithm

- `sqrt` – square root
- `sin` – sine
- `cos` – cosine
- `tan` – tangens
- `arcsin` – arc sine
- `arccos` – arc cosine
- `arctan` – arc tangens

7.1.2.2 `PatchMaster` Constants

The parser furthermore provides access to information stored in various places of PATCHMASTER. For that purposes the following definition applies, the item index is given in brackets (`[i]`):

- `f[i]` – Online function with `i` being the online function index (0..n)
- `v[i]` – Value with `i` being the value index (0..15)
- `icon[descriptor]` – Icon value with `descriptor` being the icon descriptor (as used in Macros).

7.1.2.3 Fit Parameters

Access to fit parameters as used in FITMASTER is provided by `c[i]` with `i` being the index of the fit parameter (1..9). In the fit dialog all fit parameters detected in the equation string will be activated. For each of them the following fields are provided:

- Identifier text – This text field can be used to provide a name for the fit parameter so it will be easier to interpret this later on. E.g. the text "tau" may specify that the fit parameter `c[3]` is a time constant.
- Value – Here a start value has to be given; it will contain the result when the fit is completed.

- Status – Specifies the status of the fit parameter, i.e. “fit” or “hold”.
- Store – Activate this checkbox if the fit parameter is to be stored or to be used for export purposes.

7.1.2.4 Definition Limits

Equations can be split in up to six definition ranges. Such ranges are defined by `limits`; these are expressions in curled brackets, `{..}`, that separate two equation strings. The limit expressions must not include the x-variable. It is evaluated and the result determines which equation is going to be used. For the example “`equ1 {limit1} equ2`”, `equ1` is used as long as $x \leq$ `limit1`. Otherwise `equ2` is taken.

Example The following example is the default parsed equation provided in `SeriesFit`. It describes a function that is constant (`c[2]`) for $x \leq$ `c[1]`. If x is greater, a saturating exponential function follows.

```
c[2] {c[1]} c[2] + (c[3]-c[2])*(1-exp(-(x-c[1])/c[4]))
```

The following names are given to the fit parameters: `c[1]`: “x0”, `c[2]`: “a0”, `c[3]`: “a0+a1”, `c[4]`: “tau”.

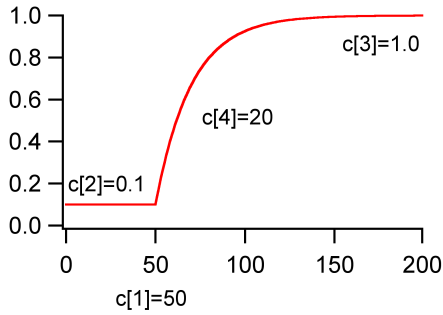


Figure 7.1: Example for a parsed equation with one limit, defined by c[1].

Note: *This default example looks very similar to a General Exponential (see 7.2.2). However, in this equation one can set the limiting value where the exponential saturates, i.e. "a0+a1", to a fixed value.*

7.1.2.5 Equation Editor

The `Edit` botton activates an equation editor. This automatically breaks up the total equation string into individual equations and limits. Based on the x-value specified in this dialog, the individual strings are evaluated to yield the individual `Results`. This is very useful for testing purposes and trouble-shooting. `Total Result` gives the result of the total equation string for the given x-value. Error messages shown in this dialog help to find syntax errors. Make sure that the results of the definition limits have an ascending order.

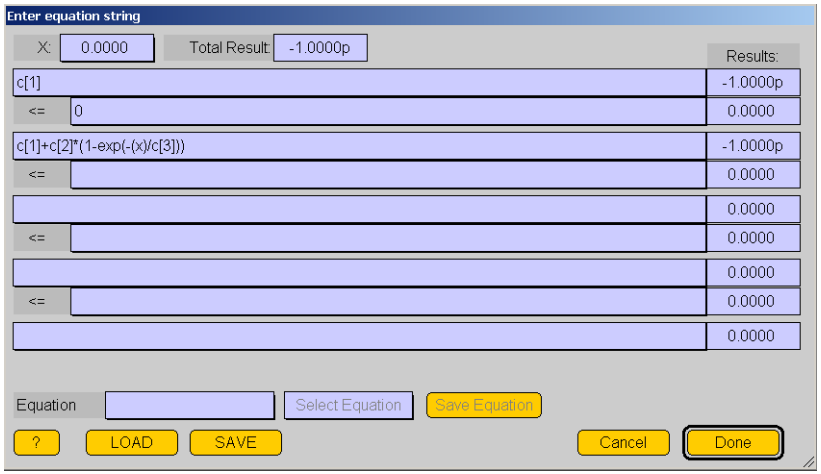


Figure 7.2: Equation Editor for default Parsed Equation of `SeriesFit` with $x = 11.0$.

`Integral` is the integral of the fit function within the cursor range for `TraceFit` and the fit function range in `SeriesFit`.

7.1.2.6 Equation File

In particular for very long equations it is sometimes easier to provide them via an `Equation File`. Such a file has standard ASCII format and holds information on up to 15 equations. For each equation there are up to two lines: the first holds the equation name and the equation itself, the second one holds a description of the fit parameters necessary for FitMaster.

The syntax of the `Equation File` is as follows:

```
@EqName c[1]+x*5.0... ; "comment"
#EqName c[1], "name", "unit", value, status, store, c[2], ...
```

Thus, the first line holds the equation name indicated with a `@` sign, followed by the equation as describe above. An optional comment is added to the line, separated from the equation by a semicolon. A line has a maximal length of 1024 characters.

The second line holds information on the parameters. It starts with the equation name indicated with a `#` sign, then followed by a comma-separated list of:

name: Name of the parameter, e.g. "Amplitude" (in quotes)

unit: Unit of the parameter, e.g. "V" (in quotes)

value: Starting value of the parameter. Note that the starting values are important to ensure faithful convergence of the fit algorithm.

status: Indicator specifying if the parameter is to be kept constant during the fit ("hold"), or is optimized ("fit").

store: Specifies whether the parameter is going to be stored as fit result ("store") or not ("nostore").

The following controls are used to handle equations and the equation file:

Equation Name: In this field the name of the active equation is shown. It can be edited to store a new or a modified equation under a new name.

Select Equation: This is a list of the available equations found in the `Equation File`; it is used to select an equation to be shown in the editor.

Save Equation: The equation presently active in the editor is stored to

file under the given equation name.

?: Show help information in the parser syntax in the `Notebook Window`.

LOAD/SAVE: Load or save a new `Equation File`.

Cancel: Leave the `Equation Editor` without any changes.

Done: Leave the `Equation Editor` and return to `TraceFit` or `SeriesFit` with the active equation loaded.

