CurveExpert Professional Documentation

Release 2.7.3

Daniel G. Hyams

Oct 17, 2020

CONTENTS

CHAPTER

ONE

WHAT IS CURVEEXPERT PROFESSIONAL?

CurveExpert Professional is a cross-platform solution for curve fitting and data analysis. Data can be modelled using a toolbox of linear regression models, nonlinear regression models, smoothing methods, or various kinds of splines.

Over 60 models are built-in, but custom regression models may also be defined by the user. Full-featured publication-quality graphing capability allows thorough examination and presentation of the curve fit. The process of finding the best fit can be automated by letting CurveExpert Professional compare your data to each model to choose the best curve. The software is designed with the purpose of generating high quality results and output while saving your time in the process.

CurveExpert Professional compares very favorably to the competition and does so at a lower price; for an interactive comparison, visit http://www.curveexpert.net/products/compare-to-the-competition. The major features of the software are enumerated below:

- **Multiplatform**: runs on Windows, Mac, and Linux. Saved files are portable between platforms.
- **Easy-to-use User Interface**: most mathematically-intensive applications are very difficult to use. CurveExpert Professional has a very intuitive interface, which allows you to import your data, generate results, and create publication-quality plots with very minimal effort. In fact, to import a file takes only four clicks, and generating a battery of results with associated graphs takes two more.
- **Robust file import**: data files come in many shapes and sizes, and CurveExpert Professional makes importing your data files very easy. The smart file reader avoids non-data areas of your file dynamically, and attempts to find labels for each column of data in your file.
- **Publication quality graphs**: an arbitrary number of graphs can be created and saved. The rendering of the plots is of publication quality, with full antialiasing support and the ability to extensively customize each graph. Graphs can be saved to a variety of graphics file formats, and they may be directly copied and pasted into another application. For the purposes of 3D plotting, scatterplots, surfaces, and contour plots are supported. Graph themes allow you to customize a look that you prefer, and reuse it. Graphs are interactive, with zooming, panning, autoscaling, and view stack capability.
- **Multivariate**: linear and nonlinear regressions can be computed for datasets with multiple independent variables, without any limits.
- **Multicore**: put your extra CPU cores to work and compute results more quickly. A typical speedup is 3.5X on a 4 core system.
- **Built-in models**: over 60 built-in nonlinear models, with high-quality automatic initial guesses, are available for use. The provided models cover all of the major families.
- **Detailed Result Query**: parameter values and uncertainties, covariance matrix, parameter histories, residual plots. Forward and backward evaluation of the model's value, differentiation, integration, and table generation. Residual runs test, F-Tests, Akaike's Information Criterion Test. All results formatted beautifully in HTML reports, with fully customizable numerical formatting.

- **Custom models**: you can also define models yourself, using a very large library of built-in mathematical functions, and parameters in your models can take any name that you like. For more advanced users, a model can be as complex as necessary, as long as it can be expressed as a Python function. This allows the creation of extremely complicated models, instead of the normal one-line-equation models limitation in other software. Further, the number of parameters is unlimited.
- **Functions**: much like models, a suite of built-in functions (over 40) can be evaluated, and you can define your own custom functions as well.
- **Ranking of results**: results are automatically ranked by your choice of score, correlation coefficient, standard error, or coefficient of determination.
- **Weighting**: Support for six different weighting schemes for weighted linear and nonlinear regression.
- **Validated**: validated against the Statistical Reference Datasets Project of the National Institute of Standards and Technology. These datasets can be downloaded directly at http://www.itl.nist.gov/div898/strd/general/dataarchive.html, but are also included verbatim in the CurveExpert Professional distribution for you to use yourself.
- **Quality spreadsheet**: the built-in spreadsheet allows you to manually enter data and/or modify it with a suite of data transformation tools. Data entry and cutting and pasting capabilities are as easy as Excel.
- **Localization**: Importing data or interoperating in European-style environments (which use a comma as a decimal) is extremely easy; regional settings are automatically obeyed, or can be selectively enabled in order to match your particular workflow.
- **Logging**: a log of actions is kept across sessions of the software, in case you need to recreate a particular result. A messages pane keeps you informed of the status of every computed result.
- **Documentation**: Extensive documentation in HTML and PDF format, available both directly from the software and online at https://www.curveexpert.net/support/documentation/.

CHAPTER

TWO

INSTALLATION AND ACTIVATION

2.1 Installation

2.1.1 Windows

To install on Windows platforms, simply run the installer, either from the command line or by double clicking it in the Windows explorer. The automated installer will start and guide you through the installation process; just follow the on-screen prompts. CurveExpert Professional will install for all users (if you are the Administrator), or for you only (if you are not). This allows you to install CurveExpert Professional without any Administrator privileges.

Windows automated installs

The installer for Windows operating systems recognizes two command line switches: /S for silent installs, and /D=dirname to specify where the program should be installed. For example, in an environment where a system administrator would want to install the program silently to the directory “d:\sharedprogs”, one would do:

```
> CurveExpertPro-2.6.4-win64.exe /S /D=d:\sharedprogs
```

Windows automated uninstalls

The uninstaller for Windows operating systems can be used for silent uninstalls. If the software was installed in the directory “c:\sharedprogs”, one would do:

```
> "c:\sharedprogs\CurveExpert Professional\uninstall" /S
```

Note that there are no quotes used in the last _? option, even if there are spaces in the path… index:: single:
mac single: osx

2.1.2 Mac/OSX

To install on an Apple Macintosh, double click the downloaded .zip archive. If this file is in your Downloads folder, the installation process will start immediately. Otherwise, a .pkg file will be created. In that case, run the .pkg file by double clicking it, and the installation process will then start. Then, follow the on-screen prompts to complete your installation.

CurveExpert Professional is *not* an Apple-signed application. Therefore, **Gatekeeper** (if you have OSX Mountain Lion or later) will complain about this fact. This behavior is normal, and it is safe to ignore the complaints from *Gatekeeper*.

2.1.3 Linux

CurveExpert Professional is a 64-bit application (for 32-bit users, see the note below). To install on a Linux machine, first download the .run file from the CurveExpert Professional web site. Usually, this file will not download with the correct permissions to be run directly. So, give the file execute permissions and then run the downloaded package directly from the command line:

```
% chmod +x CurveExpertPro-2.5.3-Linux-x86_64.run
% ./CurveExpertPro-2.5.3-Linux.run
```

Alternatively, you can double-click the downloaded .run file in your file manager, but before you do this, you must give the file execute permissions by right clicking on the file, selecting *Permissions*, and checking *Give file executable permissions* (or *Allow executing file as program*) before you can run the file directly from the file manager.

Note: CurveExpert Professional is also still available as a 32-bit application for legacy users. If you have a 32-bit version, this requires 32-bit libraries to be installed, and in some cases, the 32-bit libraries must be present even before the CurveExpert Professional installation process. If you are running a pure 64-bit Linux installation, you will likely have to install your distribution's 32-bit compatibility libraries. Usually, you can install the 32-bit version of zlib, gtk libraries, and standard c++ library, and that will be enough (on Debian/Ubuntu distributions, this is packages: ia32-libs-gtk, lib32z1, and lib32stdc++6).

Then, follow the on-screen prompts to complete your installation.

CurveExpert Professional has been tested on the following Linux distributions. If a particular distribution is not listed, that does not necessarily mean that the application will not run properly; as long as the distribution is reasonably recent, the software should work with no problems.

- SuSe 11.0 [WORKS]
- SuSe 11.1 [WORKS]
- SuSe 11.2 [WORKS]
- SuSe 11.3 [WORKS]
- SuSe 11.4 [WORKS]
- Kubuntu 10.04 [WORKS]
- Kubuntu 10.10 [WORKS]
- Kubuntu 8.10 [WORKS]
- Ubuntu 10.10 [WORKS]
- Ubuntu 12.10 [WORKS]
- Xubuntu 12.10 [WORKS]
- Fedora 13 [WORKS]
- Fedora 14 [WORKS]
- CentOS 5.0 [FAILED: libc version is too old: 2.7 required]
- CentOS 5.5 [FAILED: libc version is too old: 2.7 required]
- Mandriva 2010 Spring [WORKS]
- Linux Mint Debian 6 [WORKS]
- Debian 6 [WORKS]

In the Linux arena, there are two primary desktop systems: GNOME and KDE. CurveExpert Professional integrates well with GNOME, offering a desktop icon, appropriate menu entries and icons, file association between .cxp files, and icons for .cxp files. Under KDE, desktop integration is not fully developed. Desktop icons and menu entries are created, but the icons for these do not yet work. File association also works, but the icons for .cxp files are not yet functional.

To run the software from the command line, run the **cepro.sh** script. For example, if CurveExpert Professional was installed in your home directory, the command would be:

```
~/CurveExpertPro/cepro.sh
```

2.2 Starting a Trial

If you want to evaluate CurveExpert Professional on a trial basis, you can start a trial period by selecting *Help->Activate Trial Period*; note that an Internet connection is required for the starting of the trial period, but is not required once the trial period has been activated.

Note: If you use a proxy for Internet access, please set the appropriate proxy settings in *Edit->Preferences->Advanced* (see *Proxy settings*).

This trial period is active for approximately 30 days from activation date. During this period, all features of CurveExpert Professional are accessible. When the software is out of its trial period (either before or after), most output functions are disabled, and a watermark appears on all graphs.

After evaluation via the trial period, please consider purchasing the software at http://www.curveexpert.net/order.

2.3 Purchasing

To purchase CurveExpert Professional, visit http://www.curveexpert.net/order, and choose the product that you would like to purchase.

2.4 After Purchase: licensing

If you have purchased a CurveExpert Professional license, you will receive the license information via email. Licenses are fulfilled within 24 hours of receipt; average turnaround time between purchase and license fulfillment is 2 hours.

Proceed to *Help->Enter License*, and paste the entire email into the box. The fields below the box (Name, Company, etc.) will update, and you can then press OK to complete the process.

If you are not sure how to copy and paste the email, highlight the text of the email with your mouse (click and drag), and press Ctrl+C (Cmd+C on a Macintosh). Then, just press the *Paste* button in the *Help->Enter License* window.

Thank you for your support of CurveExpert Professional!

2.5 Automated licensing

System administrators may need a means to apply the license via the command line. To do this, save the licensing email as a text file, and then pass the text file to CurveExpert Professional on the command line as follows:

```
> cepro.exe --license the_license_filename.txt --quit
```

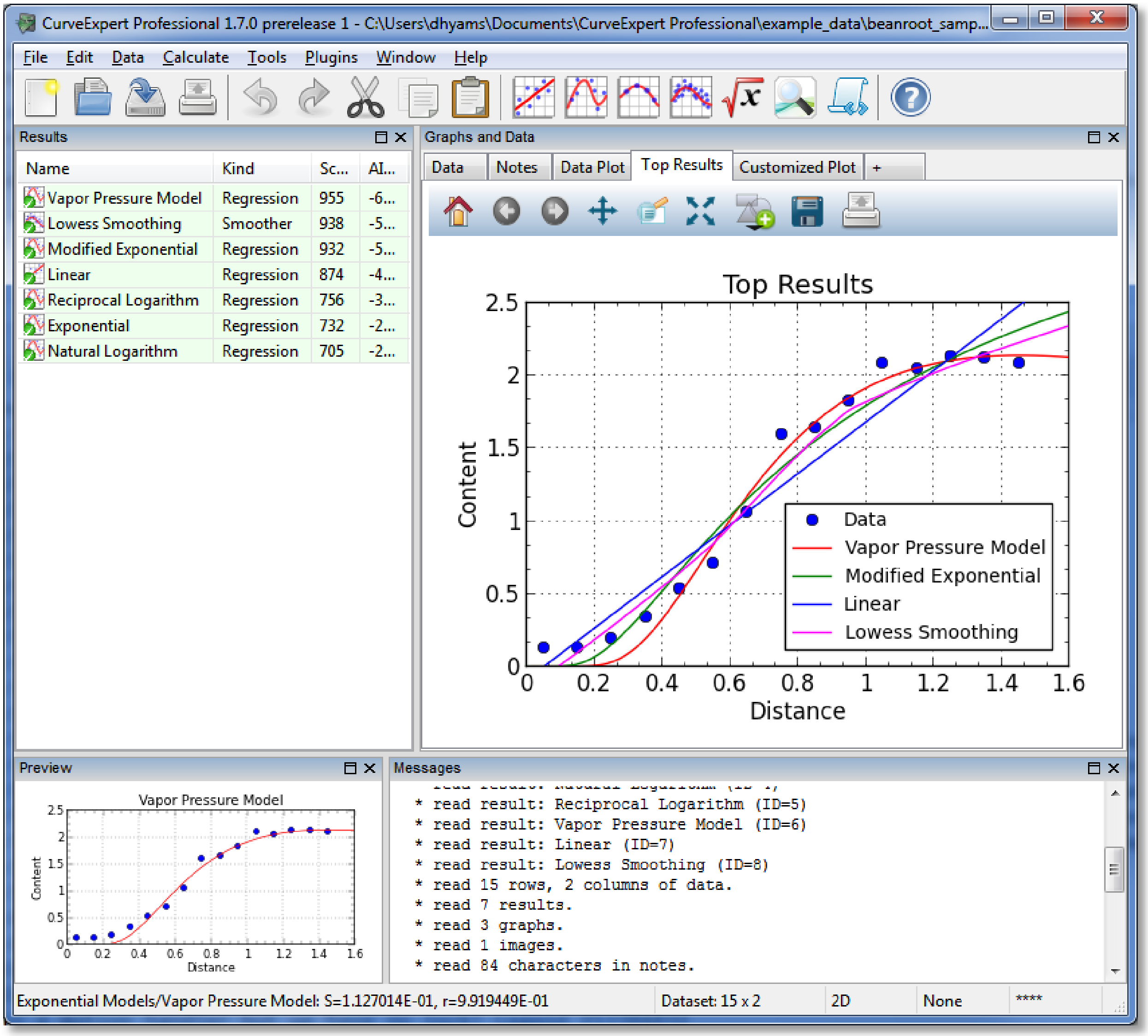
Make sure that you are in the same directory as the executable when running this command, so that the executable can find its required shared libraries. This procedure works under both Mac and Linux; under Linux, simply remove the .exe from the commands above. On macOS, you will have to use the "open" command:

```
> open /Applications/CurveExpertPro.app --args --license the_license_filename.txt --quit
```

The application will write a file "license.log" that prints the results of the licensing (success or failure).

GETTING STARTED

This guide leads you through the bare minimum of information you need to know in order to run CurveExpert Professional successfully. See the *User Interface* chapter for a full explanation of the interface.



Name	Kind	Sc...	Al...
Vapor Pressure Model	Regression	955	-6...
Lowess Smoothing	Smoother	938	-5...
Modified Exponential	Regression	932	-5...
Linear	Regression	874	-4...
Reciprocal Logarithm	Regression	756	-3...
Exponential	Regression	732	-2...
Natural Logarithm	Regression	705	-2...

```
* read result: Reciprocal Logarithm (ID=5)
* read result: Vapor Pressure Model (ID=6)
* read result: Linear (ID=7)
* read result: Lowess Smoothing (ID=8)
* read 15 rows, 2 columns of data.
* read 7 results.
* read 3 graphs.
* read 1 images.
* read 84 characters in notes.
```

Exponential Models/Vapor Pressure Model: S=1.127014E-01, r=9.919449E-01 | Dataset: 15 x 2 | 2D | None | ****

3.1 Load data into CurveExpert Professional

Read a sample file supplied with CurveExpert Professional. Choose *File->Open*, double click on “example_data”, set the file filter to *.dat, and double click the file “beanroot.dat”. When the File Import dialog appears, simply click OK. (see *Reading Data*)

3.2 Perform a Nonlinear Regression

Choose *Calculate->Nonlinear Model Fit*. Select the Sigmoidal Models, which automatically selects all of the members of that model family for computation. Click OK. (see *Calculating Results*)

3.3 Examine the Results

As a result of the computation, the results will be shown in the left pane in CurveExpert Professional, ranked in order from the best fit to the worst fit (see *Working with Results*). Double click, or right-click and select *Details…*, on any of the results to see details (see *Querying Result Details*). Also, you can click the tabs in the graph stack (the right hand pane in the application window) to see visualizations of the results. Later, you can add graphs of your own.

3.4 Export a Table

Double click on a result in the Results pane to see its details, and click the *Table* tab in order to generate a table for that result (see *Table*).

3.5 Visualize your Results

In the “Graphs and Data” pane, two graphs have been automatically created for you; one that shows the data only, and one that shows the data plotted with the top few results that have been computed. To add a graph that you can manipulate freely, click the “+” tab in the “Graphs and Data” pane (see *Graphing*), and add results by simply dragging and dropping them from the “Results” pane.

CHAPTER

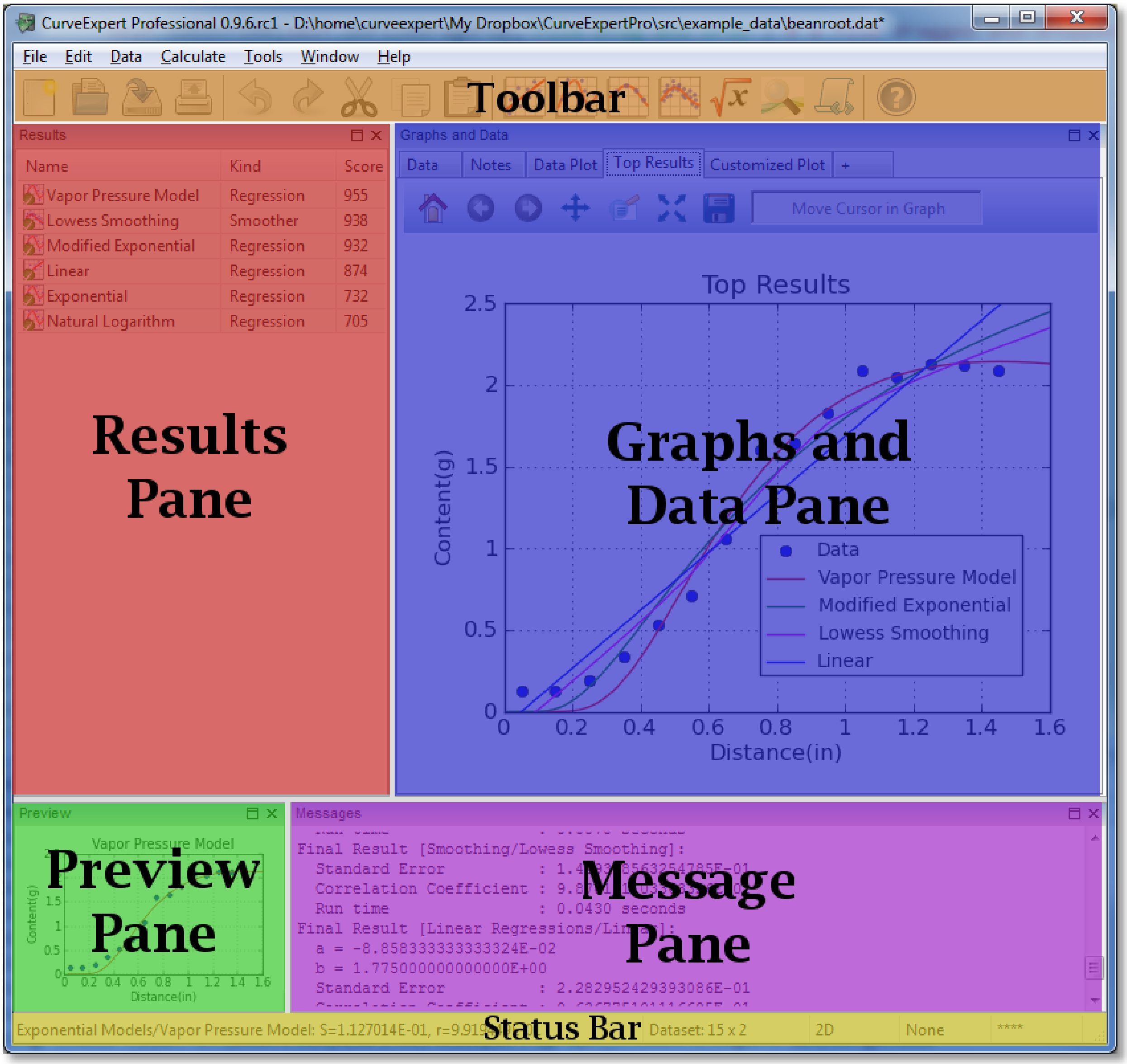
FOUR

USER INTERFACE

The CurveExpert Professional user interface is divided into panes:

- Results
- Graphs and Data
- Preview
- Messages

The panes, as well as the toolbar and status bar, are labeled in the graphic below:



Name	Kind	Score
Vapor Pressure Model	Regression	955
Lowess Smoothing	Smoother	938
Modified Exponential	Regression	932
Linear	Regression	874
Exponential	Regression	732
Natural Logarithm	Regression	705

```
Final Result [Smoothing/Lowess Smoothing]:
  Standard Error          : 1.4[illegible]8563254785E-01
  Correlation Coefficient : 9.8[illegible]
  Run time                : 0.0430 seconds
Final Result [Linear Regressions/Linear]:
  a = -8.858333333333324E-02
  b = 1.775000000000000E+00
  Standard Error          : 2.282952429393086E-01
```

Exponential Models/Vapor Pressure Model: S=1.127014E-01, r=9.91[illegible] | Dataset: 15 x 2 | 2D | None | ****

All of the panes can be shown or hidden via the *Window* menu in the menu bar. Further, panes (except for the Graphs and Data pane) can be torn out from the main window; grab the pane by its title bar, and tear it out. The panes can be redocked in the main window by the reverse operation; drag the window to one of the docking indicators, and a transparent hint will appear to show where the window will be docked. Using this procedure, the panes can be rearranged inside the main window to taste.

Panes can be quickly maximized by clicking the small box at the top right of each pane; this allows the pane to fill the entire main window. Restoring the size of the pane can be performed by clicking the double-box at the top-right of the pane. This gives a quick means of viewing a particular pane (most commonly the Graphs and Data pane) in full size.

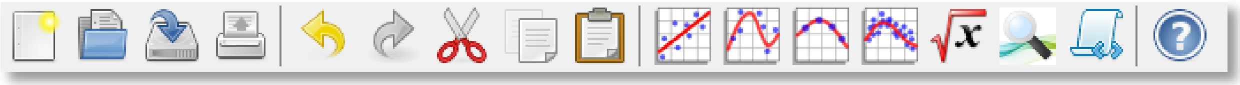
All panes can be hidden by clicking the 'X' at the top right of the pane, or by deselecting the corresponding pane name in the *Window* menu. To show a pane after being hidden, select the corresponding pane name in the *Window* menu.

4.1 Menus

If current conditions are such that a particular result cannot be computed, it will be dimmed in its *Calculate* menu. For example, with only one point in the dataset, it is not possible to compute any regressions So, all linear and nonlinear regressions in the *Calculate* menu will be dimmed and therefore unselectable. See *Menu Reference* for a quick explanation of each of the menu choices.

4.2 Toolbar

The toolbar can be shown or hidden via "Window->Toolbar".



Each button on the toolbar has a one-to-one mapping to an item in the menus. The toolbar is divided up into the following sections:

- File
- Edit
- Calculate
- Tools
- Help

The buttons are each documented below.

Icon	Meaning
	File->New. Reinitializes the application with an empty dataset and no results. When New is selected, you must input the
	File->Open. Opens a new (text) data file or CurveExpert file.
	File->Save. Save the current data, results, graphs, and notes into a CurveExpert file.
	File->Print. Prints the contents of the spreadsheet as a data table. If you would like to print all graphs, hold SHIFT while
	Edit->Undo. Undo the last appropriate action.
	Edit->Redo. Redo the last appropriate undone action.
	Edit->Cut. Cuts the selected item(s) to the clipboard as appropriate.
	Edit->Copy. Copies the selected item(s) to the clipboard as appropriate.
	Edit->Paste. Pastes data from the clipboard as appropriate.
	Calculate->Linear. Calculates a straight line regression for the data.
	Calculate->Nonlinear Model Fit. Shows the nonlinear regression picker so that you can select a model or models to comp
	Calculate->Cubic Spline. Calculates a cubic spline for the data.
	Calculate->Lowess Smoothing. Immediately calculates a Lowess smoothing of the data.
	Calculate->Function. Shows the function picker so that you can select a function or functions to compute.
	Tools->CurveFinder. Runs the CurveFinder to scan models for the best fit for your data
	Tools->Custom Models(Simple). Opens the custom model creator (simple interface)
?	Help->Manual. Opens the documentation.

4.3 Results Pane

The results pane displays a list of the currently calculated results. This list is color-coded and badge-coded; see *Working with Results* for an explanation of the color codes and badges.

The items displayed in the columns of the results pane can be customized by right-clicking the column headers. The following items may be selected for display:

- Name (required; cannot be deactivated)
- Kind
- Family
- Score
- R (correlation coefficient)
- R^2 (coefficient of determination)
- Std_Err (standard error)

Remember that you can send results to a plot, remove a result, or view result details by right-clicking a particular result. See *Working with Results* for details.

4.4 Graphs and Data Pane

The "Graphs and Data" pane is a notebook where the dataset and graphs reside (as well as any notes that you would like to save). reside. Please refer to *The spreadsheet* for information on how to work with the dataset via the built-in spreadsheet. Refer to *Graphing* for documentation concerning how to work with graphs.

Two graphs are always present: "Data Plot" and "Top Results". As the name implies, the Data Plot always displays your raw data with no other results. Top Results displays the top *n* results from the set of calculations that the user has run; *n* is configurable via *Edit->Preferences->Graphing->Automatically plot top n results*.

To add another graph, simply press the "+" tab; this adds another graph for you to build as you see fit. You may also rename any of your custom plots by right-clicking on the tab and selecting *Rename*; this assists in keeping track of multiple graphs. Also, you can delete a graph entirely by right-clicking the graph tab and selecting *Delete*. Finally, you can copy (clone) a graph by right-clicking the appropriate graph's tab and selecting *Clone*.

Your notebook can also be rearranged to taste. Select one of the notebook pages (by clicking on it), and then hold the shift key along with a left or right arrow key to move the selected notebook page left or right, respectively. Using the left/right arrow keys without holding shift simply selects the page to the left or right, respectively.

Remember that, for a graph that you have created, you can drag and drop results from the results pane to plot any results of interest.

4.5 Preview Pane

The preview pane displays the result that is pointed to in the Result pane, or, if a result is pointed to in a graph, the preview pane also updates to the pointed-to-result. While convenient, the preview window can cause the user interface to be sluggish on older computers; to disable the preview, simply hide it by deselecting *Window->Preview*, or more straightforwardly, click the 'X' at the top right of the preview pane.

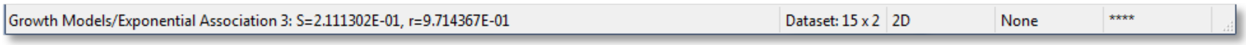
4.6 Messages Pane

The **Messages Pane** is an area where CurveExpert Professional sends all notifications for your perusal. For example, when a file is read, the a summary of the file contents appears in the Messages pane. Also, when any calculation is undertaken, a summary of the result appears in the Messages pane.

The content of the messages pane can be copied and pasted to another application by highlighting a portion, right-clicking, and selecting *Copy*, or by using Ctrl+C (on a Mac, Cmd+C).

Error messages appear in the Messages pane in red, and special messages appear in blue. All messages that appear in the Message pane are automatically logged, and be accessed after-the-fact via *Tools->Log Viewer* (see *Log Viewer*).

4.7 Status bar



Growth Models/Exponential Association 3: S=2.111302E-01, r=9.714367E-01 | Dataset: 15 x 2 | 2D | None | ****

The status bar is divided into five sections, which show the following items:

Section	Displays
1	typically shows vital statistics for the currently pointed-to result, or displays help when appropriate.
2	size of the currrent dataset (nrows x ncolumns)
3	dimension of the dataset, which determines the mode of the application
4	current locale
5	number of helper processes present that perform calculations (see *Max number of cores to utilize*). Each asterisk represents one helper process.

4.8 Menu Reference

Note: On OSX, some menu items appear both in the *CurveExpertPro* menu at the top left of the menubar (in order to conform to Apple UI guidelines), and in their standard locations as documented below. Specifically, *Preferences* (documented throughout this manual as *Edit->Preferences*), *About* (documented below as *Help->About*), and *Quit CurveExpert Pro* (documented below as *File->Exit*).

4.8.1 File menu

Menu choice	Purpose
File->New	Reinitializes the application with an empty dataset and no results. When New is selected, you must input the desired number of columns in the new dataset.
File->New from Clipboard	Reinitializes the application with a dataset imported from the clipboard. The file import dialog allows you to fine-tune the importing process.
File->Open	Opens a new (text) data file or CurveExpert file.
File->Recent files	Opens a previously opened file.
File->Close	Closes the current file.
File->Save	Save the current data into a file.
File->Save As	Save the current data into a file, under a different name.
File->Export	exports the current dataset as a text file.
File->Print	Prints the contents of the spreadsheet as a data table. If you would like to print all graphs, hold SHIFT while selecting File->Print.
File->Print Preview	Shows a print preview of the spreadsheet data, optionally printing it as well. If you would like to print preview for all graphs, hold SHIFT while selecting File->Print Preview.
File->Page Setup	allows setting of preferences for margins, paper orientation, etc., for printing.
File->Exit	close CurveExpert Professional

4.8.2 Edit menu

Menu choice	Purpose
Edit->Undo	Undo the last appropriate action. For data in the spreadsheet, see *Undo and Redo*.
Edit->Redo	Redo the last appropriate undone action. For data in the spreadsheet, see *Undo and Redo*.
Edit->Cut	Cuts the selected item(s) to clipboard as appropriate. For selections of data, see *Cutting, Copying, and Pasting*.
Edit->Copy	Copies the selected item(s) to clipboard as appropriate. For selections of data, see *Cutting, Copying, and Pasting*.
Edit->Paste	Pastes items from the clipboard as appropriate. For pasting of data, see *Cutting, Copying, and Pasting*.
Edit->Clear	clears the selected data as appropriate, without placing it on the clipboard.
Edit->Select All	Selects all items as appropriate.
Edit->Preferences	Opens a dialog where the application settings/preferences can be viewed and/or modified. See *Application Preferences*.

4.8.3 Data menu

Menu choice	Purpose
Data->Sort	sort the selected data by the first column. See *Operating on Data*.
Data->Scale	scale the selected data by a factor. See *Operating on Data*.
Data->Translate	translate (add) a constant to the selected data. See *Operating on Data*.
Data->Transform	Apply a general transformation to the selected data. See *Operating on Data*.
Data->Crop	Crop the data to a selected minimum and maximum; removes points from the dataset. See *Operating on Data*.
Data->Clamp	Clamp the selected data to a minimum and maximum. See *Operating on Data*.
Data->Rotate	Rotate the columns in a dataset to the left. See *Operating on Data*.
Data->Add/Remove STDDEV Column	Add or remove a standard deviation column to/from the dataset, as appropriate.
Data->Statistics	Show the basic statistics for the dataset. See *Data statistics*.

4.8.4 Calculate menu

Menu choice	Purpose
Calculate->Linear Fit	Calculates a straight line regression for the data. See linreg.
Calculate->nth Order Polynomial	Calculates a polynomial fit for the data, with a user-selectable degree. See linreg.
Calculate->Nonlinear Fit	Shows the nonlinear regression picker so that you can select a model or models to compute. See nonlinreg.
Calculate->Linear Spline	Calculates a linear spline for the data (dot-to-dot). See interpolation
Calculate->Cubic Spline	Calculates a cubic spline for the data. See interpolation
Calculate->Polynomial Spline	Calculates an nth-degree polynomial spline for the data. See interpolation
Calculate->Tension Spline	Calculates a tension spline for the data. See tensionsplines
Calculate->Moving Average	Calculates a moving average for the data, with parameters such as the window size and type to be specified. See movingaverage
Calculate->Lowess Smoothing	Calculates a Lowess smoothing of the data. See lowess.
Calculate->Function	Shows the function pick so that you can select a function or functions to compute. See functions.

4.8.5 Tools menu

Menu choice	Purpose
Tools->CurveFinder	Runs the CurveFinder to scan models for the best fit for your data. See *CurveFinder*.
Tools->Custom Models(Simple)	Opens the custom model creator with the simple one-line-equation interface. See *Custom Models*.
Tools->Custom Models(Advanced)	Opens the custom model creator with the advanced interface. See *Custom Models: Advanced Usage*.
Tools->Custom Functions(Simple)	Opens the custom function creator with the simple one-line-equation interface. See *Custom Functions*.
Tools->Custom Functions(Advanced)	Opens the custom function creator with the advanced interface. See *Custom Functions: Advanced Usage*.
Tools->Manage Graph Themes	Create/delete graph themes. See *Graph Themes* and *Graph Theme Manager*.
Tools->Run NIST Validation	Check CurveExpert Professional against certified NIST results. See *Validation*.
Tools->View Logs	View the log files. See *Log Viewer*.

4.8.6 Window menu

Menu choice	Purpose
Window->Results	Toggle the Results pane on and off. See *User Interface*.
Window->Graphs and Data	Toggle the Graphs and Data pane on and off. See *User Interface*.
Window->Messages	Toggle the Messages pane on and off. See *User Interface*.
Window->Preview	Toggle the previewing pane on and off. See *User Interface*.
Window->Toolbar	Toggle the main toolbar on and off. See *Toolbar*.
Window->Dock All	Dock all floating panes back into the main window. See *User Interface*.
Window->Restore to Default	Restores all panes the the default layout (docked, Results pane top left, Preview bottom left, Graphs and Data top right, and Messages bottom right). See *User Interface*.

4.8.7 Help

Menu choice	Purpose
Help->Manual	Displays the CurveExpert Professional manual (HTML version).
Help->Web Site	Displays the Hyams Development web site.
Help->Discussion Forum	Displays the CurveExpert user-to-user discussion forum.
Help->Tutorial Videos	Displays the available video tutorials for CurveExpert Professional
Help->Activate Trial Period	activates a 30 day trial period for CurveExpert Professional; see *Installation and Activation*.
Help->Enter License	after purchase, enter your license for the software; see *Installation and Activation*.
Help->Order	direct link to the website for ordering.
Help->Check for Updates	Checks to see if your version is the latest one; requires Internet access.
Help->Show Diagnostics	Displays diagnostic information in the Messages pane.
About	Displays the application about box.

CHAPTER

FIVE

READING DATA

5.1 Introduction

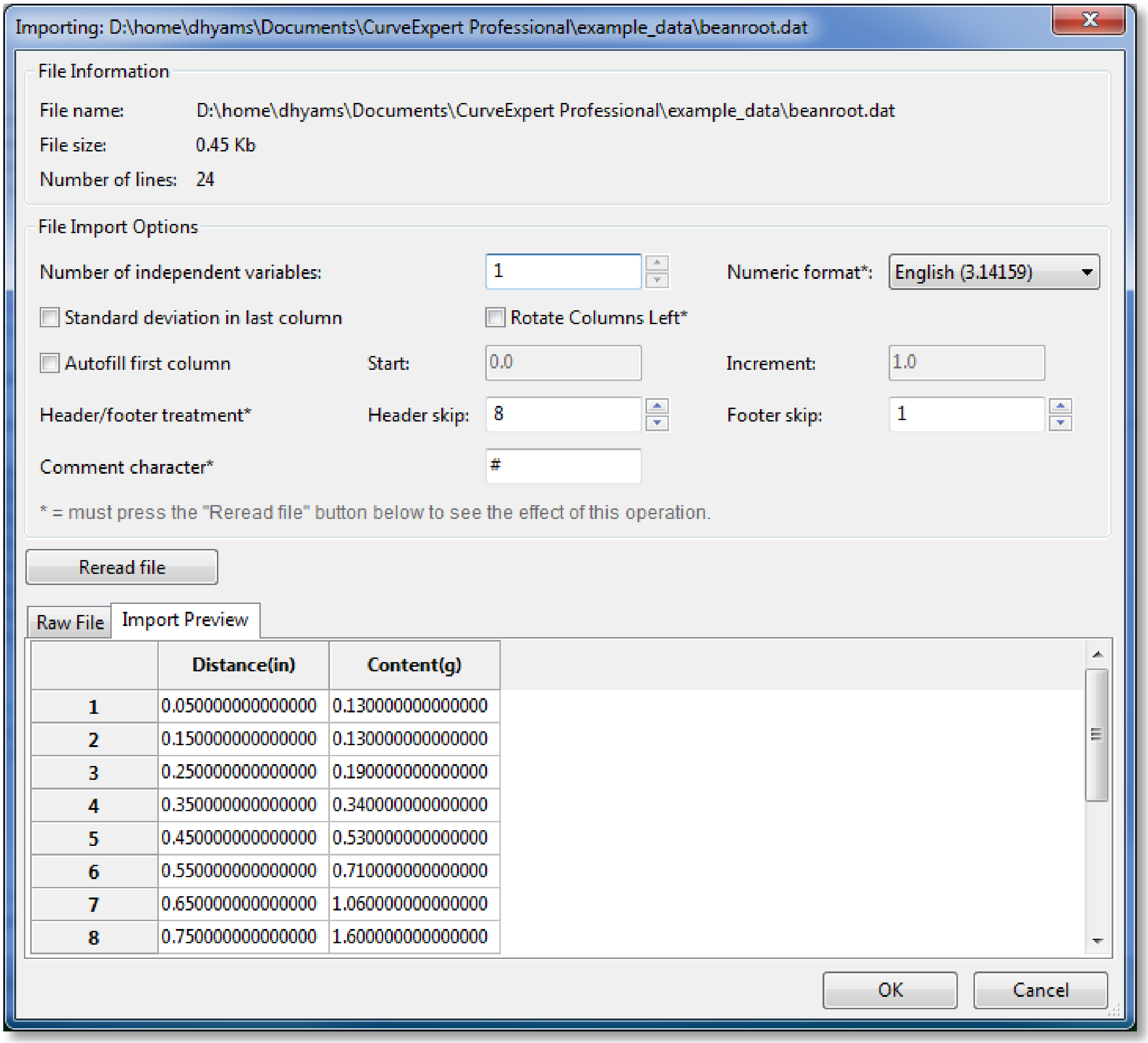
The first task in any data analysis program like CurveExpert Professional is to read or import your data into the software. For this, CurveExpert Professional provides a robust file import mechanism, which can be accessed simply by choosing *File->Open*. You can read in both native CurveExpert files (.cxp) and generic data files in text format.

Note: To be able to choose files with extensions other than .cxp, you must select the appropriate extension in the bottom right corner of the file choosing dialog.

Most of the time, the data to be read in is contained within a text file of some sort, with headers and comments interspersed. CurveExpert Professional tries to make the job of importing this data as easy and painless as possible. The file import intelligently examines your raw file in order to find the data, as well as finding column headers for that data, if present.

5.2 Raw file import

To read in a text file (common extensions are .dat and .txt, but can be anything that you choose), simply choose *File->Open*, and then the filename. Alternatively, drag and drop the file to the CurveExpert Professional spreadsheet. The file import window will display, as follows:



Importing: D:\home\dhyams\Documents\CurveExpert Professional\example_data\beanroot.dat

File Information

File name: D:\home\dhyams\Documents\CurveExpert Professional\example_data\beanroot.dat

File size: 0.45 Kb

Number of lines: 24

File Import Options

Number of independent variables: 1

Numeric format*: English (3.14159)

□ Standard deviation in last column

□ Rotate Columns Left*

□ Autofill first column

Start: 0.0

Increment: 1.0

Header/footer treatment*

Header skip: 8

Footer skip: 1

Comment character* #

* = must press the "Reread file" button below to see the effect of this operation.

	Distance(in)	Content(g)
1	0.050000000000000	0.130000000000000
2	0.150000000000000	0.130000000000000
3	0.250000000000000	0.190000000000000
4	0.350000000000000	0.340000000000000
5	0.450000000000000	0.530000000000000
6	0.550000000000000	0.710000000000000
7	0.650000000000000	1.060000000000000
8	0.750000000000000	1.600000000000000

Basic information about the file is shown at the top of the window, where the complete path to the file, file size, and the number of lines detected in the file is displayed. The middle section of the window is where you set parameters necessary for reading in the file. Most of the time, default options are appropriate, and there is not much more to do except to glance at the spreadsheet preview in the bottom third of the window, and click the OK button. However, in cases where the defaults are not appropriate, there are a range of settings you can manipulate to read your file in successfully.

5.2.1 Number of independent variables

First, the number of independent variables must be set. CurveExpert Professional normally guesses the number of independent variables to be the number of columns detected in your file minus one. You may change this setting if the default is not correct. Initially, in the absence of any column labels in the file, the independent variables are always called X1, X2, X3, and so on, and the dependent variable is called Y. The result of your operation is shown instantly in the spreadsheet preview.

Note: CurveExpert Professional always expects the independent variable column(s) to be to the left of the dependent variable column.

5.2.2 Numeric format

By default, the numeric format is set according to the settings in *Edit->Preferences->Localization* (see *Localization*). The two available settings are English (use dots for decimals) and European (use commas for decimals). Select the appropriate numeric format for your file.

5.2.3 Standard deviation in last column

If the last column of your dataset is standard deviation data, make sure to check the "*standard deviation in last column* checkbox. A typical situation is that, on import of a three column dataset, CurveExpert guesses that there are two independent variables and no standard deviation data. To remedy the situation, simply set the number of independent variables to 1, which will leave the last column unassigned. Then, select *Standard deviation in last column* The result of your operation is shown instantly in the spreadsheet preview.

5.2.4 Rotate columns left

Some raw datasets list the dependent variable first. If this is the case, checking *Rotate columns left* will shift the data in the first column to the last, moving all other columns to the left. This is the ordering that CurveExpert Professional expects (independent variables first, and the dependent variable following). For a two column dataset, this operation is equivalent to swapping the columns. To see the effects of your choice here, the *Reread file* button must be pressed.

5.2.5 Autofill

Sometimes, the independent variable is implied by the ordering of the data. If you would like to generate the first column of data as a simple range and increment, select *Autofill first column*, and then choose the starting value and increment between each data point. The result of your operation is shown instantly in the spreadsheet preview. The fill is computed as follows:

$$x = x_{start} + i\Delta x$$

where i is the point id (starting at zero for the first point), x_{start} is the starting value input, and Δx is the increment input.

5.2.6 Treatment of Header and Footer

Often, data files have text that is not part of the dataset, and this text appears either before or after the real dataset (or both). To accommodate for this, you can set the *Header Skip* and *Footer Skip*, which is the number of lines in the file that are skipped at the top and bottom, respectively, before any attempt is made at reading data. This capability can also be used to intentionally skip over certain data in your file.

The header and footer skip can be thought of as a mask that is applied to your file before any reading takes place.

CurveExpert Professional uses a sophisticated procedure to try to guess the header and footer skip correctly. In all but the most esoteric cases, the header and footer skips will be correct already, and therefore need no interaction from the user.

Note: A newline at the end of your file counts as just that; a new line. This line is skipped by CurveExpert Professional because it is a line with no data. Therefore, it is common for the footer skip to be set to 1 by the automatic header/footer algorithm.

5.2.7 Comment character

Most data files have comments interspersed throughout them, sometimes in the middle of the data. In order to skip these comments, the character that designates the comment must be specified. In the data file, then, anything that appears after a comment character is ignored. Common comment characters are '#' and '%'.

5.2.8 Refreshing the preview

If rotating columns or changing the header/footer treatment, you must press the *Reread file* button to refresh the preview.

5.2.9 File import preview and raw file preview

The file import preview shows the result of the reading of the file, augmented by the current settings discussed above. This preview shows you exactly what CurveExpert Professional will read and place in the main spreadsheet.

The raw file preview shows you a copy of the file, with no modification, so that you can see the contents. Line numbers are annotated on the left, the the header and footer skip settings are shown by dimming the areas of the file that are masked away by these settings.