> ***Note:*** *Equation strings up to a length of 1024 can be edited in the* ***`Equation Editor`*** *and/or provided via the* ***`Equation File`****. However, the equation name in the* FitMaster *analysis file structure only has a length of 256 characters. Thus, if longer equation strings are encountered, the equation string is trimmed and the name of the equation is appended as a comment. In this way, the original full equation string can always be retrieved from the* ***`Equation File`****.*

7.1.3 Polynomial

A Polynomial of a given order n (maximum = 9). This fit does not require any iteration as the result can be directly calculated. $Coeff_i$ are the fit parameters.

$$y(x) = \sum_{i=0}^{n} \mathrm{Coeff}_i \, x^i$$

> ***Note:*** *A polynomial with the order of 0 is a constant, with the order of 1 a straight line, and the order of 2 will result in a parabola.*

The derived parameters `X peak` and `Y peak` denote the x and y values of the first detected peak; this only applies for polynomials with the order of 2 or greater. `X(0)` gives the x value of the first zero crossing within the cursor range. In case such derived parameters cannot be calculated, the corresponding "skip" flags are set.

7.1.4 Exponential

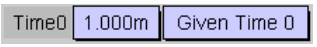
Exponential with the steady-state value Amp_0 and a maximum of 3 components. The amplitudes Amp_i are the linear, the time constants Tau_i the non-linear parameters of this function.

$$y(x) = Amp_0 + \sum_{i=1}^{3} Amp_i \, e^{\frac{-x}{Tau_i}}$$

The derived parameter `X(0)` denotes the x value where the fit curve crosses zero within the cursor limits. `Y(0)` is the y value extrapolated to time zero. `Integral` is the integral of the fit function within the cursor range.

7.1.4.1 Time Base

For exponential functions in `TraceFit` the time base can be selected. The options are:



- Segment Start – Time starts at the left edge of the relevant segment.
- Cursor Position – Time starts at the left cursor.
- Given Time 0 – Time starts at a value (counting from the beginning of the `Trace`) specified in the `Time0` field.

7.1.5 1-Exponential

One minus Exponential with the steady-state value Amp_0 and a maximum of 3 components. The amplitudes Amp_i are the linear, the time constants Tau_i the non-linear parameters of this function.

$$y(x) = Amp_0 + \sum_{i=1}^{3} Amp_i \, \left(1 - e^{\frac{-x}{Tau_i}}\right)$$

Derived parameters and `Time0` fields are as for `Exponential`.

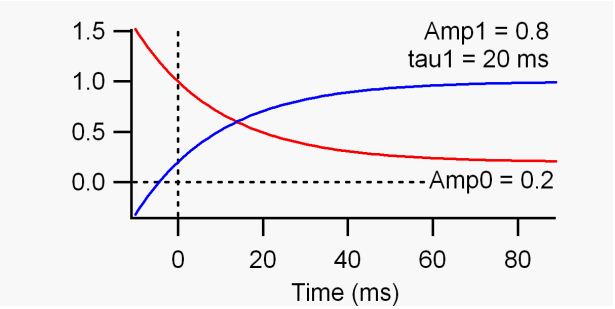


Figure 7.3: Example of Exponential (red) and 1-Exponential (blue) functions with one component.

7.1.6 Hodgkin-Huxley Formalism

This equation is used to fit ion channel activation traces according to the classical Hodgkin-Huxley formalism. This means that the current flowing through an ensemble of ion channels upon a step depolarization is described by a number of independent activation (m) and inactivation (h) gates. In the function provided by FITMASTER there are the additional options of including a time offset ($Delay$), gating charge (Q_{on}) relaxation, and a second inactivating component (τ_{h2}). The time constant for activation gates are τ_m, the time constant for the primary inactivation gates τ_h. h_∞ denotes the steady-state non-inactivating fraction of current, i.e., $h_\infty = 0$ means that the current is completely inactivating, $h_\infty = 1$ stands for no inactivation. r_{12} is the fraction of channels that inactivates according to the primary inactivation process. Thus, $1 - r_{12}$ is the fraction of channels that inactivates according to the secondary inactivation process. If all features of this function are selected, it looks like the following:

$$
\begin{aligned}
t &= x - Delay \\
I_{gat}(t) &= \frac{Q_{on}}{\tau_m}\, e^{\frac{-t}{\tau_m}} \\
r_{12} &= \frac{a_1}{a_1 + a_2}
\end{aligned}
$$

$$
\begin{aligned}
m(t) &= 1 - e^{\frac{-t}{\tau_m}} \\
h(t) &= h_\infty + (1 - h_\infty)(r_{12}\, e^{\frac{-t}{\tau_h}} + (1 - r_{12})\, e^{\frac{-t}{\tau_{h2}}}) \\
\\
I(t) &= I_{gat} + I(0)\, m(x)^m\, h(x)^h
\end{aligned}
$$

The peak current at time zero, $I(0)$, is treated as a linear parameter, all other parameters are non-linear. The time starts at the left edge of the selected relevant segment.

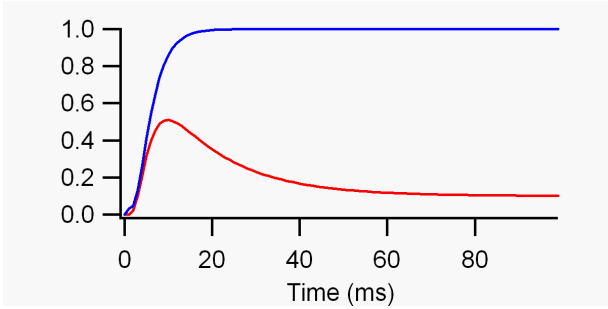


Figure 7.4: Example of Hodgkin&Huxlex functions with (red) and without (blue) an inactivating component. Parameters are: $I(0) = 1$, $Delay = 1ms$, $m = 3$, $\tau_m = 3\,ms$, red curve: $h = 1$, $\tau_h = 15\,ms$, $h_\infty = 0.1$; blue curve: $Q_{on} = 0.1$.

> ***Note:*** *The fit is not constrained to physically realistic values. Thus, for example, it can happen that Delay becomes negative or h_∞ becomes greater than 1 or smaller than 0. For some purposes this flexibility can be useful. If the parameters drift into an unrealistic region, one needs to manually constrain them by using the* `hold` *function.*

The derived parameters `X peak` and `Y peak` are the coordinates of the first detected minimun or maximum. `X(0)` is the x value of the first zero crossing.

7.1.7 Gaussian

A sum of up to three Gaussian components can be fitted here. Amp_i are the amplitudes of the Gaussian distributions, $Mean_i$ are the centers of the distributions on the X-axis, and SD_i are the standard deviations of the distribution (widths). Amp_i are the linear, $Mean_i$ and SD_i the non-linear parameters of this function.

$$y(x) = \sum_{i=1}^{3} Amp_i \, e^{\frac{-(x-Mean_i)^2}{SD_i^2}}$$

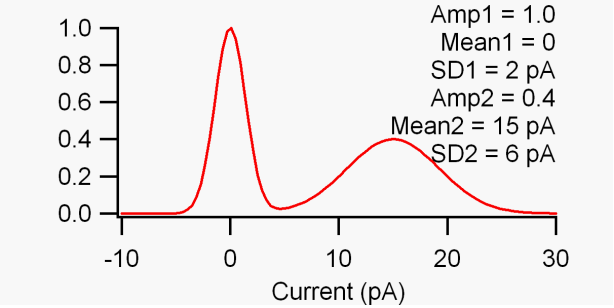


Figure 7.5: Example of Gaussian function with two components

`Integral` is the integral of the fit function within the cursor range.