5.2.10 Column Labels

CurveExpert Professional attempts to detect the labels of your columns, such that they are correctly read into the software. To detect the column labels, the file reader examines one line above the skipped header, if such a line exists. In other words, if the header skip is set to 10, line 10 is examined for the column labels. First, the line is trimmed of any white space and comment characters, and then broken into individual tokens. If the number of items on this line is the same as the detected number of columns, these tokens are treated as the column names, as long as they "look like" names. To look like a name, the labels must begin with an alphabetical character, and only contain legal punctuation "()+-*". So, your data file can look something like this:

```
Comments at the top of the file are automatically
ignored by the file reader, because
it can tell
that there is no numeric data there.
# or you can comment with a designated comment character

time(s) voltage(volts)
1.0 4.5
2.0 4.23
3.2 4.1
```

In the example above, the column labels will be automatically detected as "time(s)" and "voltage(volts)".

5.2.11 Other file metadata

To force the localization of a given text file, see *File overrides* for the required header information.

5.2.12 Finishing

When the file import preview looks like the desired dataset, simply hit OK; your data will be read and placed into the main spreadsheet.

5.3 CurveExpert Professional files

CurveExpert Professional files contain the data, results, graphs, and notes (notes are available in CurveExpert Pro only). To save a .cxp file, simply choose *File->Save* or *File->Save As* as appropriate. Correspondingly, to read a .cxp file, choose *File->Open*.

These files are portable across platforms (Mac, Windows, and Linux), so you can read a file in on a Mac that was generated in Windows, for example. Also, .cxp files are compatible between CurveExpert Professional and CurveExpert Basic 2.0 or later. CurveExpert Basic can read CurveExpert Pro files, and vice versa.

Also, if a particular model or function is saved as a result in a .cxp file and it is not present on the system reading the file, the model/function will automatically be created on your behalf. This newly created model/function will then be available for use in the “Imported” family, and will appear appropriately in the Nonlinear model or function pickers.

CHAPTER

SIX

WORKING WITH DATA

6.1 Introduction

The raw data in CurveExpert Professional is handled as a simple matrix of numbers; the dataset is made up of a number of rows and a number of columns. Usually, the last column is interpreted as the dependent variable, and the ncol-1 columns before this are all of the independent variables. However, if you have selected in the file import that the last column is a column of standard deviation data per point, the last column is appropriately treated in this manner.

6.2 Data statistics

Simple statistics for the current dataset can be obtained by selecting *Data->Statistics*. The resulting window will display the number of points and columns, and the following stats for each column of your dataset:

- minimum
- maximum
- range
- average (mean)
- standard deviation

The average is computed with the standard formula:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Standard deviation is computed with the following formula, which is called the *standard deviation of the sample*:

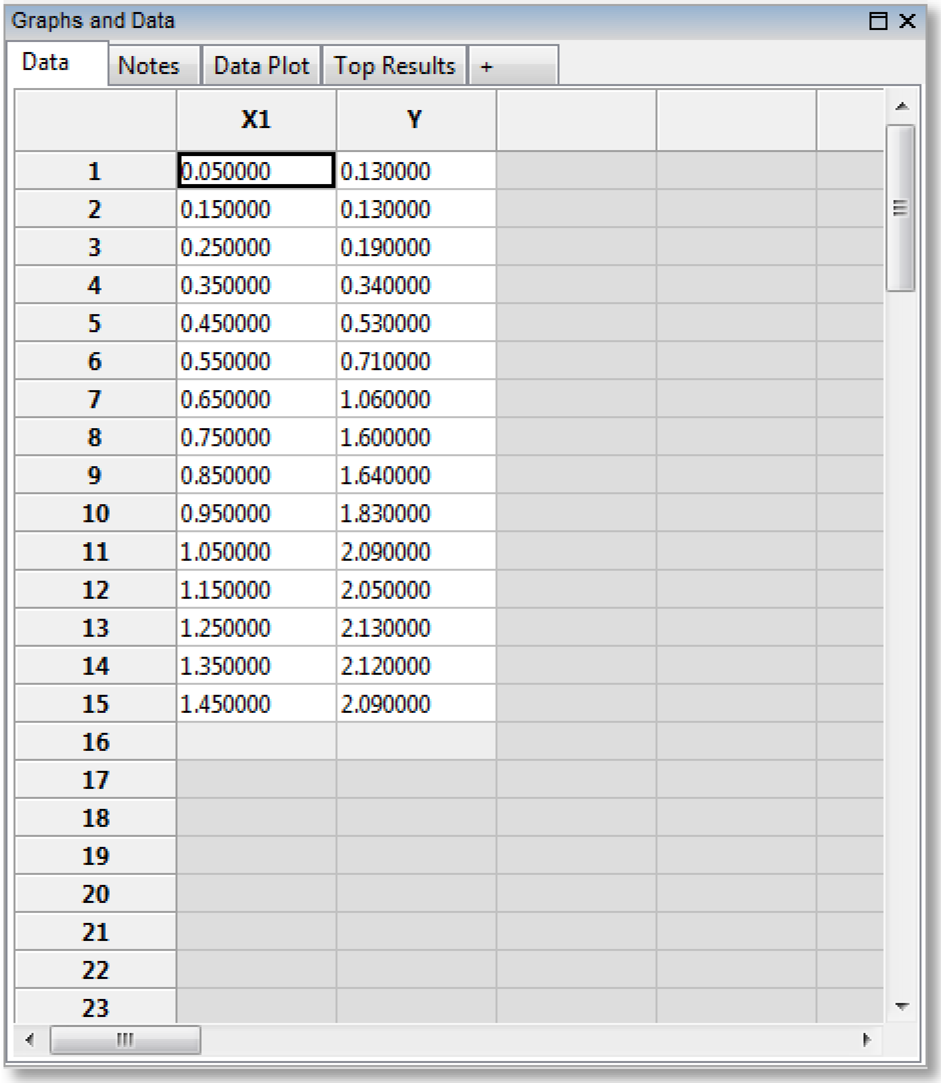
$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

To copy the statistics information to the clipboard, press the *Copy* button at the bottom of the dialog.

6.3 The spreadsheet

The spreadsheet resides in the "Graphs and Data" pane in CurveExpert Professional.

The spreadsheet allows manual entry or modification of a dataset either entry-by-entry, or by cutting and pasting from within CurveExpert Professional, or from other sources. An example of the spreadsheet is shown below;



	X1	Y
1	0.050000	0.130000
2	0.150000	0.130000
3	0.250000	0.190000
4	0.350000	0.340000
5	0.450000	0.530000
6	0.550000	0.710000
7	0.650000	1.060000
8	0.750000	1.600000
9	0.850000	1.640000
10	0.950000	1.830000
11	1.050000	2.090000
12	1.150000	2.050000
13	1.250000	2.130000
14	1.350000	2.120000
15	1.450000	2.090000
16		
17		
18		
19		
20		
21		
22		
23		

All cells that hold data are white, and cells that are not part of the dataset are colored in gray. At the end of the dataset is one light-gray row, which signifies that the cells may be typed into (in order to add data points), but are not yet part of the dataset.

Right-click the spreadsheet in order to access operations that you can perform on the data itself, in-place.

The name of a column of data can be changed by right-clicking the appropriate column header and selecting *Rename Column* (alternatively, double click the column header). These names are for user information only, but are used if an entire dataset with multiple dependent variables is sent to a graph; the column names are used to differentiate the various curves in the legend.

The name of a row of data can be changed by right-clicking the appropriate row header and selecting *Rename Row* (alternatively, double click the row header). These names are currently for user information only.

6.3.1 Cursor Movement

Cursor movement in the spreadsheet is normally accomplished with the TAB and RETURN keys; TAB moves the cursor to the right, and RETURN moves the cursor downward. Of course, the arrow keys and PAGEUP/PAGEDOWN may also be used to move the cursor. If it is desired to move the cursor with a carriage-return like approach (in which the RETURN key moves the cursor to the next line if at the end of a row), this can be selected in *Edit->Preferences->General->Excel style cursor movement*.

To reverse the direction of cursor navigation, hold the SHIFT key.

6.3.2 Cutting, Copying, and Pasting

When copying data, the current selection in the spreadsheet is placed on the clipboard (and obeys your regional settings while doing so). If no data is selected, all data is copied. The delimiter used between columns, if applicable, is the delimiter chosen in *Edit->Preferences->General->Default Delimiter*.

A "cut" must involve all columns of the dataset. Therefore, the "Cut" item in the data menu is dimmed if your selection only involves some of the columns. This behavior is due to the fact that a cut will remove points from the dataset, and your selection should reflect that reality.

A paste from the clipboard informs CurveExpert Professional to attempt to intelligently read what is on the clipboard and turn that into an intermediate dataset for subsequent pasting. Different rows in this intermediate dataset are delimited by newlines (or carriage returns), and columns are delimited by whitespace or commas (unless using European format for incoming numbers, in which case commas are not valid as a delimiter). This intermediate dataset will then be inserted at the current cursor location in the spreadsheet, overwriting parts of your existing dataset and extending your existing dataset as necessary. A paste operation can be viewed as an insertion of a block of data at the current cursor location, following by a subsequent discarding of any data that happens to fall outside the current number of columns of your dataset.

6.3.3 Inserting Rows

To insert a row into your dataset, right click on the spreadsheet and select "Insert Row". This will cause all rows at or below the current cursor location (indicated by the black outline) to be shifted downwards, and a new row will be inserted at the current location.

To insert multiple rows at once, select a range in the spreadsheet, right click, and select "Insert Row" as before. In this case, all rows touched by the selection will be shifted downwards, and new rows are inserted at the current cursor location. The number of inserted rows will equal the number of rows in the original selection. For example, to insert three rows at row 2, select rows 2-4 (inclusive) and select "Insert Row".

6.3.4 Removing Rows

To remove rows from the dataset, select a set of rows to be removed, right click, and select "Remove Rows". This is functionally similar to cutting the rows (see above), except that the data is not copied to the clipboard.

6.3.5 Undo and Redo

All data operations are undoable and redoable to a number of levels possible by the *Edit->Preferences->Advanced->Peak Undo Memory Usage* setting. All data operations are undoable and redoable to a number of levels possible by the *Edit->Preferences->Advanced->Number of undo levels* setting. So, you can feel comfortable manipulating data knowing that any transformation, clear, or cut/paste operation can be undone.

6.4 Number Formatting

The values shown in a spreadsheet may be formatted by selecting a block via the mouse, right clicking, and selecting **Format Numbers** from the resulting menu. A dialog will appear as follows:

A **General** formatting prints a number naturally as most practitioners would write the number. **Scientific** formatting prints the number always with a mantissa and exponent part, as in "1.532e+04". **Engineering** formatting is almost the same as scientific, but the exponent is always guaranteed to be a multiple of three. **Percentage** formatting shows a decimal as a percentage, with a following percentage sign. For example, the number 0.05 would be shown as 5.0%.

The **Decimal places** field allows the user to specify the number of decimal places shown; this setting is relevant for all formatting settings listed above except for **General**.

The **Leader** and **Trailer** fields allow the user to place customized text before and/or after the number. These fields are very useful for adding units to numbers (use the trailer) or prepending a dollar sign for currencies (place $ in the Leader field).

6.5 Operating on Data

CurveExpert Professional provides several ways that you can operate on raw data in the spreadsheet. To perform these operations, simply highlight the data that you want to work on, and select one of Sort, Scale, Translate, Transform, Crop, or Clamp from the Data menu. Alternatively, you can right click on the spreadsheet after making a selection. If no selection is made, the operation is assumed to apply to the entire dataset (all rows and all columns).

6.5.1 Sort

To sort your data against the first column of the dataset, select **Sort** from the Data menu (the entire dataset must be selected in the spreadsheet). If CurveExpert Professional has detected that the data is already sorted, this menu item will be dimmed. If the choice is nondimmed, however, it does not necessarily mean that the dataset is already sorted.

Sorting your dataset is crucial for splining of your data or other operations that depend on the data being sorted.

6.5.2 Scale

To multiply by or divide the selected data by a factor, select *Data->Scale*.

6.5.3 Translate

To add a constant value to the selected data, select *Data->Translate*.

6.5.4 Crop

To drop points from the selected part of your dataset that are above or below a certain threshhold, select *Data->Crop*. After a cropping operation, there is likely to be fewer data points in your data set than before the crop. Usually, cropping is used to eliminate outliers from a dataset.

6.5.5 Clamp

To clamp points in the selected part of your dataset to a certain threshhold value, select *Data->Clamp*. Clamping is usually used to correct values that are slightly over or under a valid domain for that value.

6.5.6 Rotate

To rotate columns in a dataset, select *Data->Rotate*. For a dataset with two columns, this is exactly the same as a column swap. Note that the rotation always proceeds to the left in a circular fashion; the leftmost column becomes the rightmost one after the rotation, and all other columns are displaced by one to the left.

6.5.7 Transforming Data

Data->Transform allows you to map the selected part of your dataset through a general function. If this menu item is selected, you can enter an expression, as a function of x, and all selected data will be evaluated via this function. Note that "x", in this context, just refers to the selected data.

For example:

```
x + 5
pow(x*100,0.75)
degrees(x)
```

In the transformation, you may use any functions listed in *Appendix A: Math Functions*. Note that both the Scale and Translate operations above can just as easily be done via a transformation here.

CHAPTER

SEVEN

CALCULATING RESULTS

CurveExpert Professional supports four distinct classes of results:

- regressions
  - linear regressions (linear and polynomial fits)
  - nonlinear regressions (built-in and custom models)
- interpolations
  - linear spline
  - cubic spline
  - polynomial spline
  - tension spline
- smoothings
  - lowess smoothing
  - moving averages
- functions (built-in and custom functions)

Calculation of these results can be accessed through the *Calculate* menu in CurveExpert Professional, or optionally via the corresponding buttons on the toolbar (see *Toolbar*).

Note: A distinction is made between “models” and “functions”. Models have free parameters, and are therefore dependent on the independent variables and the parameters: $f(\vec{x};\vec{a})$. Functions are dependent on the independent variables only, with no free parameters: $f(\vec{x})$.

7.1 General Guidelines

7.1.1 Scaling Datasets

For best results, always scale your data sets to the order of one. You may do this prior to reading in your data file, or after your data file has been imported using the scaling feature in CurveExpert Professional (see *Operating on Data*).

Imagine a data set with x values ranging from 1000 to 10000 and a regression model where the term exp(a*x) is involved. Unless a very good initial guess for a is given, chances are that an exponential to a very large power (eg. exp(1000)) will be taken in the course of the nonlinear regression algorithm. The calculation will overflow, and the regression will fail as a result. Even if the regression happens not to fail, the regression algorithm will have an exceedingly tough time finding the correct parameters, since a small change in the free parameters will cause a tremendous change in the size of the term. The moral of the story: always scale your data!

As another example, if you have data that describes atmospheric pressure at different elevations, you might have (in metric units) a data set that looks like:

```
x=[-100, 0, 100, 500, 1000, 4000] meters
y=[102000,101325,101000,100500,100000,99900] Pascals
```

Using the *Data->Scale* feature in the Data menu of CurveExpert Professional, you can scale this data using a scale factor of 0.001 on the x data, and 0.00001 on the y data. So, you would then have the following data set:

```
x=[-0.1, 0, 0.1, 0.500, 1, 4] kilometers
y=[1.02,1.01325,1.01,1.005,1.0,0.999] bars
```

The second example is much more likely to allow nonlinear regressions to converge, and also will allow higher order polynomial fits to be performed with more accuracy. If you have a data set that seems particularly ill-behaved, scaling can help solve this problem.

Note that CurveExpert Professional is able to perform correctly on data in any scale, as long as the calculations do not overflow or underflow. So, if a data set is giving problems, scaling it should be the first action to take.

7.1.2 Weighting

Weighting has to do with defining how “important” a particular point is in the optimization calculation; i.e., accounting for the fact that in some cases, certain data in the dataset is more precise than other data. Thus, a greater importance should be placed on the more precise points. More precisely defined points (those with less uncertainty) should be given greater influence (a greater weighting) over the final fitted parameters.

There are six choices for weighting in CurveExpert Professional:

- **Default**: No weighting unless a standard deviation column is defined in the dataset; in that case, weight by the square of the standard deviation column. This is equivalent to weighting by the uncertainty (if available).
- **None**: No weighting under any circumstance, even if a standard deviation column is present.
- **Y**: Weight each data point by the absolute value of its Y (dependent variable) value.
- **Y^2**: Weight each data point by the square of its Y (dependent variable) value.
- **X**: Weight each data point by the absolute value of its X (independent variable) value.
- **X^2**: Weight each data point by the square of its X (independent variable) value.

Note that if there is more than one independent variable, weighting by **X** and **X^2** will not be available.

7.1.3 Set the tolerance parameter reasonably

Don’t set the tolerance parameter (in *Edit->Preferences->Regression*) too low. In regression modeling, not much advantage is to be gained by setting a very strict tolerance. Its main purpose in life is to prevent the nonlinear regression algorithm from converging on local minima, not to make the calculated parameters more accurate.

7.1.4 Data should be appropriate to the model

Make sure that the data is appropriate to the model. Especially look out for using logarithmic or exponential families of models with data that contains zeros or negatives. For example, it is not possible, in any shape or form, to obtain a negative or zero with the basic exponential model (y=ae^(bx), assuming a is positive; the inverse problem exists for a negative a). So, it is not wise to use a model that cannot reflect the trends in the data.

7.1.5 For interpolations, data points must be sorted

If you are using an interpolation-type curve fit, the data points must be sorted by ascending x. Use the *Data->Sort* feature in CurveExpert Professional to sort your data points correctly.

7.1.6 Sometimes, it is unavoidable

Some curve fits are simply ill-behaved, i.e. prone to divergence. For some data sets, it may not be possible to converge certain nonlinear regression models.

7.2 Interpolation

7.2.1 Introduction

An interpolation, by definition, passes through every data point, and as such, the correlation coefficient will always be 1, and the standard error will always be zero. CurveExpert Professional supports polynomial splines and tension splines. All splines are defined in a piecewise fashion between data points.

Note: The dataset must be sorted (based on the independent variable) in order for any spline interpolation to work. Select *Data->Sort* from the main menu in order to sort your dataset if necessary. If CurveExpert Professional detects that your dataset is not sorted, it will not allow spline interpolations to be selected.

Note: Currently, CurveExpert Professional supports interpolation for datasets with one independent variable only. For multivariate datasets, all interpolations will be disabled.

7.2.2 Linear Splines

To calculate a linear spline, select *Calculate->Linear Spline* from the main menu. The linear spline is simply a polynomial spline of order 1. It appears as a “dot-to-dot” connecting each point with a straight line segment. Linear splines only guarantee continuity of the spline at the data points.

7.2.3 Cubic Splines

To calculate a cubic spline, select *Calculate->Cubic Spline* from the main menu. The cubic spline is simply a polynomial spline of order 3; cubic splines are the most common form of spline. Cubic splines guarantee continuity in the spline, and continuity in the first and second derivatives of the spline at the data points. At the endpoints, the second derivative is set to zero, which is termed a "natural" spline at the endpoints, as the curvature goes to zero.

7.2.4 Polynomial Splines

To calculate a generic polynomial spline, select *Calculate->Polynomial Spline* from the main menu. A prompt will appear to ask for the degree of the polynomial spline; by way of example, a degree of 1 would be a linear spline, and a degree of 3 would be a cubic spline.

7.2.5 Tension Splines

To calculate a tension spline, select *Calculate->Tension Spline* from the main menu. A prompt will appear to ask for the amount of tension desired. Tension splines are based on hyperbolic functions, and simulate a cord being stretched with a defined tension (amount of force) between the data points. An extremely high tension approaches a linear spline, and low tensions will appear correspondingly "loose" around the data points, resembling a cubic spline.

7.3 Linear Regression

7.3.1 Introduction

Linear regressions, as a class of results, can be calculated directly, and do not need an iterative process like nonlinear regressions do. See *Linear Regression* in the appendices for a more in-depth explanation. A linear regression can be constructed from any model that is a linear combination of functions; the coefficients in the linear combination are the parameters to be found.

There are two types of linear regressions supported in CurveExpert: linear and polynomial. All other regressions, even if they could be calculated as a linear-type regression through a variable transformation, are computed with the nonlinear regression engine.

7.3.2 Linear Fit

In CurveExpert Professional, you can choose to calculate a straight line regression:

$$y = ax + b$$

by choosing *Calculate->Linear Fit*. For multivariate datasets (more than one independent variable), a straight line regression is computed as

$$y = \sum_{j}^{NVAR} a_j x_j + b$$

For example, for a 3D data set,

$$y = ax_1 + bx_2 + c$$

7.3.3 nth Order Polynomial Fit

To calculate a polynomial via linear regression, choose *Calculate->nth Order Polynomial Fit*. A prompt will appear to ask for the degree of polynomial desired. Also, in the same prompt, you can choose to force the polynomial through the origin, which forces the intercept to zero. Also, you may choose the desired weighting for each point in the dataset (see weighting for further information) After entering the degree (and origin forcing or weighting as appropriate), the polynomial will be computed and added to the result list.

Note: Polynomial fits computed via linear regression are supported only for 2D datasets. To compute a similar polynomial for multivariate datasets, use the nonlinear regression capability, selecting or defining a model as necessary.

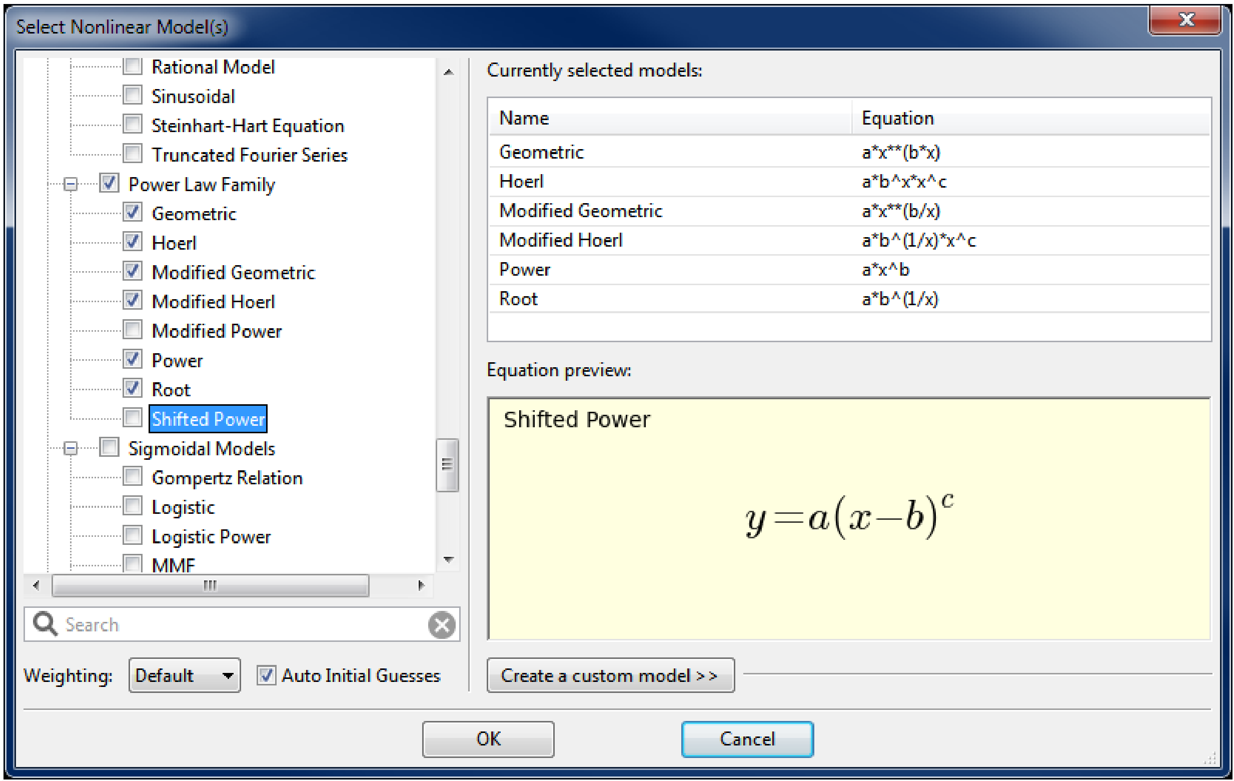
7.4 Nonlinear Regression

7.4.1 Introduction

Nonlinear regressions are solved with the Marquart-Levenberg method as documented in *Nonlinear Regression* in the appendices. To calculate a nonlinear regression, select *Calculate->Nonlinear Model Fit* from the main menu.

7.4.2 Selecting a Model or Models

Upon selecting *Calculate->Nonlinear Model Fit*, the nonlinear regression picker will appear; all models appear in this picker (built-in models and custom models) that are appropriate for the number of independent variables in the dataset. A screenshot of the model picker is below:



- □ Rational Model
- □ Sinusoidal
- □ Steinhart-Hart Equation
- □ Truncated Fourier Series
- ☑ Power Law Family
  - ☑ Geometric
  - ☑ Hoerl
  - ☑ Modified Geometric
  - ☑ Modified Hoerl
  - □ Modified Power
  - ☑ Power
  - ☑ Root
  - □ Shifted Power
- □ Sigmoidal Models
  - □ Gompertz Relation
  - □ Logistic
  - □ Logistic Power
  - □ MMF

Weighting: Default ☑ Auto Initial Guesses

Currently selected models:

Name	Equation
Geometric	a*x**(b*x)
Hoerl	a*b^x*x^c
Modified Geometric	a*x**(b/x)
Modified Hoerl	a*b^(1/x)*x^c
Power	a*x^b
Root	a*b^(1/x)

Equation preview:

Shifted Power

$$y = a(x-b)^c$$

On the left side of the dialog is where the desired models can be selected. Models can either be selected individually (by clicking the checkboxes next to the model), or a family at a time (by selecting the checkbox next to the desired family). All models can be selected by clicking the "root" of the tree labeled "Nonlinear Models".

A search field appears below the picker, so that you can filter the models to a certain specification. For example, typing "sig" in the search field will show all models that have "sig" in their name, or belong to a family with "sig" in its name. The search field allows you to quickly find a desired model in the hierarchy.

For convenience, the list of currently selected models will appear in the **Currently Selected Models** list in the upper right region of the picker. A preview of the equation for the currently pointed-to model is rendered in the **Equation preview** region in the bottom right area of the picker.

The *Automatic Initial Guesses* checkbox allows you to enable or disable automatic initial guessing for the calculation of the selected nonlinear models. If this box is enabled, CurveExpert Professional will provide high-quality initial guesses for built-in models, and for custom models, the custom model initialization (if any) will be called. If this box is disabled, you will be prompted for initial guesses for every model selected.

Note: if automatic initial guesses are disabled, the multicore capability, if enabled, will not be used for the currently selected batch of models.

Creating a Custom Model

The nonlinear model picker also provides for a quick way of creating a custom model (normally, custom models are created with *Tools->Custom Models*, see *Creating Custom Models and Functions*). To utilize this feature, simply click on the **Create a Custom Model** expander. This will open a small area in which you can type the name and equation for a model, and save it. Upon saving, the new custom model will be immediately available in the left pane for selecting.

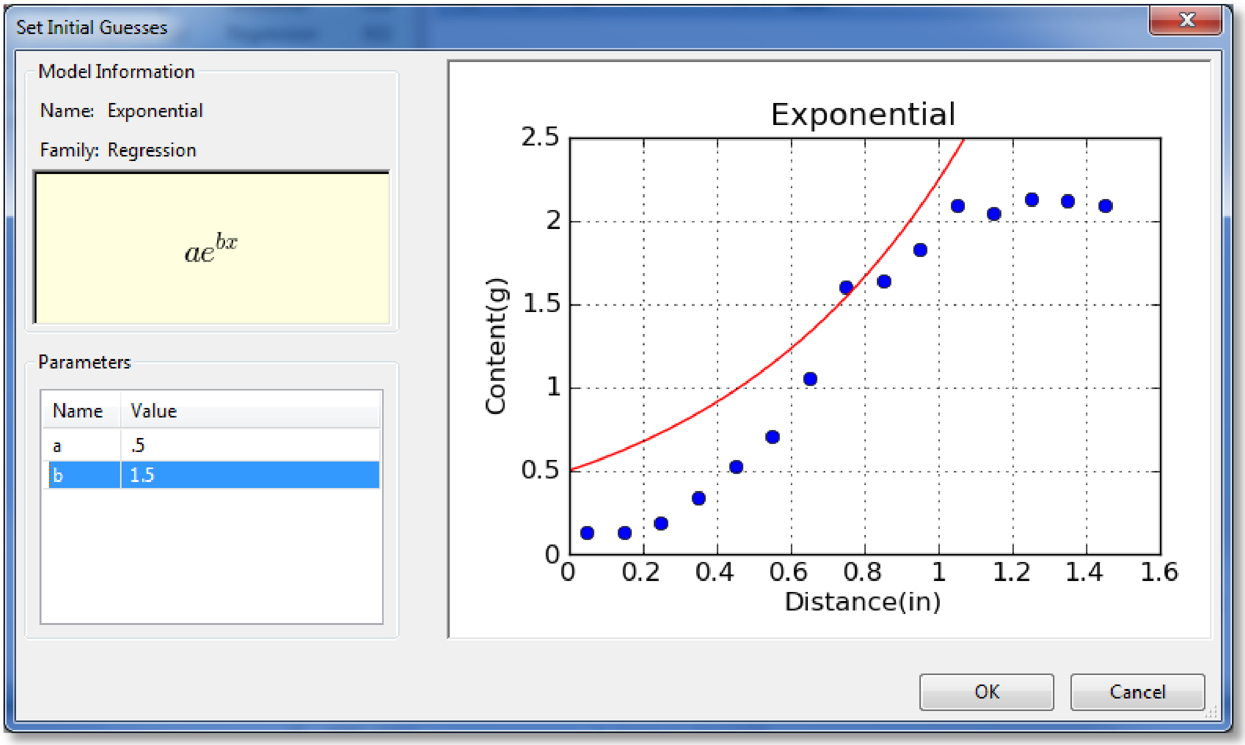
7.4.3 Setting the Weighting Scheme

To set the weighting scheme desired for the models that are to be computed, select the weighting scheme from the chooser at the bottom left of the dialog. See weighting for further details on the weighting schemes.

7.4.4 Manually setting initial guesses

Two situations cause the manual initial guess window to appear; one is if you choose to disable automatic initial guesses in the nonlinear model picker. The other is if a nonlinear regression fails, and you choose to set the initial guesses yourself in an effort to successfully calculate that model.

The manual initial guess window is shown below:



Set Initial Guesses

Model Information

Name: Exponential

Family: Regression

ae^{bx}

Parameters

Name	Value
a	.5
b	1.5

For informational purposes, the name, family, and equation for the nonlinear regression is shown in the upper left quadrant of the window. The parameters can be adjusted in the bottom left quadrant by clicking on the entries. As you adjust parameters, the graph drawn on the right will adjust accordingly. This gives real-time feedback on the parameter adjustment so that you can quickly refine the initial guesses into a reasonable state.

7.5 Smoothing

7.5.1 Introduction

Smoothings generate a new set of data points from the existing dataset. In CurveExpert Pro, Lowess smoothing is supported, as well as moving averages with an averaging window defined by the user.

7.5.2 Lowess Smoothing

To calculate a Lowess smoothing, select *Calculate->Lowess Smoothing* from the main menu. Lowess smoothing (which stands for LOcally WEighted Scatterplot Smoothing) builds on regular linear regression in order to smooth an existing dataset; in this context, an entirely new dataset will be generated that represents the smoothed data. Lowess smoothing generates a polynomial fit to a subset of data around each point in the dataset. More weighting is given to points near the point of interest, and less weight is given to points far away. After the polynomial is determined, a “smoothed” y data value is obtain by evaluating the polynomial at x.

The weighting function used is the tri-cube weighting function for $|x| < 1$:

$$w(x) = \left(1 - |x|^3\right)^3$$

where $x = (x_i - x_p)/s$, and the weighting function is zero for $|x| >= 1$. x_p is the point of interest, and s is the maximum distance of any point in the local dataset to x_p.

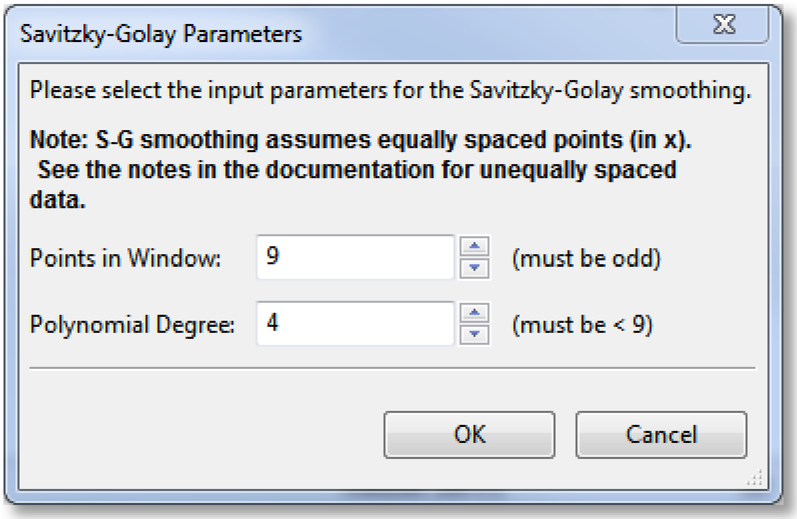
The polynomial used by CurveExpert Pro is a simple degree-one linear regression. Once the smoothed values are obtained, a linear spline is used to connect the points for visualization and evaluation purposes.

Note: William S. Cleveland: "Robust locally weighted regression and smoothing scatterplots", Journal of the American Statistical Association, December 1979, volume 74, number 368, pp. 829-836.

Note: William S. Cleveland and Susan J. Devlin: "Locally weighted regression: An approach to regression analysis by local fitting", Journal of the American Statistical Association, September 1988, volume 83, number 403, pp. 596-610.

7.5.3 Savitzky-Golay Smoothing

To calculate a Savitzky-Golay smoothing, select *Calculate->Savitzky-Golay Smoothing* from the main menu. Savitzky-Golay (SG) smoothing combines a moving-average-type averaging with a locally-fitted polynomial in order to smooth an existing dataset. The user has the opportunity to select the number of points in the moving average window (the number of points must be odd), and the degree of the polynomial used to fit the data within each window. Additionally, the degree of the The input dialog for the SG parameters is shown below.



Please select the input parameters for the Savitzky-Golay smoothing.

Note: S-G smoothing assumes equally spaced points (in x). See the notes in the documentation for unequally spaced data.

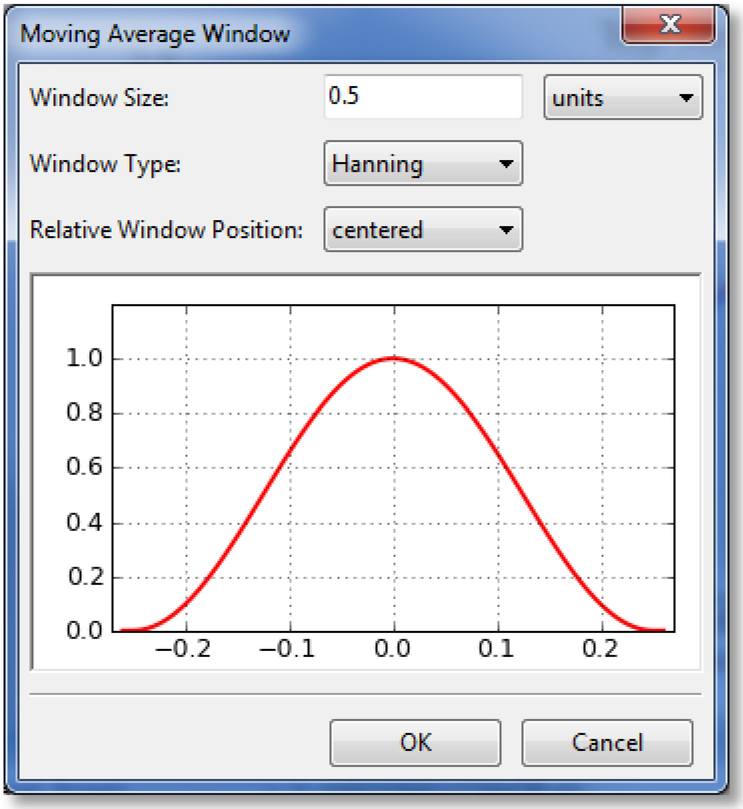
Points in Window: 9 (must be odd)

Polynomial Degree: 4 (must be < 9)

SG smoothing is primarily intended for data that is equally spaced in x (typically, x is time). In cases where the x data is not equally spaced, CurveExpert Pro still assumes that it is for the purposes of performing the smoothing. Doing so virtually shifts the data points to equally spaced positions within the averaging window. In cases where smoothing is useful in general, this shift is the equivalent of introducing an additional source of noise in the function values; however, this noise will often be much smaller than the noise already present in the samples. Thus, the smoothing is still useful, but should be used with care. The nature of smoothing in general is just to guide the eye through a forest of points, so the user can determine if the output smoothing performs its desired task rather easily.

7.5.4 Moving Averages

Moving average applies an averaging window to each point in a dataset, in turn, in order to generate a new set of data. The size and type of window determine the amount of averaging that will take place. To calculate a moving average, select *Calculate->Moving Average* from the main menu. A dialog will appear that will allow you to select the window to utilize for the averaging.



Moving Average Window

Window Size: 0.5 units

Window Type: Hanning

Relative Window Position: centered

The first parameter to select for the window is its size. Here, you can select the size (sometimes called the extent) of the window in terms of units (meaning a distance along the x axis), or in terms of number of points. The (0.0) location on the preview is always defined to be at the x location of the point being window averaged. The window preview will continually update in order to show the currently defined window. Note that the underlying dataset must be sorted in order for the window to be specified in terms of a number of points. If the dataset is not sorted, the *points* choice is disabled.

The window type can also be selected from the following:

- rectangular
- Bartlett (triangular)
- Blackman
- Hamming
- Hanning

The window type affects the weighting factor that is applied to each point within the window that is to be averaged. For example, a rectangular window applies a weight of 1 to all points within the window.

Finally, the relative window position can be set. Usually, a centered window is used, which places the center of the window at the same x location as the data point being averaged (location 0.0). Alternatively, you can select a leading or lagging window, which places the window ahead of the data point or behind the data point, respectively.

After tuning the window parameters to the desired settings, click “OK”, and the moving average will compute and display.

7.6 Functions

7.6.1 Introduction

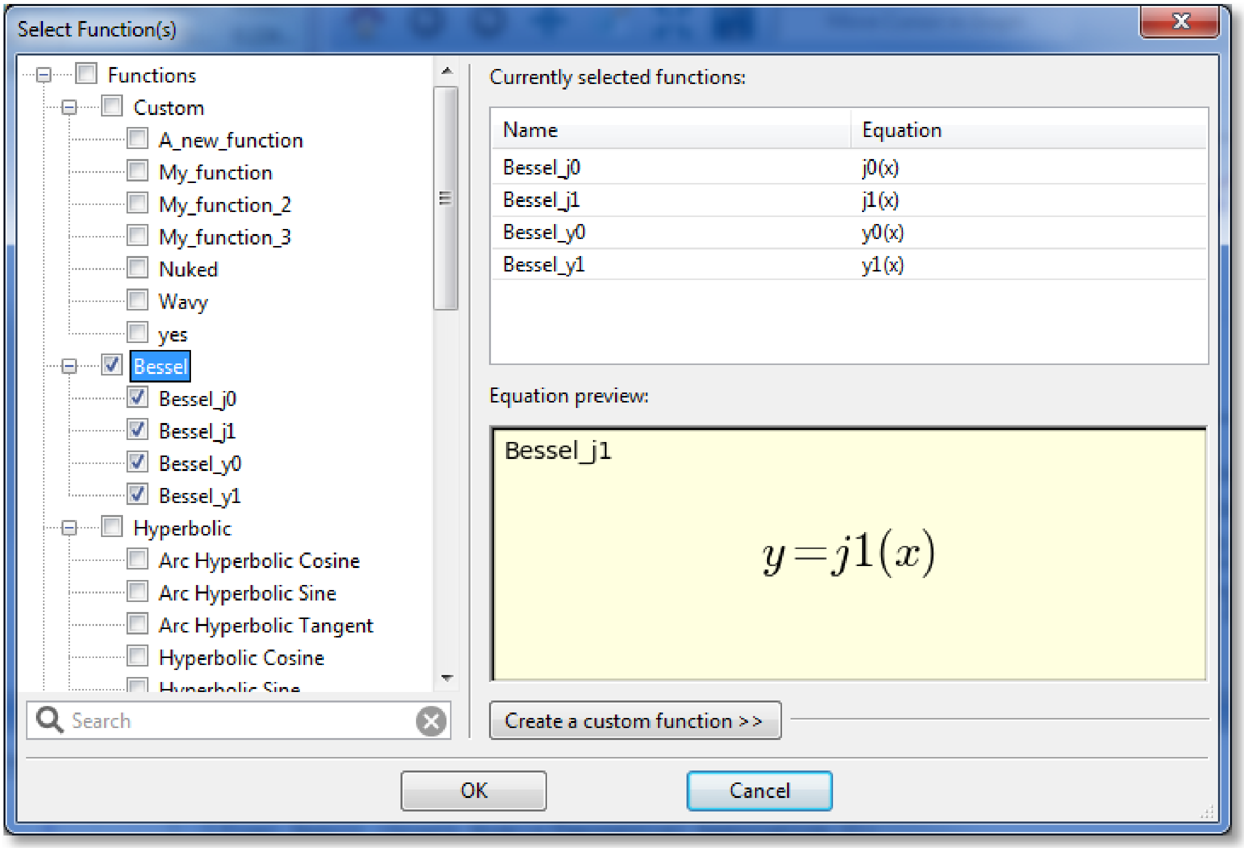
Functions in CurveExpert Professional depend on the independent variables (x) only, with no free parameters. Functions are available in CurveExpert Pro for convenience, and you can perform all of the normal operations on functions that you can on any other results.

To calculate a function, select *Calculate->Functions* from the main menu. There does not have to be a dataset present in order to compute a function.

Note: A distinction is made between “models” and “functions”. Models have free parameters, and are therefore dependent on the independent variables and the parameters: $f(\vec{x};\vec{a})$. Functions are dependent on the independent variables only, with no free parameters: $f(\vec{x})$.

7.6.2 Selecting a Function or Functions

Upon selecting *Calculate->Functions*, the function picker will appear; all functions appear in this picker (built-in functions and custom functions) that are appropriate for the number of independent variables in the dataset. A screenshot of the function picker is below:



- □ Functions
  - □ Custom
    - □ A_new_function
    - □ My_function
    - □ My_function_2
    - □ My_function_3
    - □ Nuked
    - □ Wavy
    - □ yes
  - ☑ Bessel
    - ☑ Bessel_j0
    - ☑ Bessel_j1
    - ☑ Bessel_y0
    - ☑ Bessel_y1
  - □ Hyperbolic
    - □ Arc Hyperbolic Cosine
    - □ Arc Hyperbolic Sine
    - □ Arc Hyperbolic Tangent
    - □ Hyperbolic Cosine
    - □ Hyperbolic Sine

Currently selected functions:

Name	Equation
Bessel_j0	j0(x)
Bessel_j1	j1(x)
Bessel_y0	y0(x)
Bessel_y1	y1(x)

Equation preview:

Bessel_j1

$$y = j1(x)$$

On the left side of the dialog is where the desired functions can be selected. Functions can either be selected individually (by clicking the checkboxes next to the model), or a family at a time (by selecting the checkbox next to the desired family). All functions can be selected by clicking the 'root' of the tree labeled 'Functions'.

A search field appears below the picker, so that you can filter the functions to a certain specification. For example, typing "bes" in the search field will show all functions that have "bes" in their name, or belong to a family with "bes" in its name. The search field allows you to quickly find a desired function in the hierarchy.

For convenience, the list of currently selected functions will appear in the **Currently Selected Functions** list in the upper right region of the picker. A preview of the equation for the currently pointed-to function is rendered in the **Equation preview** region in the bottom right area of the picker.