7.1.8 Rise Time

This function does not provide a fit, it is only a tool to measure time intervals that elapse while currents change from a `Low Level` to a `High Level`. These numbers are specified in percent. 100 percent is given by the maximal (or minimal) current value found inside the cursor range. Minima and maxima are found with respect to a `Baseline` current. This is specified according to the following settings of the status flag:

- fit – The baseline current value is measured automatically according to the cursor settings specified below. These settings are % values with respect to the relevant segment. I.e., if the baseline is to be determined *before* the relevant segment, the cursor settings could be −10 and 0, for example.

- hold – In this case the baseline current as input by the user into the `Baseline` field is going to be taken.

- off - The baseline current is set to zero.

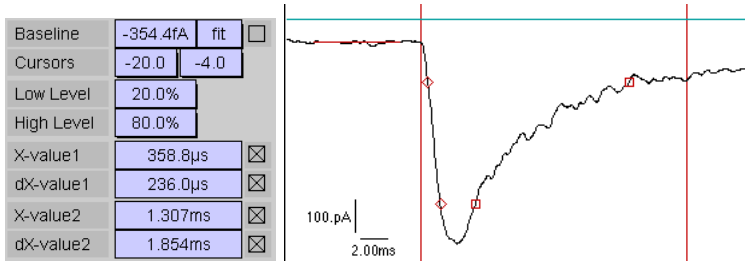


Figure 7.6: RiseTime fit. Left: Parameter list. Right: Example showing the determined basline current and level crossings superimposed to the data trace.

Cursors: These are the left and right bounds used to measure the baseline current. The values are given in % of the relevant segment. The baseline is drawn into the `Oscilloscope` window upon a fit operation.

`RiseTime` fit can produce up to five results:

Baseline: The baseline current; in particular useful as a result when determined automatically.

X-value1: This is the time from the relevant segment start to the first crossing of the `Low Level` threshold.

dX-value1: This is the risetime, i.e. the time interval from the first crossing of the `Low Level` threshold to the first crossing of the `High Level` crossing.

X-value2: This is the time from the relevant segment start to the first crossing of the `High Level` threshold *after* the maximum/minium of the selected fit range.

dX-value2: This is the *falling* risetime, i.e. the time interval from the first crossing of the `High Level` threshold to the first crossing of the `Low Level` crossing *after* the maximum/minium of the selected fit range. Note that *rising* or *falling* do not refer to the current direction; when a minimum is detected like in the example shown, *rising* means increasing negative current.

The times determined have a higher precision than the sample interval because they are linearly interpolated between adjacent data points. In case a *falling* threshold crossing cannot be detected (if there is no inactivation), the corresponding values are set to zero and the status flag to "skipped".

The example shows a typical application of a `RiseTime` measurement with an activating (dX-value1) and an inactivating (dX-value2) component. The events found are indicated in the `Oscilloscope` window: rising threshold crossings are indicated by diamonds, falling threshold crossings by squares. In addition, the baseline current is indicated by a line within the cursor limits used to measure the baseline current.

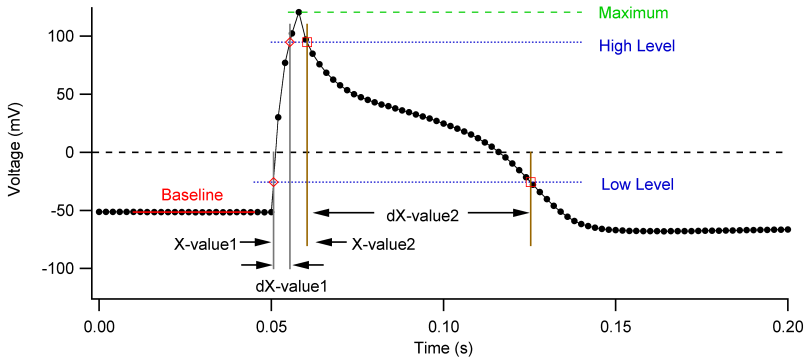
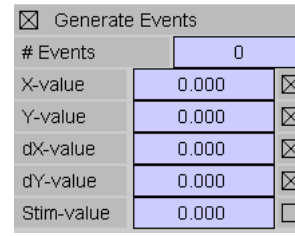


Figure 7.7: Evoked action potential with explanation of the parameters used in risetime determinations.

7.1.9 Measure

This function does not provide a fit, it is a tool for manually measuring point coordinates and distances with the aid of the mouse.



Generate Events: If this checkbox is selected, multiple measurements are stored to the analysis tree as `Events`. If it is not selected, the measurements are stored as `Results`.

Events: Shows the actual number of `Events` associated with the active `Trace`.

X-value: Measures the X-coordinate of the mouse position with respect to the beginning of the stored data when releasing the mouse key.

Y-value: Measures the Y-coordinate of the mouse position when releasing the mouse key.

dX-value: Measures the distance along the X-axis between start point (first click on the mouse key) and end point (release of the mouse key).

dY-value: Measures the distance along the Y-axis between start point (first click on the mouse key) and end point (release of the mouse key).

Stim-value: Determines the stimulation value associated to the measured X-value (e.g., determines the voltage within a ramp segment that corresponds to the time measured).

After selection of the `Sweep` for analysis, start the measure mode by clicking on the `Fit` button in the `TraceFit` window. This activates the `Measure` mode in the `Oscilloscope` window. You can then perform two types of measurements:

- **Single Point Measurement:** Move the mouse tip to the point of interest an perform a single mouse click. Upon accepting this coordinate with `To TraceFit` in the oscilloscope window, the coordinates of this point will be displayed as `X-value` and `Y-value` in the `Measure` parameters of the `TraceFit` window and the point will be marked in

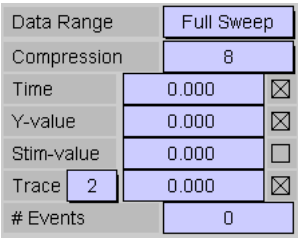
the oscilloscope.

- **Measurement of Distances:** Move the mouse tip to the starting point, press the mouse key and keep it pressed. Then move the mouse tip to the end point. Finalize the measurement by releasing the mouse key. Upon accepting these coordinates with `To TraceFit` in the oscilloscope window, the final coordinate (end point) will be stored as `X-` and `Y-value` and the displacements as `dX-value` and `dY-value`. In addition, the point and displacement will be marked in the oscilloscope.

Use the up and down arrow in the `TraceFit` window to select the next sweep for analysis and perform the next measurement starting with pressing the `Fit` button.

7.1.10 Trace Copy

Extract raw data from the `Trace` and store them as `Events` in the analysis tree.



Data Range: A data range can be specified. You can select between `Full Sweep` and `Cursor Bounds`.

Compression: Specifies the compression factor. The copied data are the mean values averaged for `Compression` points.