Creating a Custom Function

The function picker also provides for a quick way of creating a custom function (normally, custom functions are created with *Tools->Custom Functions*, see *Creating Custom Models and Functions*). To utilize this feature, simply click on the **Create a Custom Function** expander. This will open a small area in which you can type the name and equation for a function, and save it. Upon saving, the new custom function will be immediately available in the left pane for selecting.

CHAPTER

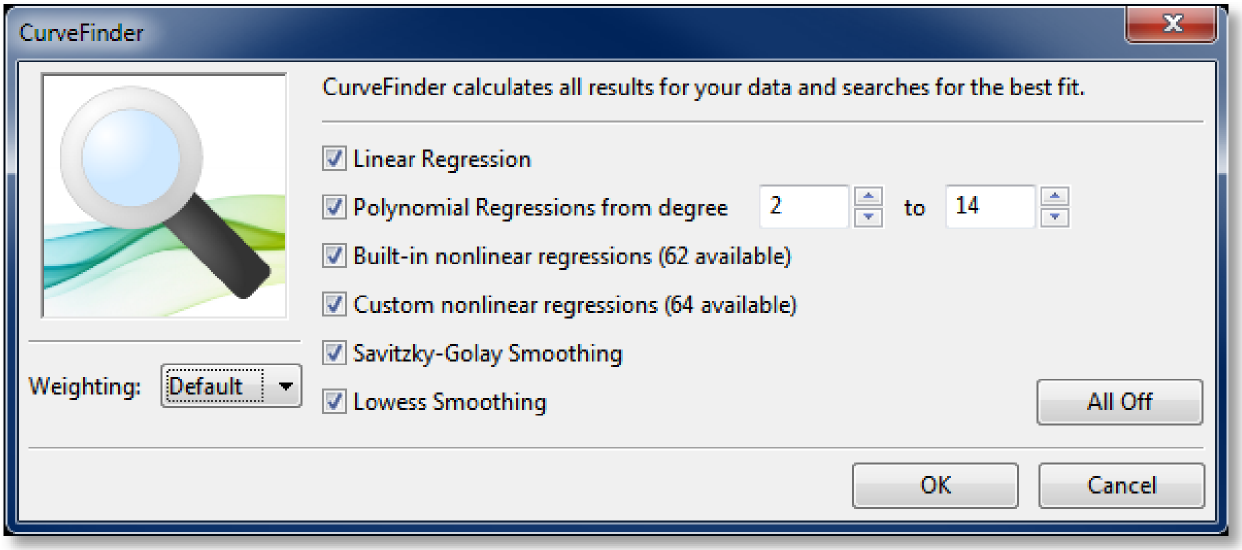
EIGHT

CURVEFINDER

8.1 Usage

CurveFinder is a tool that you can use to search all available regressions (and Lowess smoothing) for the best fit to your data. When CurveFinder is run, it will compute a fairly large list of results, and they will be shown in ranked order in the Results pane.

The use of the CurveFinder tool is quite straightforward; simply choose *Tools->CurveFinder* from the main menu, and then select the results that you want included in the calculation.



CurveFinder calculates all results for your data and searches for the best fit.

☑ Linear Regression

☑ Polynomial Regressions from degree 2 to 14

☑ Built-in nonlinear regressions (62 available)

☑ Custom nonlinear regressions (64 available)

☑ Savitzky-Golay Smoothing

☑ Lowess Smoothing

Weighting: Default

In this context “Linear Regression” refers to a polynomial of degree 1, which is the same straight line obtained from choosing *Calculate->Linear*. You can also choose the range of polynomials that should be searched, which ranges from 2 up to the lesser of 20 and the number of points in the dataset minus one. Nonlinear regressions (either built-in, custom, or both) can also be searched. Also, a Lowess smoothing can be searched.

For convenience, you can use the “All On”/”All Off” button to turn all results on or off, as appropriate. After selecting the set of results that should be computed, simply press the OK button.

You may also select a weighting that can be used for all fits generated by the CurveFinder tool. To do, select the appropriate weighting scheme in the chooser on the left side of the dialog. See weighting for further details.

8.2 Cautions

Just because a particular model is ranked high in the Results pane does not mean that it is necessarily a good choice. Make sure to visually inspect the curve that represents your model. Polynomials, in particular, are very prone to oscillation, and can veer wildly away from the dataset before returning to be near the data points at each *x* value. Make sure that the model that you choose as best is a reasonable representation of your data.

CHAPTER

NINE

WORKING WITH RESULTS

9.1 Introduction

Once any results have been calculated, whether they are curve fits (regressions), interpolations, smoothings, or functions, they appear in the result pane ranked in order of their score (the result pane normally is shown in the upper left part of the CurveExpert Professional application, see *User Interface*). The score reflects how closely the model adheres to the underlying data. Note that functions (as opposed to models) do not have a score, as they are not associated with the data directly in any way.

In the results pane, the results can also be sorted by any of the items displayed in the column header (name, kind, family, score, correlation coefficient, coefficient of determination, or standard error) simply by clicking the appropriate column header. Note that you can right-click on the column header in order to customize what is displayed in the results list; by default, the name, kind, and score is displayed.

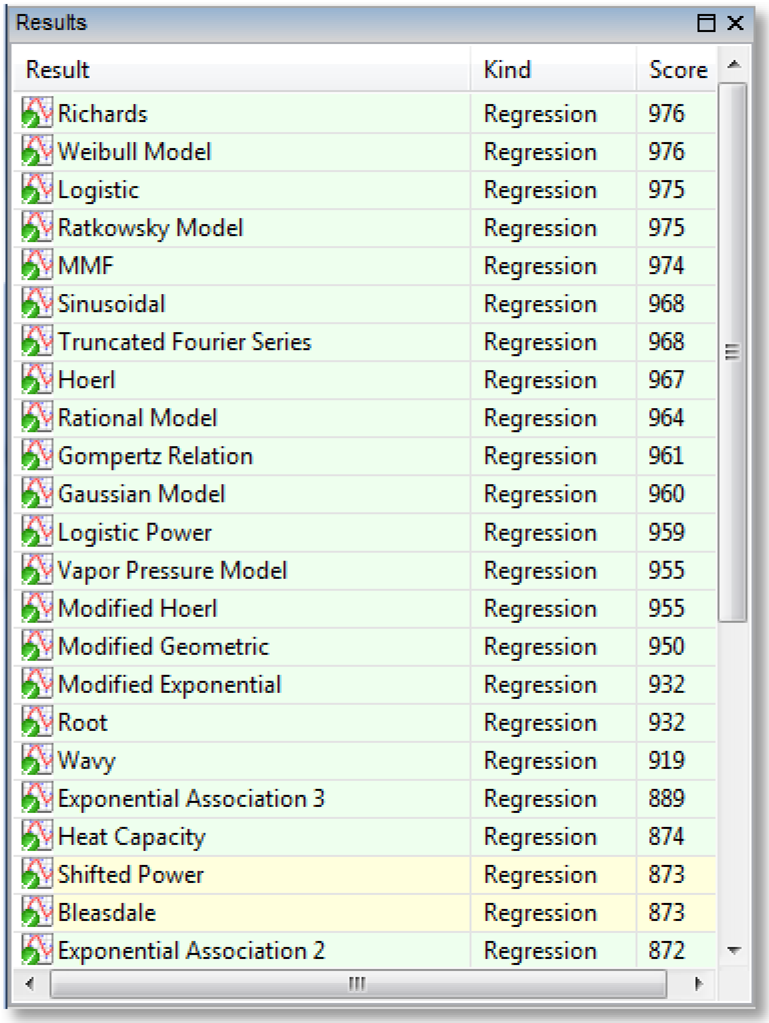
When selecting results, you can simply click on the result (which selects it), or use multiple selection by holding down SHIFT or CTRL when clicking on results. This allows you to apply an operation to batches of results at once.

Every result in the results pane has a unique icon indicating the kind of result that it is. The table below shows meaning of each icon.

Icon	Meaning
📈	Linear regression model (includes polynomials)
∿	Nonlinear regression model (curve fit)
⌒	Data smoothing
√x	Function

9.2 Color Coding

Also in the results pane, each result has a color coding of green or yellow. An example is shown below:



Result	Kind	Score
Richards	Regression	976
Weibull Model	Regression	976
Logistic	Regression	975
Ratkowsky Model	Regression	975
MMF	Regression	974
Sinusoidal	Regression	968
Truncated Fourier Series	Regression	968
Hoerl	Regression	967
Rational Model	Regression	964
Gompertz Relation	Regression	961
Gaussian Model	Regression	960
Logistic Power	Regression	959
Vapor Pressure Model	Regression	955
Modified Hoerl	Regression	955
Modified Geometric	Regression	950
Modified Exponential	Regression	932
Root	Regression	932
Wavy	Regression	919
Exponential Association 3	Regression	889
Heat Capacity	Regression	874
Shifted Power	Regression	873
Bleasdale	Regression	873
Exponential Association 2	Regression	872

A green color coding indicates that the result was calculated successfully; it does **not** mean that the result is necessarily good; that is indicated by its score. The green coloring only indicates that the calculation of the result proceeded smoothly, and it is not possible to get a better result than what is given. In the common parlance of nonlinear regression, the parameters of the fit are said to be "converged".

A yellow color coding indicates that the result should only be used with caution. It indicates that the calculation succeeded, but with reservations. The most common cause for this is a nonlinear regression calculation that did not converge before the maximum number of iterations (configurable via *Edit->Preferences->Regression*) was reached. Again, the score will indicate how well the model adheres to the data.

If a result is presented in yellow, it is highly recommended to examine the residual and parameter histories to see if the parameters would have changed further if the max number of iterations is higher. Double click on the result in the results pane, and select the *Convergence* tab in the resulting dialog. Also examine the *PHist* tab.

9.3 Badges

Every result is badged with either a green checkmark or a yellow exclamation point. The intention here is to inform the user when a result becomes invalid due to a change in the underlying dataset. For example, if a nonlinear regression is calculated, and then the dataset is scaled (using, say, the tools in the Data menu), that nonlinear regression becomes invalid, and an attention badge is placed on that result in the results pane to signify this.

To remedy the situation, you can update the result yourself by right clicking on the result and selecting "Update", or, if you would like to keep the old result and compute a new one, simply reselect the desired result from the *Calculate* menu, and an entirely new result will be generated.

Badge	Meaning
✓	Data unchanged since calculation of this result
⚠	Data changed since calculation of this result

9.4 Previewing

Pointing at a result will show its preview in the preview pane, usually at the bottom left of your application. Also, the standard error and correlation coefficient, if applicable, will display in the status bar at the bottom of the application window.

9.5 Placing Results on a Graph

There are several ways for a result to be shown on a graph; the first is automatic, in that the top scoring results (usually the top 3), are automatically placed in the "Top Results" graph whenever a new calculation is performed.

To place a specific result on a graph that you have created (press "+" in the Graphs and Data pane), you can:

- right click a result in the Results pane, and select *Send to New Plot*. This will create a new graph with the default graph theme (as selected

in the application preferences), and add the selected results to that plot. * right click a result in the Results pane, and select *Send to Current Plot*. This will plot the selected results directly on the active plot. * drag and drop the result from the Results pane into a graph, and that will add the result to the graph.

Note: To send multiple results to a graph at once, you can multiple select items in the result pane with either the shift or control key. Then drag and drop, or right-click and select *Send to New Plot* or *Send to Current Plot* as normal.

9.6 Removing Results

If you decide that a result is no longer needed, select it, right click, and pick *Remove*. This action is not undoable, but certainly the result can be calculated again by simply picking it from the *Calculate* menu.

Like placing results on a graph, you can remove multiple results at once by shift-clicking or control-clicking to select the results that you want to remove. If the results are shown in any graph, they are removed from the graph as well.

9.7 Creating/Exporting a Table

Directly from a result, you can generate a table that consists of a column of x (independent variable) data, and the corresponding column of the result evaluated at each of the points.

Here, the user can specify the x (independent variable) data, and CurveExpert Professional will fill in the y (dependent variable) data. There are four methods to specify the x data: 1) by min, max, and increment for evenly spaced data, 2) by min, max and number of points for evenly spaced data, 3) use the x data from the currently loaded dataset, or 4) by reading a file. Set the first pulldown to “Range/incr”, “Range/npts”, “Data”, or “File” as appropriate to select the method by which you would like to specify the data.

For specifying by **Range (with the spacing specified)**, select **Range/incr** and set the min, max, and increment of your table, click *Update*, and the table spreadsheet will update to show the calculated values. To write the table, press *Save…* at the bottom. Alternatively, if you want your table on the clipboard for subsequent pasting into another application, press *Copy*.

For specifying by **Range (with the number of points specified)**, select **Range/npts** and set the min, max, and number of points (rows) of your table, click *Update*, and the table spreadsheet will update to show the calculated values. To write the table, press *Save…* at the bottom. Alternatively, if you want your table on the clipboard for subsequent pasting into another application, press *Copy*.

For specifying by **Data**, there is no further work to do aside from setting the first pulldown to “Data”. The independent variable (x) will be extracted from the current dataset and utilized.

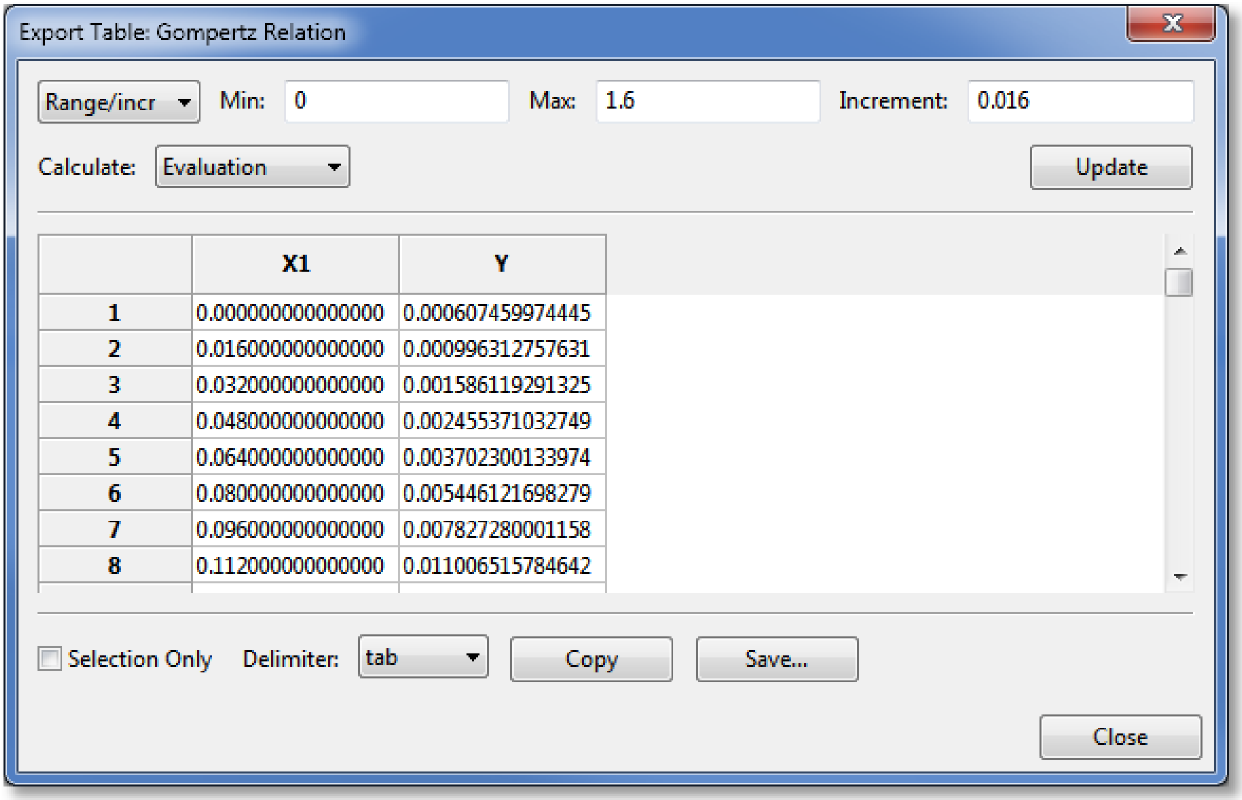
Likewise, for specifying by **File**, set the first pulldown to *File*, and click the *Browse* button to locate the file that you would like to read. Ideally, the file should be made up of a single row or a single column of data that will become the x data. If the file is made up of multiple columns, CurveExpert Professional assumes that the first column should be used as the x data. This capability allows the user to read datasets related to the original easily. As soon as a new filename is entered into the control, it will be read. To write the table, press *Save…* at the bottom. Alternatively, if you want your table on the clipboard for subsequent pasting into another application, press *Copy*.

Normally, an “Evaluation” style table is appropriate; an evaluation table simply evaluates the results at each of the points in the independent variable column. An “Accumulation” table computes the integral

$$\int_{x_0}^{x_i} f(x;\vec{a})\,dx$$

for every point in the dataset, where x_0 is the min value specified for the table.

At the bottom of the table, before you copy or save, you can pick *Selection only* to only save the data that you have currently selected in the table spreadsheet. Otherwise, the entire table is saved or copied. Also, you can select the delimiter to use between values on the same row. The default delimiter can be set at *Edit->Preferences->General->Default delimiter*.



Range/incr Min: 0 Max: 1.6 Increment: 0.016

Calculate: Evaluation

	X1	Y
1	0.000000000000000	0.000607459974445
2	0.016000000000000	0.000996312757631
3	0.032000000000000	0.001586119291325
4	0.048000000000000	0.002455371032749
5	0.064000000000000	0.003702300133974
6	0.080000000000000	0.005446121698279
7	0.096000000000000	0.007827280001158
8	0.112000000000000	0.011006515784642

□ Selection Only Delimiter: tab

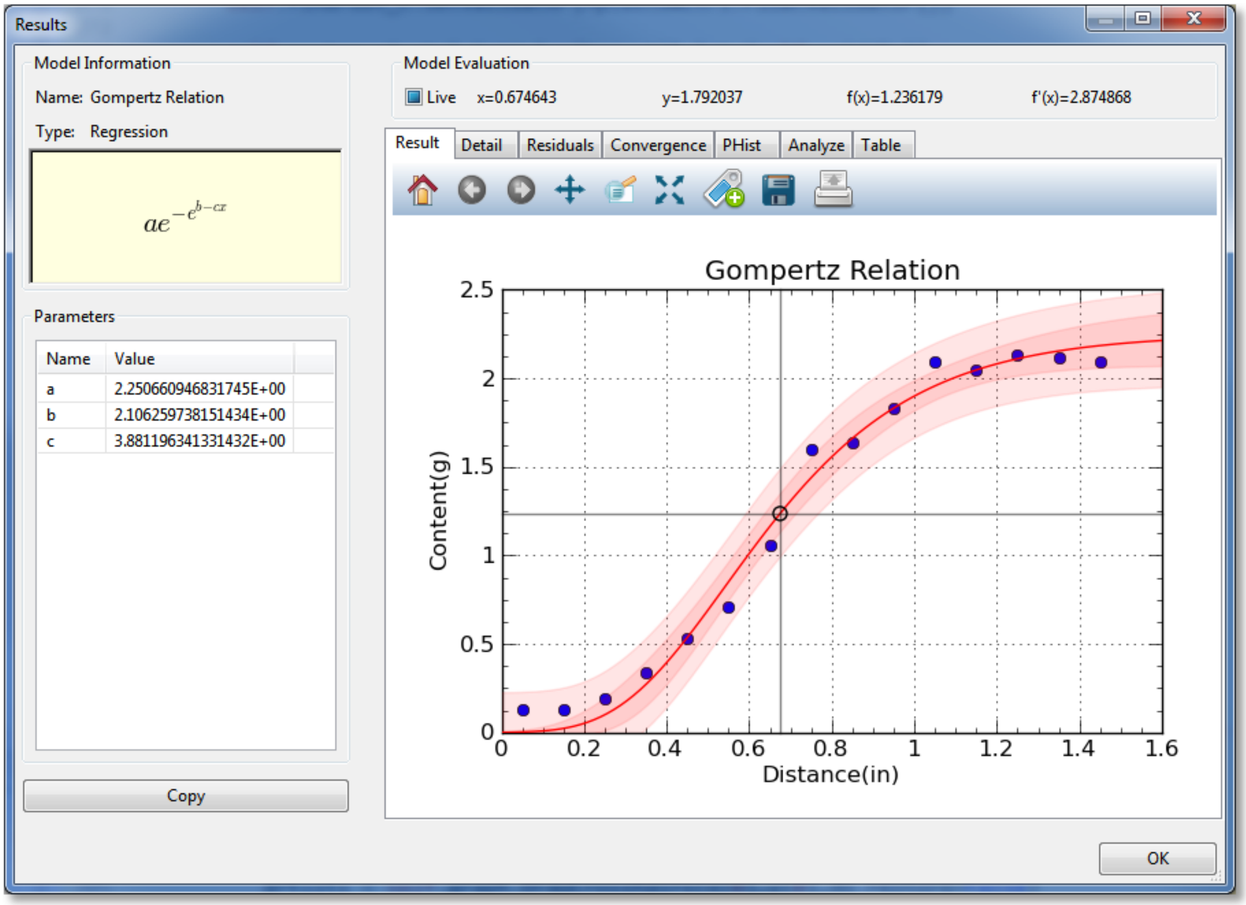
9.8 Copying Result Information

Right clicking on a result will allow you to select *Copy All Info*, which will copy a report about the selected result(s) to the clipboard.

If the selected result is a regression, the entry *Copy Parameters* will also appear. Select this to copy only the parameter name/value pairs to the clipboard, separated by the delimiter selected in the application preferences.

9.9 Querying Result Details

If you want to get more detail on a calculated result, either double-click it in the Results pane, or right-click and select *Details*. This will display a window very similar to the following:



Results

Model Information

Name: Gompertz Relation

Type: Regression

$$ae^{-e^{b-cx}}$$

Parameters

Name	Value
a	2.250660946831745E+00
b	2.106259738151434E+00
c	3.881196341331432E+00

Model Evaluation

☑ Live x=0.674643 y=1.792037 f(x)=1.236179 f'(x)=2.874868

In the result details window, you can examine a particular result much more closely. The left pane shows the name of the result, its family, and an equation for the result, if applicable. For a model, the fitted parameters are also shown on the left. To copy result details to the clipboard in text format, simply click the *Copy* button.

The notebook on the right contains several items of interest, most notable of which is the result graph (in which a graph is shown with just the data and the single result), and the table tool, in which a data table can be generated with this result. The following sections describe each section of the notebook.

9.9.1 Result Graph

The result graph simply shows the data and the current result, so that you can visualize the result without the (possible) clutter of having other results on the same plot. All of the graphing capabilities (see *Graphing*) are also available here, including zooming, panning, changing the graph properties, saving the plot to file or clipboard, and printing.

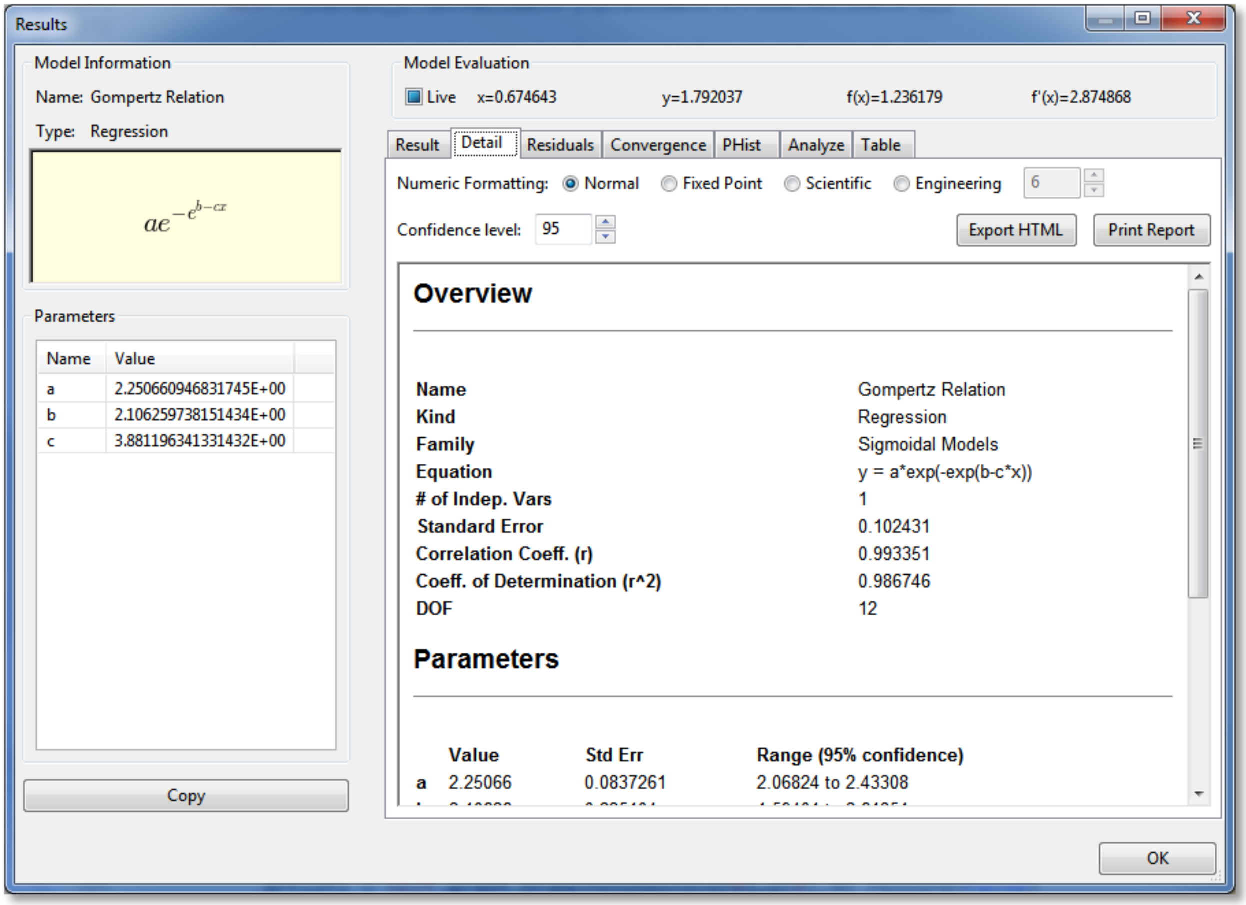
As you move your pointer over the graph, the live model evaluation tool above the notebook activates and displays your current xy location, the result evaluated at x, and the result's derivative evaluated at x. A marker follows the result depedending on where the mouse pointer is placed. This behavior can be disabled by unchecking the "Live" box above the graph, or it can be made to only activate when the left mouse button is pressed (by setting the "Live" box to its intermediate state).

9.9.2 Detail

The Detail notebook section contains a formatted report that shows the basic statistics of the result, such as the standard error, correlation coefficient, and coefficient of determination. Also, the covariance matrix and parameter uncertainties are displayed as appropriate.

The style of numeric formatting can be adjusted using controls at the top of the report, where you can choose from normal, fixed point, or scientific notation (and choose the precision for each as appropriate). Also, you can choose the confidence level with which the parameter uncertainties are reported. Any adjustment to any of these controls will cause the report to be regenerated automatically.

Press the *Export HTML* button in order to export an HTML file of the report. Press *Print Report* in order to send the report directly to the printer.



Name: Gompertz Relation

Type: Regression

$$ae^{-e^{b-cx}}$$

Parameters

Name	Value
a	2.250660946831745E+00
b	2.106259738151434E+00
c	3.881196341331432E+00

☐ Live x=0.674643 y=1.792037 f(x)=1.236179 f'(x)=2.874868

Numeric Formatting: ● Normal ○ Fixed Point ○ Scientific ○ Engineering 6

Confidence level: 95

Overview

Name	Gompertz Relation
Kind	Regression
Family	Sigmoidal Models
Equation	y = a*exp(-exp(b-c*x))
# of Indep. Vars	1
Standard Error	0.102431
Correlation Coeff. (r)	0.993351
Coeff. of Determination (r^2)	0.986746
DOF	12

Parameters

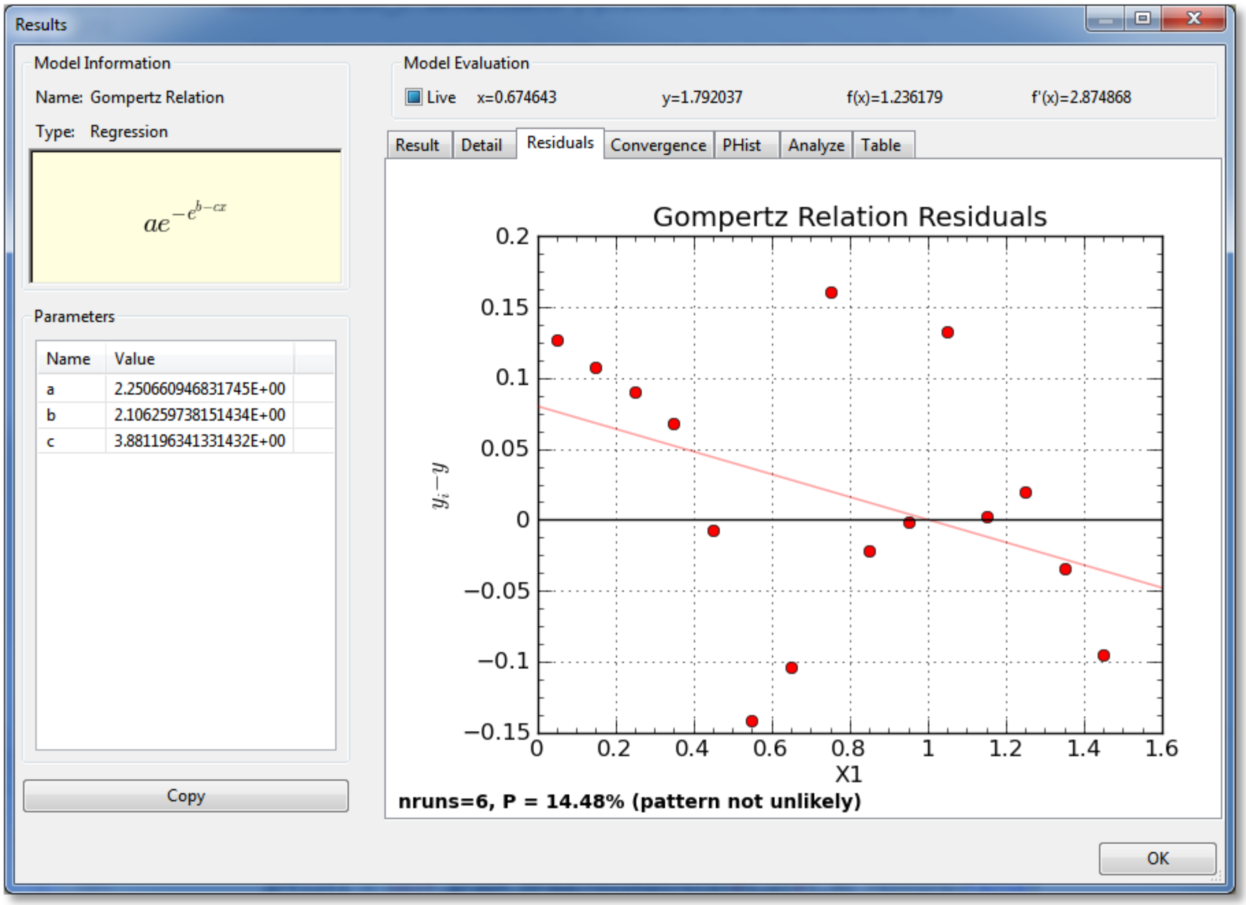
	Value	**Std Err**	**Range (95% confidence)**
a	2.25066	0.0837261	2.06824 to 2.43308

9.9.3 Residuals

The Residuals notebook section shows the difference between the result and the data, as a function of the independent variable. All of the graphing capabilities (see *Graphing*) are also available here, including zooming,panning, changing the graph properties, saving the plot to file or clipboard, and printing.

Also displayed on the residuals plot is a straight-line fit to the residual points. This regression line (colored light red) shows whether there is an upward or downward trend to the data (indicated by the slope of the line) as well as showing whether the residuals bias upward or downward.

Also, a Wald-Wolfowitz runs test is performed on the residuals. At the bottom of the plot, the observed number of runs is listed, as well as the likelihood that this observed number of runs could occur if the model used to fit the data was correct (i.e., the residuals are randomly distributed around the curve). If the probability is less than 5%, CurveExpert Professional states that the run pattern of the residuals is unlikely; if greater than 5%, the pattern is not unlikely (which is distinctly different than being likely, which cannot be claimed). A higher likelihood is desirable, obviously.



Name: Gompertz Relation

Type: Regression

$ae^{-e^{b-cx}}$

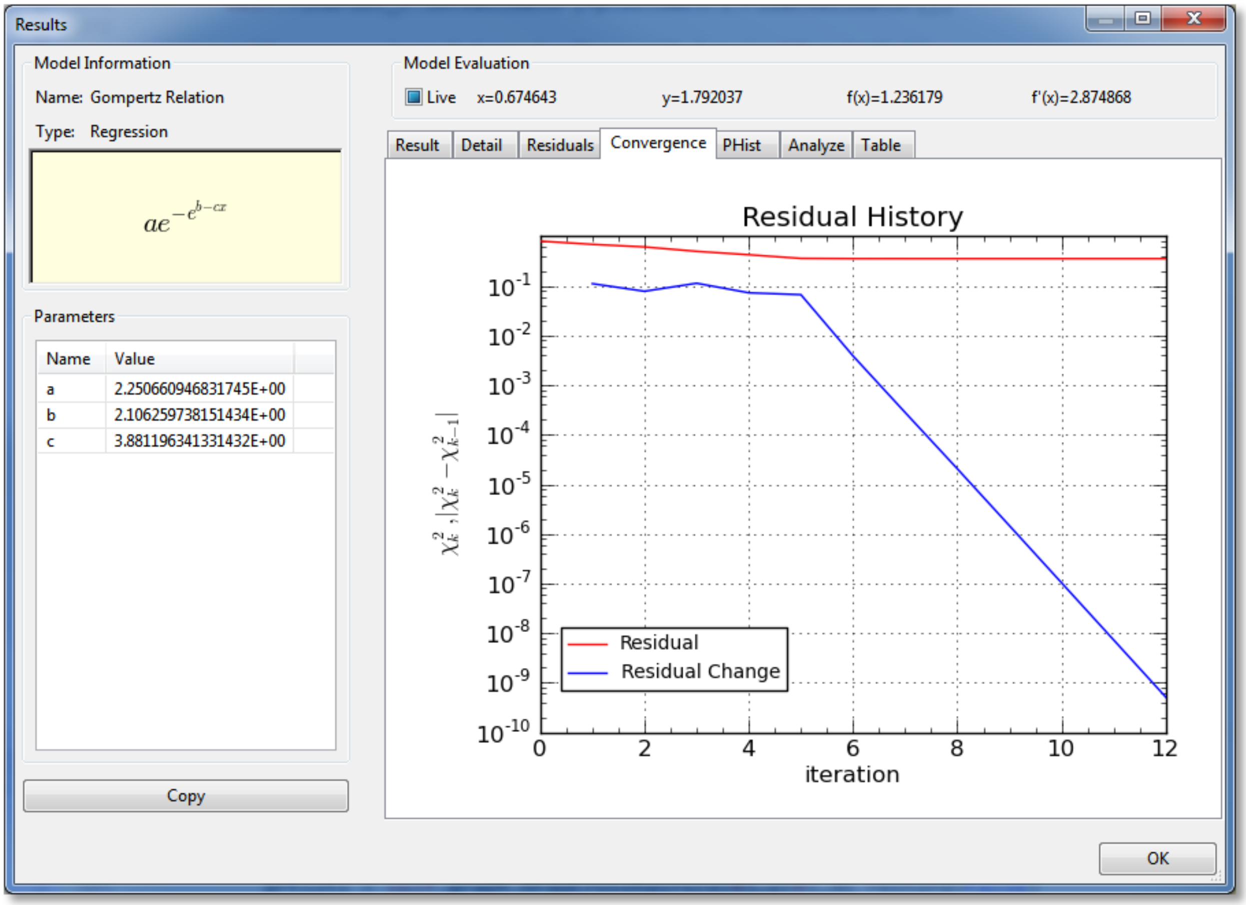
Name	Value
a	2.250660946831745E+00
b	2.106259738151434E+00
c	3.881196341331432E+00

☑ Live x=0.674643 y=1.792037 f(x)=1.236179 f'(x)=2.874868

nruns=6, P = 14.48% (pattern not unlikely)

9.9.4 Convergence

For nonlinear regressions (which is the only type of result for which this page appears), a graph of the convergence history is displayed. The norm of the residual (difference between the result and the data) is shown as a function of iteration number, and the change in the residual is also shown as a function of the iteration number. If the iteration has converged, the residual on the last iteration should be to the level set in the application preferences. One exception to this is if the iteration has terminated based on lack of change in parameters, rather than lack of change in the residual.



Name: Gompertz Relation

Type: Regression

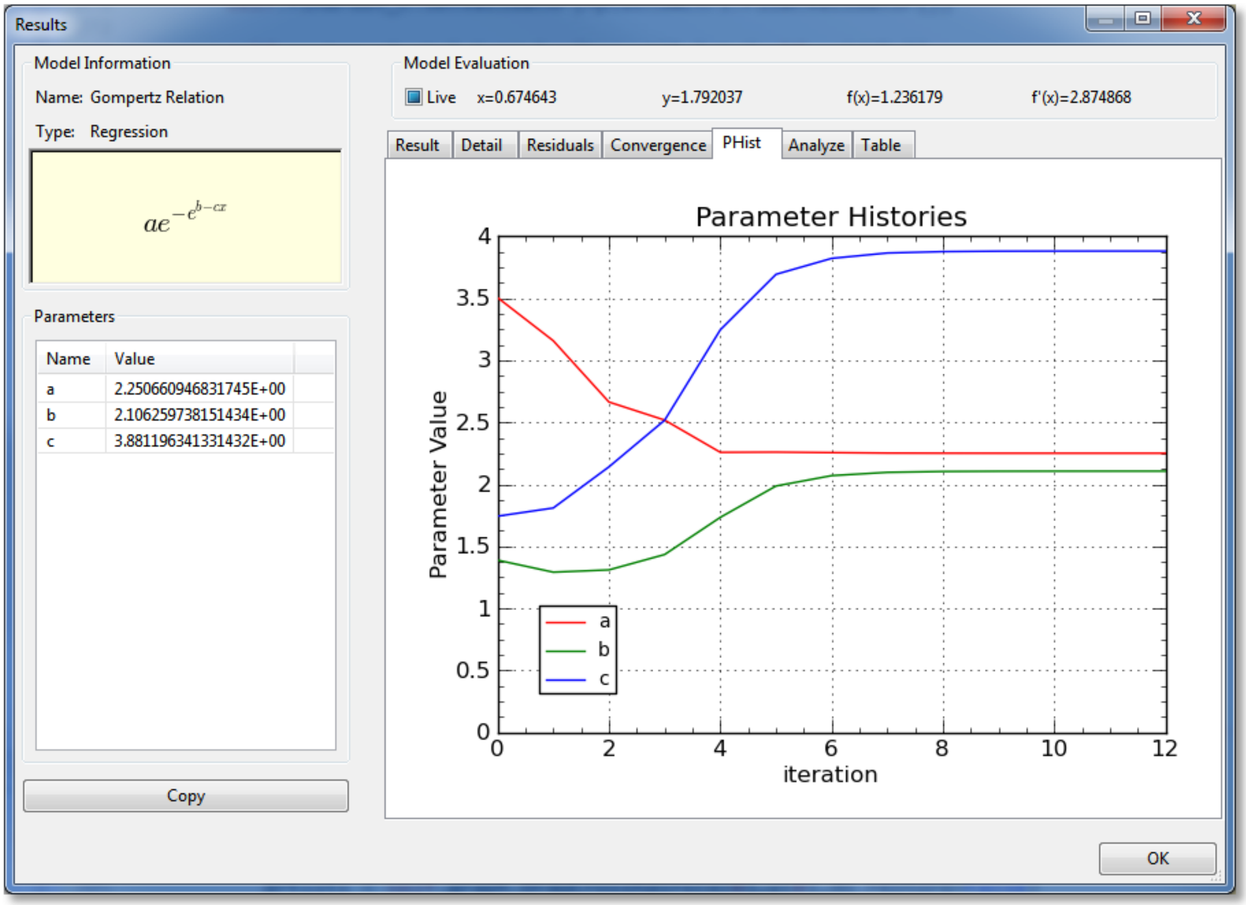
$ae^{-e^{b-cx}}$

☐ Live x=0.674643 y=1.792037 f(x)=1.236179 f'(x)=2.874868

Name	Value
a	2.250660946831745E+00
b	2.106259738151434E+00
c	3.881196341331432E+00

9.9.5 Parameter Histories (PHist)

For nonlinear regressions (which is the only type of result for which this page appears), a graph of the parameter histories is displayed. The value of each parameter is shown as a function of iteration number. Here, you can see whether or not the parameters have "settled" on a particular value before the iteration was terminated. Unless the iteration is terminated due to exceeding the maximum number of iterations set in the application preferences, the parameters are always flat at the right side of the plot, indicating that they are settled.



Model Information

Name: Gompertz Relation

Type: Regression

$ae^{-e^{b-cx}}$

Model Evaluation

■ Live x=0.674643 y=1.792037 f(x)=1.236179 f'(x)=2.874868

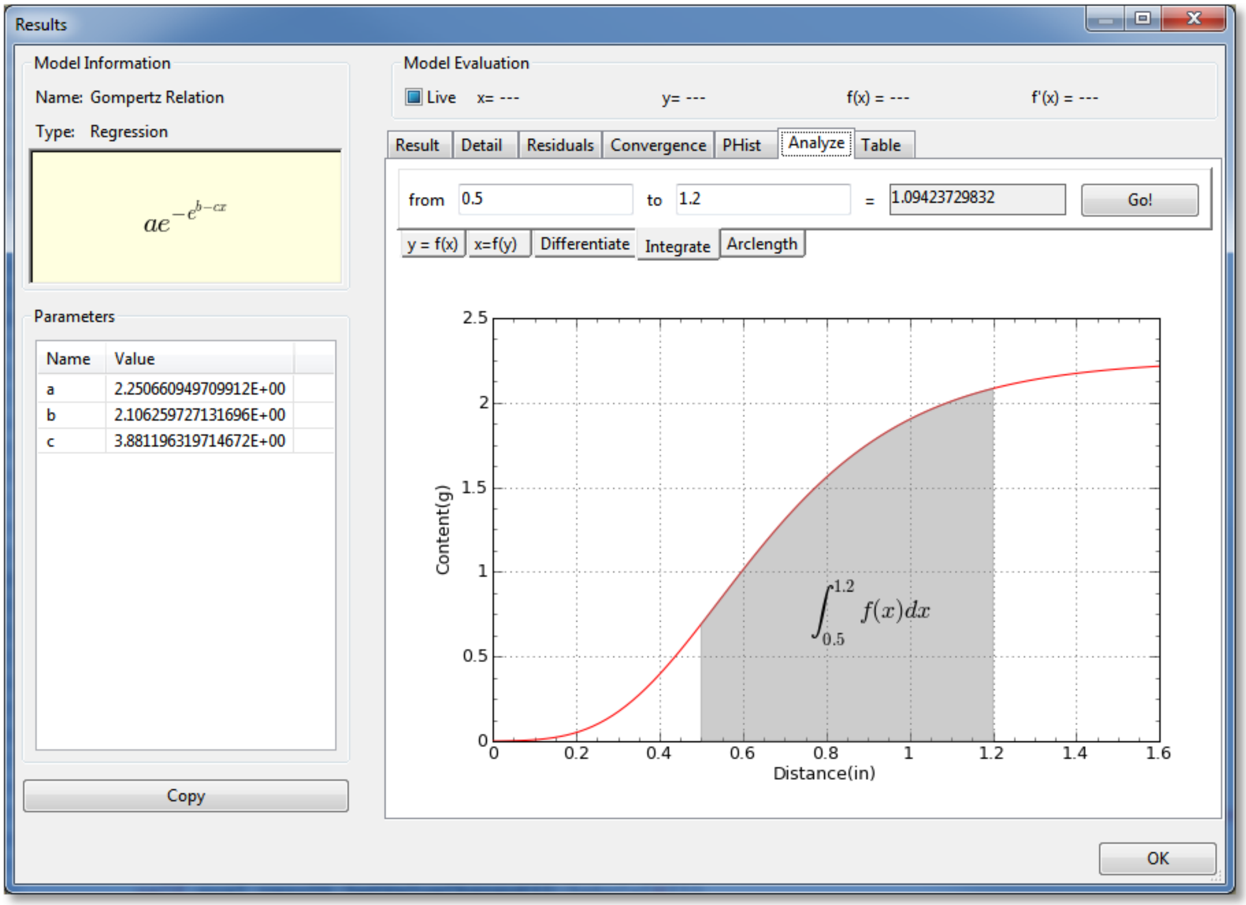
Parameters

Name	Value
a	2.250660946831745E+00
b	2.106259738151434E+00
c	3.881196341331432E+00

9.9.6 Analyze

The Analyze notebook page is a toolbox of mathematical operations that you can apply to the current result. Select the operation that you want to enact by selecting the appropriate sub-tab. You can calculate y as a function of x, x as a function of y, differentiate the result, integrate the result, or compute the arclength of the result. Simply select the appropriate sub-tab, fill in the numbers, and hit *Go*. Also note that pressing Enter in any of the fillable fields is the same as hitting the *Go* button; this allows you to quickly change the numbers being computed without having to reach for the mouse.