Time: Time of data point relative to the start of the `Trace`. (In case the X-axis does not hold time values, this is just the X-value.)

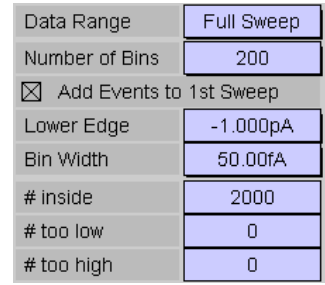
Y-value: Averaged Y-value.

Stim-value: Equivalent stimulation value, e.g. voltage in a ramp segment.

Trace, Trace Number: Here you can select a second `Trace` by specifying its number. The average of this trace will be computed according to the compression of the primary `Trace`. Note that this only yields reasonable results if the second trace has the same sample interval and segmenting as the first one.

#Events: Displays the number of `Events` stored.

7.1.11 Amplitude Histogram



Compiles a point-to-point histogram.

Data Range: A data range can be specified. You can select between `Full Sweep` and `Cursor Bounds`.

Number of Bins: Defines the number of bins for the histogram.

Add Events to 1st Sweep: If this flag is on, all `Events` of a `Series` are added to the events list of the first `Sweep`. This feature is very useful for reducing the file size.

Lower Edge: Defines the X-value of the first bin.

Bin Width: Defines the width of the bins.

Upon execution of "Fit", it will be analyzed how many events are inside the ranges of the histogram.

 # inside: Number of events that are inside the histogram range.

 # too low: Number of events that are too small.

 # too high: Number of events that are too large.

> ***Note:*** *Such histograms are compiled separately for each Sweep. Histograms from individual Sweeps can then be accumulated on the level of* `SeriesFit`*.*

The output of the amplitude histogram is stored as follows.

Storage on the level of an `Event` of an individual trace:

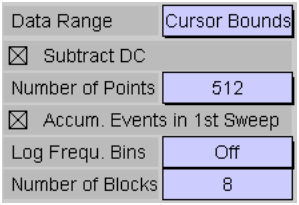
- **Amplitude:** Contains the mean value of the bins; usually used as X-axis in the `SeriesFit` graph.

- **Count:** Contains the number of events per corresponding bin; usually used as Y-axis in the `SeriesFit` graph.

Storage on the level of a `Result` of an individual trace:

- **Amplitude:** Contains the number of events that fall outside the histogram at the low side (`# too low`).
- **Count:** Contains the number of events that fall outside the histogram at the high side (`# too high`).

7.1.12 Power Spectrum



The Power Spectrum is used to perform a spectral analysis of raw data traces. The power spectrum is computed in sections of the indicated number of data points as they are displayed (i.e. after digital filtering, leak and zero subtraction). Points have to be a power of 2 between 16 and 1,024. For example, if the cursors delimit a data range of 1,000 samples and Points is set to 256, then three power spectra are computed and accumulated (i.e. Number of Blocks is 3). The trailing 232 sample points are not considered for analysis.

Data Range: A data range can be specified. You can select between `Full Sweep` and `Cursor Bounds`.

Subtract DC: Optionally, the mean value of the data can be subtracted before calculation of the power spectrum. This is required to get a meaningful value of the power spectrum at the frequency of 0 Hz.

Number of Points: Number of data points to be taken for computation of the power spectrum. The pull-down menu contains the available numbers that can be selected.

Accum. Events in 1st Sweep: If this flag is on, all `Events` of a `Series` are accumulated to the events list of the first `Sweep`, i.e. mean spectra are generated. This feature is very useful for reducing the file size.

Log Frequ. Bins: Enter a number of bins that are created uniformly distributed on a logarithmic frequency scale. This improves the fit of the power spectrum on the level of `SeriesFit`.

Number of Blocks: Number of data blocks that were used for the computation of the spectrum.

> ***Note:*** *The power spectra are calculated and stored for each sweep separately. Power spectra from individual sweeps can then be averaged on the level of* `SeriesFit`*.*

The output of the power spectrum is stored as follows.

Storage as `Events` of an individual trace:

- **Frequency:** Contains the frequency of the power spectrum; usually used as X-axis in the `SeriesFit` graph.
- **SpecDens:** Spectral density of the power spectrum in units of A^2/Hz; usually used as Y-axis in the `SeriesFit` graph.

Storage as `Result` of an individual trace:

- **SpecDens:** Contains the `Number of blocks` per trace. I.e. number of averaged spectra per trace.

7.1.13 Single Channel

When this function is selected, the data sweeps are searched for single-channel events according to the definitions specified in the `Event Detection` dialog.

Please refer to chapter *Event Detection* on page 97.

7.1.14 Action Potential

When this function is selected, the data sweeps are searched for action potential events according to the definitions specified in the `Event Detection` dialog.

Please refer to chapter *Event Detection* on page 97.

7.2 SeriesFit Functions

Available `SeriesFit` functions are:

- Parsed Equation (see section 7.1.2 of `TraceFit` functions)
- Polynomial (see section 7.1.3 of `TraceFit` functions)
- 2 Regressions
- Exponential (see section 7.1.4 of `TraceFit` functions)
- 1-Exponential (see section 7.1.5 of `TraceFit` functions)
- (Exp(x-x0))^n
- Boltzmann
- Dose-Response
- Gaussian (see section 7.1.7 of `TraceFit` functions)
- Current-Voltage
- Spectra

7.2.1 Two Regressions

With this function one can fit two linear functions to the data in a two-step process. Both linear functions have two parameters, the value at $x = 0$ (Coeff0) and the Slope (Coeff1). When the checkbox "2nd Component" is *off*, the first component is fitted, otherwise the second. The individual fits result in individual residuals (Residual1 and Residual2). The crossing of the two lines is calculated: X intersept and Y intersept.

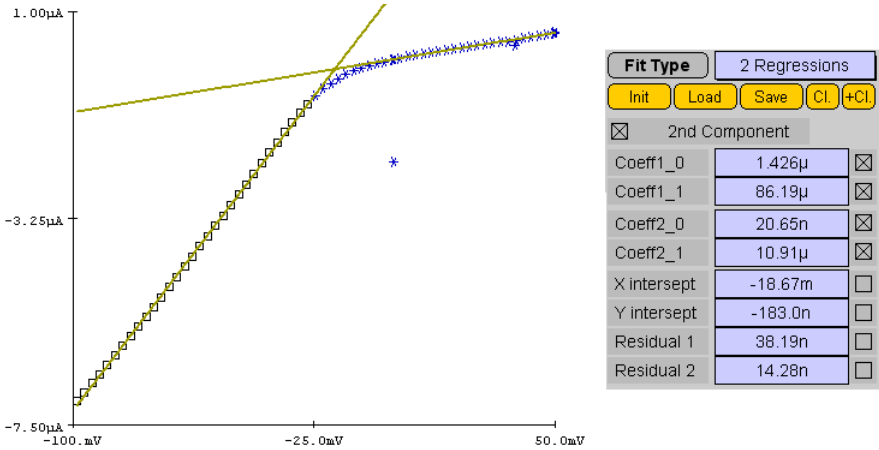


Figure 7.8: Left: Fit of two regression lines to a data set. The second component was fit to the currently active data points. Right: Dialog for "2 Regressions".