For each type of result, a helpful graphic will be drawn on the graph. For example, if integrating, the area under the curve between the two limits of integration will be shown.



Name: Gompertz Relation

Type: Regression

$ae^{-e^{b-cx}}$

x= --- y= --- f(x) = --- f'(x) = ---

from 0.5 to 1.2 = 1.09423729832

Name	Value
a	2.250660949709912E+00
b	2.106259727131696E+00
c	3.881196319714672E+00

9.9.7 Table

The Table notebook page allows you to create data tables corresponding to your result. Here, the user can specify the x (independent variable) data, and CurveExpert Professional will fill in the y (dependent variable) data. There are four methods to specify the x data: 1) by min, max, and increment for evenly spaced data, 2) by min, max and number of points for evenly spaced data, 3) use the x data from the currently loaded dataset, or 4) by reading a file. Set the first pulldown to "Range/incr", "Range/npts", "Data", or "File" as appropriate to select the method by which you would like to specify the data.

For specifying by **Range (with the spacing specified)**, select **Range/incr** and set the min, max, and increment of your table, click *Update*, and the table spreadsheet will update to show the calculated values. To write the table, press *Save...* at the bottom. Alternatively, if you want your table on the clipboard for subsequent pasting into another application, press *Copy*.

For specifying by **Range (with the number of points specified)**, select **Range/npts** and set the min, max, and number of points (rows) of your table, click *Update*, and the table spreadsheet will update to show the calculated values. To write the table, press *Save...* at the bottom. Alternatively, if you want your table on the clipboard for subsequent pasting into another application, press *Copy*.

For specifying by **Data**, there is no further work to do aside from setting the first pulldown to "Data". The independent variable (x) will be extracted from the current dataset and utilized.

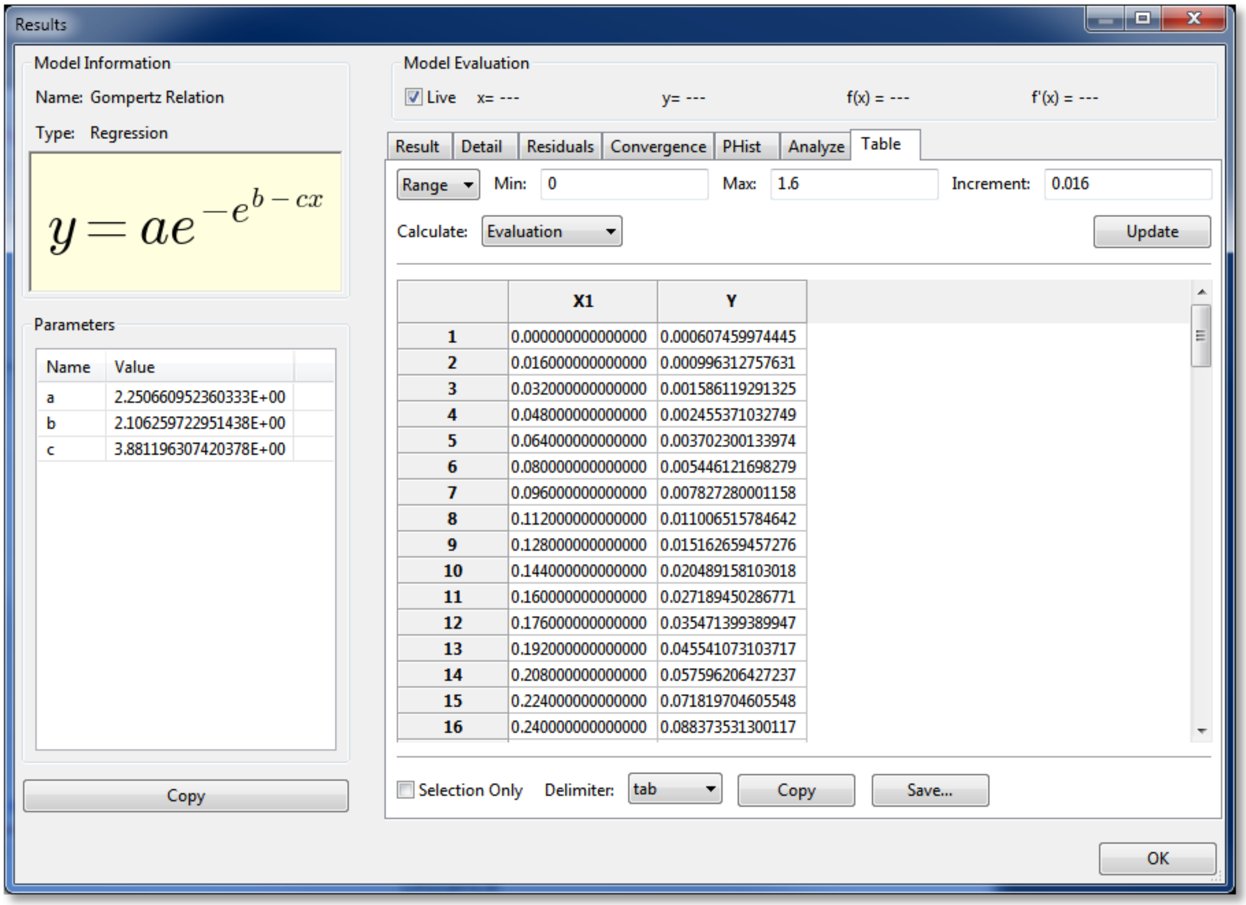
Likewise, for specifying by **File**, set the first pulldown to *File*, and click the *Browse* button to locate the file that you would like to read. Ideally, the file should be made up of a single row or a single column of data that will become the x data. If the file is made up of multiple columns, CurveExpert Professional assumes that the first column should be used as the x data. This capability allows the user to read datasets related to the original easily. As soon as a new filename is entered into the control, it will be read. To write the table, press *Save…* at the bottom. Alternatively, if you want your table on the clipboard for subsequent pasting into another application, press *Copy*.

Normally, an “Evaluation” style table is appropriate; an evaluation table simply evaluates the results at each of the points in the independent variable column. An “Accumulation” table computes the integral

$$\int_{x_0}^{x_i} f\left(x; \vec{a}\right) dx$$

for every point in the dataset, where x_0 is the min value specified for the table.

At the bottom of the table, before you copy or save, you can pick *Selection only* to only save the data that you have currently selected in the table spreadsheet. Otherwise, the entire table is saved or copied. Also, you can select the delimiter to use between values on the same row. The default delimiter can be set at *Edit->Preferences->General->Default delimiter*.



Name: Gompertz Relation

Type: Regression

$$y = ae^{-e^{b-cx}}$$

Parameters

Name	Value
a	2.250660952360333E+00
b	2.106259722951438E+00
c	3.881196307420378E+00

☑ Live x= --- y= --- f(x) = --- f'(x) = ---

Range Min: 0 Max: 1.6 Increment: 0.016

Calculate: Evaluation

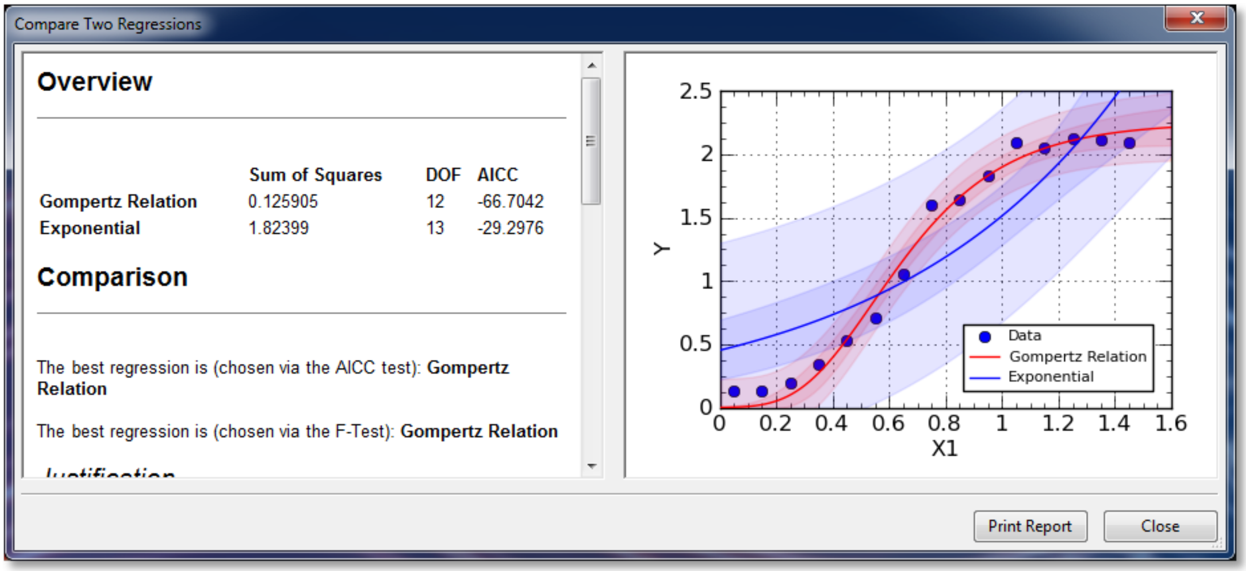
	X1	Y
1	0.000000000000000	0.000607459974445
2	0.016000000000000	0.000996312757631
3	0.032000000000000	0.001586119291325
4	0.048000000000000	0.002455371032749
5	0.064000000000000	0.003702300133974
6	0.080000000000000	0.005446121698279
7	0.096000000000000	0.007827280001158
8	0.112000000000000	0.011006515784642
9	0.128000000000000	0.015162659457276
10	0.144000000000000	0.020489158103018
11	0.160000000000000	0.027189450286771
12	0.176000000000000	0.035471399389947
13	0.192000000000000	0.045541073103717
14	0.208000000000000	0.057596206427237
15	0.224000000000000	0.071819704605548
16	0.240000000000000	0.088373531300117

☐ Selection Only Delimiter: tab

9.10 Comparing Two Regressions

CurveExpert Professional has the capability of comparing two regressions (curve fits) to provide some basis for the superiority of one versus the other. To compare two regressions, simple right click on a regression in the Results Pane, and pick *Compare To*.

The compare tool will appear, which presents a report in the left side, and the graph, with both regressions drawn, in the right side. By default, both regressions will have their confidence and prediction bands shaded in matching colors with their curve. If desired, these can be turned off by modifying each curve's properties via the normal properties dialog (right click and select Properties, or as a shortcut, press 2 or 3).



Overview

	Sum of Squares	DOF	AICC
Gompertz Relation	0.125905	12	-66.7042
Exponential	1.82399	13	-29.2976

Comparison

The best regression is (chosen via the AICC test): **Gompertz Relation**

The best regression is (chosen via the F-Test): **Gompertz Relation**

Justification

To compare two regressions, CurveExpert Professional uses both the Akaike Information Criterion (AIC) test, and the more commonly known F-Test. The better of the two fits, as judged by these two tests, is shown at the top of the report in the Comparision section. Following the best fit choices, a justification is given as to the reason for the choice, as well as supporting calculations.

One should note that the F-Test is only valid in cases where one model is a subset of the other; i.e., the less complex model can be obtained from the more complex one by judicious choice of the model parameters. CurveExpert Professional, at this time, has no means of determining whether or not this is true, so the interpretation of the F-test results is left up to the user. the AIC test is valid in any case, and is therefore the preferred method of comparision in CurveExpert Professional.

If you would like to print the report, simply click *Print Report* at the bottom of the comparison tool. Also, any part of the report can be copied out with the normal copying procedure (highlight and then pres Ctrl+C).

9.11 Confidence and Prediction Bands

For any regression (linear or nonlinear), the confidence bands and prediction bands can be shown. In order to show these bands, and/or change their appearance, right click the plot, select *Properties->Series*, and then select the appropriate series (alternatively and much more quickly, if the regression is the 3rd entry in the legend, press 3). If the series is a regression, options to control the appearance of the confidence band and prediction band will be present. See *Series Page* for more information on changing the appearance of the bands.

The confidence band is the area that has a certain likelihood (typically 95%, but you can adjust the level to your liking) of containing the true curve that fits the data.

The prediction band is the area that has a certain likelihood (typically 95%, but you can adjust the level to your liking) of containing any future data points. The prediction band is always wider than the confidence band.

CHAPTER

TEN

GRAPHING

10.1 Introduction

CurveExpert Professional contains a fully-featured graphing engine. The graphs are publication quality and are fully antialiased for the best possible appearance. Graphs are also themeable in order to save the user time in the production of a large number of similar plots.

Both 2D and 3D graphing is supported. The 3D graphing is activated when a dataset is read in or created that has two independent variables. When a dataset with more than two independent variables is read in, graphing is disabled.

10.2 Graph Types

There are three types of graphs supported by CurveExpert Professional. These three types are XY, Polar, and 3D.

10.2.1 XY Plots

XY plots, by definition, have two axes in Cartesian space. XY plots are appropriate for showing relationships between cause and effect, or equivalently, the relationship between one independent variable and one dependent variable.

Note that XY plots can also display contours (both filled and unfilled), and is the appropriate plot type to use for contour plots. Contour plots show the relationship between two independent variables and one dependent variable.

10.2.2 Polar Plots

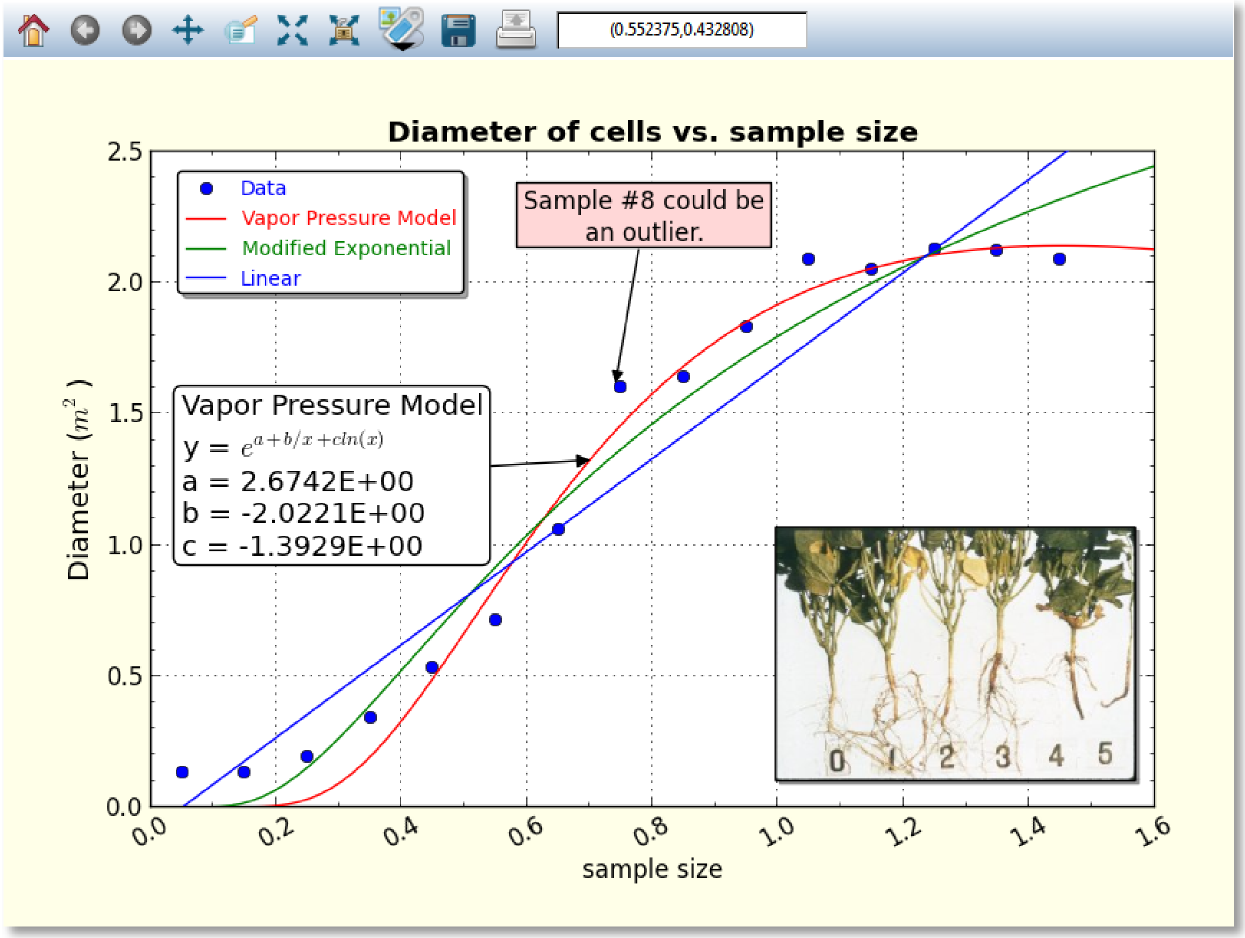
Polar plots have one radial axis and one angular axis. Polar plots always treat the angular axis in radians, although the numbers shown on the angular axis can easily be formatted as degrees. In other words, a dataset that ranges between $[0, 2\pi]$ will completely wrap around a polar plot.

10.2.3 3D Plots

3D plots have three axes; two for independent variables X1 and X2, and the third in the vertical direction for the dependent variable (Y).

10.3 Basics

The graphing area is divided into two primary portions; the toolbar, which is situated along the top of a graphing area, and the graph canvas, where the plot is drawn. The toolbar allows interactivity with the graph, such as panning, zooming, and autoscaling. Right clicking on the plot canvas shows the graphing menu, where you can customize items on the plot like the axis labels, title, and the look of the graph. Also from this menu, you can print, copy to clipboard, or save your graph in one of many picture formats. An example of a graph is shown below:



Graphs occur in quite a few places throughout CurveExpert Professional, and you can always interact with them in a very similar way. Just right-click on plots wherever you see them, in order to find out what sorts of interaction are possible.

Right clicking on elements in the graphs (such as legends, images, annotations, and colorbars) show appropriate menus for manipulations of those objects. Also, remember that anywhere that you can change the text on a graph (such as the title or axis labels), you can use mathtext for high quality rendering of mathematical expressions and symbols. For example, in the above graph, mathtext was used to render the `m^2` part of the y axis label. See mathtext for details.

The hotkeys (keyboard shortcuts) for use with graphs are documented in *Keyboard Shortcuts for Graphs*.

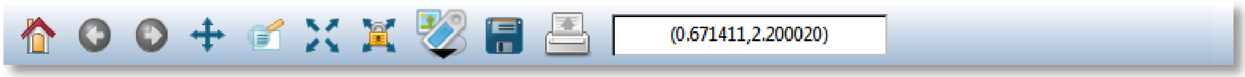
10.4 Interacting with a graph

Interacting with a graph is quite straightforward; the mouse cursor always gives a hint of what can be done to various elements on a graph; a cursor that consists of four arrows indicates that the pointed-to-element is moveable, activateable, and possibly resizeable. A hand cursor means that the pointed-to-element can be activated. In this context, “activating” means to show a dialog box to modify the properties of the element in question.

Further, the frame of the graph can be moved and resized by pressing shift (or ctrl) and clicking anywhere inside of the frame. Equivalently, the ‘r’ key can be pressed while the mouse pointer is inside of the frame. This displays a standard set of resizing handles that you can use to move the edges of the graph frame into the desired locations.

10.4.1 Graph Toolbar

The first style of interaction with a graph is to use the toolbar. A sample of a toolbar is shown below:



(0.671411,2.200020)

All buttons have tooltip help, so if you can’t remember what a button does, either watch the status bar at the bottom of the main window as you point to the buttons, or hover your mouse over a button to see its function.

Home, Back, Forward

As the graph view changes interactively, each view is saved so that you can return to a previous one at any given time. If you change the view via the zoom/pan/autoscale controls, you can return to a previous view by clicking the *Back* button, and once you have done this, you can move forward to the next view by clicking the *Forward* button. Pressing the *Home* button will return you to the initial view of your graph.

Pan

The pan button allows you to move the graph extents with your mouse. Press the pan button to begin a pan operation, press and drag in the plot to move it around, and then press the pan button again to end the pan operation. Using the right mouse button to drag during the pan manipulates each axis independently in a pseudo-zooming operation.

Polar Panning

For polar plots, the pan button allows zooming in on the radial axis. With the right button, the location of the radial axis labels can be moved by right clicking and dragging them into place.

3D Panning

For 3D plots, the pan button allows more transformations than just a simple move, depending on the mouse button that is depressed.

The left mouse button moves the entire XYZ axes box around on the background canvas, and is analgous to panning a virtual camera pointed where the view direction is unchanging.

The right mouse button stretches axes in the direction in which you are dragging the mouse, and is also affected by the location at which the drag starts (called the grab point). For example, if you right click and hold at the center of the axes box (grab point at the center), and drag right, the X1 and X2 axes expand in size. If you grab in the center of the axes box and drag upward, the Y axes expands. Drag the opposite direction for a contraction. The grab point, in fact, is a fixed point. Grabbing at the right of a plot, for example, and dragging right, will contract the lateral directon of the axes, but will keep the right side reasonably fixed in the canvas.

A useful way of dragging is to grab in the center, and drag diagonally. This expands/contracts all three axes at the same rate, and has the effect of interactively moving the camera toward or away from the axes box.

In summary, the right mouse button manipulates the viewing frustum in regard to the axes box.

Zoom

The zoom button allows you to zoom in or out on a graph with your mouse. Press the zoom button to begin a zoom operation, and draw a box from top left to bottom right in order to zoom in, by clicking and dragging your left mouse button. To zoom out, drag a box with the right mouse button instead of the left. Note that the zoom box is not available for polar plots.

Autoscale

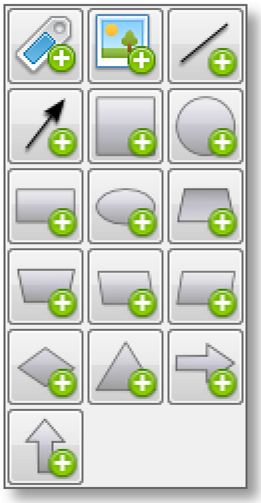
The autoscale button chooses new limits and new tick intervals for your graph that fit the current dataset. See also *Autoscaling*.

Autoscale lock

CurveExpert Professional will automatically autoscale your plot as data or results are added, changed, or removed. This is done according to *Graphing Preferences*. If you click the autoscale lock, it will prevent any of these automatic autoscales from affecting this particular plot. Note that as soon as CurveExpert Professional detects that you have manually set the axis limits or tick location properties (see *Axis pages*), the autoscale lock is set on your behalf. To release the lock, obviously, just click on the autoscale lock button again.

Add Artistic Element (Annotations, Images, Arrows)

The "add artistic element" button allows opens a tool palette in order for you to add a new drawing element to your graph. The three main classes of drawing elements that can be added are *Images*, *Annotations*, and *Arrows*. Note that "shapes" are just annotations without any text displayed inside of them; text may be added inside if you wish. Pressing this button shows the following tool palette:



The annotation button allows you to add an annotation (text box with an optional arrow) at any time. Pressing this button immediately adds an annotation, which you can then drag and drop to the desired location, as well as edit by right clicking the annotation and selecting *Annotation Properties*. For more information, see *Annotations*.

The image button allows you to add images at any time. Pressing this button allows you to browse for an image file (PNG, JPG, BMP, TIFF, GIF, etc.), which will be added, at its native resolution, to the graph. You can then interactively drag and resize the image much like a similar operation in regular photo/image packages. The image can also be edited by right clicking the image and selecting *Image Properties*. For more information, see *Images*.

There are two arrow-style buttons; one to add an arrow, and one to add a line. The only difference between the two is that an arrow has a single arrowhead at the destination by default, while the line (obviously) does not by default. The arrowheads, as well as other properties, may be changed by right clicking the arrow and selecting *Arrow Properties*. For more information, see *Arrows*.

The rest of the buttons in the panel allow you to add a shape to the graph, of the type depicted on the button. The list of available shapes (via the buttons) are: square, circle, rectangle, ellipse, trapezoid, inverted trapezoid, left parallelogram, right parallelogram, diamond, triangle, horizontal arrow patch, and vertical arrow patch. The properties of any of these shapes, including the type of shape, may be modified by right clicking the shape in the graph and selecting *Annotation Properties*. For more information, see *Annotations*.

Note: All shapes are considered *Annotations*, and can have text inside of them as well as an optional arrow.

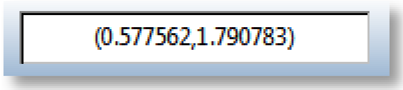
Save

The save button allows you to save your graph in a variety of picture formats. Pressing this button shows a file chooser, so that you can select the file type and where to save the graph. See *Saving graphs* for more information, including supported image formats.

Print

The print button allows you to quickly output a hardcopy of the currently visible graph.

Locator



(0.577562,1.790783)

The locator always reports the location of your mouse, in data coordinates. Note that the locator is only activated for 2D plots only.

10.4.2 Dragging Elements on Graphs

Annotations, legends, images, graph titles, and colorbars are all draggable; meaning that they can be moved directly via interaction with the mouse. If any of these elements are not where you desire them be, simply click and drag it to the desired position.

10.4.3 Picking

If you point to a series on your graph, it will become highlighted. If you then double-click the mouse, a dialog will appear to allow you to change the properties of that series (color, marker style, etc.).

Further, if you right click on a series, you can also select *Details* in order to show the details for the underlying dataset or result associated with that particular curve. When right-clicking on a curve, there is also the opportunity to automatically add an annotation with information about that particular curve's result; choose *Autoannotate*. This generates an annotation with the name of the result, equation, and coefficients (if applicable). The content of this annotation can then be edited just like any other annotation by right clicking it and selecting *Annotation Properties*.

Copy underlying data

If you right click on a curve, you can also access the data that the particular curve is built from. Pick *Copy underlying data*, and a table of numbers will be written to the clipboard. Now, you may paste these numbers into another application for further analysis and/or exploration.

In most cases, the table has the x data in the first column, and the y data in the second. However, if your dataset has an associated set of error bars, there will be a third and fourth column which list the values of the lower error bar, and the upper error bar, respectively.

If your curve has confidence or prediction bounds, the lo/hi values of the confidence or prediction bounds will be placed in additional columns (columns 3 and 4) as appropriate as well. In the case that both the confidence and prediction bounds are present, the lo/hi for the confidence bounds is listed in columns 3 and 4, and the lo/hi for the prediction bound is listed in columns 5 and 6.

10.4.4 Autoscaling

Every series on a graph has its own default minimum and maximum value along each axis. Autoscaling a graph means that each (visible) series will be tested, and the extents of each axis will be set to values that encompass all of the series.

If the series is a dataset, the minimum and maximum along each axis is determined from the dataset's minimum and maximum along the corresponding axes (naturally). If the series is a continuous entity, such as a model or function, a specified "default domain" must be set in each direction for the autoscaler to use. This is set when the function is defined; see *Creating Custom Models and Functions* for details. Note that this default domain is for autoscaling purposes only, and the graph extent along any axis can be set by the user to the desired values in the graph properties dialog, as described in *Axis pages*.

Exactly when the plot automatically autoscales is determined by the current preference settings in *Graphing Preferences*. If you click the autoscale lock, it will prevent any of these automatic autoscales from affecting this particular plot. Note that as soon as CurveExpert Professional detects that you have manually set the axis limits or tick location properties, the autoscale lock is set on your behalf. To release the lock, obviously, just click on the autoscale lock button again.

10.4.5 Adding images and annotations

Adding images and/or annotations can be done straightforwardly by clicking the appropriate "Add" button in the graph toolbar palette (see *Graph Toolbar*). However, there are other ways of adding annotations and images.

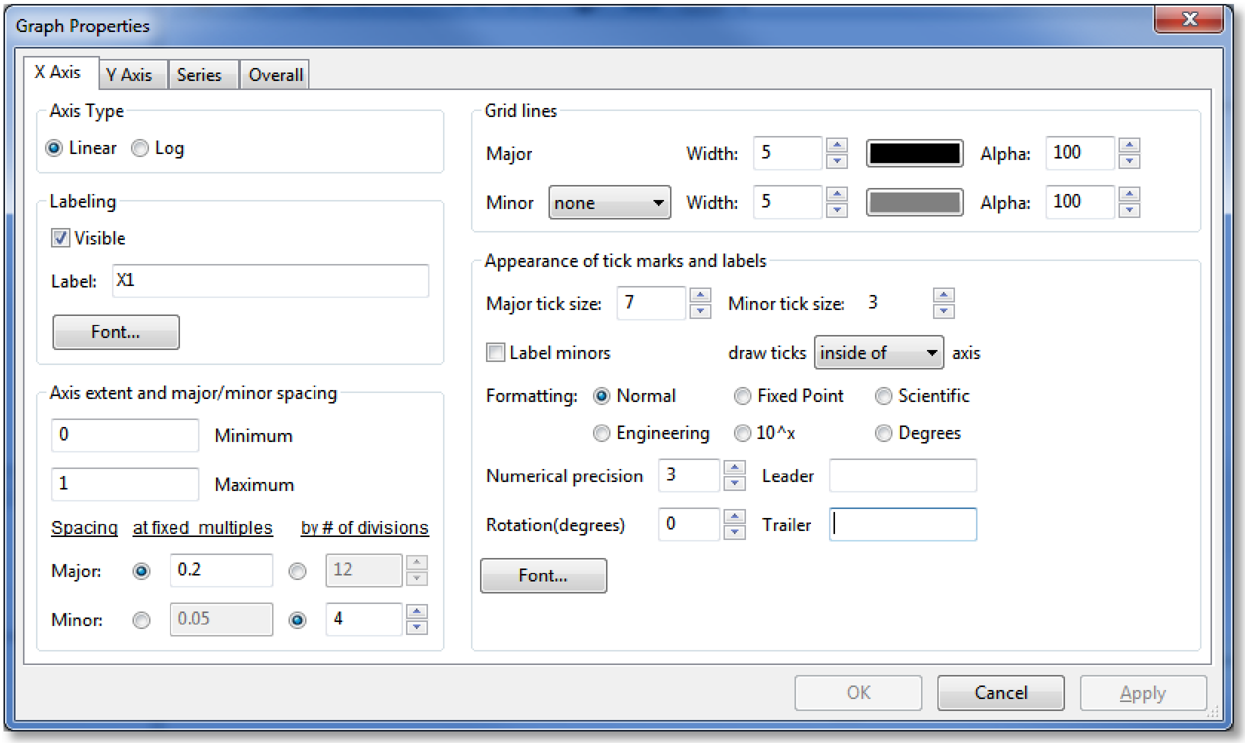
If text is on the clipboard, you can press Ctrl+V (Cmd+V on a Mac) on a graph, or right-click and select *Paste*. The text on the clipboard automatically becomes an annotation. You can also drag and drop text from another source for the same effect.

If an image is on the clipboard, you can press Ctrl+V (Cmd+V on a Mac) on a graph, or right-click and select *Paste*. The image on the clipboard will then be pasted to the graph canvas in its native resolution. You can also drag and drop an image file to the graph canvas for the same effect.

10.5 Changing Graph Properties

The graph properties window is divided by functionality into pages.

10.5.1 Axis pages



Axis Type
● Linear ○ Log

Labeling
☑ Visible
Label: X1

Axis extent and major/minor spacing
0 Minimum
1 Maximum
Spacing at fixed multiples by # of divisions
Major: ● 0.2 ○ 12
Minor: ○ 0.05 ● 4

Grid lines
Major Width: 5 Alpha: 100
Minor none Width: 5 Alpha: 100

Appearance of tick marks and labels
Major tick size: 7 Minor tick size: 3
☐ Label minors draw ticks inside of axis
Formatting: ● Normal ○ Fixed Point ○ Scientific
○ Engineering ○ 10^x ○ Degrees
Numerical precision 3 Leader
Rotation(degrees) 0 Trailer

The axis pages (there can be two or three depending on whether you are working with a 2D or 3D graph) allow you to set properties such as the axis limits, labeling, and grid lines.

The **Axis Type** section allows you to set your axes to a normal (linear) scale, or a logarithmic scale. Note that logarithmic scales are not available on 3D graphs.

The **Labeling** section can be used to either hide the axis label, or specify the label itself. Also, the font and font color of the axis label can be selected by pressing the *Font* button in this section.

The **Axis extent and major/minor spacing** section allows you to specify the minimum and maximum value of the axis (determining its limits). To have your axis run in reverse, set the maximum of the axis less than the minimum.

The major and minor spacing can each be set in one of two ways: by a fixed amount of spacing between each tick, or by the number of divisions an interval is divided into. To select the kind of spacing, click the radio buttons under the *at fixed multiples* or *by # of divisions* column as desired. If *at fixed multiples* is chosen, each tick mark (either major or minor) is set each time the multiple appears on the axis. For example, if the major ticks are set at fixed multiples of 0.2, there will be a major tick at -0.2, 0, 0.2, 0.4, etc. If *by # of divisions* is chosen, the ticks are placed such that the appropriate interval (the entire axis in the case of minor ticks, and the interval between two majors in the case of major ticks) is divided into the specified number of parts. For example, if the minor ticks are set to a # of divisions of 4, there should be four intervals between each major tick (which means that three minors will be visible).

The **Grid Lines** section allows you to specify the line style, line width, color, and transparency (alpha) of both the major and minor grid lines.

The **Appearance of tick marks and labels** section is where the size and style of tick marks, the font and color of the tick mark labels, and the numerical format of the tick mark labels can be specified. The **Major/Minor** tick size determines how large the markers are that are placed at each tick, in points. Also, the **Draw ticks inside of axis** setting determines if these tick marks are placed completely inside, outside, or across the axis. **Label minors** determines whether or not text labels are placed at each minor demarcation (most useful for log scaled plots).

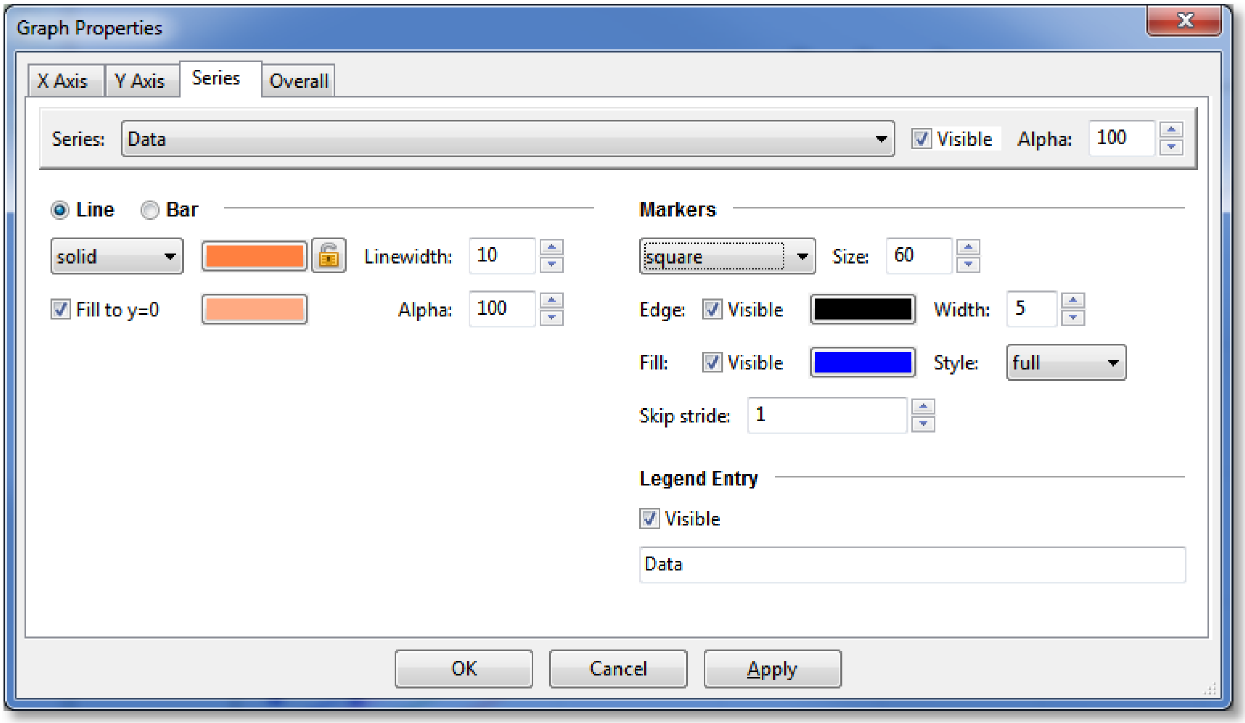
The format of the tick mark labels is determined by the combination of the **Formatting** setting and the **Numerical precision** setting. **Formatting** can be set to normal for a generic style of numerical formatting, fixed point, scientific, engineering, 10^x, or degrees. In all cases except for normal formatting and 10^x, the **Numerical precision** field determines exactly how many places are shown after the decimal in the label.

- Normal: a generic style of formatting that lets the software determine best appearance. e.g. 10245.1234
- Fixed point: exactly like normal formatting, except that the numerical precision is honored. e.g. 10245.1 (coupled with a numerical precision of 1)
- Scientific: a mantissa followed by an exponent. e.g. 1.02e4 (coupled with a precision of 2)
- Engineering: a mantissa followed by an exponent that is a power of 3. e.g. 10.25e3 (coupled with a precision of 2)
- 10^x: most useful for log scale plots, where the user desires to display the number on the axis as 10 raised to the appropriate exponent. e.g. 10^2
- Degrees: most useful for polar plots, where the user desires to display a number *in radians* as a number *in degrees*. e.g. 60° (coupled with a precision of 0)

Any axis tick label can be preceded or followed by customized text (including mathtext) as you specify. For text to appear before each axis label, place the text in the **Leader** field; for text to appear after each axis label, place the text in the **Trailer** field. By default, no text appears before or after the axis tick label. Examples of common usage of this feature are to place a dollar sign in front of the numbers to indicate currency, or units after the label.

Finally, the set of tick labels can be rotated by a certain number of degrees, which is also configurable in this section via the **Rotation** setting.

10.5.2 Series Page



Series: Data ☑ Visible Alpha: 100

● Line ○ Bar

solid Linewidth: 10

☑ Fill to y=0 Alpha: 100

Markers

square Size: 60

Edge: ☑ Visible Width: 5

Fill: ☑ Visible Style: full

Skip stride: 1

Legend Entry

☑ Visible

Data

The Series page is where the settings for individual series can be changed. For quick help with any of the settings in this page, hover your mouse over any control.

The first item to choose is the **Series**; this determines which curve the settings in the dialog apply to. As the series is changed, the controls in the rest of the dialog will update appropriately. Next to the series chooser, you can set the *Visible* flag, and the overall transparency of the series via the *Alpha* control (100 being opaque, and 0 being fully transparent).

The next section can be either **Line** or **Bar**, depending on the selection of the user. A **Line** section indicates that the series should be drawn as a line with markers at each data point (both line and markers are optional, though, based on your specific settings for these). A **Bar** section indicates that the series should be drawn as a sequence of vertical bars extending upwards from y=0. If **Bar** is selected, there is no concept of markers or confidence/prediction bands (see below), so these sections are hidden.

The **Line** section obviously sets properties for the line; the linestyle can be chosen, as well as the color and width. The *Resolution* parameter determines how many pieces make up the curve, from the current minimum extent to maximum extent of the graph. Higher resolutions look more attractive, but take longer to draw. If your curve changes sharply over a short distance, increase the resolution in order to avoid a segmented look to the curve.

The lock button adjacent to the line color selector determines whether or not the marker edge color and fill color (see below) remain the same as the line color. If left unlocked, the marker edge color and fill color may be adjusted independently from line color. If locked, the marker edge color and fill color are always governed by the line color.

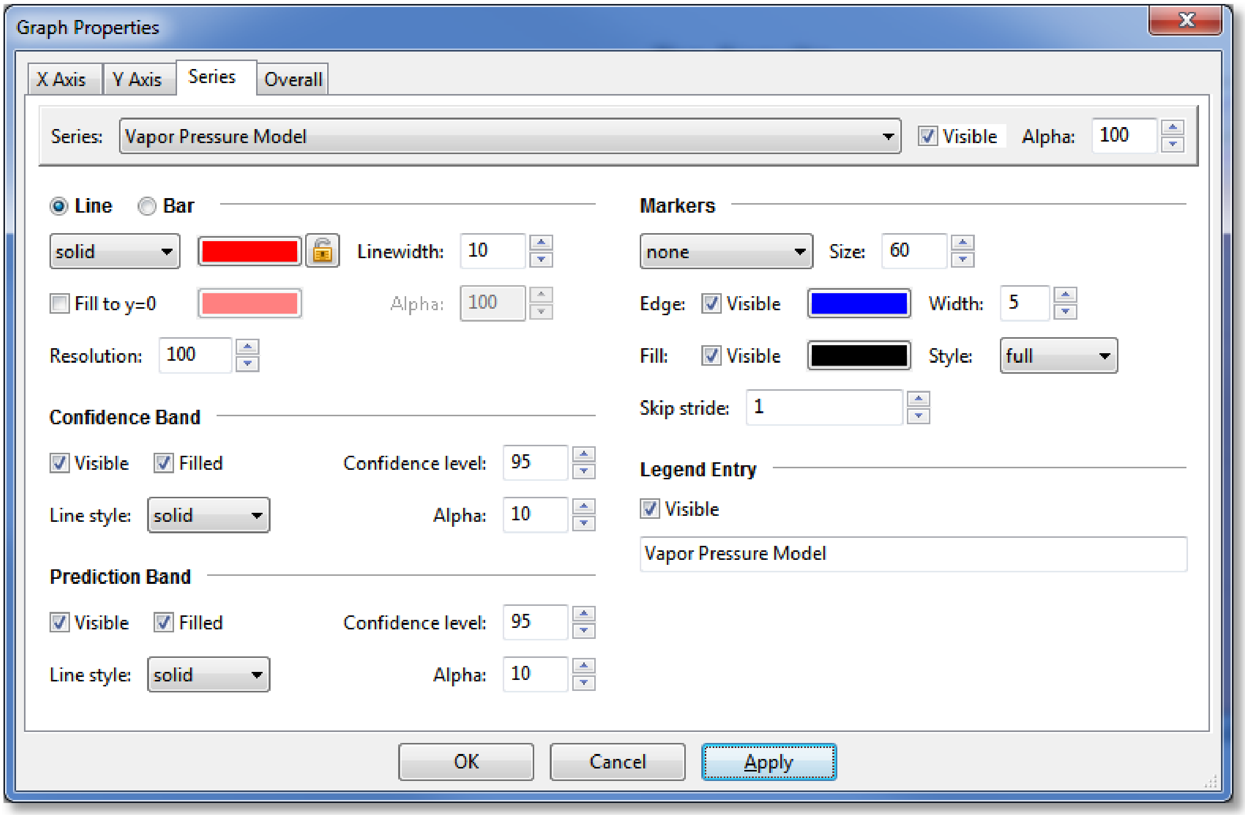
The **Bar** section sets properties for the bar sequence; the color of the bar border, linestyle of the bar borders, and the line width of the bar borders can be chosen. Further, the bar can be filled with a particular color that does not necessarily match the bar borders described above. The resolution determines the number of divisions to create over the current x range of the graph. Finally, the bar width can be either prescribed specifically as a horizontal extent, or set to *Auto*, which sets the width of each bar such that each bar touches the bar before and after.

The **Marker** section obviously sets properties for markers (this section is not visible while **Bar** is selected). The marker type can be chosen from a set of 22 different markers, and the marker size can be set with the Size control.

Most markers (such as circles, squares, triangles, etc.) consist of both edge and fill; the edge is the outline of the shape, and the fill is the interior of the shape. Other markers (such as X, +, etc.) consist of only an edge, and the fill property has no effect. The edge and fill properties can be set independently in order to customize the markers to the maximum degree possible. Also, the markers can be half-filled or fully-filled according to the *Style* setting. If the marker distribution is too dense, the *Skip Stride* parameter can be increased in order to decrease the marker density on any particular series. Note that for a dataset, there is one marker per data point; for a continuous result, there is one marker per resolution increment set in the **Line** section. Also note that the marker edge and fill colors may only be adjusted if the "lock" button in the line section (see above) is unlocked.

The **Legend Entry** section gives control over the series's appearance in the legend. Here, you can exclude the series from appearance in the legend, and also change the label that appears in the legend for the active series.

If the selected series at the top of the dialog is a regression, there will also be controls to manipulate the appearance of the confidence and/or prediction bands (see *Confidence and Prediction Bands*). By default, confidence and prediction bands are not drawn on normal graphs, but are in the result details window (see *Querying Result Details*), and in the regression comparison window (see *Comparing Two Regressions*). A sample of the available controls for the confidence and prediction bands is shown below:



Series: Vapor Pressure Model ☑ Visible Alpha: 100

● Line ○ Bar

solid 🟥 Linewidth: 10

☐ Fill to y=0 Alpha: 100

Resolution: 100

Confidence Band

☑ Visible ☑ Filled Confidence level: 95

Line style: solid Alpha: 10

Prediction Band

☑ Visible ☑ Filled Confidence level: 95

Line style: solid Alpha: 10

Markers

none Size: 60

Edge: ☑ Visible 🟦 Width: 5

Fill: ☑ Visible ⬛ Style: full

Skip stride: 1

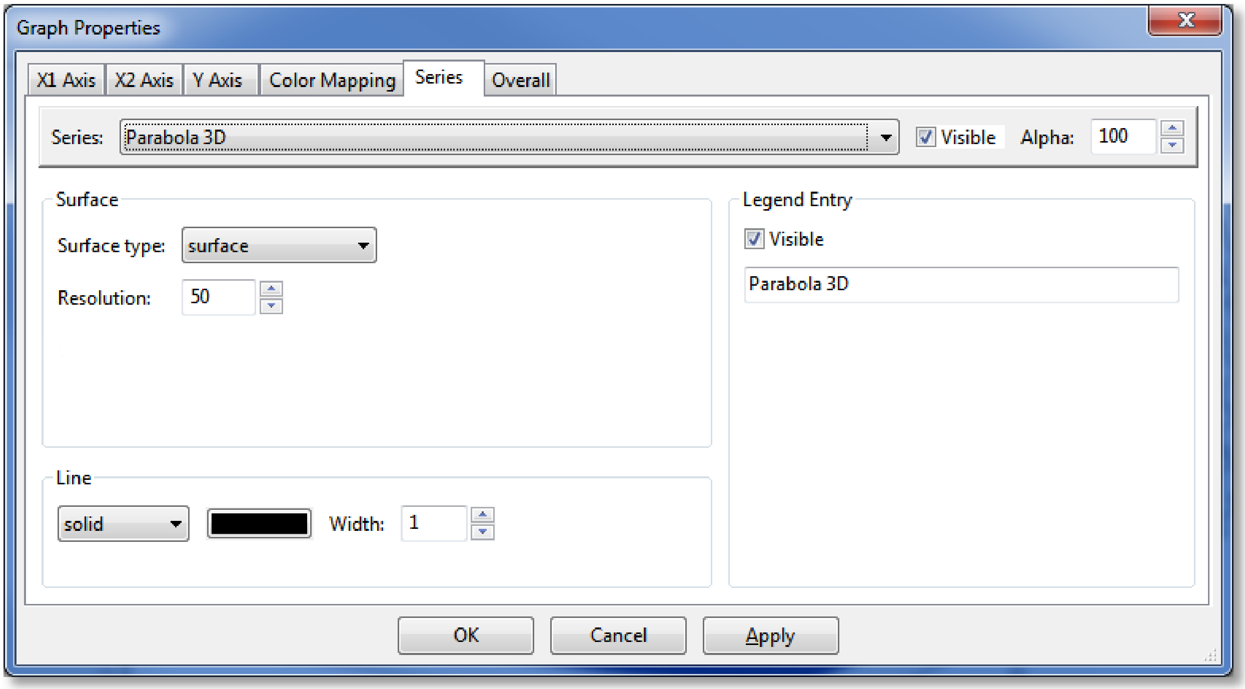
Legend Entry

☑ Visible

Vapor Pressure Model

In both the **Confidence band** and **Prediction band** sections (these sections are not visible unless the series is a regression, and further, are not visible when **Bar** is selected), the visibility of the respective bands can be set via the *Visible* checkbox; if visibility is off, none of the other settings are significant. the *Filled* checkbox determines if the bands are drawn as a filled region or a pair of lines. The *Confidence level* allows you to set the desired probability level for the confidence bands or prediction bands; the default value for this probability is set via the *Application Preferences*. If the bands are drawn as a pair of lines, the *Linestyle* setting determines the style for those lines; the line color always matches the line color of the corresponding regression curve. If the bands are drawn as a filled region, the **Alpha** parameter determines the transparency of that region.

If the selected series at the top of the dialog is a 3D series, the **Surface** section becomes active, the the **Marker** section is hidden:



Series: Parabola 3D ☑ Visible Alpha: 100

Surface type: surface

Resolution: 50

solid Width: 1

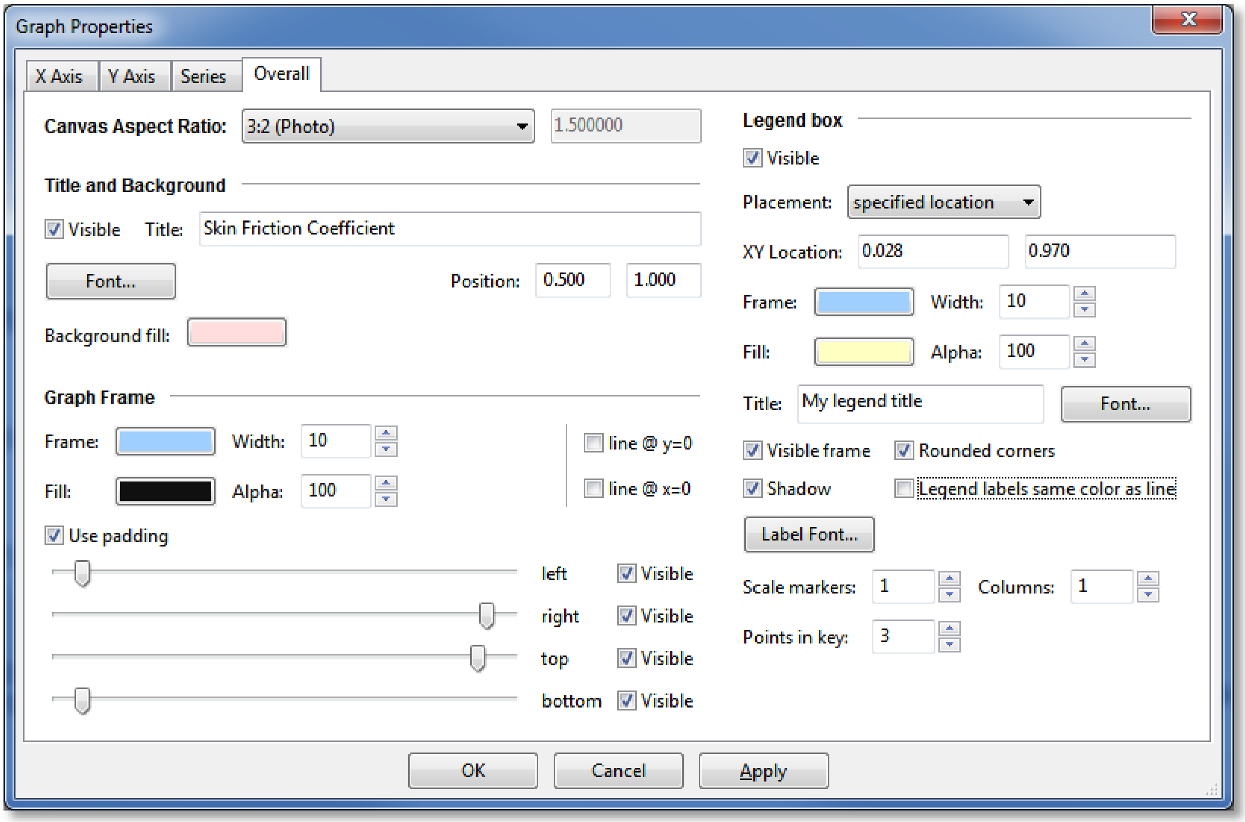
☑ Visible

Parabola 3D

The **Surface** section allows you to set parameters for the 3D series. Here, the surface type can be selected as a scatter, wireframe, triangulated wireframe, surface, triangulated surface, 3D contour, or 3D filled contour. Like the corresponding parameter in the **Line** section, the *Resolution* parameter determines the number of intervals used in the X1 and X2 directions in order to form a mesh on which to evaluate the 3D function for visualization. Higher resolutions in this case will increase drawing time severely, as the amount of work to be done is proportional to the square of the resolution. Note that the color mapping applied to your 3D surface is set via the *Color mapping* section; see *3D Graphing* for further information.

The **Line** section is still active for a 3D surface, because it affects the lines drawn around each of the patches that make up the 3D surface. In particular, for a wireframe surface, the colormap (which affects fill) is irrelevant, while the line style and width set in the **Line** section control the appearance of the lines that make up the wireframe.

10.5.3 Overall Page



Canvas Aspect Ratio: 3:2 (Photo) 1.500000

Title and Background

☑ Visible Title: Skin Friction Coefficient

Position: 0.500 1.000

Background fill: 🟥

Graph Frame

Frame: 🟦 Width: 10

Fill: ⬛ Alpha: 100

□ line @ y=0

□ line @ x=0

☑ Use padding

left ☑ Visible

right ☑ Visible

top ☑ Visible

bottom ☑ Visible

Legend box

☑ Visible

Placement: specified location

XY Location: 0.028 0.970

Frame: 🟦 Width: 10

Fill: 🟨 Alpha: 100

Title: My legend title

☑ Visible frame ☑ Rounded corners

☑ Shadow □ Legend labels same color as line

Scale markers: 1 Columns: 1

Points in key: 3

The overall settings page is where the global settings for a plot are changed, such as the title, background, and legend box appearance. For quick help with any of the settings in this page, hover your mouse over any control.

The **Canvas Aspect Ratio** setting determines the aspect ratio of your plot. If set to Dynamic, which is the default, the plot will always fill the window that it resides in. However, there are many cases where the aspect ratio of the plot should be fixed; there are several predefined aspect ratios in the **Canvas Aspect Ratio** pulldown to select from, that fit common aspect ratios in use in various modern applications. If one of these fixed aspect ratios is set, your plot will automatically size itself in its parent window (leaving part of the parent window unpainted) appropriately in order to retain the desired aspect. If *Custom* is selected as the aspect ratio, you may type in the desired aspect (width/height) as a floating point number next to the pulldown.

The **Title and Background** section allows the title, font and color to be set. The position of the title is governed by the *Position* setting; this position is in the axes coordinate system, which is a system that is (0,0) at the bottom left corner of the axes frame, and (1,1) at the top right corner. The default position is (0.5,1.02), which places the title horizontally centered and above the top of the axes frame. Note that the title may be dragged and dropped interactively on the plot itself. The title can be turned completely off by unchecking the *Visible* box. Also, the background color (the area around the graph frame) can be set with the *Background fill* parameter.

In the **Graph Frame** section, the graph frame's line color and width, along with its fill and transparency, can also be set. The position of the graph frame's sides can be set by sliding the appropriate sliders near the bottom of this section; each side of the graph frame can also be turned on and off individually. The *Use padding* setting determines whether or not extra padding is added around the graph in order to make sure that all text (i.e., tick labels, axis labels, and titles) is visible. Turning the padding off provides finer control over the location of the axes frame (because there is not an automatic sizer involved; all sizing is explicit), but also will allow you to set the frame such that certain textual elements are not visible. Note that if you move/resize a frame interactivity in the plotting window (by pressing 'r' or shift-clicking the axes area), the padding switch is automatically turned off. A solid line, of the same thickness and color of the frame, can be drawn specifically at x=0 or y=0 by checking the appropriate box.

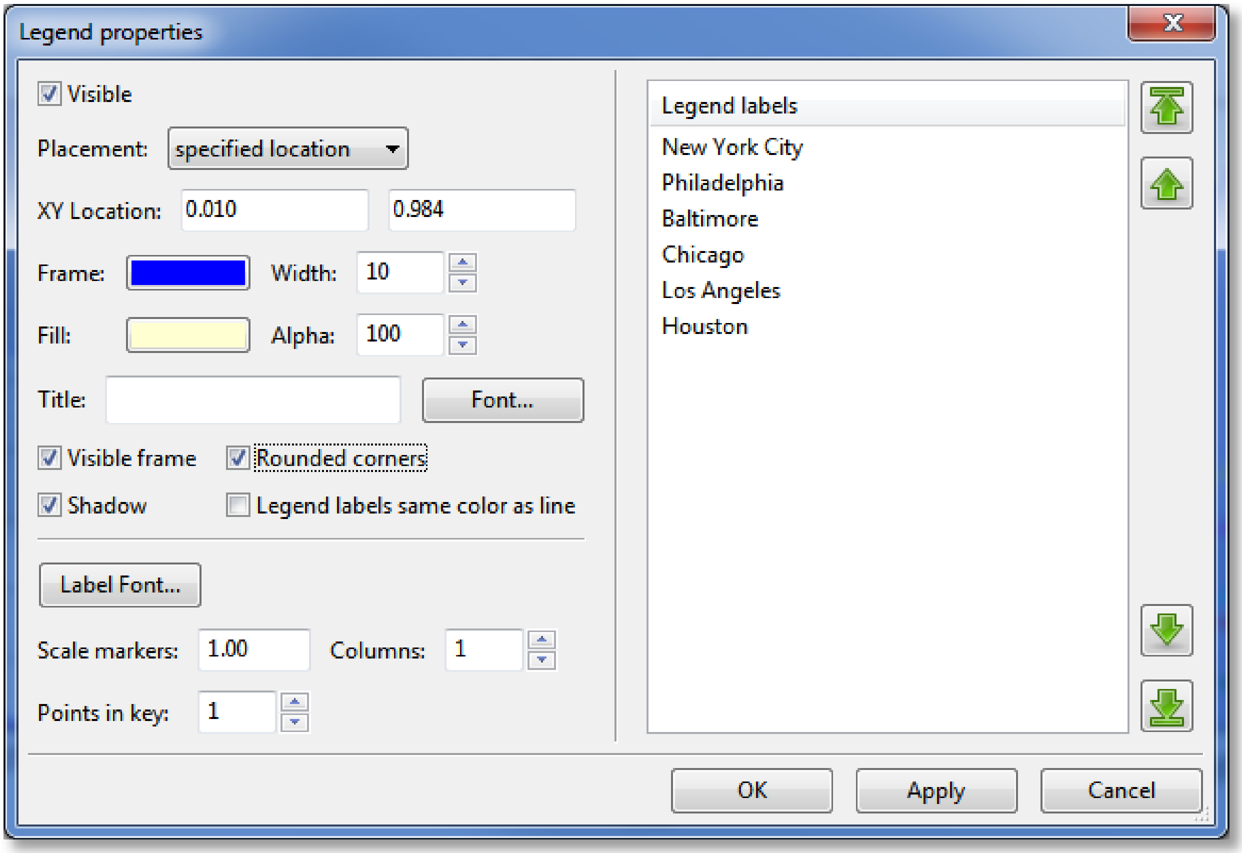
The **Legend Box** section is located in the right side of the *Overall* page. See *Legend Properties* for details.

10.6 Legend

The legend shows the mapping between a drawn entity on the plot and the dataset/function/model that it represents. Right click on a legend for the actions that be performed on the legend itself; one can change the look of the legend or reorder items in the legend (*Legend Properties*), or raise/lower legend itself in order for it to be drawn appropriately with other items on the graph.

10.6.1 Legend Properties

The legend properties can be accessed via the normal graph properties dialog (right click on the graph, choose *Properties*, and then go to the **Overall** tab), or can be accessed by right clicking on a legend itself. The legend properties dialog appears as below:



Legend properties

☑ Visible

Placement: specified location

XY Location: 0.010 0.984

Frame: 🟦 Width: 10

Fill: ▭ Alpha: 100

Title:

☑ Visible frame ☑ Rounded corners

☑ Shadow □ Legend labels same color as line

Scale markers: 1.00 Columns: 1

Points in key: 1

Legend labels
New York City
Philadelphia
Baltimore
Chicago
Los Angeles
Houston

The entire legend box can be made invisible, and its placement in the graph can be chosen from the Placement chooser. If the placement is set to “specified location”, the XY location of the legend’s upper left corner can be set. The XY location is in graph coordinates, which run from (0,0) at the bottom left of the graph frame to (1,1) at the top right of the frame. The XY location can certainly be set to numbers greater than 1 or less than 0, in order to move the legend into the desired position.

Still in the **Legend Box** section, the color and linewidth of the legend frame can be selected, and the fill/transparency can also be selected. A legend title may be specified (along with a font and color), that is displayed above the legend entries and inside the legend box. A set of checkboxes allow the legend frame to be visible/invisible, have a drop shadow, have rounded corners, or have the labels in the legend assume the same colors as the line that they refer to.

At the bottom of the **Legend Box** section, the font and color of the legend labels can be selected. If markers need to be made smaller or larger in the legend, a *Scale markers* setting takes care of this. To lay your legend out in a multicolumn format, increase the number of columns in the *Columns* setting. To include more than one marker in a legend key, increase the *Points in key* setting.

10.6.2 Legend label editing

Legend labels may be changed in the **Legend labels** section, by clicking an already-selected label. The label will become editable, and you can change the text to whatever is desired. Remember that mathtext can be used in a legend label as well. Also, the label can be made blank so that the series does not appear in the legend. Editing a label in this manner is exactly the same as doing so via the *Series Page* in the **Graph Properties dialog**.

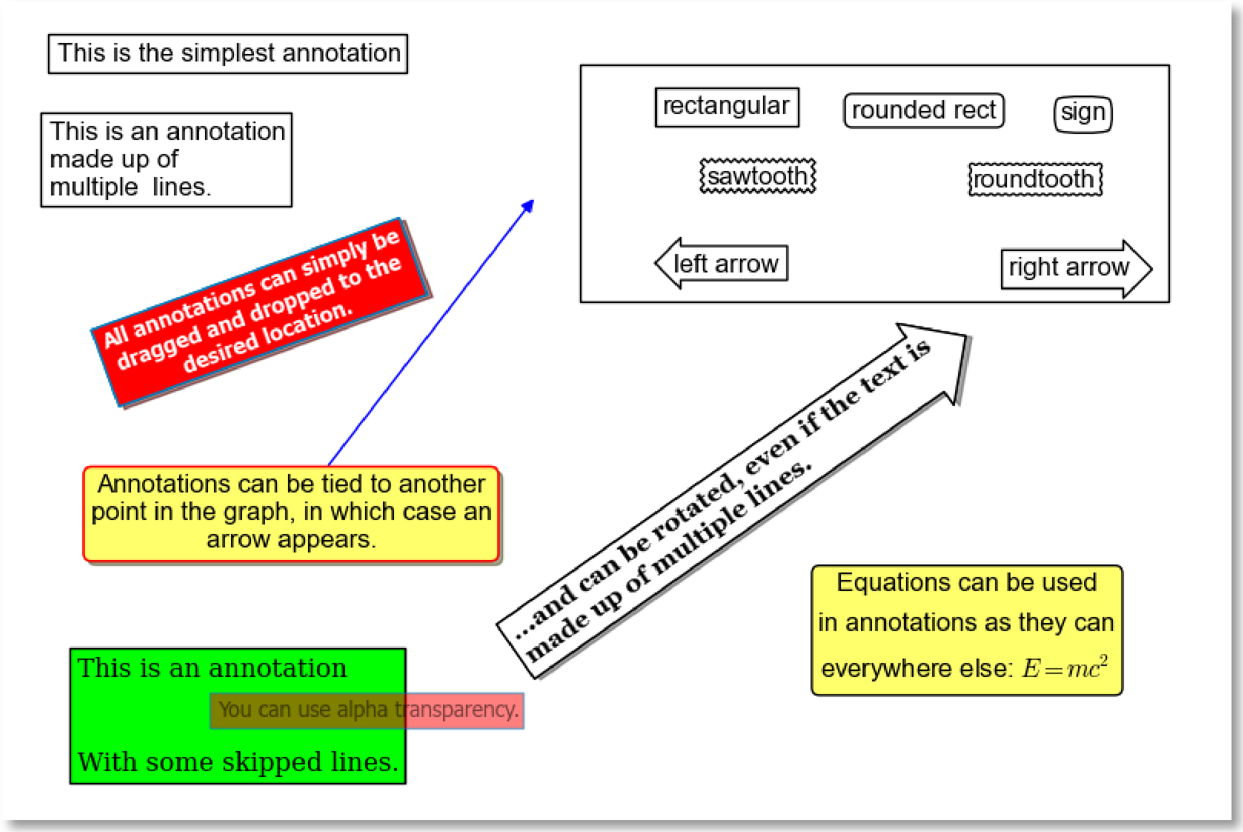
10.6.3 Legend Ordering

The order of the series in of a legend can be changed by selecting the desired series's legend label in the **Legend labels** section, and pressing the appropriate arrow buttons to arrange the ordering of the series. The topmost arrow button moves the selected series to the top; while the bottommost arrow button moves the selected series to the bottom. The inner two arrow buttons move the selected series up or down by one slot as appropriate.

The ordering of items in the legend also has another important role: it determines the order in which each series is drawn in the graph. So, if you would like a particular series to be drawn on top, move it to the front in the legend ordering. Series are drawn in reverse order as the legend ordering, which means that the item appearing on top of the legend order will also appear on top of other series in the graph.

10.7 Annotations

Annotations are essentially a text box (the text is optional, in which case an annotation can be more accurately described as a "shape") with an optional arrow that points to any desired location. Annotations add any information content to a graph that is desired by the user, including equations. mathtext is valid to use within an annotation wherever desired. Some annotation samples are shown below:


This is the simplest annotation
This is an annotation made up of multiple lines.
All annotations can simply be dragged and dropped to the desired location.
rectangular
rounded rect
sign
sawtooth
roundtooth
left arrow
right arrow
...and can be rotated, even if the text is made up of multiple lines.
Annotations can be tied to another point in the graph, in which case an arrow appears.
Equations can be used in annotations as they can everywhere else: $E = mc^2$
This is an annotation
You can use alpha transparency.
With some skipped lines.

To add a new annotation, press the Annotation button in the *Graph Toolbar*, or press the N key while in a graph. When adding an annotation, the annotation properties dialog will appear, which is documented below. Another way to create an annotation is to clone an already existing one, by right clicking an annotation and picking **Clone** from the resulting menu.

Once an annotation has been added to the graph, it may be freely dragged and dropped to its desired location. If an annotation has its *Fit to text* property set (see below in the documentation for annotation properties), it may only be moved to a particular position; the size of the annotation is fixed. If *Fit to text* is not set, grab-handles will appear on the annotation when clicked, and these can be used to resize the annotation's box as desired. In this mode, moving of the annotation proceeds as usual; just click and drag near the middle of the box.

The order in which the annotations are drawn (called the z-order) can be manipulated by right clicking an annotation and selecting one of the *Send to back*, *Bring to front*, *Send backward*, or *Bring forward* options. Note that the graph legend, arrows, and images also participate in the z-ordering process, so annotations can be placed in front of or behind the legend, other images, arrows, or other annotations as desired.

To edit an annotation, right click it and select *Annotation Properties*; alternatively, you can right click anywhere in the graph, select the *Edit Annotation* submenu, and select the annotation that you desire to edit.

To remove an annotation, right click it and select *Remove*, or press the DEL key after selecting the desired annotation. Alternatively, you can right click anywhere in the graph, select the *Remove Annotation* submenu, and select the annotation that you desire to delete.

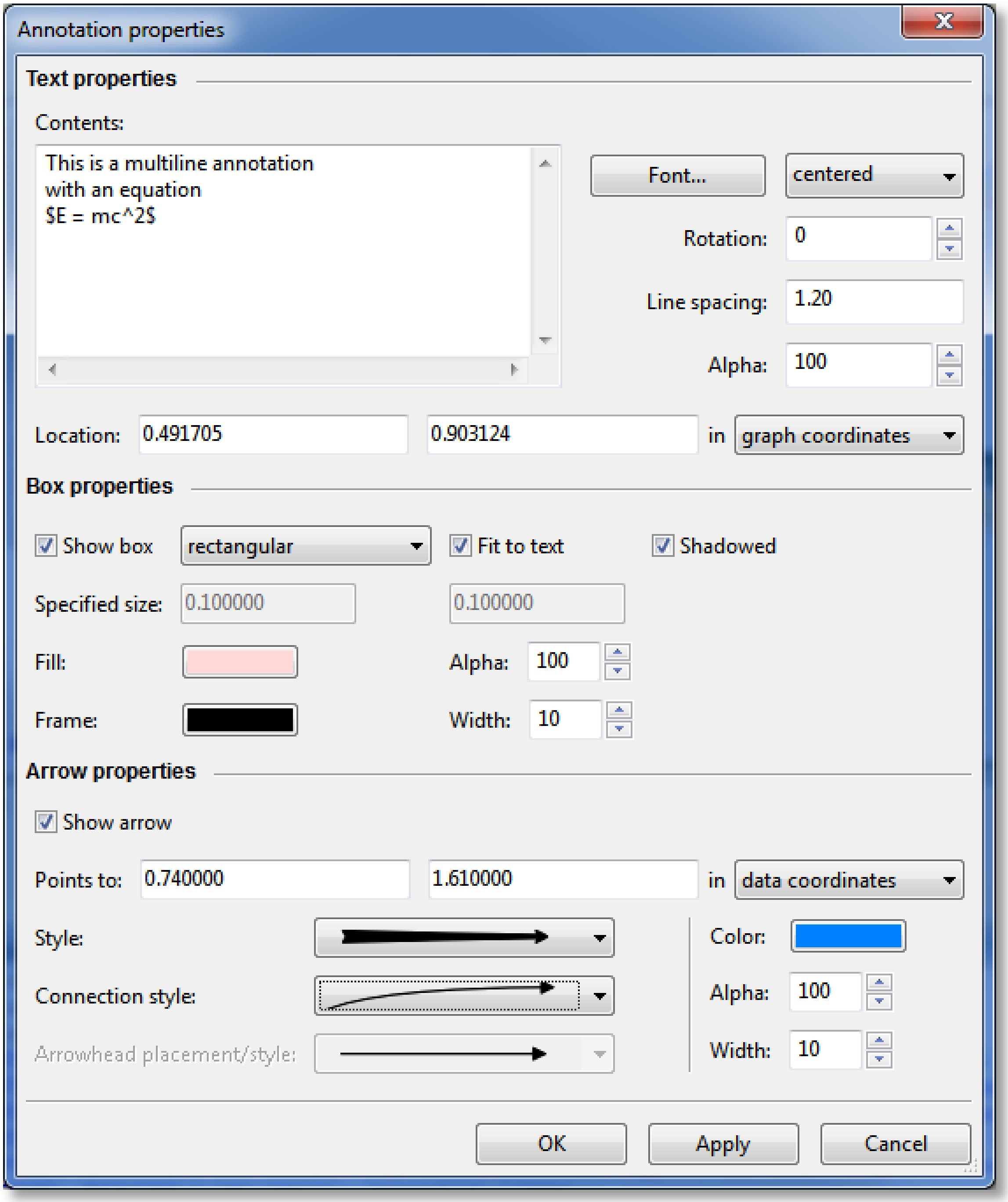
10.7.1 Cutting, Copying, and Pasting Annotations

Cut/Copy/Paste is also supported for annotations; right click the desired annotation, and pick the desired cut or copy operation. Then, right click the plot, and if there is an annotation on the clipboard (from a previous cut/copy of an annotation), the **Paste** option will appear in the menu. Annotations may be freely copied/pasted between documents or even between invocations of CurveExpert Professional.

One trick that can be used is to paste an annotation as text into another application (such as notepad) for storage. You can then highlight the pasted text and copy to the clipboard, so that you can paste the annotation into a CurveExpert Professional graph at a later time. This is useful for annotations that are commonly used; you can build a library of common annotations as just text entries in a file that you store.

10.7.2 Annotation properties

When adding a annotation, or editing its properties, the following dialog will appear:



Annotation properties

Text properties

Contents:

```
This is a multiline annotation
with an equation
$E = mc^2$
```

- Font...: centered
- Rotation: 0
- Line spacing: 1.20
- Alpha: 100
- Location: 0.491705, 0.903124 in graph coordinates

Box properties

- ☑ Show box: rectangular
- ☑ Fit to text
- ☑ Shadowed
- Specified size: 0.100000, 0.100000
- Fill:
- Alpha: 100
- Frame:
- Width: 10

Arrow properties

- ☑ Show arrow
- Points to: 0.740000, 1.610000 in data coordinates
- Style:
- Connection style:
- Arrowhead placement/style:
- Color:
- Alpha: 100
- Width: 10

Text properties

The annotation properties dialog allows you to set all available properties for an annotation (except for the z-order, which will be discussed shortly). The **Contents** section is the text content of the annotation; recall that mathtext is valid (place the mathtext between dollar signs), such that equations can be added if desired. The font of the annotation text can be adjust with the **Font** button. Horizontal alignment can be set to left aligned, centered, or right aligned. Any annotation can be rotated an an arbitrary angle as specified in the **Rotation** control; spacing between lines of a multiline annotation is given in **Line spacing** (a value of 1.2 is approximately equal to single spacing). The transparency of the text can be set in the **Alpha** control.

The (x,y) location of the center of the annotation box is specified by the **Location** fields. By default, the coordinate system used is the graph coordinate system, which is (0,0) at the bottom left of the axis frame, and (1,1) at the top right of the axis frame. However, if desired, the location can be expressed in data coordinates, in which case the center of the annotation box will always be pinned to the given (x,y) coordinate in the same system as the data set.

Note: In 3D plots, the only available coordinate system is the graph coordinate system (no data coordinate system supported).

Box properties

The middle section of the annotation properties dialog specifies the properties of the box surrounding the annotation text. The **Show box** control determines if the box is visible at all. The box style can be set to one of rectangular, square, ellipse, circle, diamond, rounded rectangular, sign, roundtooth, sawtooth, right facing arrow, left facing arrow, triangle, parallelogram-right, parallelogram-left, trapezoid, or inverted trapezoid.

The **Fit to text** control determines whether or not the box will be sized only to fit the text present (if any), or if the box will be sized according to the **Specified size** settings. The size given as a horizontal and vertical extent in the same coordinate system as given for the **Location** of the annotation box in the *Text Properties* section. Another way of specifying the size of the box is interactively on the graph itself. If an annotation has its **Fit to text** property set, it may only be moved to a particular position interactively; the size of the annotation is fixed. If **Fit to text** is not set, grab-handles will appear on the annotation when clicked, and these can be used to resize the annotation's box as desired. In this mode, moving of the annotation proceeds as usual; just click and drag near the middle of the box.

A drop shadow behind the annotation box can be activated by selecting **Shadowed**. The **Fill** and **Frame** color selectors determine the color of the interior and frame of the box, respectively.

The transparency of the box can be set via the **Alpha** control, and the line width of the box frame is given via the **Width** control.

To create a box with no text, just leave the **Contents** field blank.

Arrow properties

The last section of the annotation properties dialog specifies the properties of the arrow, if any. If **Show Arrow** is checked, an arrow is drawn from the middle of the annotation box to the (x,y) point specified in the **Points to** fields. Like the location of the annotation textbox, the location of the annotated point can be in either graph coordinates or data coordinates.

Note: In 3D plots, the only available coordinate system for the annotated point is the graph coordinate system.

The appearance of the arrow is given by the **Style**, **Connection style**, and the **Arrowhead placement/style** fields. The arrow style can be one of plain, thick, wedge, or fancy; each arrow style has its own distinct appearance that can be most easily previewed by clicking the control itself. The connection style is how the arrow connects between the annotation and the point being annotated; the connection style can be a simple straight line, an arc, or a line that travels only in the X and Y directions appropriately to reach between the source and destination. The arrowhead placement/style determines the appearance of the arrowheads themselves, and whether or not the arrowheads are placed at both ends of the connecting line, at the annotation end of the connecting line, or the destination end of the connecting line.

Note that arrowhead placement/styles are only valid for "plain" arrow styles. Also, the XY connection style is only valid for the "plain" arrow style. Finally, the connection style for the "thick" arrow style is always straight. The graphical interface will prevent you from selecting invalid combinations of arrow style, connection style, and arrowhead style.

The line color, transparency, and width of the arrow can be set via the **Line**, **Alpha**, and **Width** controls, respectively.

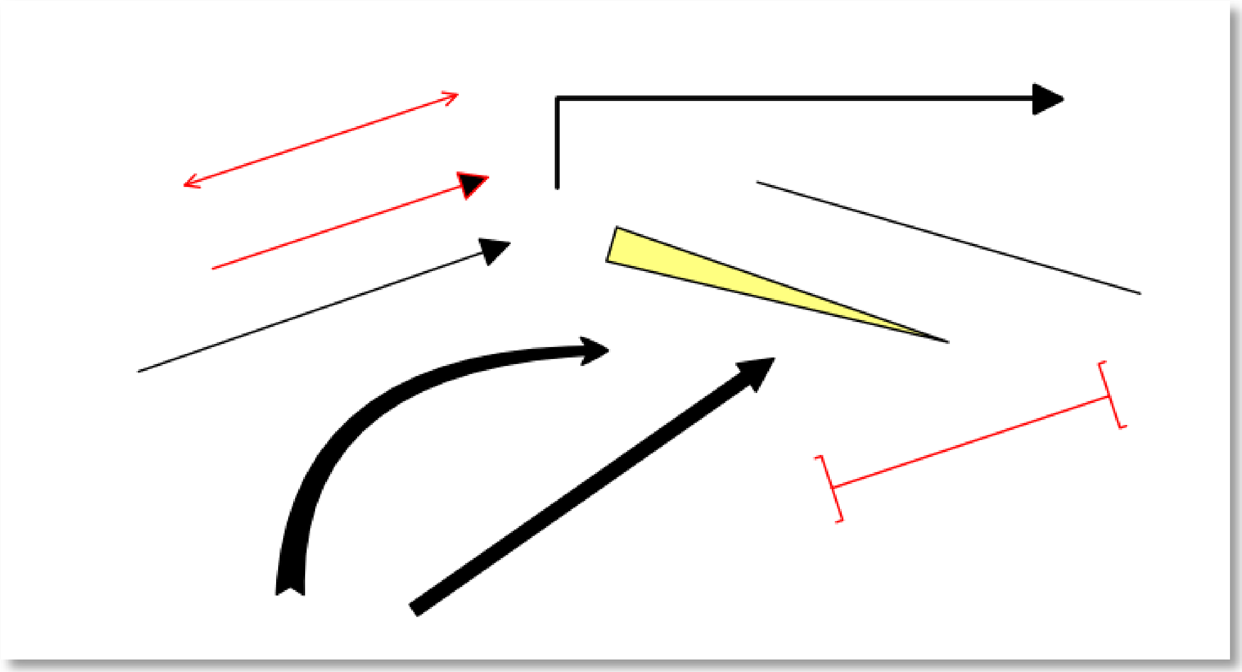
10.8 Automatic annotations

An automatic annotation is an annotation with the name, formatted equation, and coefficients (if applicable) of a particular result. The content of this annotation can then be edited just like any other by right clicking it and selecting **Annotation Properties**.

For results that are plotted on a graph, an autoannotation can be generated in two ways. The first method is to right click the graph, and pick the appropriate result under the **Autoannotate** submenu; all results available on that graph will be available for selection. Secondly, you can generate an automatic annotation by pointing to the result (it will become highlighted), right clicking, and selecting **Autoannotate**.

10.9 Arrows

Arrows are essentially lines that connect two points on a graph to indicate association between them, or to indicate a direction of information flow. Some arrow examples are shown below:



To add an arrow, press one of the "Add Arrow" buttons in the *Graph Toolbar*. When adding an arrow in this manner, an arrow will immediately appear near the center of your graph; you can then edit the arrow's properties by right clicking on it and selecting *Arrow Properties*. Another way to create an arrow is to clone an already existing one, by right clicking an arrow and picking *Clone* from the resulting menu.

Once an arrow has been added to the graph, it may be freely dragged, resized and dropped to its desired location. The order in which the arrows are drawn (called the z-order) can be manipulated by right clicking an arrow and selecting one of the "Send to back", *Bring to front*, *Send backward*, or *Bring forward* options. Note that the graph legend, annotations, and images also participate in the z-ordering process, so arrows can be placed in front of or behind the legend, other images, annotations, or other arrows as desired.

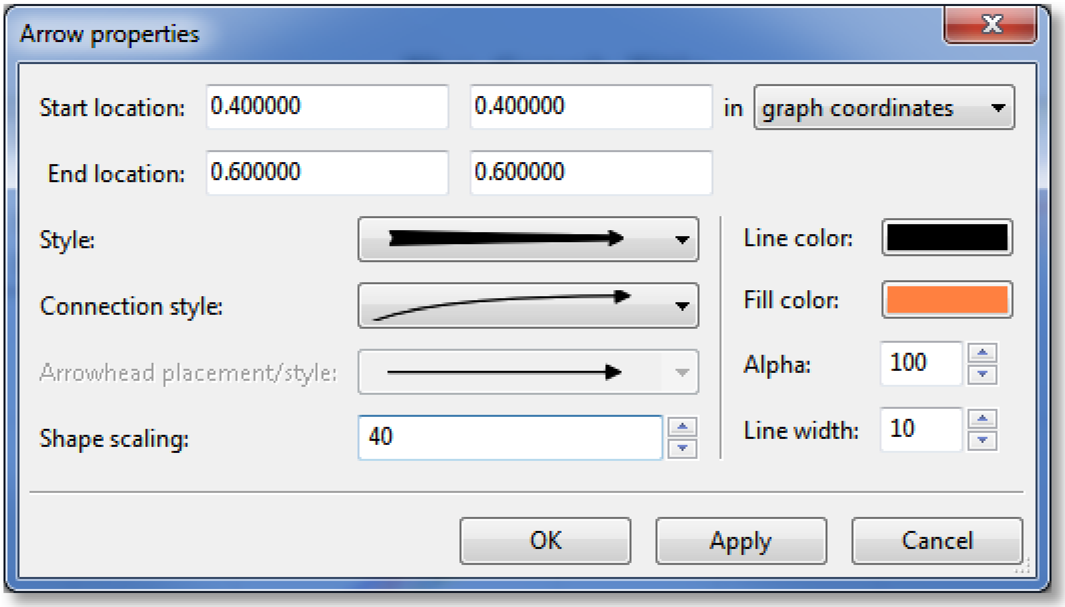
To edit an arrow, right click it and select *Arrow Properties*. To remove an arrow, right click it and select *Remove* or press the DEL key after selecting the desired arrow.

10.9.1 Cutting, Copying, and Pasting Arrows

Cut/Copy/Paste is also supported for arrow; simply right click the desired arrow, and pick the desired cut or copy operation. Then, right click the plot, and if there is an arrow on the clipboard (from a previous cut/copy of an arrow), the *Paste* option will appear in the menu. Arrows may be freely copied/pasted between documents or even between invocations of CurveExpert Professional.

10.9.2 Arrow properties

When editing the properties of an arrow, the following dialog will appear:



Arrow properties

Start location: 0.400000 0.400000 in graph coordinates

End location: 0.600000 0.600000

Style:

Connection style:

Arrowhead placement/style:

Shape scaling: 40

Line color: ■

Fill color: 🟧

Alpha: 100

Line width: 10

The starting and ending xy locations (origin and destination) are given in the **Start location** and **End location** fields; each are in the coordinate system specified (graph or data coordinates).

Note: In 3D plots, the only available coordinate system for the source and destination point of the arrow is the graph coordinate system.

The appearance of the arrow is given by the **Style**, **Connection style**, and the **Arrowhead placement/style** fields. The arrow style can be one of plain, thick, wedge, or fancy; each arrow style has its own distinct appearance that can be most easily previewed by clicking the control itself. The connection style is how the arrow connects between the origin and destination points; the connection style can be a simple straight line, an arc, or a line that travels only in the X and Y directions appropriately to reach between the source and destination. The arrowhead placement/style determines the appearance of the arrowheads themselves, and whether or not the arrowheads are placed at both ends of the connecting line, at the starting end of the connecting line, or the destination end of the connecting line.

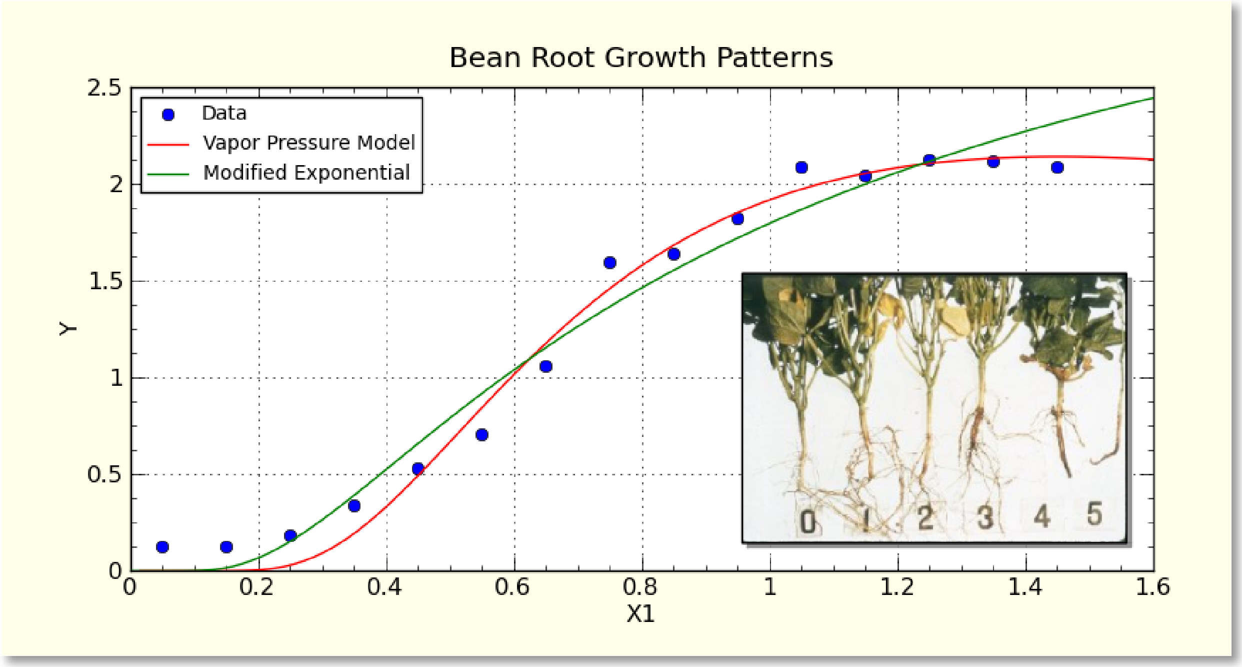
Note that arrowhead placement/styles are only valid for "plain" arrow styles. Also, the XY connection style is only valid for the "plain" arrow style. Finally, the connection style for the "thick" arrow style is always straight. The graphical interface will prevent you from selecting invalid combinations of arrow style, connection style, and arrowhead style.