7.2.2 General Exponential

In this General Exponential function the X-origin is free to fit. In addition, the mode (i.e. exponential or 1-exponential) can be selected as well as an exponent for each exponential. With this function many biologically relevant processes can be described. If for all exponential components the checkbox $1 - exp_i$ is selected, the fit function looks as follows:

$$\begin{aligned} t &= x - x_0 \\ y(t \le 0) &= Amp_0 \\ y(t > 0) &= Amp_0 \\ &+ \sum_{i=1}^{3} Amp_i \left(1 - e^{\frac{-t}{Tau_i}}\right)^{n_i} \end{aligned}$$

Amp_0 is a linear, all others are non-linear parameters of this function.

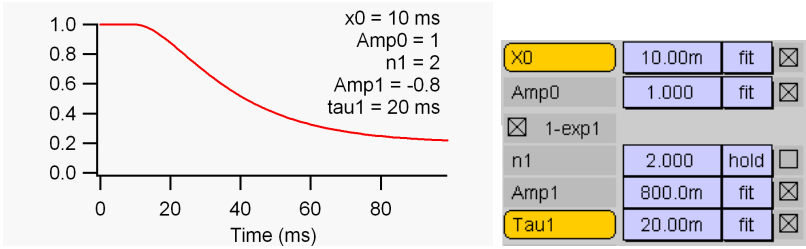


Figure 7.9: Example of General Exponential with one component in the "1-exp" mode.

7.2.3 Boltzmann

This function is used to fit Boltzmann distributions, e.g. voltage-dependence of steady-state activation/inactivation curves. It has up to 2 components with the amplitudes Amp_i, the mid-points x_{hi}, and the slope factors $Slope_i$. Each component can have an INTEGER exponent n_i. In addition, an *Offset* is provided. The *Offset* and the amplitudes are linear parameters, the others are non-linear.

$$y(x) = \text{Offset} + \sum_{i=1}^{2} \frac{Amp_i}{\left(1 + e^{\frac{-(x-x_{hi})}{Slope_i}}\right)^{n_i}}$$

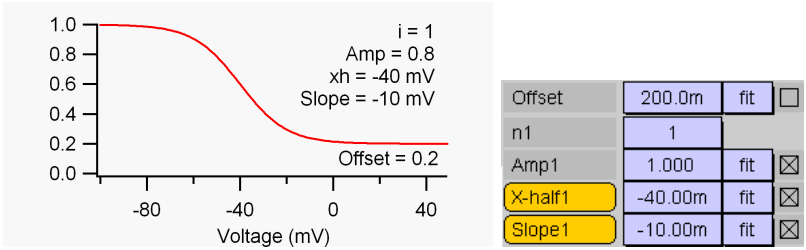


Figure 7.10: Example of Boltzmann function with one component

7.2.4 Dose-Response

This function describes a Hill equation with an amplitude *Amp*, an *Offset*, a dissociation constant KD, and a Hill coefficient *Hill*. The amplitude and the offset are linear parameters, the others are non-linear.

$$y(x) = \text{Offset} + \frac{Amp}{1 + \left(\frac{x}{KD}\right)^{Hill}}$$

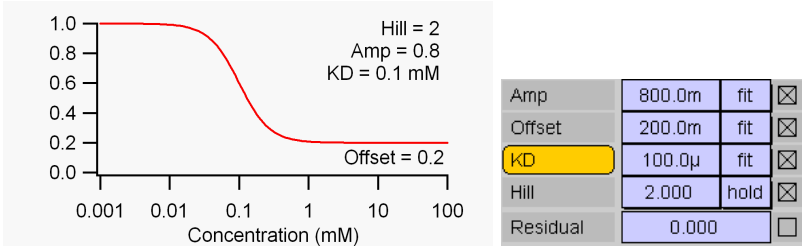


Figure 7.11: Example of Dose-Response function with logarithmic X-scaling

7.2.5 Current-Voltage

This function is used to describe the voltage-dependent activation of ion channels. The open-probability of the channels, P_{open}, is calculated according to a Boltzmann distribution with m independent gating particles.

$$P_{open}(x) = \frac{1}{\left(1 + e^{-(x - V_m)/k_m}\right)^m}$$

with the half-activating voltage per subunit, V_m (or V-half), and the slope factor, k_m (or Slope). The maximal current is determined by a total conductance, Γ (Conduct.), multiplied with the driving force, which contains the single-channel iV characteristic. This can either be Ohmic or according to Goldman-Hodgkin-Katz (GHK).

- Ohmic driving force:

$$v(x) = x - E_{rev}$$

- Goldman-Hodgkin-Katz driving force:

$$v(x) = x \, \frac{1 - e^{-(x-E_{rev})/25mV}}{1 - e^{-x/25mV}}$$

with the reversal potential, E_{rev}. Note that the variable x is supposed to be a voltage, measured in volts (V). Thus, the constants in the GHK equation reflect kT/e, i.e. 25.5 mV at room temperature.

In addition to channel activation and permeation characteristics, the IV-fit also provides means for simulating voltage-dependent channel block at either low or high voltages. These block functions are simple Boltzmann distributions for charged blocking particles:

$$P_{PosUnbl} = \frac{1}{1 + e^{(x-V_{PosBl})/k_{PosBl}}}$$

and

$$P_{NegUnbl} = \frac{1}{1 + e^{-(x-V_{NegBl})/k_{NegBl}}}$$

In total, this yields a function for the current as a function of voltage, or generally, for $y(x)$:

$$y(x) = \Gamma \, v(x) \, P_{open} \, P_{PosUnbl} \, P_{NegUnbl}$$

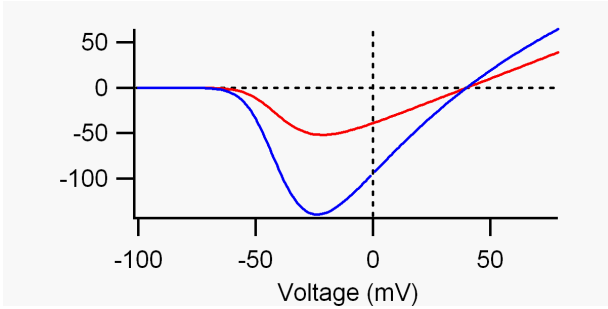


Figure 7.12: Example Current-Voltage relationship for sodium channels. The single-channel characteristics was chosen to be linear (red) and of Goldman-Hodgkin-Katz type (blue). The other parameters were: $\Gamma = 1$, $E_{rev} = 40\,mV$, $m = 3$, $V_m = -50\,mV$, $k_m = 10\,mV$.

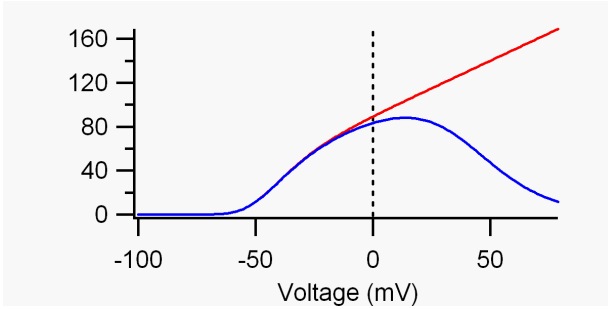


Figure 7.13: Example Current-Voltage relationship for potassium channels. The single-channel characteristics was chosen to be linear. In the red curve no block was considered, while the blue curve contains a block at positive voltages with $V_{block+} = 40\,mV$, $k_{block+} = 15\,mV$. The other parameters were: $\Gamma = 1$, $E_{rev} = -90\,mV$, $m = 4$, $V_m = -60\,mV$, $k_m = 10\,mV$.

7.2.6 Spectra

This function is used to fit power spectra. It describes the dependency of the spectral density on the frequency.

The spectral density can be expressed as the sum of three functions:

Shot Noise: A constant noise contribution.

$$S(x) = S_{inf}$$

Lorentzian:

$$S(x) = \frac{SL_0 1}{1 + (\frac{x}{fL_c 1})^2} + \frac{SL_0 2}{1 + (\frac{x}{fL_c 2})^2}$$

One over Frequency (1/f):

$$S(x) = \frac{SO_0}{1 + (\frac{x}{fO_c})}$$

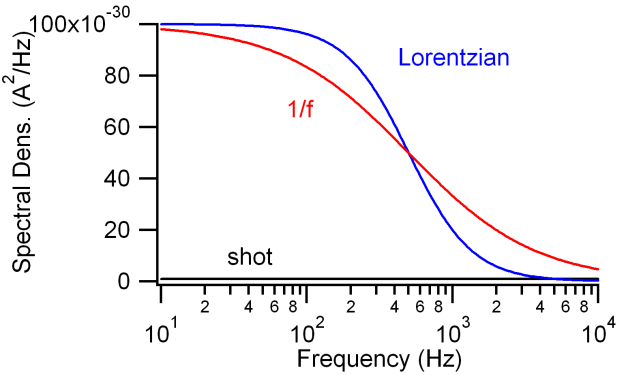


Figure 7.14: Spectra fit functions. Black: Shot noise with $S_{inf} = 10^{-30}$; red: One-over-f noise with $SO_0 = 10^{-28}$ and $fO_c = 500\ Hz$; blue: Lorentzian with $SL0_1 = 10^{-28}$ and $fL_{c1} = 500\ Hz$.

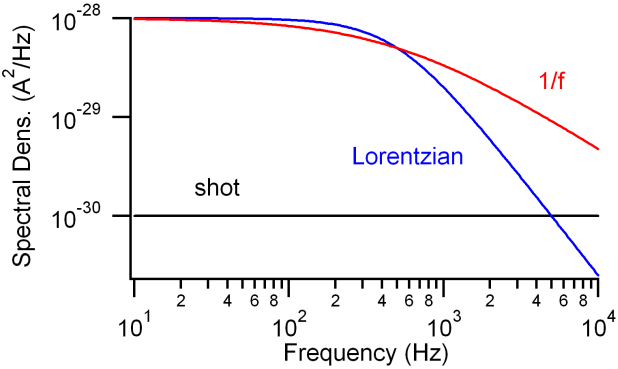


Figure 7.15: Spectra fit functions. As figure 7.14 but as log - log plot.

Chapter 8

Event Detection

In this chapter we describe how event detection is performed in FITMASTER.

8.1 Overview

`Events` in the terminology of FITMASTER refers to a data structure in the analysis file. These are values appended to the results. I.e. for each `Sweep` in a data file, we can have many `Parameters` and `Results`. These structures are then extended by an arbitrary number of `Events`. *Events* in a physical sense are certain structures in a stream of data (e.g. current versus time) that have a certain property. A typical application is a single-channel event, i.e. an jump in a single-channel current signal from one level to another. Such a physical *event* can, for example, be described by two numbers, the time when the transition happens and a current level. In FITMASTER we would then define two data structures holding this information. For other applications, e.g. detection and characterization of an action potential, two parameters may not be sufficient. In such cases more data are stored simultaneously (in form of `Events`) to describe the physical *event* (e.g. by specifying time, duration, amplitude, slope, after-potential, etc.). Thus, the FITMASTER data structure provides full flexibility for defining *events* of arbitrary complexity. Therefore, the first thing to be done in `Event Detection` is to select the type of detection and then to specify by which parameters the detected event is to be characterized.

Presently the following options for event detection are available:

- Single Channels: Analyze single-channel events.

- Action Potential: Analyze action potentials.

The data structure of `Events` are also used by the `TraceFit` functions `Measure`, `Trace Copy`, `Amplitude Histogram`, and `Power Spectrum`. They are presented and discussed separately.

When executing `Fit` or `AutoFit`, the `Event Detection` dialog opens.

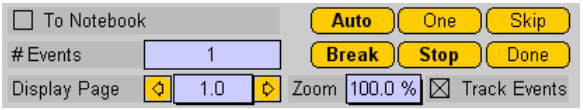


Figure 8.1: Common functions of the Event Detection dialog.

All event types have the following common controls:

To Notebook: If checked, the event parameters are written to the notebook.

 # Events: Current number of events detected.

Do All: This button only appears when `Event Detection` dialog was called via the `AutoFit` function. It starts an automatic analysis of the current `Trace` (see *Auto* below) and proceeds through the rest of the selected `Series`. This operation can be discontinued with the `Stop` button.

Skip: Mark the detected event as "skipped". It will then be excluded from the `SeriesFit` analysis. In the `Oscilloscope` window the corresponding `Event` will be indicated in black. Note that only the very last `Event` can be skipped with this function. "Earlier" `Events` can only be skipped on the level of the `SeriesFit` analysis.

One: Analyzes just the next event and indicate it in the `Oscilloscope` window with symbols.

Auto: Automatically detects all events in the given data range.

Break: Interrupt analysis for the selected `Trace`.

Stop: Interrupt analysis for the selected `Trace` and also interrupt analysis of a `Series` in case analysis was started via the `AutoFit` function.

Done: Closes the dialog.

For the analysis of a large number of `Events` it is very useful to assign keys to the *One*, *Auto*, and *Skip* buttons. For key assignments please refer to the PatchMaster manual.

Parameters and results for the different event types are defined or listed in the corresponding panes.

8.1.1 Display Controls

The last line on the `Event Detection` dialog provides buttons to control the display while performing event detection. Upon return to `TraceFit`, the original display settings are restored. The display is controlled via two variables: a `Page` number and a `Zoom` factor. An entire selected data traces fills exactly one page when the zoom factor is 100%. When the zoom factor is 5%, the entire data trace extends over a total of 20 pages.

Display Page: The currently displayed page. 1.0 means that the first page is displayed, i.e. the left edge is the beginning of the trace. 2.5 means that the left edge is set to 1.5 pages. With the arrows one can navigate through the trace in steps of 0.5 pages.