The line color, transparency, and linewidth of the arrow can be set via the **Line**, **Alpha**, and **Width** controls, respectively.

The **Shape scaling** parameter determines the "fatness" of the arrow shape. Lower scalings lead to thinner arrow bodies (in cases where the arrow body is a shape, and not just a line), and smaller arrow heads. Higher scalings conversely lead to thicker arrow bodies and correspondingly larger arrow heads.

Note: To create a simple line on your graph, set the arrow style to plain, connection style to straight, and set the **Arrowhead Placement/Style** to the first selection (no arrowheads). This is done for you automatically via the line button in the *Graph Toolbar* palette.

10.10 Images



Images can be added to any graph, and will be saved along with the graph just with any other graph element. Images can be added by pressing the "Add Image" button in the *Graph Toolbar* (which will allow you to browse for the image file), via a clipboard paste from another application, or via drag and drop of an image file. Images are always first added at their native resolution, and then can be interactively moved and/or resized by clicking on them and dragging the appropriate handles. Images with embedded transparency will have that transparency respected.

Further image properties can be modified by right-clicking the image and selecting *Image Properties*. From the properties dialog, an image can be placed precisely, turned into a background, or have other image processing techniques applied to achieve a desired effect.

Supported images file formats are: PNG, BMP, JPEG, GIF, PCX, PPM, TGA, TIFF, WMF, XBM, XPM, and Photoshop 2.0/3.0 PSD files.

Once an image has been added to the graph, it may be freely dragged and/or resized to its desired location. The order in which the image are drawn (called the z-order) can be manipulated by right clicking an image and selecting one of the *Send to back*, *Bring to front*, *Send backward*, or *Bring forward* options. Note that the graph legend and annotations also participate in the z-ordering process, so images can be placed in front of or behind annotations, arrows, or the legend as desired.

To edit an image, right click it and select *Image Properties*; alternatively, you can right click anywhere in the graph, select the Edit Image submenu, and select the image that you desire to edit.

To remove an image, right click it and select *Remove* or hit the DEL key after selecting the image. Alternatively, you can right click anywhere in the graph, select the *Remove Image* submenu, and select the image that you desire to delete.

10.10.1 Cutting, Copying, and Pasting Images

Cut/Copy/Paste is also supported for annotations; right click the desired images, and pick the desired cut or copy operation. Then, right click the plot, and if there is an image on the clipboard (from a previous cut/copy of an images), the *Paste* option will appear in the menu. Images may be freely copied/pasted between documents or even between invocations of CurveExpert Professional.

To quickly delete an image, simply press the Delete key after selecting an image.

10.10.2 Image properties

When editing the properties of an image, the following dialog will appear:

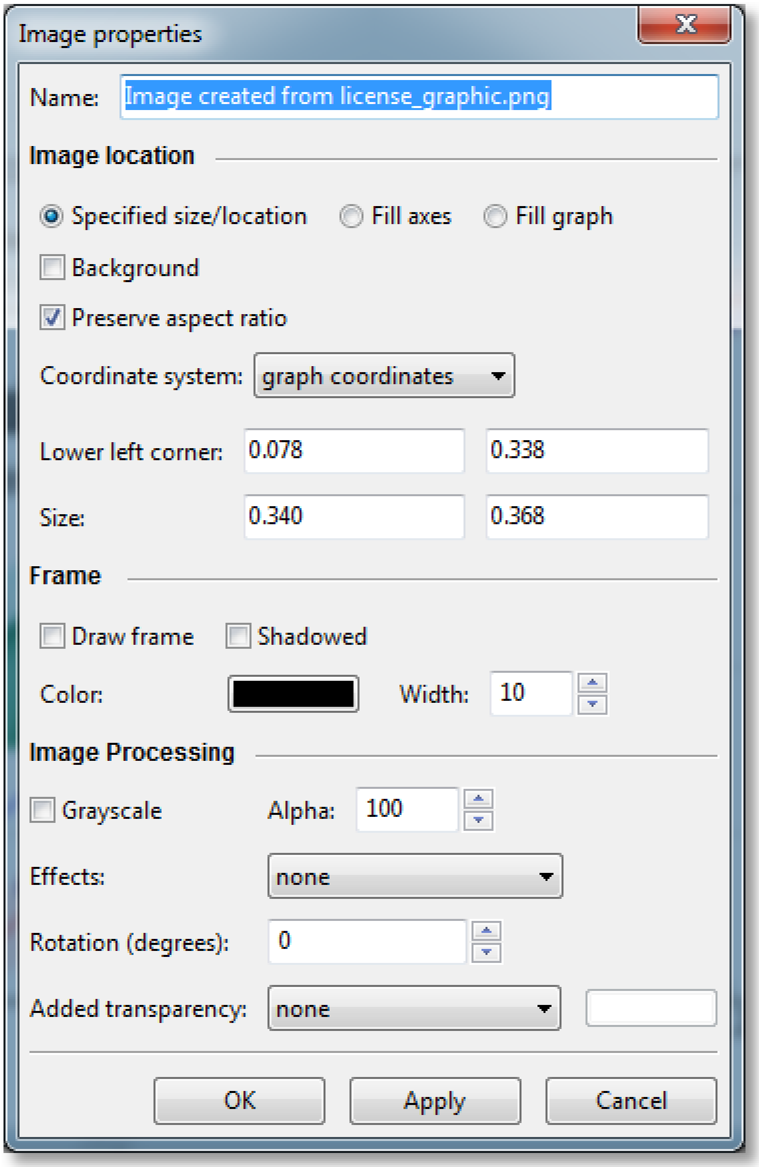


Image properties

Name: Image created from license_graphic.png

Image location

● Specified size/location ○ Fill axes ○ Fill graph

□ Background

☑ Preserve aspect ratio

Coordinate system: graph coordinates

Lower left corner: 0.078 0.338

Size: 0.340 0.368

Frame

□ Draw frame □ Shadowed

Color: ■ Width: 10

Image Processing

□ Grayscale Alpha: 100

Effects: none

Rotation (degrees): 0

Added transparency: none

The **Name** field is only for identification of the image (at times, the image name will appear in the right-click graph properties menu to allow you to edit or delete the image directly). This name can be anything that you desire it to be.

Images always obey any built-in transparency (alpha) values. So, you can (for example) read in a PNG with transparency, and it will be displayed properly within CurveExpert Professional.

Image location

Images can be utilized on a graph in a number of different ways. By default, an image is treated similarly to any drawing program, where the location is specified by the *Lower left corner* and *Size* fields. The lower left corner and size can be changed directly, or more commonly, the image can be dragged and resized to the desired location on the graph. The interpretation of the coordinate numbers can be modified by choosing the appropriate coordinate system in the *Coordinate system* choice list. Graph coordinates are (0,0) at the lower left corner of the graph and (1,1) at the top right corner; axes coordinates are (0,0) at the lower left corner of the frame and (1,1) at the top right corner, and data coordinates are coordinates in the current x/y dataset.

Normally, images share z-ordering with annotations and legends, and can be placed as desired with the *Bring to front*, etc. right-click menu items. However, if you want the image to be placed behind everything on the graph, select the *Background* checkbox.

The *Preserve aspect ratio* property governs how the image behaves as the image is resized; if selected, the image always retains its aspect ratio; if not, the image will be free to stretch to retain its specified coordinates.

If the *Fill axes* checkbox is set, the image will be placed into the background and scaled such that it fills the axes frame. Likewise, if the *Fill graph* checkbox is set, the image will be placed into the background and scaled such that it fills the entire graph. If either of these options are selected, the image is automatically placed into the background and does not share z-order with other graph elements.

Image Frame

Any image can have a simple frame drawn around it, with the *Draw frame* and *Shadowed* settings. The frame color can be specified with the *Color* setting, and the frame's line width can be modified with the *Width* setting.

Image Processing

CurveExpert Professional offers image processing facilities in order to create desired effects with your images. Your image is never modified with the image processing options; only its appearance. The *Grayscale* switch causes the image to be grayscaled before display. The *Alpha* setting determines the tranparency of the image.

The *Effects* setting sets a particular image filter that operates on the original image's pixels, and outputs a changed image, much like filters in popular image processing software packages. Filters available are blur, contour, detail, edge enhance, emboss, find edges, smooth, and sharpen.

The image can be rotated arbitrarily with the *Rotation* setting. The rotation is given in degrees, and can range between positive and negative 360 degrees.

The *Added transparency* setting is a way to generate a transparency mask on an existing image. The color mask method allows you to select a particular color, and all pixels of that color are turned transparent. *Floodfill from corners* applies a flood fill at each of the four corner pixels of the image, filling with transparency until a different color is encountered.

Note: In 3D plots, the only available coordinate system is the graph coordinate system (no data coordinate system supported).

10.11 Saving graphs

To save a graph, either choose the *Save* button in the graph toolbar, or right click the graph and select *Save*. A file chooser will appear that allows you to select the location and filename of the graph to be saved, and the format of the picture file to be saved. For image (raster) files, supported filetypes are:

- portable network graphics (PNG)
- JPEG
- TIFF
- raw RGBA bitmaps (RGBA or RAW)

Supported vector graphics file formats are:

- scalable vector graphics (SVG)

- enhanced metafiles (EMF)
- encapsulated postscript (EPS)
- postscript (PS)
- portable document format (PDF)

Of these formats, the PNG format is the most common for raster-type images, and SVG is most common for vector graphics. A PNG file can be read by any graphics application, and by applications such as Microsoft Word and Powerpoint. SVG files can be read by computer illustration software, such as Adobe Illustrator or Inkscape. It should be noted that an SVG file written by CurveExpert Professional retains all object information, so you can add or remove elements from the graph after-the-fact with an illustration package, if desired.

10.12 Copying graphs

To copy a graph, right click it and select *Copy*. An image will be copied to the clipboard, which can be subsequently pasted into any application that can receive an image from the clipboard, such as Microsoft Word.

10.13 Printing graphs

To print a graph, right click it and select *Print Preview* or *Print*, depending on the desired action. Alternatively, press the Print button in the graph toolbar associated with the desired graph.

10.14 Graph Themes

A graph theme gives the user the opportunity to set up the “look” of a graph once, and reuse these settings in subsequent graphs. A graph theme is a snapshot of the parameters set in the Graph Properties window (accessible by right-clicking a graph and selecting *Properties*) as well as the properties of elements on the graph, such as annotations and arrows. In fact, so that the appearance of annotations, legends, arrows, and images match the look of your graph, it is recommended to create one each element and configure them before saving the theme.

10.14.1 Saving a theme

To save the current look of your graph, right click the graph and select *Save Theme*. Give your theme a name, and select OK.

The second way to save a theme is to select *Tools->Manage Graph Themes* from the CurveExpert Professional main window; this gives access to a sandbox where you can change the graph look freely, and save the results when you are finished. See *Graph Theme Manager*.

10.14.2 Applying a theme

To apply a saved theme to your current graph, right click the graph and select *Apply Theme*, which will give access to all of the available saved themes. Select the desired theme, and the graph will change its look.

10.14.3 Copying/Pasting a theme

A theme can be copied by right clicking the graph and selecting *Copy Current Theme*. This operation internally stores the theme of the current graph so that you can use this theme on a another graph. To apply a previously copied theme, right click the graph and select *Paste Theme*. While this operation is extremely similar to a standard cut/paste clipboard operation, it does not actually use the clipboard; so, any item that you have on the clipboard remains intact. This does mean, however, that you cannot copy/paste themes between applications; if this is desired, just save and then apply the theme as documented above.

10.15 2D Contour Graphing

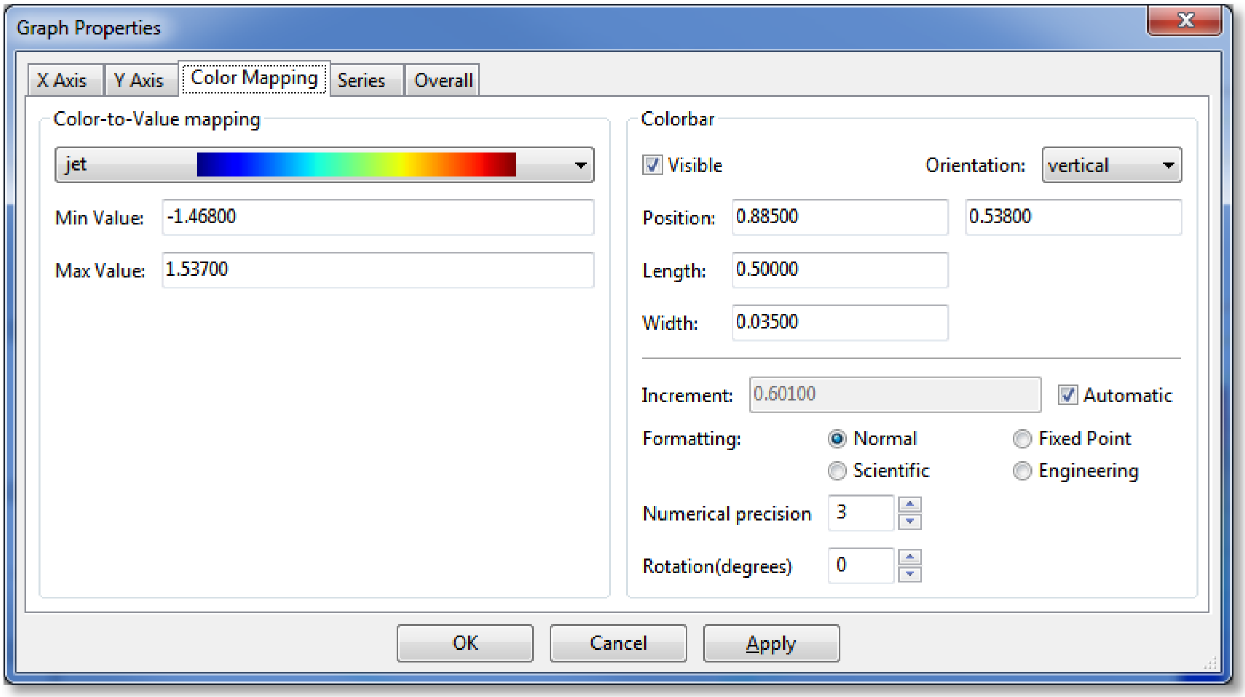
The 2D contour graphing capability is activated when there are two independent variables present. So, either a dataset must be read in with two independent variables, or the user can select *File->New*, and specify three columns in the dataset. The typical method of creating a 2D contour graph is to select the "+" tab in the "Graphs and Data" pane, followed by a selection of "2D Contour". This creates a graph axes that is appropriate for both line and filled contours.

10.15.1 Interactive Viewing

Mouse manipulations, panning and zooming in a contour plot is identical to that described above for normal XY graphs.

10.15.2 Changing 2D Contour Graph Properties

To access the graph properties, right click the graph, and select *Graph Properties* as usual. The main changes that you will see are that there is an extra tab for **Color Mapping**; so that you can set the colorbar appearance, as well as changing the mapping from values in the z direction (height) in the contour to a color. The color mapping panel (within the Graph Properties dialog) is shown below:



Color-to-Value mapping

jet

Min Value: -1.46800

Max Value: 1.53700

Colorbar

☑ Visible

Orientation: vertical

Position: 0.88500 0.53800

Length: 0.50000

Width: 0.03500

Increment: 0.60100 ☑ Automatic

Formatting: ● Normal ○ Fixed Point ○ Scientific ○ Engineering

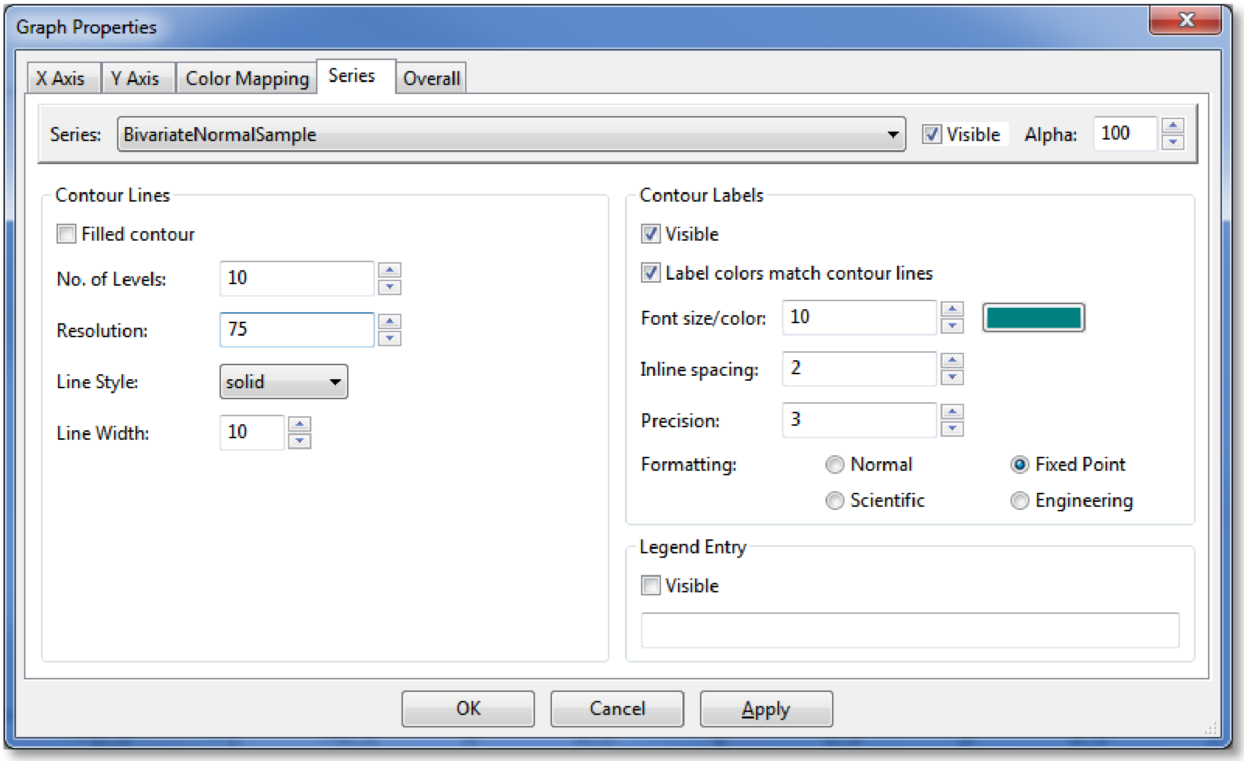
Numerical precision 3

Rotation(degrees) 0

The **Colorbar** section allows the specification of the colorbar that is shown on the graph (as long as the *Visible* checkbox is set) to show the mapping between contour height (z) and a color. The *Orientation* selection determines if this colorbar is drawn in a vertical or horizontal orientation. The position, length, and width of the colorbar is given in the next section of fields; the position as well as the length and width are given in terms of axes coordinates, where (0,0) is the bottom left corner of the axes pane, and (1,1) is the upper right corner. The *Font* button allows the specification of the font and color of the text used to label the ticks on the colorbar.

The **Color mapping** section determines the mapping between contour height (z) and color. This mapping can be selected in the first pulldown; there are currently 130 choices for a colormap to use. The minimum value and maximum value given are mapped to the left and right end of the selected colormap, respectively. The tick increment can either be automatically determined or specified; the tick increment affects how many ticks are drawn in the colorbar.

Further, the Series section holds the settings for the contour series, when that series is selected in the *Series* pulldown. The relevant panel is shown below:



Series: BivariateNormalSample ☑ Visible Alpha: 100

Contour Lines

☐ Filled contour

No. of Levels: 10

Resolution: 75

Line Style: solid

Line Width: 10

Contour Labels

☑ Visible

☑ Label colors match contour lines

Font size/color: 10 🟦

Inline spacing: 2

Precision: 3

Formatting: ○ Normal ● Fixed Point ○ Scientific ○ Engineering

Legend Entry

☐ Visible

The **Contour Lines** section determines the overall appearance of the contour lines (or filled areas as appropriate). The *Filled contour* checkbox determines if the contour is drawn as a set of curves, or as a set of filled regions. The *No. of Levels* field determines the resolution with which the z (height) direction is filled with contour lines or filled regions, as appropriate. The *Resolution* field increases the number of line segments that are used to define the contour curve at a specific height (a higher resolution means a less segmeneted-looking contour curve). The *Line Style* and *Line Width* are active only when the contour is not filled, and determines the style and width of the lines used to draw the contour lines.

For more contour lines, increase *No. of Levels*. For a better quality draw of the contour lines, increase *Resolution*.

The **Contour Labels** section controls the inline height(z) labeling of the contour lines, and is only applicable when using unfilled contours. The *Visible* flag toggles whether or not the labels are drawn. *Label colors match contour lines* determines if the contour labels are colored the same as the contour line that they are associated with, or colored with the font color as selected in the color picker on the next line (labeled “Font size/color”). The *Font size/color* section determines the size of the labels and their color, provided that *Label colors match contour lines* is left unchecked. The *Inline spacing* determines the size of the break in the contour line in which the label is inserted; smaller sizes mean that the contour lines practically touch the label. The *Precision* and *Formatting* settings determine the number of decimal places used in the label, and the formatting style of the labels, respectively.

The **Legend Entry** section indicates whether this series should be listed in the legend, and if so, what label should be used (the default label in the legend is the name of the series, which is used if the legend label field is left empty).

10.16 3D Graphing

The 3D graphing capability is activated when there are two independent variables present. So, either a dataset must be read in with two independent variables, or the user can select *File->New*, and specify three columns in the dataset.

Note that pointing at a series in a 3D plot will not highlight it as is done in the 2D plots; this operation is too expensive to be carried out in an interactive way.

10.16.1 Interactive Viewing

For the most part, plotting in 3D is exactly the same as in 2D, except with more interactivity with the viewing angle.

Clicking and dragging the left mouse button on a 3D plot will dynamically change the viewing angle. Middle click and drag zooms the 3D plot.

Note: If you would rather zoom with the right mouse, go to *Tools->Preferences->Graphing* and check the appropriate box. However, the context menu (which normally takes a right click to access), is then mapped to the middle mouse button.

More transformations for the 3D plot are available via the pan button; please see *Pan* for details.

10.16.2 Changing 3D Graph Properties

To access the graph properties, right click the graph (or middle click, if you have remapped the mouse buttons for graphs), and select *Properties* as usual. The main changes that you will see are that there are three tabs for the axes, so that you can set scale and properties on the X1, X2, and Y axes appropriately.

Controls throughout the graph properties that are not applicable for 3D graphs are either hidden from view or disabled.

The other difference for 3D plots is within the **Series** section, where the controls have changed appropriately to deal with a 3D entity instead of a 2D line. You can still choose line width and color, which determines the line properties of the 3D object being drawn. The new **Surface** section allows you to set properties for your surface; the type can be set to the following:

- scatter
- wireframe
- triangulated wireframe
- surface
- triangulated surface
- contour
- filled contour

The resolution determines the size of the mesh used to reconstruct the graphical object; i.e., if the resolution is 50, then a 50x50 mesh is created on which the 3D function is evaluated. The drawing expense rises dramatically with the resolution, so it is best to set this parameter as low as possible and still have a desirable looking graph. For a final render to picture or printer, you can raise this parameter in order to achieve the highest quality at the expense of (very) slow rendering. Note that resolution is ignored for irregular triangulated surfaces and wireframes; the points in the dataset being plotted are used directly.

The **Color mapping** section determines the mapping between the z dimension (the dependent variable) and color (often referred to as a pseudocolor map). This mapping can be selected in the first pulldown; there are currently 130 choices for a colormap to use. The min value and max value given are mapped to the left and right end of the selected colormap, respectively. Any surface that is drawn in any filled manner (i.e. not wireframed) will use these color mapping settings. The tick increment can either be automatically determined or specified; the tick increment affects how many ticks are drawn in the colorbar.

10.17 Keyboard Shortcuts for Graphs

Keyboard shortcuts can be used when a graph, or the notebook tab associated with the graph, has keyboard focus.

Keyboard Shortcut	Action
a	autoscale. Identical to clicking the autoscale button in the graph toolbar. See *Graph Toolbar* and *Autoscaling*.
p	toggle panning mode. Identical to clicking the pan button in the graph toolbar. See *Graph Toolbar*.
z	toggle zooming mode. Identical to clicking the zoom button in the graph toolbar. See *Graph Toolbar*.
r	resize the graph frame interactively (pointer must be over the frame when r is pressed).
x	Open context sensitive properties dialog. Identical to right click->Properties. See *Changing Graph Properties*.
1-9	Pressing a number button (1-9) will open the graph properties dialog directly to the corresponding series as listed in the legend.
Shift + 1-9	Pressing a number button (1-9) will open the Results Window (see *Querying Result Details*) directly to the corresponding series as listed in the legend.
Ctrl+C	if the pointer is over an image, annotation, or arrow, copy it; if over the graph, copy the graph to the clipboard as an image. Identical to right click->Copy. See *Graph Toolbar*.
Ctrl+V	paste a graph element (image, annotation, or arrow)
,	move backward to the previous view. Identical to clicking the back button in the graph toolbar. See *Graph Toolbar*.
.	move forward to the next view. Identical to clicking the forward button in the graph toolbar. See *Graph Toolbar*.
h	return to the first view. Identical to clicking the home button in the graph toolbar. See *Graph Toolbar*.
n	add a new annotation
r	add a new arrow

10.18 Mathtext

You can use a subset of TeX markup anywhere that a graph uses text (for a label, title, legend, annotation, etc.) by placing a bit of your text inside a pair of dollar signs ($). This capability allows you to typeset quite complex mathematical expressions within any text label on any plot. Regular text and math text can be freely intermixed. For example:

```
Surface Area $m^2$
```

renders as:

$$\text{Surface Area } m^2$$

Note: To make it easy to display monetary values such as $100.00, if a single dollar sign is present in the entire string, it will be displayed as a dollar sign.

Greek letters and a large number of symbols are supported.

10.18.1 Subscripts and superscripts

To make subscripts and superscripts, use the `'_'` and `'^'` symbols:

```
$\alpha_i > \beta_i$
```

$$\alpha_i > \beta_i$$

Some symbols automatically put their sub/superscripts under and over the operator. For example, to write the sum of x_i from 0 to ∞, you could do:

```
$\sum_{i=0}^\infty x_i$
```

$$\sum_{i=0}^\infty x_i$$

10.18.2 Fractions, binomials and stacked numbers

Fractions, binomials and stacked numbers can be created with the `\frac{}{}`, `\binom{}{}` and `\stackrel{}{}` commands, respectively:

```
$\frac{3}{4} \binom{3}{4} \stackrel{3}{4}$
```

produces

$$\frac{3}{4} \binom{3}{4} \stackrel{3}{4}$$

Fractions can be arbitrarily nested:

```
$\frac{5 - \frac{1}{x}}{4}$
```

produces

$$\frac{5 - \frac{1}{x}}{4}$$

Note that special care needs to be taken to place parentheses and brackets around fractions. Doing things the obvious way produces brackets that are too small:

```
$(\frac{5 - \frac{1}{x}}{4})$
```

$$(\frac{5 - \frac{1}{x}}{4})$$

The solution is to precede the bracket with `\left` and `\right` to inform the parser that those brackets encompass the entire object:

```
$\left(\frac{5 - \frac{1}{x}}{4}\right)$
```

$$\left(\frac{5 - \frac{1}{x}}{4}\right)$$

10.18.3 Radicals

Radicals can be produced with the `\sqrt[]{}` command. For example:

```
$\sqrt{2}$
```

$$\sqrt{2}$$

Any base can (optionally) be provided inside square brackets. Note that the base must be a simple expression, and cannot contain layout commands such as fractions or sub/superscripts:

```
$\sqrt[3]{x}$
```

$$\sqrt[3]{x}$$

10.18.4 Inline Fonts

The default font is *italics* for mathematical symbols.

To change fonts, eg, to write “sin” in a Roman font, enclose the text in a font command:

```
$s(t) = \mathcal{A}\mathrm{sin}(2 \omega t)$
```

$$s(t) = \mathcal{A}\mathrm{sin}(2 \omega t)$$

More conveniently, many commonly used function names that are typeset in a Roman font have shortcuts. So the expression above could be written as follows:

```
$s(t) = \mathcal{A}\sin(2 \omega t)$
```

$$s(t) = \mathcal{A}\sin(2 \omega t)$$

Here “s” and “t” are variable in italics font (default), “sin” is in Roman font, and the amplitude “A” is in calligraphy font. Note in the example above the calligraphy `A` is extremely close to the `sin`. You can use a spacing command to add a little whitespace between them:

```
s(t) = \mathcal{A}\/\sin(2 \omega t)
```

$$s(t) = \mathcal{A}\/\sin(2 \omega t)$$

The choices available with all fonts are:

Command	Result
`\mathrm{Roman}`	Roman
`\mathit{Italic}`	*Italic*
`\mathtt{Typewriter}`	$\mathtt{Typewriter}$
`\mathcal{CALLIGRAPHY}`	$\mathcal{CALLIGRAPHY}$

10.18.5 Global Fonts

In CurveExpert Professional, there are three choices for the global font to use for math symbols: Computer Modern, STIX Serif, and STIX Sans Serif. To select one of these fonts, choose *Edit->Preferences->Graphing->Global mathtext font*.

The look of each font slightly different, with the Computer Modern font resembling the traditional TeX typsetting most closely, STIX Serif meant to blend with Serif-style fonts (such as Times New Roman), and STIX Sans Serif being a mathematical font without any serifs at all.

10.18.6 Accents

An accent command may precede any symbol to add an accent above it. There are long and short forms for some of them.

Command	Result
`\acute a` or `\'a`	$\acute{a}$
`\bar a`	$\bar{a}$
`\breve a`	$\breve{a}$
`\ddot a` or `\"a`	$\ddot{a}$
`\dot a` or `\.a`	$\dot{a}$
`\grave a` or ``\`a``	$\grave{a}$
`\hat a` or `\^a`	$\hat{a}$
`\tilde a` or `\~a`	$\tilde{a}$
`\vec a`	$\vec{a}$

In addition, there are two special accents that automatically adjust to the width of the symbols below:

Command	Result
`\widehat{xyz}`	$\widehat{xyz}$
`\widetilde{xyz}`	$\widetilde{xyz}$

Care should be taken when putting accents on lower-case i's and j's. Note that in the following `\imath` is used to avoid the extra dot over the i:

```
r"$\hat i\ \ \hat \imath$"
```

$$\hat i\ \ \hat \imath$$

10.18.7 Symbols

Other symbols available in math typesetting are listed below.

Lower-case Greek

α `\alpha`	β `\beta`	χ `\chi`	δ `\delta`	$\digamma$ `\digamma`
ϵ `\epsilon`	η `\eta`	γ `\gamma`	ι `\iota`	κ `\kappa`
λ `\lambda`	μ `\mu`	ν `\nu`	ω `\omega`	ϕ `\phi`
π `\pi`	ψ `\psi`	ρ `\rho`	σ `\sigma`	τ `\tau`
θ `\theta`	υ `\upsilon`	ε `\varepsilon`	$\varkappa$ `\varkappa`	φ `\varphi`
ϖ `\varpi`	ϱ `\varrho`	ς `\varsigma`	ϑ `\vartheta`	ξ `\xi`
ζ `\zeta`				

Upper-case Greek

Δ \Delta	Γ \Gamma	Λ \Lambda	Ω \Omega	Φ \Phi	Π \Pi
Ψ \Psi	Σ \Sigma	Θ \Theta	ϒ \Upsilon	Ξ \Xi	℧ \mho
∇ \nabla					

Hebrew

ℵ \aleph	ℶ \beth	ℸ \daleth	ℷ \gimel

Delimiters

<table>
<tr><td>/ /</td><td>[[</td><td>⇓ \Downarrow</td><td>⇑ \Uparrow</td><td>‖ \Vert</td><td>\ \backslash</td></tr>
<tr><td>↓ \downarrow</td><td>⟨ \langle</td><td>⌈ \lceil</td><td>⌊ \lfloor</td><td>⌞ \llcorner</td><td>⌟ \lrcorner</td></tr>
<tr><td>⟩ \rangle</td><td>⌉ \rceil</td><td>⌋ \rfloor</td><td>⌜ \ulcorner</td><td>↑ \uparrow</td><td>⌝ \urcorner</td></tr>
<tr><td>| \vert</td><td>{ \{</td><td>‖ \|</td><td>} \}</td><td>]]</td><td>| |</td></tr>
</table>

Big symbols

⋂ \bigcap	⋃ \bigcup	⨀ \bigodot	⨁ \bigoplus	⨂ \bigotimes
⨄ \biguplus	⋁ \bigvee	⋀ \bigwedge	∐ \coprod	∫ \int
∮ \oint	∏ \prod	∑ \sum		

Standard function names

Pr \Pr	arccos \arccos	arcsin \arcsin	arctan \arctan
arg \arg	cos \cos	cosh \cosh	cot \cot
coth \coth	csc \csc	deg \deg	det \det
dim \dim	exp \exp	gcd \gcd	hom \hom
inf \inf	ker \ker	lg \lg	lim \lim
lim inf \liminf	lim sup \limsup	ln \ln	log \log
max \max	min \min	sec \sec	sin \sin
sinh \sinh	sup \sup	tan \tan	tanh \tanh

Binary operation and relation symbols

≎ \Bumpeq	⋒ \Cap	⋓ \Cup
≑ \Doteq	⋈ \Join	⋐ \Subset
⋑ \Supset	⊩ \Vdash	⊪ \Vvdash
≈ \approx	≊ \approxeq	∗ \ast
≍ \asymp	϶ \backepsilon	∽ \backsim
⋍ \backsimeq	⊼ \barwedge	∵ \because
≬ \between	◯ \bigcirc	▽ \bigtriangledown
△ \bigtriangleup	◀ \blacktriangleleft	▶ \blacktriangleright
⊥ \bot	⋈ \bowtie	⊡ \boxdot
⊟ \boxminus	⊞ \boxplus	⊠ \boxtimes
• \bullet	≏ \bumpeq	∩ \cap
· \cdot	∘ \circ	≗ \circeq
≔ \coloneq	≅ \cong	∪ \cup
⋞ \curlyeqprec	⋟ \curlyeqsucc	⋎ \curlyvee
⋏ \curlywedge	† \dag	⊣ \dashv
‡ \ddag	⋄ \diamond	÷ \div
⋇ \divideontimes	≐ \doteq	≑ \doteqdot
∔ \dotplus	⩞ \doublebarwedge	≖ \eqcirc
≕ \eqcolon	≂ \eqsim	⪖ \eqslantgtr
⪕ \eqslantless	≡ \equiv	≒ \fallingdotseq

⌢ \frown	≥ \geq	≧ \geqq
⩾ \geqslant	≫ \gg	⋙ \ggg
⪊ \gnapprox	≩ \gneqq	⋧ \gnsim
⪆ \gtrapprox	⋗ \gtrdot	⋛ \gtreqless
⪌ \gtreqqless	≷ \gtrless	≳ \gtrsim
∈ \in	⊺ \intercal	⋋ \leftthreetimes
≤ \leq	≦ \leqq	⩽ \leqslant
⪅ \lessapprox	⋖ \lessdot	⋚ \lesseqgtr
⪋ \lesseqqgtr	≶ \lessgtr	≲ \lesssim
≪ \ll	⋘ \lll	⪉ \lnapprox
≨ \lneqq	⋦ \lnsim	⋉ \ltimes
∣ \mid	⊨ \models	∓ \mp
⊯ \nVDash	⊮ \nVdash	≉ \napprox
≇ \ncong	≠ \ne	≠ \neq
≠ \neq	≢ \nequiv	≱ \ngeq
≯ \ngtr	∋ \ni	≰ \nleq
≮ \nless	∤ \nmid	∉ \notin
∦ \nparallel	⊀ \nprec	≁ \nsim
⊄ \nsubset	⊈ \nsubseteq	⊁ \nsucc
⊅ \nsupset	⊉ \nsupseteq	⋪ \ntriangleleft

⋬ \ntrianglelefteq	⋫ \ntriangleright	⋭ \ntrianglerighteq
⊭ \nvDash	⊬ \nvdash	⊙ \odot
⊖ \ominus	⊕ \oplus	⊘ \oslash
⊗ \otimes	∥ \parallel	⊥ \perp
⋔ \pitchfork	± \pm	≺ \prec
⪷ \precapprox	≼ \preccurlyeq	⪯ \preceq
⪹ \precnapprox	⋨ \precnsim	≾ \precsim
∝ \propto	⋌ \rightthreetimes	≓ \risingdotseq
⋊ \rtimes	∼ \sim	≃ \simeq
/ \slash	⌣ \smile	⊓ \sqcap
⊔ \sqcup	⊏ \sqsubset	⊏ \sqsubset
⊑ \sqsubseteq	⊐ \sqsupset	⊐ \sqsupset
⊒ \sqsupseteq	⋆ \star	⊂ \subset
⊆ \subseteq	⫅ \subseteqq	⊊ \subsetneq
⫋ \subsetneqq	≻ \succ	⪸ \succapprox
≽ \succcurlyeq	⪰ \succeq	⪺ \succnapprox
⋩ \succnsim	≿ \succsim	⊃ \supset
⊇ \supseteq	⫆ \supseteqq	⊋ \supsetneq
⫌ \supsetneqq	∴ \therefore	× \times
⊤ \top	◃ \triangleleft	⊴ \trianglelefteq

≜ \triangleq	▹ \triangleright	⊵ \trianglerighteq
⊎ \uplus	⊨ \vDash	∝ \varpropto
⊲ \vartriangleleft	⊳ \vartriangleright	⊢ \vdash
∨ \vee	⊻ \veebar	∧ \wedge
≀ \wr		

Arrow symbols

⇓ \Downarrow	⇐ \Leftarrow
⇔ \Leftrightarrow	⇚ \Lleftarrow
⟸ \Longleftarrow	⟺ \Longleftrightarrow
⟹ \Longrightarrow	↰ \Lsh
⇗ \Nearrow	⇖ \Nwarrow
⇒ \Rightarrow	⇛ \Rrightarrow
↱ \Rsh	⇘ \Searrow
⇙ \Swarrow	⇑ \Uparrow
⇕ \Updownarrow	↺ \circlearrowleft
↻ \circlearrowright	↶ \curvearrowleft
↷ \curvearrowright	⇠ \dashleftarrow
⇢ \dashrightarrow	↓ \downarrow
⇊ \downdownarrows	⇃ \downharpoonleft
⇂ \downharpoonright	↩ \hookleftarrow
↪ \hookrightarrow	⇝ \leadsto
← \leftarrow	↢ \leftarrowtail
↽ \leftharpoondown	↼ \leftharpoonup
⇇ \leftleftarrows	↔ \leftrightarrow
⇆ \leftrightarrows	⇋ \leftrightharpoons
↭ \leftrightsquigarrow	⇜ \leftsquigarrow

⟵ \longleftarrow	⟷ \longleftrightarrow
⟼ \longmapsto	⟶ \longrightarrow
↫ \looparrowleft	↬ \looparrowright
↦ \mapsto	⊸ \multimap
⇍ \nLeftarrow	⇎ \nLeftrightarrow
⇏ \nRightarrow	↗ \nearrow
↚ \nleftarrow	↮ \nleftrightarrow
↛ \nrightarrow	↖ \nwarrow
→ \rightarrow	↣ \rightarrowtail
⇁ \rightharpoondown	⇀ \rightharpoonup
⇄ \rightleftarrows	⇄ \rightleftarrows
⇌ \rightleftharpoons	⇌ \rightleftharpoons
⇉ \rightrightarrows	⇉ \rightrightarrows
⇝ \rightsquigarrow	↘ \searrow
↙ \swarrow	→ \to
↞ \twoheadleftarrow	↠ \twoheadrightarrow
↑ \uparrow	↕ \updownarrow
↕ \updownarrow	↿ \upharpoonleft
↾ \upharpoonright	⇈ \upuparrows

Miscellaneous symbols

$ \$	Å \AA	Ⅎ \Finv
⅁ \Game	ℑ \Im	¶ \P
ℜ \Re	§ \S	∠ \angle
‵ \backprime	★ \bigstar	■ \blacksquare
▲ \blacktriangle	▼ \blacktriangledown	⋯ \cdots
✓ \checkmark	® \circledR	Ⓢ \circledS
♣ \clubsuit	∁ \complement	© \copyright
⋱ \ddots	♢ \diamondsuit	ℓ \ell
∅ \emptyset	ð \eth	∃ \exists
♭ \flat	∀ \forall	ℏ \hbar
♡ \heartsuit	ℏ \hslash	∭ \iiint
∬ \iint	∬ \iint	ı \imath
∞ \infty	ȷ \jmath	… \ldots
∡ \measuredangle	♮ \natural	¬ \neg
∄ \nexists	∰ \oiiint	∂ \partial
′ \prime	♯ \sharp	♠ \spadesuit
∢ \sphericalangle	\ss	▽ \triangledown
∅ \varnothing	△ \vartriangle	⋮ \vdots
℘ \wp	¥ \yen	

CHAPTER

ELEVEN

DIGITIZING

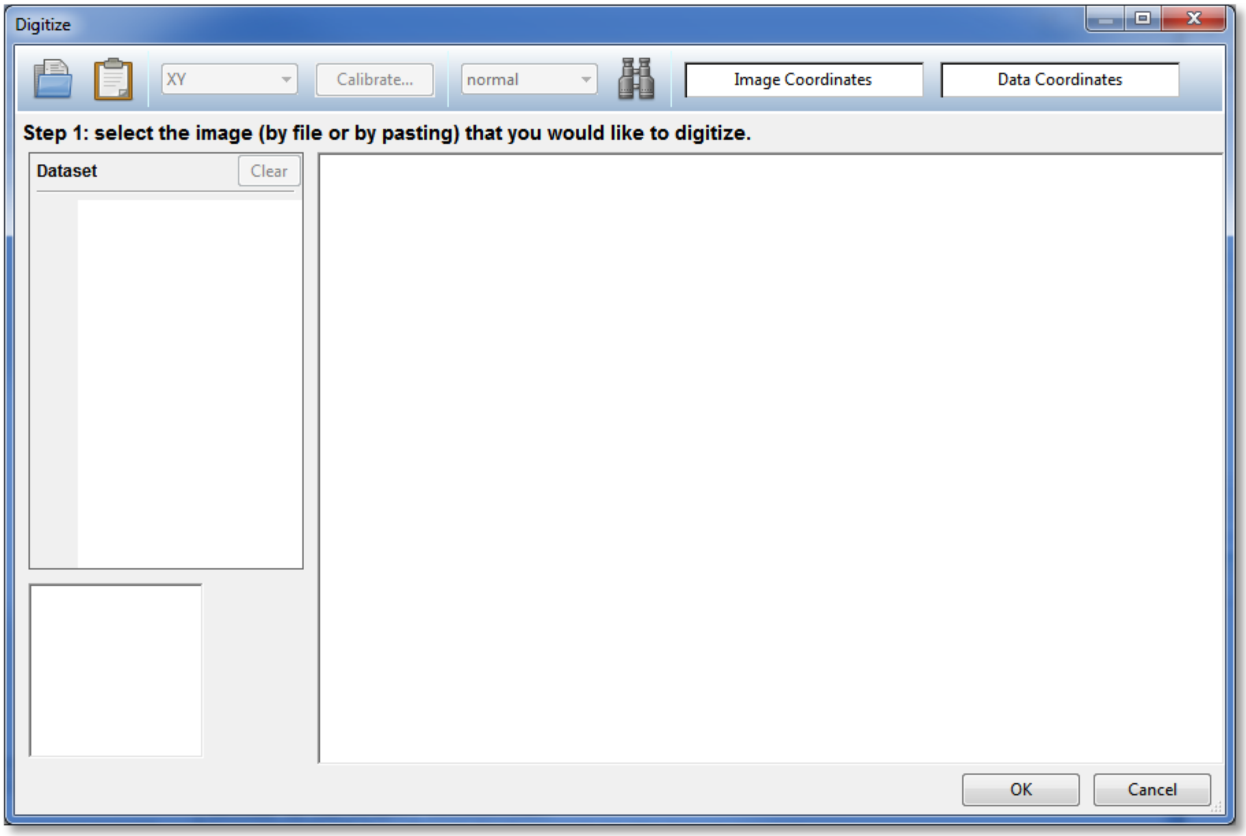
11.1 Introduction

In some cases, the data to be analyzed is in the form of a picture or image. In CurveExpert Professional, you can use the built-in digitization capabilities in order to extract the data present on an image into a dataset (to “digitize”, in this context, means to convert content from an image into a usable dataset). To successfully digitize information from an image into data, information about how pixels in an image map to real data must be collected by the application, and the digitizer leads you through this process.