Zoom: Zoom factor (in %) of the display's X-coordinate. When the zoom factor is changed, the left edge of the visible trace remains at the left edge of the display. As a consequence, `Display Page` will change simultaneously.

Any redraw operation, i.e. by changing either `Display Page` or `Zoom`, will erase all but the symbols of the very last two `Events`.

Track Events: When this checkbox is activated, the display is automatically adjusted to show the last detected `Event` (provided that `Zoom` is smaller than 100%. This function uses an imaginary page margin of 15%. When the X-coordinate of a new `Event` is outside the right margin and still inside the next page minus two margins, the data trace is scrolled forward by one page minus one margin. If the new `Event` is even outside that range, the data trace is scrolled forward such that the new `Event` will be displayed at the left margin.

8.2 Single Channels

On the `TraceFit` level the following parameters are defined:

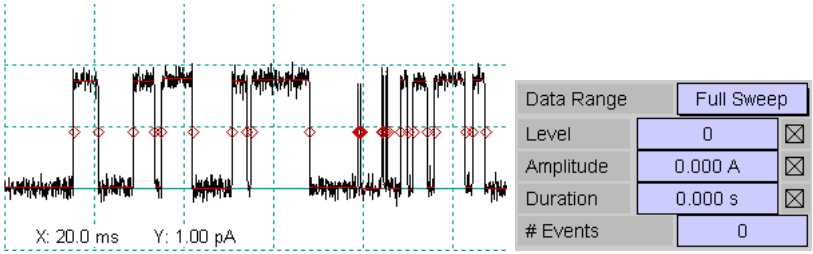


Figure 8.2: Example of single-channel analysis and `TraceFit` parameters for single-channel events.

Data Range: A data range can be specified. This can be either of the following:

- Full Sweep: Take the entire data sweep (selected trace).
- Cursor Bounds: Analyze the trace data within the cursor bounds specified in the `TraceFit` window.

Results: The results of the single-channel detection are listed. By checking the box on the right you can decide to store the corresponding result and make it available for further analysis on the `SeriesFit` level.

 # Events: The number of detected events will be given in the last line.

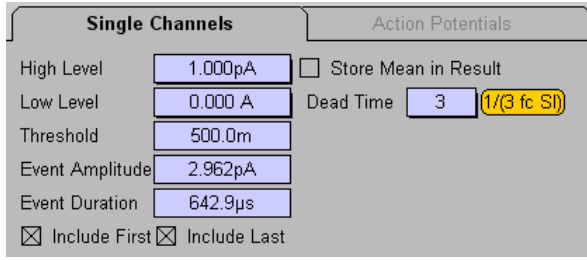


Figure 8.3: Single-channel event detection dialog.

High Level: Defines the high level of the event. Enter a value or drag the entry. The level will be graphically displayed in the `Oscilloscope` window.

Low Level: Defines the lower level of the event. Enter a value or drag the entry. The level will be graphically displayed in the `Oscilloscope` window.

Threshold: Defines the threshold between `Low Level` and `High Level`. Currently this value is fixed at 0.5 (50%).

Event Amplitude: Measured event amplitude.

Event Duration: Measured event duration.

Include First: Option to include (if checked) the first event of a trace.

Include Last: Option to include (if checked) the last event of a trace.

Store Mean in Result: Option to store the mean value of all events acquired in the `Result` data structure.

Dead Time: Number of sample points before and after a threshold crossing to be discarded before measuring the event amplitude.

1/(3 fc SI): With this button a `Dead Time` is calculated according to the bandwidth of the data trace and the selected digital filter setting: TRUNC(1/(3 f_c SI)) with the sample interval SI, and f_c, which is the cut-off frequency determined as the lowest bandwidth of the `Trace` data (e.g. as set by `Filter-2` of an EPC-10) and the digital filter setting as specified in the `Oscilloscope` window.

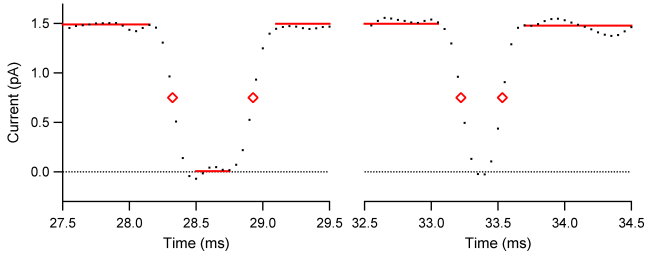


Figure 8.4: Single-channel analysis using a `Dead Time` of 3. Left: A closing event can be detected; points used to measure the single-channel amplitude are indicated by the red level line. Right: This event was detected but the duration was too short. Therefore, no amplitude level is shown and the corresponding entry in the data file is set to *skipped*. Also see how the threshold crossings, indicated by the red diamonds, are interpolated between the individual data points.

> ***Note:*** *A potential problem with single-channel detection is that the level of the very first event is not found properly. In order to avoid endless loops there is a maximum level of* ±1000. *If this maximum is exceeded the error message* "Unreasonable level. Check Low/High level settings." *appears and the search operation is interrupted. In most cases this error occurs when* `Low Level` *and* `High Level` *are not meaningful. Make sure* `Low Level` *marks the baseline, i.e. a current level where all channels are closed and* `High Level` *the current level where* one *channels is open.*

8.3 Action Potentials

On the `TraceFit` level the following parameters are defined:

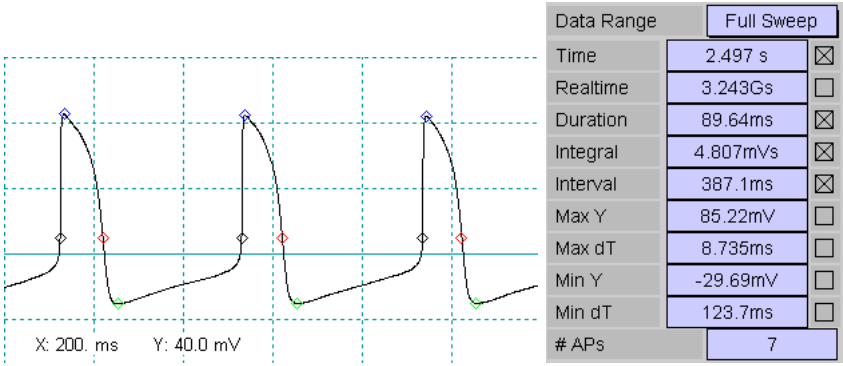


Figure 8.5: Example of Action Potential analysis and TraceFit parameters for Action Potential events.

Data Range: A data range can be specified. This can be either of the following:

- Full Sweep: Take the entire data sweep (selected trace).
- Cursor Bounds: Analyze the trace data within the cursor bounds specified in the `TraceFit Window`.

Results: The results of the action potential detection are listed. By checking the box on the right you can decide to store the corresponding result and make it available for further analysis on the `SeriesFit` level.