The digitizer supports Cartesian (XY) and polar plots; XY plots can have semilog or log-log axes. Skewed axes are also supported intrinsically by the transformations performed by the digitizer engine.

In order to invoke the digitizer, choose *Data->Digitize from image* from the main toolbar. This will begin the digitization process; first, you will be able to select the image, and then the dialog leads you through the digitization steps as documented in the following sections. In general, there are three primary steps to creating a successful digitization: * select the image * calibrate * select data points Selecting the image is straightforward; the image can either be read from a file on disk, or pasted from the clipboard. Calibration is the means by which you describe where the axes of the plot are located, and how those axes map to real numbers. Selecting of the data points tells the digitizer which locations you would like to have digitized into data.

The layout of the digitizer dialog is shown below:



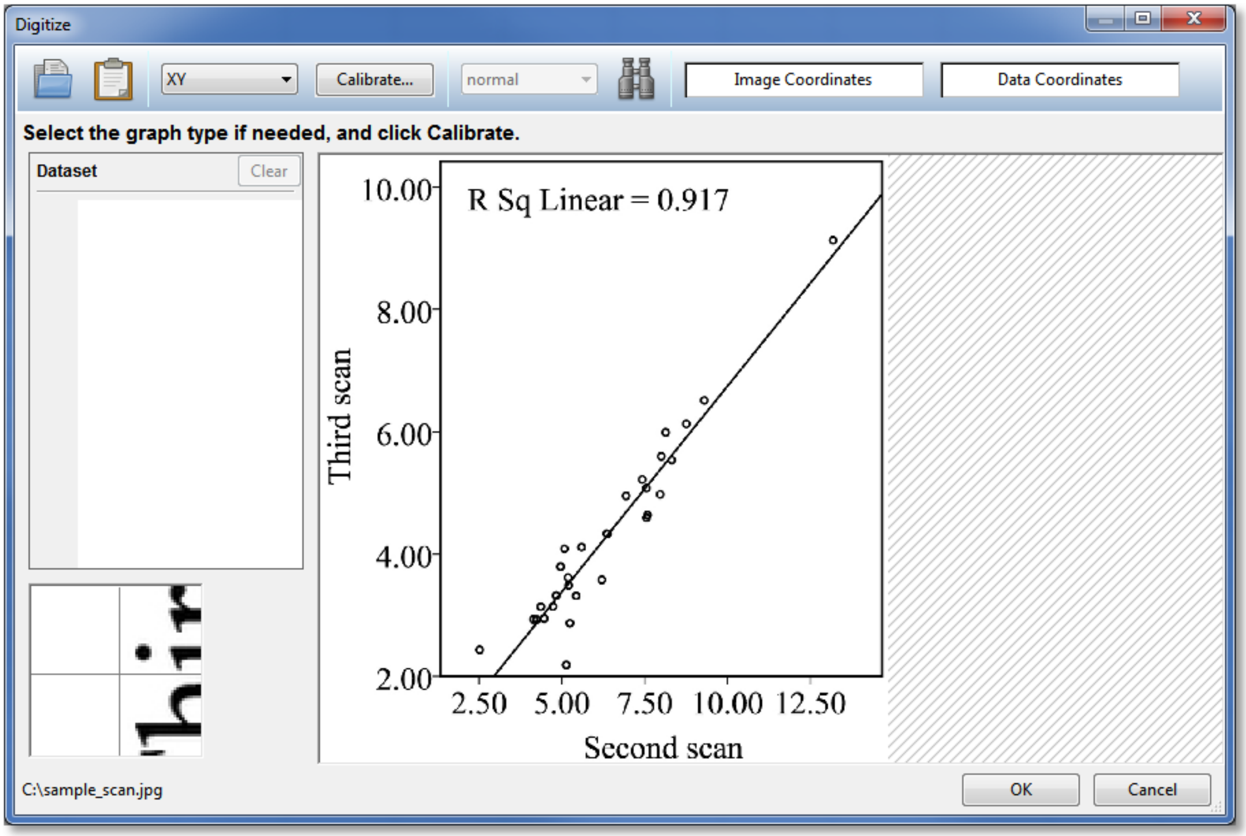
Step 1: select the image (by file or by pasting) that you would like to digitize.

The toolbar is shown across the top, with the appropriate buttons only being enabled at the appropriate times as you move through the digitization process. Beneath the toolbar, an informational message will appear that is intended to remind you what you are working on in the digitizing process. The data values that have been digitized so far are shown in the "Dataset" section. The image, current calibration, and currently selected points are shown in the canvas area in the bottom right. The bottom left area is a magnifier, where the contents of the image and overlay underneath the mouse crosshairs are magnified by a factor of 2X of the 100% image size. Finally, the path to the currently selected image is shown along the bottom of the window, to the left of the OK and Cancel buttons.

To exit the digitizer and save the current dataset, click OK; to exit the digitizer while discarding your current work on the dataset, click Cancel.

11.2 Step 1: Select the image

First, the image can either be read from a file, or pasted from the clipboard. To read from file, click the first button in the toolbar, and select your file. Supported file formats are PNG, BMP, JPEG, GIF, PCX, PPM, TGA, TIFF, WMF, XBM, XPM, and Photoshop 2.0/3.0 PSD files. If the image on the clipboard is desired, either click the Paste icon in the toolbar, or right-click the image and select *Paste Image*.



Select the graph type if needed, and click Calibrate.

C:\sample_scan.jpg

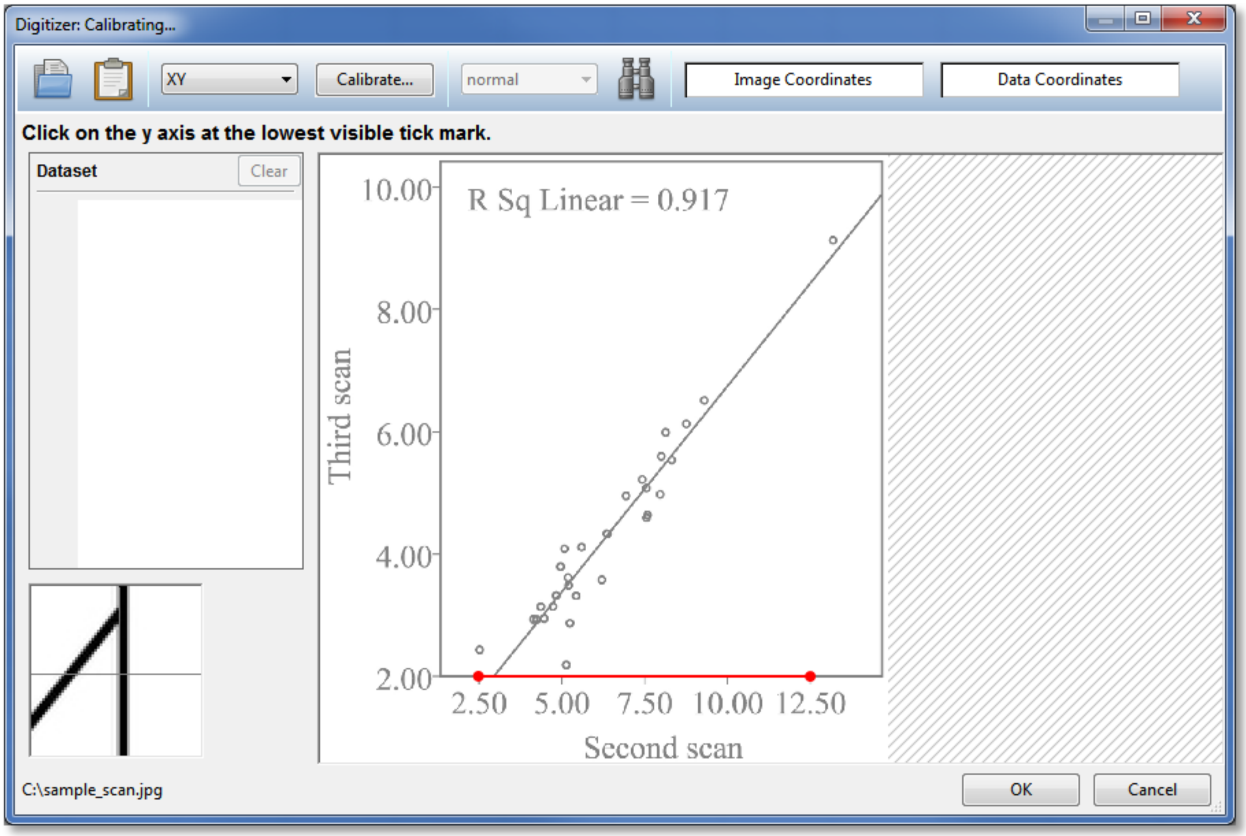
Note that once the image is imported into the digitizer, the file that it was imported from (if it was not pasted) does not need to remain on disk. The image data will be saved as part of the digitizer, and is also persistent in the CurveExpert Professional document.

11.3 Step 2: Calibrate

Once the image is in place, you may select the type of the graph in the image from the toolbar. Most commonly, plots are simple XY plots with linear x and y axes. However, CurveExpert Professional supports digitizing from XY plots with the x and/or y axis on a log scale, and also supports Polar plots.

The calibration may now be performed. Click the *Calibrate…* button, and follow the prompts that are given beneath the toolbar. For an XY plot, you will first click on the leftmost tick mark possible on the x axis, and then click on the rightmost tick mark possible on the x axis. You will soon be asked which data values (in the x direction) these two tick marks correspond to. Also note that you can fine-tune the location of the point you chose by hovering the mouse over the red dot that you just created, and using the keyboard arrows to shift the dot until it adequately lines up with both the axis and tick mark. Use the magnifier to aid in this fine tuning. A red dot will be placed to indicate each of your calibration points, connected by a red line. The canvas will look similar to the screenshot below.

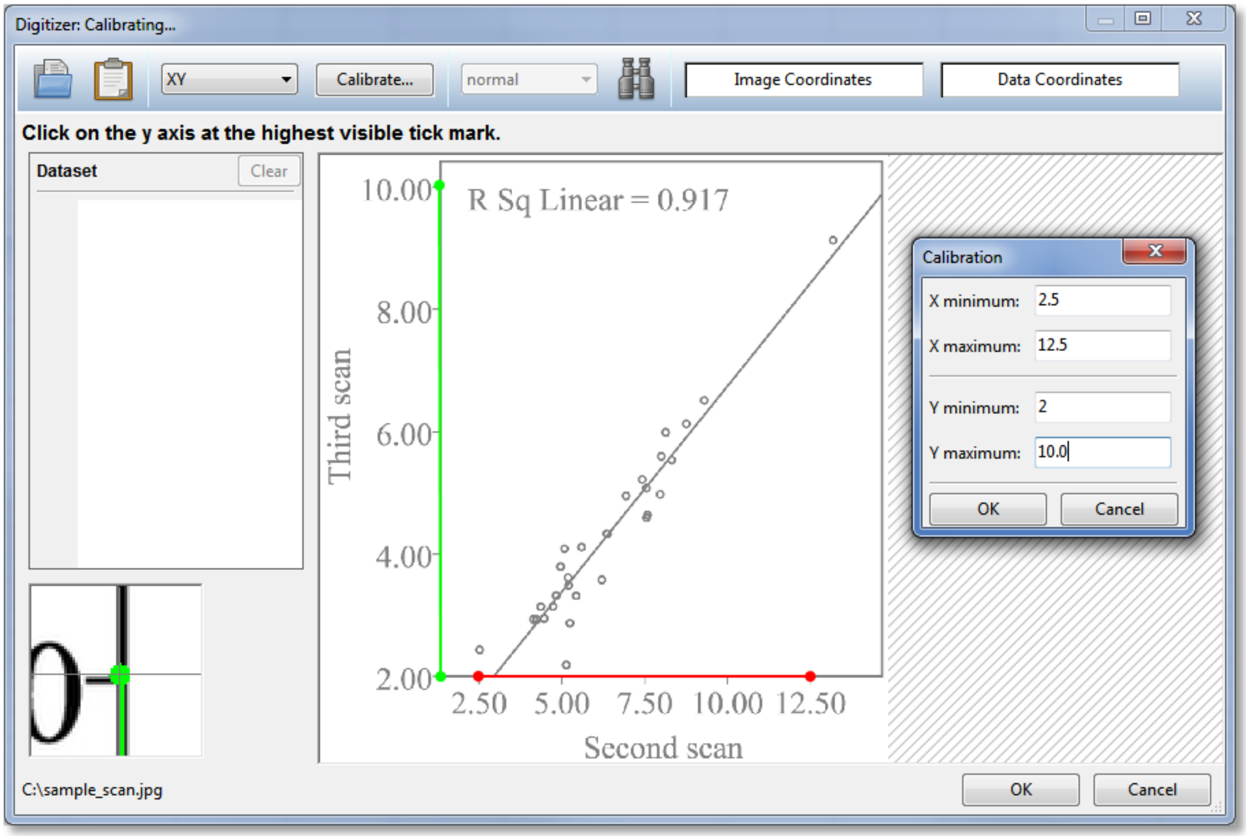
Note: You may fine-tune the calibration points at a later time as well.



Click on the y axis at the lowest visible tick mark.

After specifying the X axis calibration, we will now specify the calibration for the Y axis. In the same manner as the X axis was calibrated, click on the Y axis at the lowest possible tick mark on the Y axis, and then at the highest possible tick mark on the Y axis. The Y axis calibration points will be indicated by green points connected with a green line.

Once the second Y calibration point is selected, the calibration dialog will appear to allow you to input the data values that corresponding to each of the calibration points; this is shown in the screenshot below. For XY plots, you will specify the data values for the two points selected on the X axis, and then for the two points selected on the Y axis. After filling in this information, your calibration data values will display in red or green next to the calibration points in the image.



Click on the y axis at the highest visible tick mark.

Calibration

X minimum: 2.5

X maximum: 12.5

Y minimum: 2

Y maximum: 10.0

Remember that you can still fine-tune the location of the calibration points by pointing at them with the mouse and using the arrow keys to move the calibration point to the desired location. Any data points that you have already created will be automatically adjusted as the calibration changes.

Also, if you desire to replace the current calibration with a completely new one (as opposed to fine-tuning the existing calibration), simply click the *Calibrate…* button again.

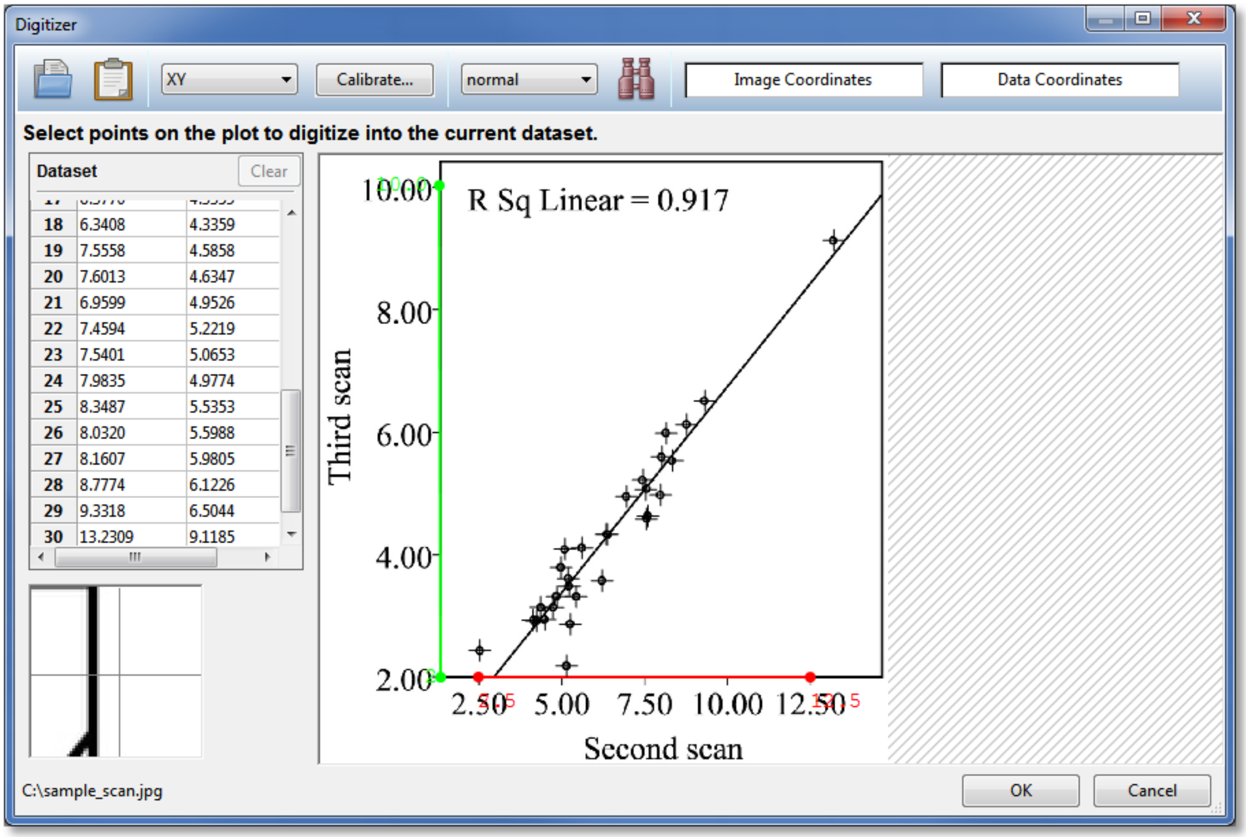
As you move the mouse over the image, you will notice that the *Image Coordinates* and *Data Coordinates* information bars will begin to show information concerning the current mouse location. The image coordinate is the pixel location within the image. The data coordinate is the transformed data value subject to the current calibration (i.e., the mapping between coordinates in the image and the real data coordinates).

11.3.1 Calibrating polar plots

The procedure for calibrating polar plots is slightly different than XY style plots; in the case of a polar plot, you will only be asked to click on the center of the polar plot, and also to click on the zero-degree axis at the tick farthest from the center. The line formed by these two points define the angle on the image that corresponds to zero degrees, and you provide the two data values (radii, in this case) that correspond to the two clicked points. The calibration dialog will query you for the Rmin and Rmax values.

11.4 Step 3: Select data points

Now that the calibration is in place, click on the image to create data points where desired. The most common method is to simply click on the middle of each data point. As you click on the image to create data points, the points are both displayed on the image as a '+' marker, and the data values show in the area of the dialog as well. Like calibration points, you can fine-tune a point's exact location in the image by pointing at its marker in the image (it will turn magenta when active), and using the keyboard arrows. The fine-tuning arrows also work immediately after the mouse click that creates the marker.



Select points on the plot to digitize into the current dataset.

18	6.3408	4.3359
19	7.5558	4.5858
20	7.6013	4.6347
21	6.9599	4.9526
22	7.4594	5.2219
23	7.5401	5.0653
24	7.9835	4.9774
25	8.3487	5.5353
26	8.0320	5.5988
27	8.1607	5.9805
28	8.7774	6.1226
29	9.3318	6.5044
30	13.2309	9.1185

C:\sample_scan.jpg

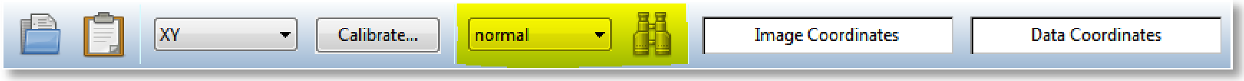
As the mouse passes over a marker, both the marker and its corresponding data value in the dataset are highlighted in magenta.

To delete an existing marker (and by extension the data value), point at it with the mouse, and press the Del key.

If in polar mode, the X value (first column) in the dataset is filled in with the angle in degrees, and the Y value (second column) in the data set is populated with the distance from center corresponding to your Rmin/Rmax calibration.

11.4.1 Point autodetection

In some cases, there are too many data points to manually pick, in these cases, the point autodetection should be utilized. There are two controls in the digitizer toolbar that control how autodetection behaves: the sensitivity setting (normal, aggressive, or conservative), and the autodetect button.



The idea behind the point autodetection is that you provide the digitizer with a small piece of the image that represents the data point (called the template), and then the digitizer searches the image for other instances of that template in the image. Therefore, you will want to provide a data point marker that is well separated from any other elements in the image, such as background text, axes, other data markers, or curves. Also, it helps for accurate selection to maximize the digitizer window (in order to maximize the size of the image canvas), or right-click the image canvas and select a larger zoom value.

During the point autodetection, the toolbar will turn red.

To provide the data point marker template, click the autodetect button on the toolbar. Draw a box around a data point marker, and as soon as the box is finished, the digitizer will start the computation to detect other similar data point markers in the image. These will be marked with a '+' overlay marker just as if you selected them yourself. For best results, try to draw a box tightly around the marker, such that the marker is centered in the box.

Inevitably, the automatically detected markers need to be cleaned up. At this time, you can clean up the results by pointing at markers that aren't quite right, and using the arrow keys to fine tune them. Markers that are missed can be added simply by clicking on the marker. Markers that are completely incorrect can be selected by pointing at them, and pressing the Del key.

11.5 Step 4: Complete the digitization

Simply press *OK* in order to import the digitization results into the CurveExpert Professional spreadsheet. At any time, you can modify the digitization by again selecting *Data->Digitize from image* and modifying your existing digitization of the image.

11.6 Using the image canvas

The image canvas is the primary means by which interaction takes place to define the digitization. You will use the canvas to select points for calibration, pick points for digitization, and interact with those points.

The image is displayed in "Fit to Window" mode by default. In this mode, you will always be able to see the entire image; however, since it is resized to fit the window, the image quality will not be optimal. Use of the magnifier at the bottom left mostly mitigates this shortcoming.

In order to view the image at fixed magnification settings, right click the image canvas and select the desired magnification of 50%, 100%, or 200%. A magnification of 100% shows the image at its native resolution, pixel-for-pixel.

As mentioned several times above, the calibration settings are shown in red and green, overlaying the base image being digitized. Digitized data points are shown with a plus ('+') marker. You may delete a marker by pointing at it and pressing the Del key. Fine-tuning a marker is performed by pointing at a marker, and pressing the arrow keys on the keyboard to move the marker into the desired position.

CHAPTER

TWELVE

GRAPH THEME MANAGER

12.1 Introduction

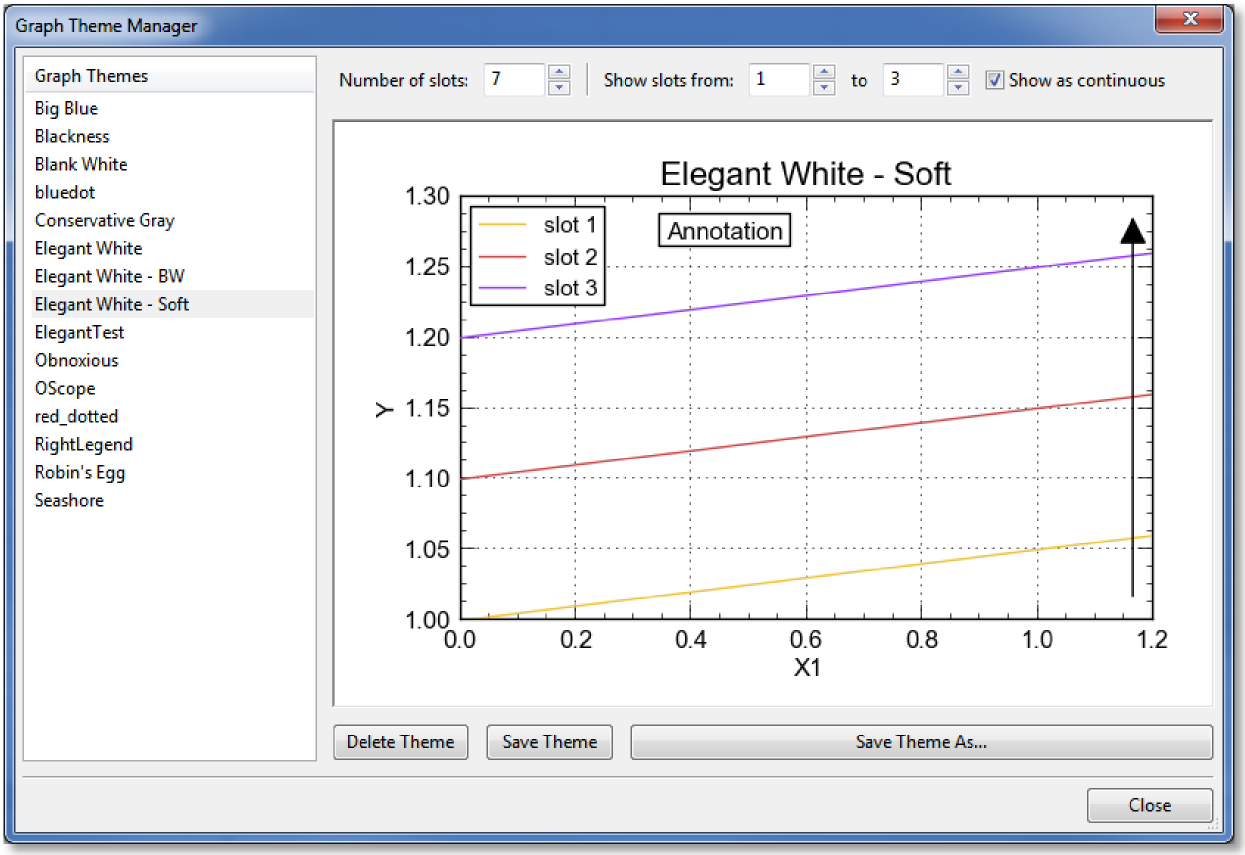
Graph themes allow you to create a custom “look” for your graph, that can be reused on other graphs as appropriate. This can save a lot of time when it comes to setting the properties for your graphs.

The graph themes manager can be accessed via *Tools->Manage Graph Themes*. The theme manager gives you the ability to preview themes (by selecting them), delete themes (select and press the delete button, or right click and select delete), or set a theme as default (right click a theme).

In the manager, you can also create new themes. Right click the graph and select *Properties* in order to modify the properties of the sample graph, and when you achieve the look that you prefer, click the *Save* button. After clicking *Save*, you can name your theme, and it will appear in the themes list to the left.

In the theme manager, there is a sample annotation, legend, arrow, and image. These items should be configured to match the rest of the graph theme if desired; the properties of these four elements are saved with the theme, and are used as default properties for any new elements created.

The theme manager is shown below.



Graph Theme Manager

Graph Themes

- Big Blue
- Blackness
- Blank White
- bluedot
- Conservative Gray
- Elegant White
- Elegant White - BW
- Elegant White - Soft
- ElegantTest
- Obnoxious
- OScope
- red_dotted
- RightLegend
- Robin's Egg
- Seashore

Number of slots: 7

Show slots from: 1 to 3

☑ Show as continuous

12.2 Series properties

12.2.1 What are slots?

CurveExpert Professional keeps track of the properties for each series in your plot by using the concept of “slots”. Each displayed series obtains its properties from slots in the same order that your series appears in the legend. What this means is that if you remove a series from a graph, its slot is vacated. The next time a series is added, the first vacated slot is automatically used to obtain the properties for that series.

When you edit properties for a series, you are really modifying the slot that the series resides. Reordering the series by changing the legend order also reorders the slots.

If a series is added to a graph when all slots are already occupied, the properties assigned to that new series “wraps” back to the front of the slots. For example, if there are three available slots and three series already displayed in the graph, the next added series will take its properties from slot 1. The next added series will take its properties from slot 2, and so forth.

Obviously, the properties awarded to new series via slots can be modified via the regular Graph Properties dialog. The slots only determine the initial properties for a newly added series.

12.2.2 Slots and themes

So, the graph theme saves the properties for each slot. The themes that ship with CurveExpert Professional all have a set of built-in slots for an attractive default set of series properties.

The graph theme manager gives you a means for editing the slots. For your own custom theme, the first parameter to decide on is the number of slots. A small number of slots (say 3) is easier and faster to work with, but does certainly mean that the slots will "wrap" more often as more series are added. Set the number of slots with the **Number of slots** control at the top of the Graph Theme Manager.

The other controls at the top of the Graph Theme Manager are for display within the manager only, just to be able to reasonably see the properties of each slot. The **Show slots from/to** control filters the lines that are shown on the graph canvas, so that you don't have to see all of your n slots at the same time. Simply set the controls appropriately to view the slots that you desire to work on.

The **Show as continuous** control changes each series from a discrete dataset to a continuous function. CurveExpert Professional uses the same slots for both, but slightly modifies the output from the slot to more desirably portray a continuous entity. When a slot's properties are awarded to a continuous function, the marker is automatically removed, and the line style is changed to a visible style if it happens to be set to none. The particular line style selected is determined by the just-disabled marker. If the marker was a square, the line style picked is dashed; if a triangle, the line style picked is dot-dashed; otherwise, the line style is solid. This mapping allows the same slot to be used reasonably regardless of whether or not the plotting entity is discrete or continuous.

Note: It is perfectly valid for a theme to contain a series property slot that is the same as the background color; this can be seen by looking at the "Elegant White" theme; there is a series that has a white line and marker set, which is very hard to see. This is done so that if the user does indeed change the background color to something else, the white series is a valid one to use. CurveExpert Professional will automatically avoid slots that have these types of colors (colors that match the background color of the graph) when awarding a new slot to a newly added series.

12.3 Graph elements

The properties of the first annotation, first arrow, and the legend are saved into a theme, and used when it is applied. Therefore, it is recommended to configure the legend, annotation, and arrow on the displayed plot appropriately to match with your desired color scheme.

12.4 What is contained in a graph theme?

So that the theme is not tied to a specific graph, the following items are *not* saved as a part of a theme:

- title
- axis labels
- axis min, max, or increment
- location of an arrow
- location of an annotation
- content of an annotation

All other parameters are saved, including the complete set of slots for all series on the graph.

CHAPTER

THIRTEEN

CREATING CUSTOM MODELS AND FUNCTIONS

13.1 Introduction

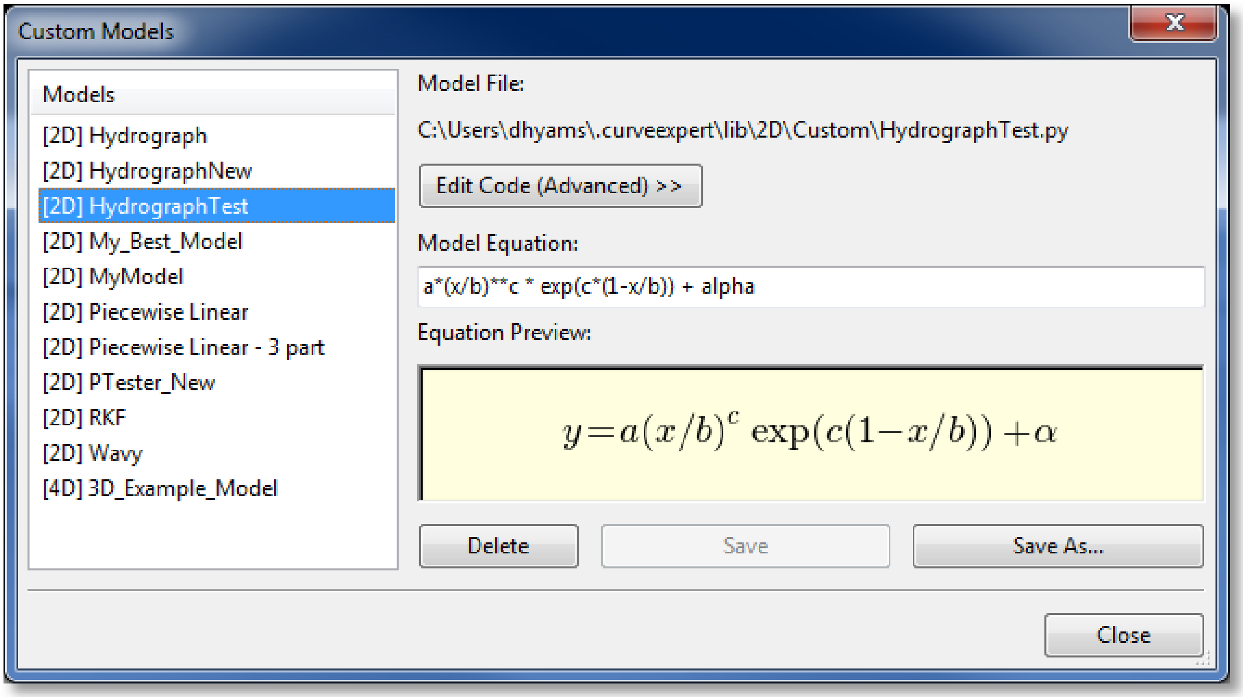
CurveExpert Professional allows very powerful construction of custom models and custom functions. By way of reminder, a “model” in CurveExpert is used to calculate nonlinear regressions, and are dependent on both the indepedent variables (x) and a vector of parameters (a). A “function” is a dependent on the independent variables only.

Creating your custom models/functions can be performed with the choices under the *Tools* menu; there are two classes of interfaces: simple interface, and advanced interface. For most needs, the simple interface suffices, as it allows the user to express a model in a one-line expression form.

Note that you can create custom models and functions at any time; they are saved so that you can use them when you compute nonlinear regressions or functions later, and they appear in the “Custom” part of the nonlinear regression picker (or function picker) when you choose *Calculate->Nonlinear Model Fit* or *Calculate->Function*.

13.2 Custom Models

To create a custom model, select *Tools->Custom Models*. This will show the model manager, which allows you to delete or save custom models.



Model File:

C:\Users\dhyams\.curveexpert\lib\2D\Custom\HydrographTest.py

Model Equation:

```
a*(x/b)**c * exp(c*(1-x/b)) + alpha
```

Equation Preview:

$$y = a(x/b)^c \exp(c(1-x/b)) + \alpha$$

To get started, simply type the expression for your model into the “Model Equation” field. Do not include “y = ” as a part of your equation; just the expression will suffice.

As you type, CurveExpert Professional will guess how to render your equation, even converting Greek letters into the appropriate representation. The Equation Preview (see *Equation Display*) shows how your equation will render in CurveExpert. Note that its guess is not adjustable in the simple interface; if the rendering is undesirable, it can be adjusted by editing “latexequation” attribute for the model directly by clicking the “Edit Code (Advanced)” button. See *Custom Models: Advanced Usage* for more details. Usually the equation rendering is reasonable and correct, but in those cases in which it is not, remember that the rendering does not matter as far as the calculation and mathematics are concerned.

Once your model is written the way that you like, simply hit the *Save* or *Save As* button, depending your intent. Give your model a name, and your model will then appear in the list on the left, signifying that it is now available for use within CurveExpert Professional, when you choose to compute nonlinear regressions.

13.2.1 Mathematical Operations

Obviously, the standard arithmetic operations are supported. Multiplication is performed with a `*`, division with `/`, addition with `+`, and subtraction with `-`. Raising a quantity to a power is performed with either `^` or `**`, with the `**` notation being preferred. Grouping is performed with parentheses `()`.

13.2.2 Reserved Words

In any model, the independent variable is always denoted by x. If you are writing a model that is a function of multiple independent variables, these are denoted as x1, x2, x3, and so on. Each of these special variables are reserved words and should not be used as a parameter name in your model.

All mathematical functions supported by CurveExpert Professional, such as sin, cos, tan, exp, etc., are also reserved words. See *Appendix A: Math Functions* for a complete list of the mathematical functions supported.

Any other name that CurveExpert Professional sees in your model is treated as a parameter. This powerful feature lets you name your parameters any name that you find appropriate, such that the model's syntax carries meaning. Thus, names such as beta, my_long_parameter_name, and b are all legal parameter names. Note also that names are case sensitive; the parameter a is distinct from the parameter A, for example.

13.3 Custom Functions

Custom functions are exactly like custom models, but without any parameters; they are only dependent on the independent variable(s). As such, if you enter a custom function, CurveExpert will flag any unknown name in your equation as an error. Also, custom functions have a default domain setting that can be set to assist the autoscaler, as documented in *Autoscaling*. Otherwise, creating a custom function is exactly the same as creating a custom model, which is documented above.

13.4 Custom Models: Advanced Usage

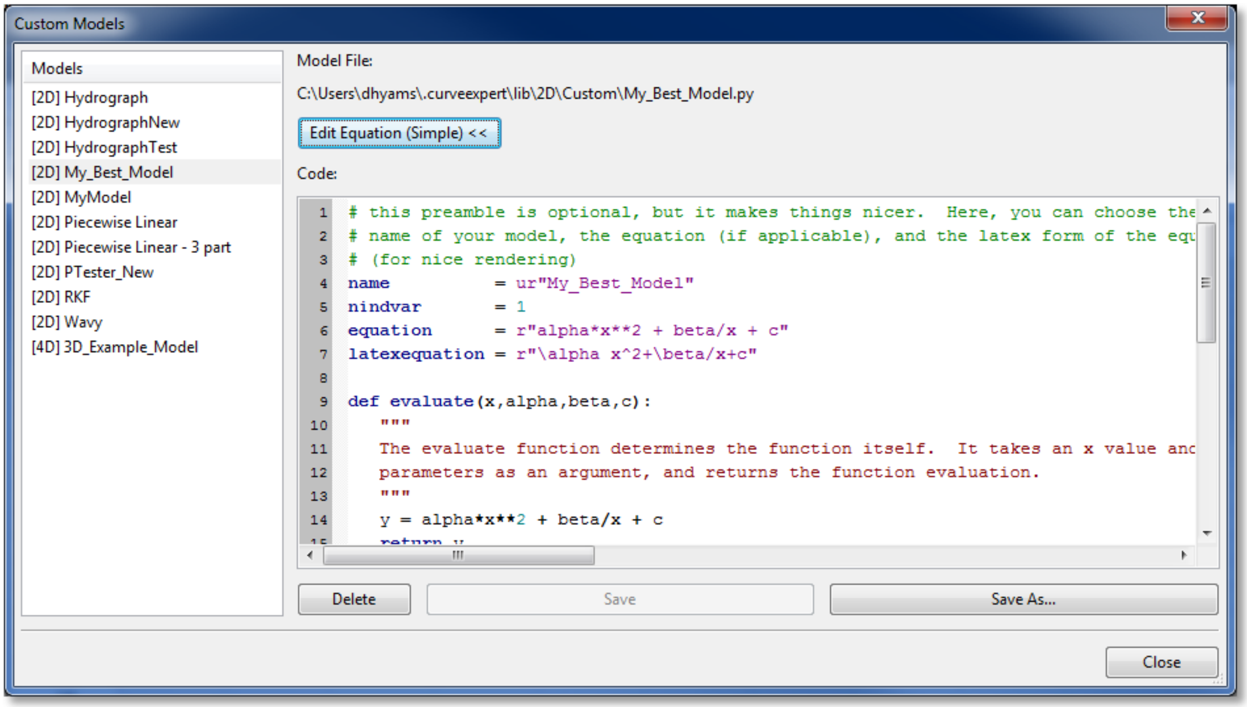
If your needs for a model are more complex than a one-line equation, CurveExpert Professional has the capability of allowing you to program a model as complicated as you like. This capability is accessible via the *Tools->Custom Models* menu choice; then, click "Edit Code" to directly access the code that implements the model.

Rather than trying to reinvent another language for the user to express a model in, CurveExpert uses Python as its base model language. Thus, anything that can be done in Python can be done in your custom models. This gives the user extreme flexibility in the custom models, even to do things like download data from a web server that influences how the model is computed. Here, your creativity is the only limit.

Documenting the entire Python language and its capabilities are obviously outside the scope of this document; for this, refer to http://docs.python.org/release/2.6.6. You do not, however, need to be proficient in Python to be reasonably productive directly editing a custom model. Just follow the examples given as a part of CurveExpert Professional, and for further reference, look in the "lib" and "libf" directories in the core CurveExpert Professional distribution (where the program files are located; see *Where files are located*). These directories contain all of the CurveExpert Professional built-in models and built-in functions.

Note: Python uses indentation to identify code blocks (for example, the code that should execute when an if statement is true). So, if you have an unexplainable syntax error in your model, make sure that your indentation is correct.

Upon selecting the "Edit Code" button, the dialog expands, and the coding window replaces the simpler interface where you can just enter the equation directly (to return to the simpler one-line equation editing, just press *Edit Equation*). The coding window has full syntax-highlighting capability, and you can use standard cut/paste shortcuts ink the window in order to facilitate code construction.



Model File:

C:\Users\dhyams\.curveexpert\lib\2D\Custom\My_Best_Model.py

```
# this preamble is optional, but it makes things nicer.  Here, you can choose the
# name of your model, the equation (if applicable), and the latex form of the equ
# (for nice rendering)
name          = ur"My_Best_Model"
nindvar       = 1
equation      = r"alpha*x**2 + beta/x + c"
latexequation = r"\alpha x^2+\beta/x+c"

def evaluate(x,alpha,beta,c):
    """
    The evaluate function determines the function itself.  It takes an x value and
    parameters as an argument, and returns the function evaluation.
    """
    y = alpha*x**2 + beta/x + c
    return y
```

Note: If you have an error in your custom model, it will have [ERROR] appended to its name for identification. To correct it, simply select it and edit the code in the coding window.

Remember also that you can view the code for your simple models through this interface, to view what is going on "behind the scenes", as well as to adjust easy things like the latexequation, in case you want to change how your model equation is rendered. In fact, when you create a model in the simple interface (one-line equation), CurveExpert Professional just writes the Python code on your behalf.

As in regular Python, any characters following a # sign are a comment, and are highlighted in green in the coding window.

Every model should have at least four items (`name`, `nindvar`, `equation`, `evaluate`), with the other two items (`latexequation` and `initialize`) being optional.

13.4.1 `name`

This is a variable to define the name of your model, which is the name used throughout CurveExpert Professional to refer to it. The name of your model is also used to generate the filename in which it is saved, the full path of which is shown above the coding window. The type of this variable is a string, and should be enclosed in quotes.

13.4.2 `nindvar`

This is a variable to define the number of independent variables that this model expects. Most of the time, this value will be 1, but can obviously be higher depending on how many independent variables you want your model to apply to.

13.4.3 `equation`

This is a variable that is a text representation of the equation. It is for information only, and is displayed by CurveExpert Professional in different areas of the software.

13.4.4 `latexequation`

This is a variable that is a mathtext representation of the equation (see mathtext and *Appendix B: Math Symbol Reference* for how to write in mathtext). In short, this is a TeX-like version of your equation that will be used every time rendering of your equation is required in CurveExpert Professional. This variable is optional, and if not present, CurveExpert Professional will attempt to guess it by looking at the "equation" variable above.