 # APs: The number of detected action potentials will be given in the last line.

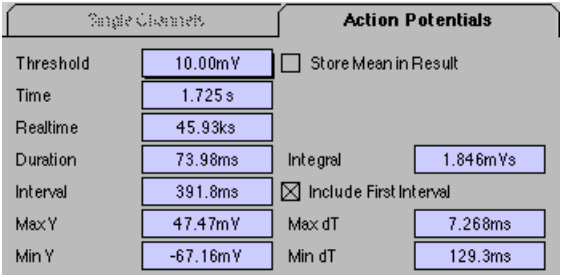


Figure 8.6: Action Potential event detection dialog.

Threshold: Defines the threshold for detection of the action potential. Enter a value or drag the entry. The level will be graphically displayed in the `Oscilloscope` window.

Time: Time of the detected action potential within the active `Trace`.

Realtime: Real time of the detected action potential.

Duration: Duration of the action potential at the level of the threshold.

Integral: Integral of the action potential from the first to the second threshold crossing, relative to the threshold amplitude.

Interval: Interval between two action potentials.

Include First Interval: Option to include (if checked) the first interval from the beginning of the `Trace` to the first action potential.

Max Y: Peak amplitude of the action potential.

Max dT: Duration from the first threshold crossing to the peak of the action potential.

Min Y: Minimum value during the repolarization phase.

Min dT: Duration from the first threshold crossing to the minimum of the repolarization phase.

Store Mean in Result: Option to store the mean value of all events acquired in the `Result` data structure. This is quite useful if, for example, mean action potential parameters are to be analyzed from a `Series` of

Traces as a function of stimulating current injections.

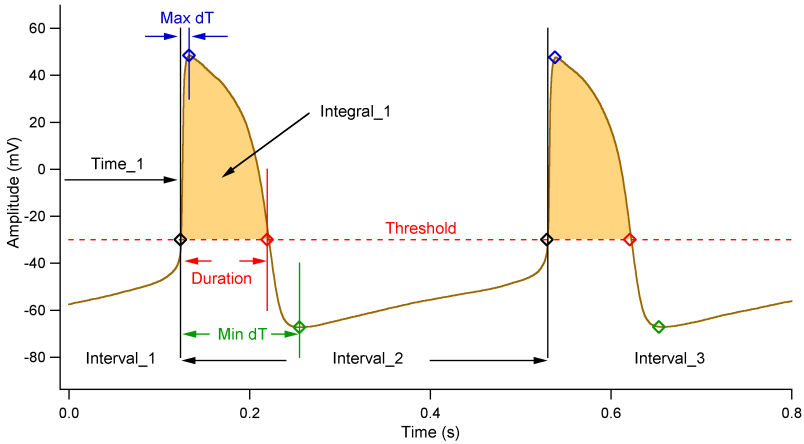


Figure 8.7: Cardiac action potentials with explanation of the parameters used.

8.4 Event Analysis in SeriesFit

Events, once detected and stored in the data file, can be analyzed to some extend using the SeriesFit functions. For this purpose, events can be loaded into the waves buffer by selecting the appropriate data source.

8.4.1 Example 1: All-Points Histogram

Suppose a Series with 10 Sweeps of single-channel data was analyzed in TraceFit with Amplitude Histograms. Then the results in SeriesFit should contain the entries "Amplitude" and "Count". Select them as X- and Y-parameter, respectively. Then select as data source "Add All Events Sweepwise" and preview all events from the entire Series. The result is an accumulated all-points histogram. An example, including a double-Gaussian fit, is shown in the following figure.

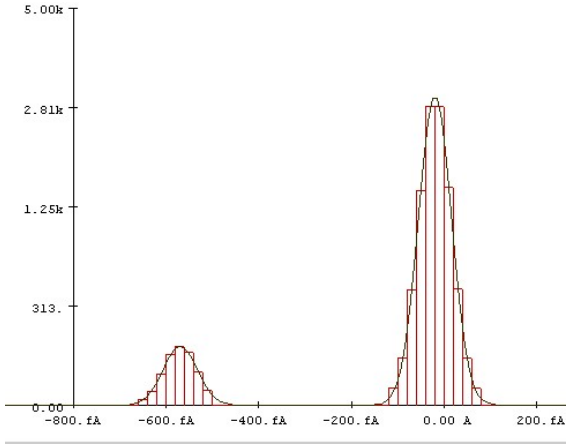


Figure 8.8: All-points histogram with superimposed double-Gaussian fit. The Y-axis is scaled according to square root.

8.4.2 Example 2: Single-Channel Open Time Distribution

Suppose a `Series` of multiple `Sweeps` of single-channel data was analyzed in `TraceFit` with `Single Channel` detection. If all result parameters have been checked on the `TraceFit` function, then the results in `SeriesFit` should contain the entries: `Level`, `Amplitude` and `Duration`. Select `Level` as X-parameter and `Duration` as Y-parameter. Then select as data source "All Events" and press `Preview`.

The durations of the low and high level are displayed in the `SeriesGraph`.

Let's assume that an open channel corresponds to the high level. Then go to the `Histogram Options` of the `SeriesFit` settings in the `Fit Configurations` and check `Select Events by Range`. Enter 0.5 as low and 1.5 as high range. The `Left Edge` set to the minimum duration (e.g. 0) and the `Bin Size` to 0.5 ms.

Now, go back to the `SeriesFit` and select as wave operation `histo preview` and then press the `Y` button next to it. The histogram of the

channel open times is displayed in the `SeriesGraph`.

For further analysis you might select the function `1-Exponential` as `SeriesFit` function and fit the exponential distribution.

Chapter 9

Keys

Many controls within windows can be accessed from the keyboard. All key commands are saved in the file `FitMaster.key` and will be read at the program start.

Note that the user can customize all commands!

Please also be aware that in case the file FitMaster.key is not available at program start, no key commands are available!

The option `Help` → `Show Keys` displays the key assignments in the various windows.

To list the keys, choose `Help` → `List Keys`.

To save the keys, choose `Help` → `Save Keys`. The keys are saved in the file `PatchMaster.key`. Old keyboard assignments will be automatically saved with an incrementing extension, e.g., .k00, .k01, .k02 ...

You can freely customize the key commands by

- editing the keys via the dialog control (see Chapter *Modifying Dialogs and Controls*, page 14) and saving them

  or by

- directly modifying the key file, e.g., in a text editor.

9.1 How to Read the Key Listing

The key macro as you can list it in the `Notebook` window uses the following notation:

; denotes remark – will not be executed

@ denotes menu command. Here you can also edit the menu entry label.

F, P.. denotes the window (F = Trace Fit, P = Pulse Generator..)

```
    ;   F -> Trace Fit
        F WindowSwitch          Key SPACE any
```

Example Listing for TraceFit keys:

```
    ...
; F -> Trace Fit
  F  WindowSwitch       Key SPACE any
  F  Clipboard          Key Char c
  F  Series             Key Char g
  F  Export             Key Char e
  F  Skip               Key CursorRight
  F  TreeUp             Key CursorUp
  F  TreeDown           Key CursorDown
  F  AutoBreak          Key Char s
  F  Break              Key Char b
  F  AutoFit            Key Char a
  F  Fit                Key Char f
```

Index