13.4.5 `evaluate(x,a,b,c,d,...)`

This is the most important part of your model; `evaluate()` is a function that is called whenever your model is evaluated with the independent variables and a set of parameters. It is this function that is called in the course of the parameter optimization process, and it is called in order to draw your plot.

The `evaluate` function must take `x` as its first argument. This `x` is the vector of independent variables, and can be multiple columns if there is more than one independent variable. Following `x` is a comma separated list of parameters, which can have any name. The job of the evaluation function is to take these pieces of data and compute an output vector y, returning it as the result of the evaluation.

An example of an evaluate function is the following (with comments removed for brevity):

```python
def evaluate(x,a,b,c):
    return exp(a + b/x + c*log(x))
```

Any other Python function (denoted by the **def** keyword) other than `evaluate` and `initialize` are ignored. This means that you can create your own Python functions and call them from your `evaluate` function as you wish.

13.4.6 `initialize(x,y)`

The initialize function is optional, but can greatly aid in the convergence of a model's parameters in the process of nonlinear regression. `initialize()` is a function that takes the independent variables as the first argument, and the dependent variables as the second (both of these quantities are based on the data currently held in the spreadsheet). With this data, you can perform any analysis necessary to compute good initial guess parameters to start with. The parameters can be returned as anything that can be translated into a Python array, but most commonly are returned simply as a comma separated list.

Good examples of using `initialize()` can be found in CurveExpert Professional's built-in models, by way of example (the built-in models are located in the "lib" directory where the program files are; see *Where files are located*). A common approach there is to transform the model into a linear one, solve a linear least squares system, and use the output from that in order to initialization the nonlinear regression.

An example of an `initialize` function is the following (with comments removed for brevity):

```
def initialize(x,y):
   try:
       n = len(x)
       A = ones((n,3))
       A[:,1] = 1.0/x
       A[:,2] = log(x)
       B = log(y)*log(x)
       soln,residues,rank,s = lstsq_solve(A,B)
       a,b,c = soln  # use Python's unpacking to extract a,b,c from the solution
   except:
       a,b,c = 1.0,1.0,1.0

   return a,b,c
```

In this case, an attempt is made to transform the model into a linear least squares system and a solve of that system is tried. If that fails (for example, if there is a zero in the data, the last two columns of `A` cannot be computed), the solution is just initialized to three ones.

13.5 Custom Functions: Advanced Usage

Again, custom functions are the same as custom models except that they have no free parameters. The evaluate() function takes only `x` as an argument, and there is no need for an initialize() function. The advanced interface for editing and creating custom functions is otherwise exactly the same as the one for custom models described above. An example evaluate function is below:

```
def evaluate(x):
    return exp(x)*sin(x)
```

For 3D and higher functions, like custom models above, x is a multicolumn vector. To extract out each column for convenience, an example is given below:

```
def evaluate(x):
   x1 = x[:,0]
   x2 = x[:,1]
   return x1*sin(x2)
```

CHAPTER

FOURTEEN

PLUGINS

Note: Plugins are only active in a licensed version of CurveExpert Professional, or while CurveExpert Professional is within its trial period.

14.1 What is a plugin?

A plugin is a tool that can be used to automate tasks within CurveExpert Professional. It is a Python script with a few specific functions (the most important one is the `run()` function, which is executed whenever the plugin is run from within CurveExpert Professional), and in these functions, you can perform whatever task needs to be accomplished. Some common Python modules are already available (such as numpy, scipy, wxPython, and obviously all of the standard modules that ship with Python) for your use, or you can import the `api` module (see *Appendix E: Plugin API*) in order to control CurveExpert Professional's behavior, such as calculating a particular result that you desire. In addition, you can import any Python modules that you have locally installed.

Plugins are extremely flexible; one can perform practically any task with them. For example, one could easily write a plugin that downloads data from a remote web server, performs a nonlinear regression on it, and writes out the result of that nonlinear regression.

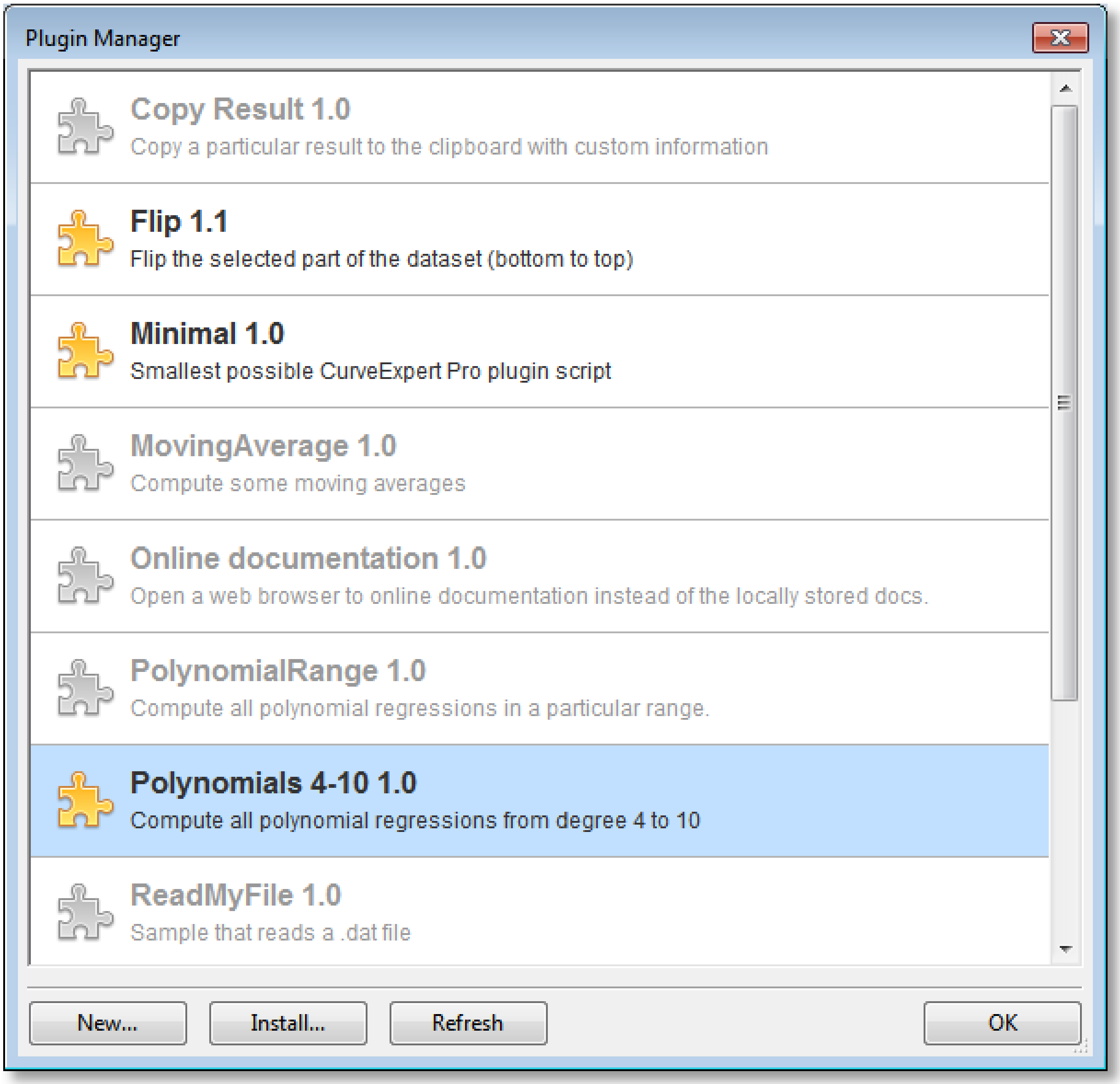
14.2 Running a Plugin

Once a plugin is installed, it can be executed by selecting it from the *Plugins* menu, or by double clicking it in the *Plugins pane*. Running the plugin calls its `run()` function. If there is no *Plugins* menu, this means that there are no plugins currently enabled; the menu will automatically appear as soon as runnable plugins are present. See *Managing Plugins* for how to enable and/or install a plugin.

In order to run a plugin from the command line, please see *Running Plugins from the Command Line* for how to perform this properly.

14.3 Managing Plugins

To access the plugins manager, select **Tools->Manage Plugins**. This displays the dialog below:



Copy Result 1.0
Copy a particular result to the clipboard with custom information

Flip 1.1
Flip the selected part of the dataset (bottom to top)

Minimal 1.0
Smallest possible CurveExpert Pro plugin script

MovingAverage 1.0
Compute some moving averages

Online documentation 1.0
Open a web browser to online documentation instead of the locally stored docs.

PolynomialRange 1.0
Compute all polynomial regressions in a particular range.

Polynomials 4-10 1.0
Compute all polynomial regressions from degree 4 to 10

ReadMyFile 1.0
Sample that reads a .dat file

All plugins that are installed are listed here; out of the set of installed plugins, enabled plugins are shown in color (disabled plugins are dimmed).

14.3.1 Enabling/disabling a plugin

To enable or disable a plugin, right click the desired plugin, and select *Enable* or *Disable* as appropriate. Disabling a plugin cause it to be dimmed in the manager, and will not appear in the **Plugins** menu or in the Plugins pane in CurveExpert Professional.

14.3.2 Creating a new plugin

Create a new plugin by clicking the **New** button at the bottom of the manager, and give the new plugin a name. This will create a skeleton plugin, with the necessary functions and attributes already present, for you to begin to work with. To edit your new plugin, either double click it in the manager, or right click it and select **Edit**.

14.3.3 Cloning a plugin

Sometimes, it's easiest to start from an existing example. To clone any plugin, right click it and select **Clone**. This will create another copy of a plugin. To edit your newly cloned plugin, either double click it in the manager, or right click it and select **Edit**.

14.3.4 Installing a plugin

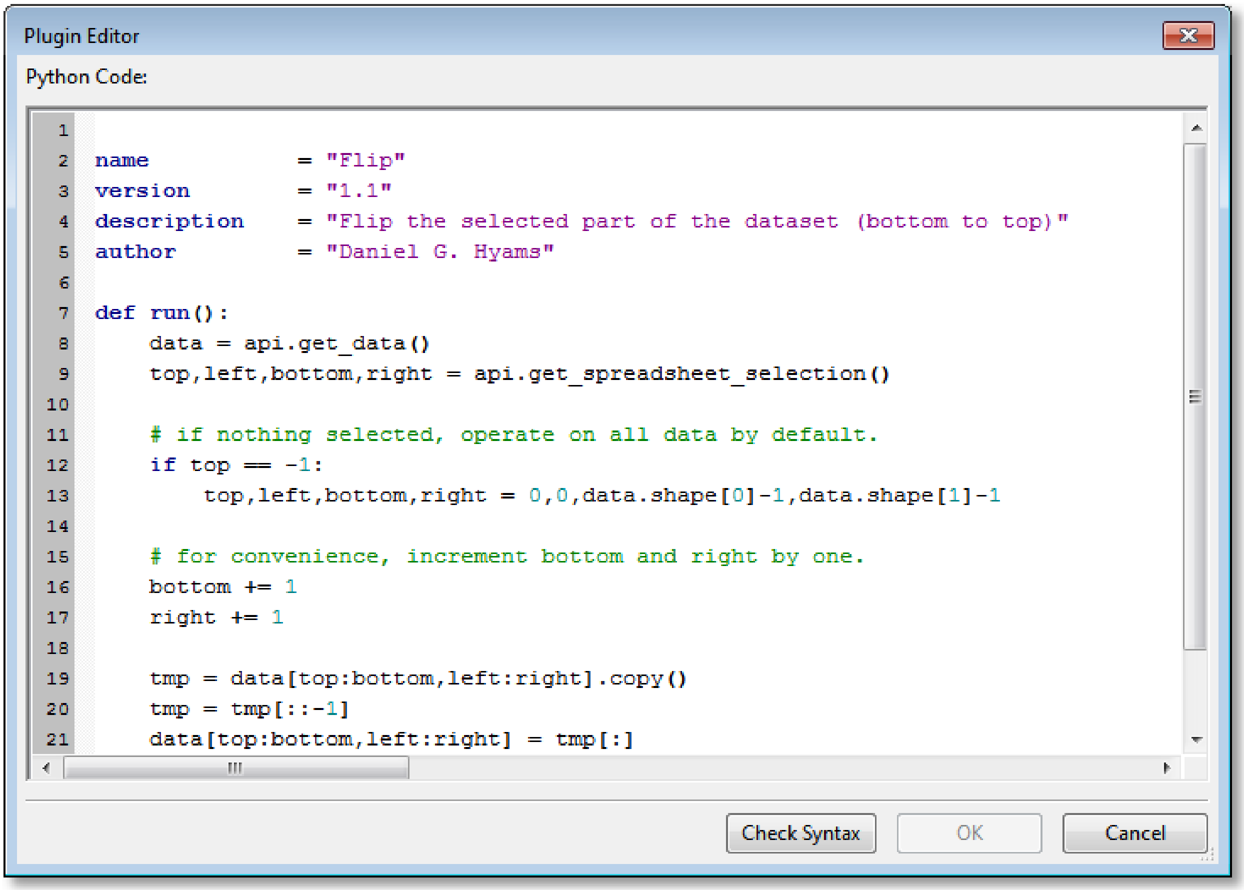
If your plugin file was created outside of CurveExpert Professional, or sent to you by a colleague, it must be installed before use. Click **Install** from within the manager, browse for the plugin file (either .py or.pyc), and open it. This will install the plugin, and it will be available just like any other. Installing essentially copies the plugin file from its current location to $HOME/.curveexpert/plugins.

14.3.5 Uninstalling a plugin

To get rid of a plugin entirely (when you know that you don't want it any more; otherwise, just disable it), right click the desired plugin and select **Uninstall**. Uninstalling deletes the plugin file from $HOME/.curveexpert/plugins (or $HOME/.curveexpert/plugins-disabled).

14.4 Writing/Editing Plugins

To edit a plugin, either double click it in the manager, or right click it and select **Edit**. This will display a Python code editor as follows:



```

name            = "Flip"
version         = "1.1"
description     = "Flip the selected part of the dataset (bottom to top)"
author          = "Daniel G. Hyams"

def run():
    data = api.get_data()
    top,left,bottom,right = api.get_spreadsheet_selection()

    # if nothing selected, operate on all data by default.
    if top == -1:
        top,left,bottom,right = 0,0,data.shape[0]-1,data.shape[1]-1

    # for convenience, increment bottom and right by one.
    bottom += 1
    right += 1

    tmp = data[top:bottom,left:right].copy()
    tmp = tmp[::-1]
    data[top:bottom,left:right] = tmp[:]
```

It should be noted that the CurveExpert Professional API module is always automatically imported for you, and is available directly in the `api` namespace. Documentation for all available API calls is given in *Appendix E: Plugin API*. Importing of other desired Python modules is easy, using the standard Python import statement.

At any point during the editing, the **Check Syntax** button can be clicked, which performs a test-import of the plugin. If there is an error, the offending line and traceback is shown at the bottom of the dialog. The editor also will not allow the plugin to be saved until it has gone through a syntax check, once editing of the plugin is detected.

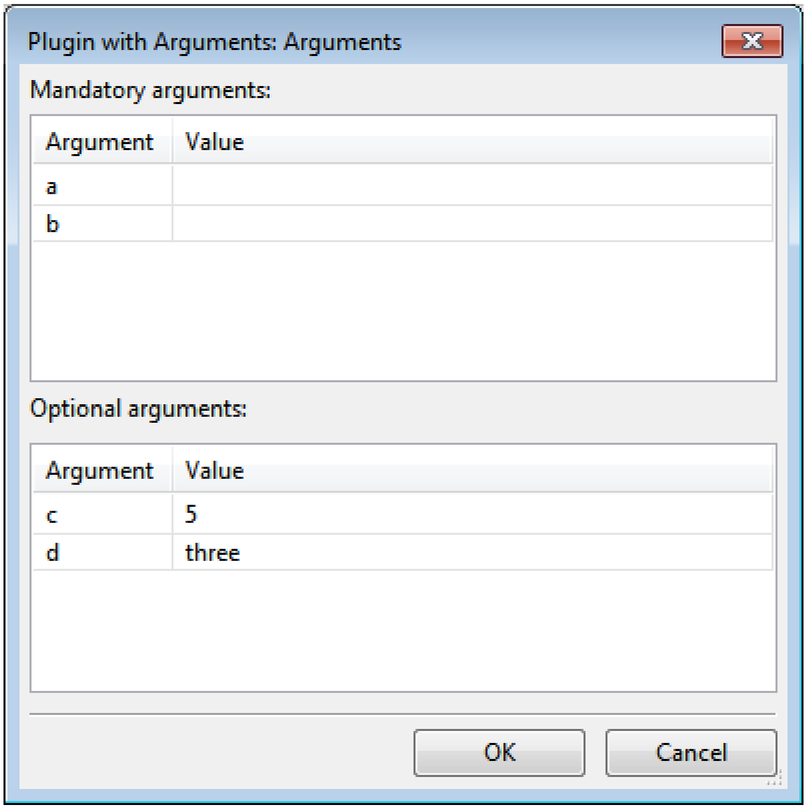
For complicated plugins, it is probably more desireable to perform the development in a real development tool, such as Eclipse or Netbeans. In this case, simply edit the plugin file and save it somewhere on disk; then install the plugin as documented above.

Your plugin may contain any number of functions (defined by the `def` keyword); however, there are several special functions that are reserved for a special purpose. These functions are `run`, `initialize`, and the set of hook functions documented below. Further, there are several module-global attributes that hold special meaning for a plugin; `name`, `version`, `description`, and `author`.

14.4.1 The run() function

Every plugin must have a `run()` function; this is the function that is executed any time the user "picks" your plugin from the **Plugins** menu.

The `run()` function can optionally have arguments. Arguments that are mandatory have no default value, and arguments that are optional do (see the "Plugin with Arguments" example plugin). If your `run()` function has arguments, the following dialog will open when the plugin is run:



Plugin with Arguments: Arguments

Mandatory arguments:

Argument	Value
a	
b	

Optional arguments:

Argument	Value
c	5
d	three

The user is given an opportunity to fill in the arguments, and your script is free to use these arguments in any way appropriate. Note that regardless of the type of your arguments given in your `run()` function, **all arguments are passed to your script as strings**. Thus, you (as the plugin writer) are responsible for checking the arguments for validity and converting them to the correct type before using them.

14.4.2 The initialize function

The `initialize()` function is called whenever the plugin is added to the *Plugins* menu. This is a good place to perform one-time initialization of data that might be needed later for the `run()` function. Note that the `initialize()` function can be called multiple times in one session, if the user enables/disables the plugin (upon each enabling, the `initialize()` function is called). This function takes no arguments.

If there is no initialization needed, this function may be omitted from your plugin.

14.4.3 Metadata

There are four attributes that every plugin should have: name, version, description, and author. The most important of these is name, which obviously defines the name of the plugin. This name is the one that appears in the Plugins menu that the user interacts with. The other attributes are informational metadata that is displayed in the Plugins manager only.

14.4.4 Hook functions

Plugins can respond to several high-level events that occur during the execution of CurveExpert Professional. To respond to an event, all you have to do is make sure that you have a function present in your plugin that services the event. These functions have specified names, as documented below.

on_writefile(file_name, userdata)

The on_writefile hook is called any time that a CXP file is being written. This hook gives the plugin author an opportunity to store information persistently in a CXP file. An example follows:

```
def on_writefile(file_name, userdata):
    api.message("A file is being written: %s"%file_name)
    userdata['my_special_plugin'] = [1,2,3,4,5]
```

The userdata parameter is a dictionary that the plugin author can add information to, and the information added to this dictionary is stored in a CXP file. Note that this dictionary can be used by other plugins as well, so it is highly recommended to store all plugin-specific information under a unique key that is specific to the plugin.

The only requirement for items placed in the userdata dictionary is that they are picklable.

on_readfile(file_name, userdata)

The on_readfile hook is called any time that a CXP file is read. This hook gives the plugin author an opportunity to read any of the plugin-specific data stored in the on_writefile hook. An example follows:

```
def on_readfile(file_name, userdata):
    api.message("A file is being read: %s"%file_name)
    something_i_saved = userdata['my_special_plugin']
```

The userdata parameter is a dictionary that the plugin author can read information from, that was saved via the plugin in the on_writefile hook.

on_datachanged()

The on_datachanged hook is called any time that the dataset is changed. An example follows:

```
def on_datachanged():
   api.message("Detected a change in the data.")
```

on_resultsadded(list_of_result_objects)

The on_resultsadded hook is called any time that results are added. The hook should take a single argument, which is the list of added results. An example follows:

```
def on_resultsadded(list_of_results):
   for result in list_of_results:
      do_something(result)
```

14.5 Examples

Example plugins are given in the resources/default_files/plugins directory of the CurveExpert Professional distribution. These plugins are installed automatically by default, but are initially disabled. To see these examples, open the Plugins manager, and double click one of them. These examples will get you started with the concept of writing plugins for CurveExpert Professional. Remember that anything that you want can be done in a plugin, because they are essentially Python codes that can be as complex as need be to accomplish a task.

14.6 Miscellaneous

At the time of this writing, the version of embedded Python with CurveExpert Professional is version 2.6.6. As such, the Python language features offered by this application are identical to this Python version.

It can be convenient for different plugins to share the same code. To support this, you have three choices:

- place the code that you want to share in a Python file, and put the file in $HOME/.curveexpert/plugins-common. This path is always searched when importing Python modules within plugins.
- place the code that you want to share in a folder/directory of your choice, and set the environment variable AUX_MODULE_PATH to this directory before running CurveExpert Professional.
- place the code that you want to share in a folder/directory of your choice, and set the environment variable PYTHONPATH to this directory before running CurveExpert Professional.

Note that both of the environment variables PYTHONPATH and AUX_MODULE_PATH can be set to multiple paths, and all of these paths will be searched. Under Windows, separate each path with a semicolon; otherwise separate with a colon. For example, under Linux, you might do (depending on your shell):

```
% export AUX_MODULE_PATH=/my/first/path:/my/second/path
```

14.7 Running Plugins from the Command Line

As of version 2.6.5, CurveExpert Professional has the ability to run a user-defined plugin from the command line, with no need to open the graphical interface. This feature allows users to make use of the computational features of the software in automated scripts.

Note: Since there is no user interface present, none of the plugin API's that require a window handle, or operate on the UI, can be used.

To run a plugin from the command line, execute the following on a Windows platform:

```
cepro.exe --plugin <path to your plugin > --parg <argument 1> --parg <argument 2> ...   [Windows]
```

On a Linux platform:

```
./cepro.sh --plugin <path to your plugin > --parg <argument 1> --parg <argument 2> ...   [Linux]
```

and on a macOS platform:

```
open /Applications/CurveExpertPro.app  --args --plugin <path to your plugin>  --parg <argument 1> .
↪..
```

for example:

```
cepro.exe --plugin c:\my_plugins\test_plugin.py --parg c:\my_data
./cepro.sh --plugin /home/user/my_plugins/test_plugin.py --parg /home/user/my_data
open /Applications/CurveExpertPro.app  --args --plugin /Users/me/my_plugins/test_plugin.py  --parg␣
↪~/test_data
```

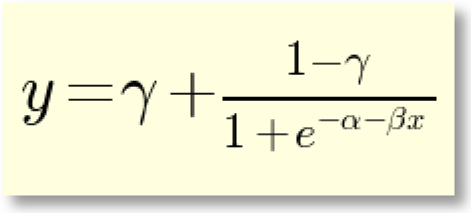
Log messages from CurveExpert Pro as it runs your custom plugin will be available in the standard log file, and is useful for debugging. See *Where files are located* for there the log files are located on your particular system.

Due to technical reasons that cannot be adequately explained briefly, under Windows and macOS CurveExpert Pro always runs asynchronously. This means that the application will return almost instantaneously, even though the work is not complete yet. If this is a problem, you should use a synchronization mechanism in your plugin, the simplest of which is a file that you will create or update when the plugin's work is complete. This way, your driving script knows when it can continue on, by monitoring the existence of this "sync" file. Under Linux, CurveExpert Pro runs synchronously as expected.

CHAPTER FIFTEEN

EQUATION DISPLAY

In various dialogs throughout CurveExpert Professional, the equation for a model or function is displayed. This equation is the "latexequation" attribute of the model, which can be set explicitly via *Custom Models: Advanced Usage*. The contents of this attribute is mathtext that describes how the equation should be rendered. An example of an equation display is below:



$$y=\gamma+\frac{1-\gamma}{1+e^{-\alpha-\beta x}}$$

The equation display has a couple of small, convenient features. If the mouse is right-clicked on the display, you will be given the option to copy the image onto the clipboard (so that you can use this in some other application). You can also zoom in or out on the equation, in order to make it temporarily larger or smaller than its default size.

Note: When copying the equation, it is copied using a white background.

CHAPTER

SIXTEEN

WRITING DATA

16.1 Introduction

After performing some analysis, a typical task is to write data, in some form, to disk for later retrieval. CurveExpert Professional supports several ways to extract data and graphs to a file or the clipboard.

16.2 Writing a CurveExpert (CXP) file

If you want to save all of your data, results, graphs, and notes to disk, you can choose *File->Save* or *File->Save As* in order to write a CXP file to disk. These files can be read in later using *File->Open*, obviously. CXP files are portable across platforms (Windows, Mac, Linux) and save all possible information about your current session. CXP files are also interchangeable between CurveExpert Professional and CurveExpert Basic.

Also, if a particular model or function is saved as a result in a CXP file and it is not present on the system reading the file, the model/function will automatically be created (on the system reading the file) on your behalf. This newly created model/function will then be available for use in the "Imported" family, and will appear appropriately in the Nonlinear model or function pickers.

16.3 Exporting a dataset to text file

To save your dataset into a text file, choose *File->Export*. The dataset will be exported using the default delimiter chosen in *Edit->Preferences->General->Default delimiter*.

16.4 Writing a graph to a picture file

To save a particular graph to a file, right click it and select *Save*. Any graph can be saved wherever it appears via this procedure. Alternatively, click the *Save* icon in the graph toolbar (if present). See *Saving graphs* for more details on supported file formats.

The same graph can be copied to the clipboard by right clicking and selecting *Copy*.

16.5 Printing

All graphs in the Graphs and Data pane can be printed by choosing *File->Print* or *File->Print Preview*. To print an individual graph, right click it and select *Print* or *Print Preview*.

If the margins or paper orientation is not correct, you can set these in the *File->Page Setup* dialog.

CHAPTER

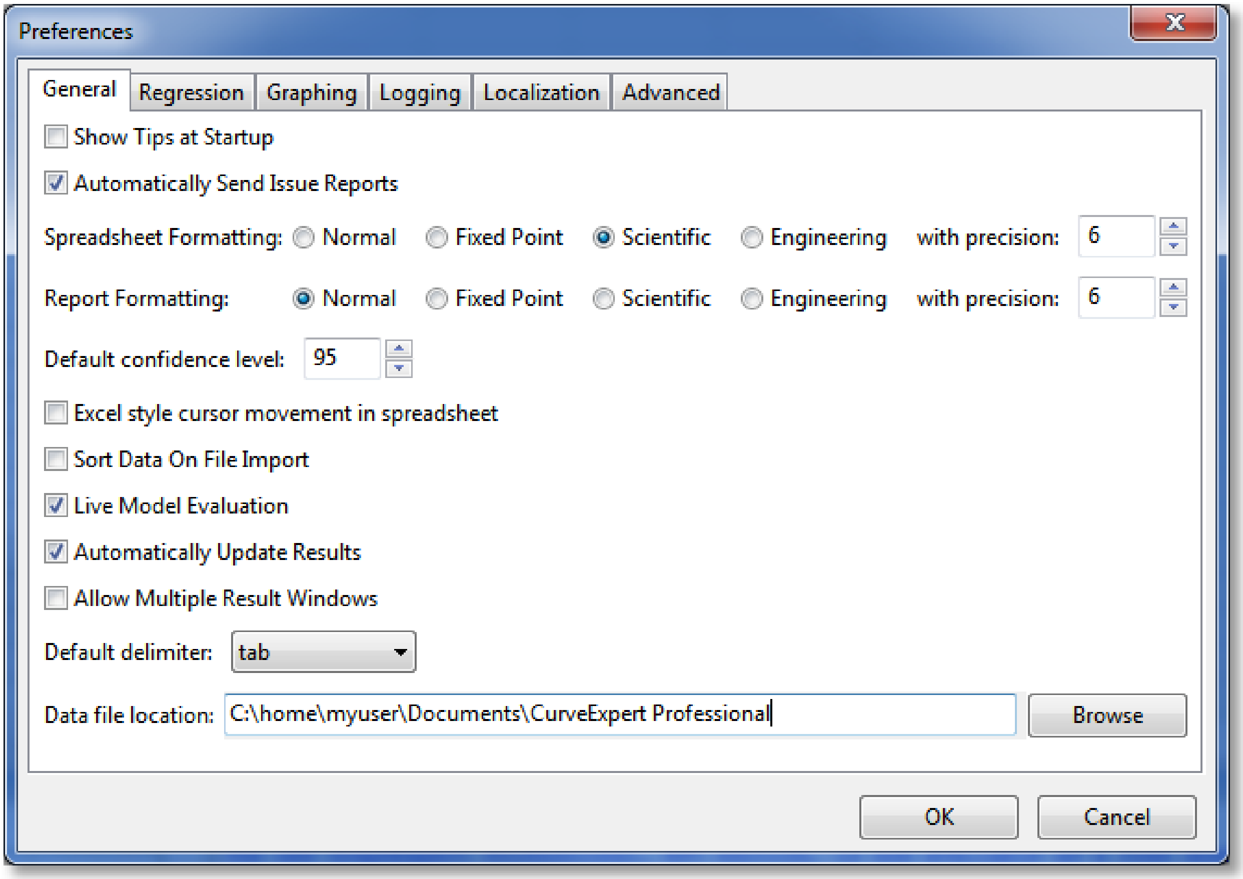
SEVENTEEN

APPLICATION PREFERENCES

17.1 Introduction

Application preferences can be changed by choosing *Edit->Preferences* from the menu.

17.2 General Preferences



□ Show Tips at Startup

☑ Automatically Send Issue Reports

Spreadsheet Formatting: ○ Normal ○ Fixed Point ● Scientific ○ Engineering with precision: 6

Report Formatting: ● Normal ○ Fixed Point ○ Scientific ○ Engineering with precision: 6

Default confidence level: 95

□ Excel style cursor movement in spreadsheet

□ Sort Data On File Import

☑ Live Model Evaluation

☑ Automatically Update Results

□ Allow Multiple Result Windows

Default delimiter: tab

Data file location: C:\home\myuser\Documents\CurveExpert Professional

General settings apply to general items of interest in CurveExpert Professional.

17.2.1 Show Tips at Startup

When the software starts, a helpful tip can be shown. However, if you find this either unhelpful or annoying, it may be turned off (or back on) by selecting this checkbox.

17.2.2 Automatically send issue reports

CurveExpert Professional has the ability to automatically send an issue report should something go wrong with the software. The objective of doing is to find and correct problems in the most timely manner possible.

An issue report, when and if one is sent, contains no personally identifying information or any of your data, and the report itself is submitted anonymously. However, if you would like to opt-out of this behavior, uncheck this box.

17.2.3 Spreadsheet Formatting

The *spreadsheet formatting* setting allows you to set your preference for how data is displayed, by default, in the spreadsheet. The data may be displayed in a fixed point, scientific, or engineering format, with a precision (number of digits after the decimal) as specified in the *with precision* field. This default number formatting can always be overridden on a per-cell basis by selecting the cells in the spreadsheet, right clicking, and selecting *Format Numbers* as documented in *Number Formatting*.

The engineering format is the same as scientific, except that the exponent is always a multiple of three.

17.2.4 Report Formatting

CurveExpert Professional displays formatted reports for things such as the result details and comparing regressions. When displaying numerical data in these reports, a particular formatting must be used for this data. The style can be chosen in this application preference.

The data may be displayed in a fixed point, scientific, or engineering format, with a precision (number of digits after the decimal) as specified in the *with precision* field.

The engineering format is the same as scientific, except that the exponent is always a multiple of three.

Note that this option is just a default, and you will be given the chance to change the formatting when the report is generated.

17.2.5 Default confidence level

In result details and in displaying confidence/prediction bands, a certain confidence level is needed. The default confidence level set in the preferences here is the starting level set throughout the application in these situations. You will be given the opportunity to change the confidence level on a result-by-result basis.

17.2.6 Excel style cursor movement in spreadsheet

By default, both the TAB and ENTER keys are treated as carraige returns in CurveExpert Professional. In other words, the cursor is moved to the left until it bumps into the right side of the dataset, in which case the cursor is moved to the beginning of the next row.

If you would rather that the cursor go right on a TAB (which is the same as the default TAB behavior), and down on an ENTER, select this box. Note that CurveExpert Professional will still keep your cursor within the bounds of the dataset.

17.2.7 Sort Data On File Import

Normally, CurveExpert Professional reads your data directly from file, accounts for the settings in the File Import window while importing, and places it in the spreadsheet. If you would also like to automatically sort the data against the first column, check this box.

This capability is most useful if you frequently use splines to fit data; the data must be sorted for a spline to compute correctly.

17.2.8 Live Model Evaluation

In the result details window, an evaluator follows the location of your mouse to inform you of the x/y location of the mouse, the model/function evaluated at the pointed-to x coordinate, and the first derivative of the model/function. If this is not desired or is too slow, uncheck this box.

There also an intermediate state for this checkbox; if set to intermediate, the live evaluator will activate when the left mouse button is pressed only (by default).

17.2.9 Automatically Update Results

If you have already calculated results and then change the data, the results become invalid (see *Working with Results*). This state is indicated by a yellow caution badge placed on the results in the Results pane. To update these results, you can select some (or all), right click, and select *Update*. To automatically update results without having to do it yourself, check this box. As soon as a change in the data is detected, all results will be recomputed.

17.2.10 Allow Multiple Result Windows

Allow the opening of multiple result windows at one time, instead of allowing only one. While convenient, it can lead to a window management problem if you open too many at once.

Note: This feature is disabled on OSX (Mac) platforms.

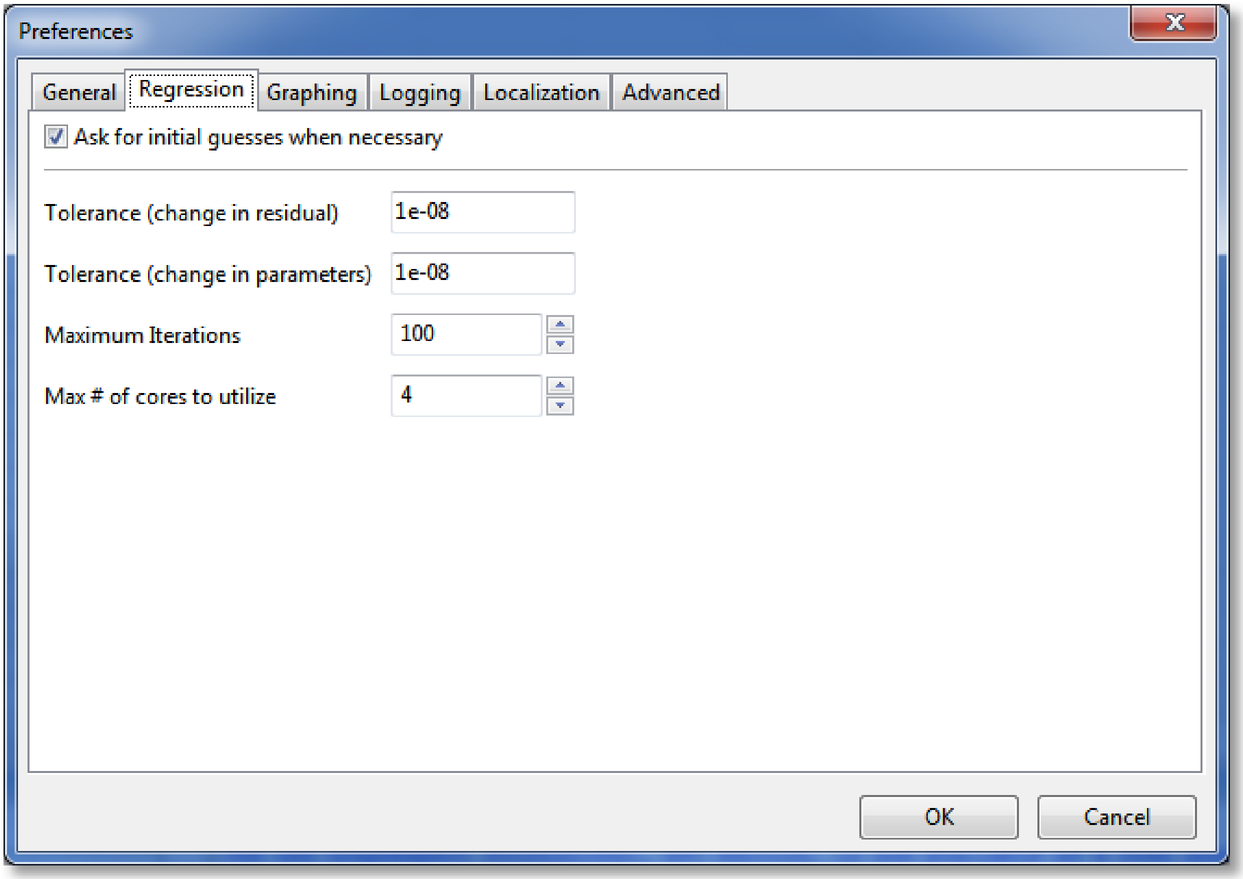
17.2.11 Default delimiter

In several places in the software, a delimiter is used to separate items in the same row. For example, in copying data to the clipoboard, or in generating a table, a delimiter between data items must be selected. If you have a favorite delimiter, set it here.

17.2.12 Data file location

Default directory where data files are stored. Initially, the data file location is set to $DOCUMENTS/CurveExpert Professional, and some example data and example documents are placed there at install time.

17.3 Regression Preferences



☑ Ask for initial guesses when necessary

Tolerance (change in residual) 1e-08

Tolerance (change in parameters) 1e-08

Maximum Iterations 100

Max # of cores to utilize 4

17.3.1 Ask for initial guesses when necessary

At times, when calculating a list of nonlinear regressions at once, the calculation of the initial guess will fail for some (for various numerical reasons; a simple example is when the natural log is taken in the process of calculating the initial conditions, and there is a zero or negative in the dataset). When this happens, CurveExpert Professional will prompt you to ask if you want to attempt initial guesses manually, and give a list of the fits that failed in this manner.

If you want CurveExpert Professional to just treat initial guess failures as general model failures, uncheck this box.

17.3.2 Tolerances

The tolerances determine when CurveExpert Professional will stop calculating a given model; in any nonlinear regression, the solution is an iterative process in which the model parameters are adjusted in order to reach the best possible set. The iteration will terminate when either the overall change in the residual is below the given tolerance, or when the change in the parameters is below the given tolerance. The residual is defined as

$$\Re = \sqrt{\sum_{i}^{N} (y_i - f(\vec{x}_i; \vec{a}))}$$

17.3.3 Maximum Iterations

The iteration described in “Tolerances” above can also be terminated when the number of iterations reaches a certain threshhold. That threshhold is specified in this field.

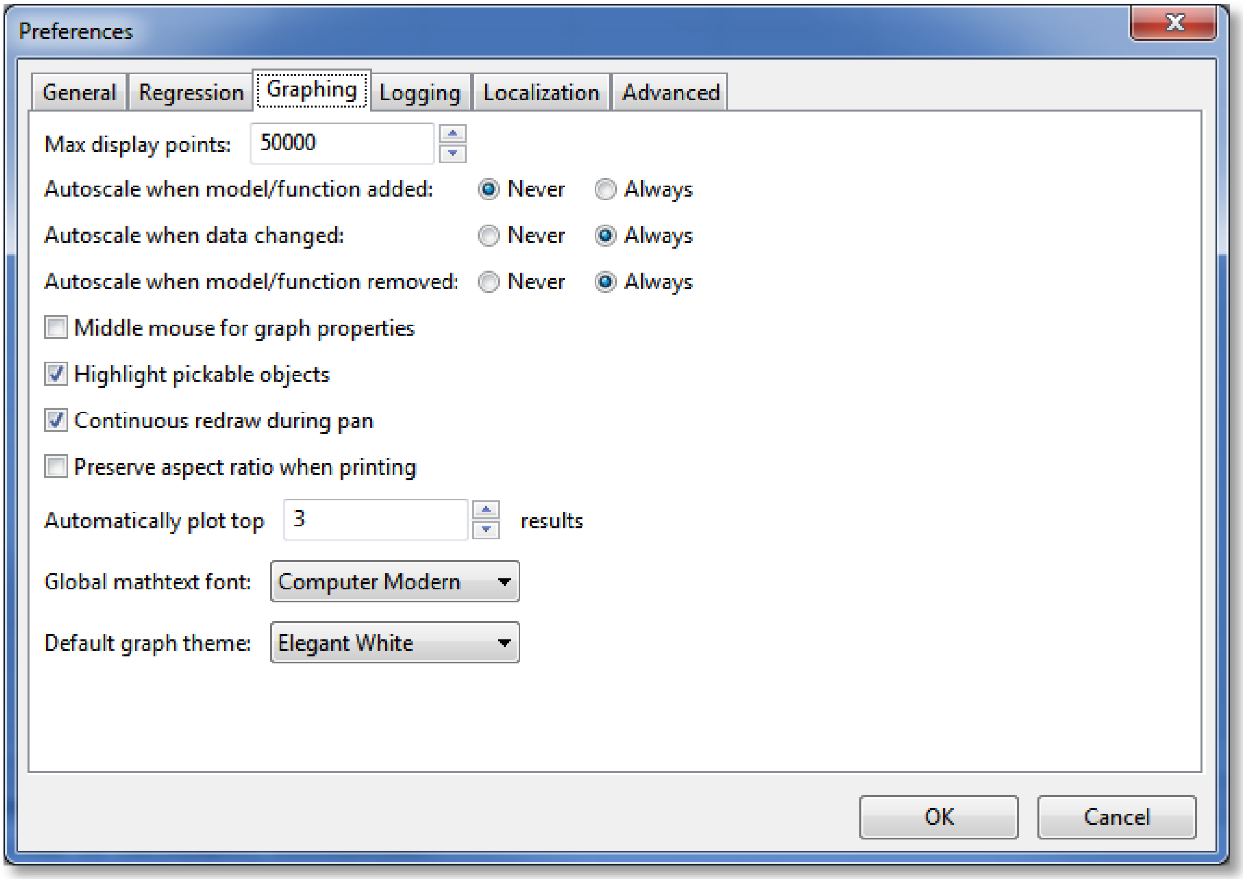
17.3.4 Max number of cores to utilize

CurveExpert Professional can make use of multiple cores on your system when computing batches of nonlinear regressions at a time. The number of cores to utilize can be specified here.

Note: If you only compute one nonlinear regression at a time, and not batches of them, it is best to just leave the multicore setting at zero.

Note: A multicore setting of zero is different than one. Zero signifies that computations take place in the same process as the GUI, and one means that there is one worker process present (which consumes memory) in which the computations are performed.

17.4 Graphing Preferences



Max display points: 50000

Autoscale when model/function added: ● Never ○ Always

Autoscale when data changed: ○ Never ● Always

Autoscale when model/function removed: ○ Never ● Always

☐ Middle mouse for graph properties

☑ Highlight pickable objects

☑ Continuous redraw during pan

☐ Preserve aspect ratio when printing

Automatically plot top 3 results

Global mathtext font: Computer Modern

Default graph theme: Elegant White

17.4.1 Max display points

Sets the maximum number of points, per series, to display when drawing a discrete dataset on a graph. If the dataset has more points than this, it will be thinned to this number of points. The thinning is based on an appropriate stride to use in order to reduce the total number of points drawn to below the threshhold given here. This option is in place to prevent extremely slow graph redraws due to attempting to draw an excessive number of points.

This option affects discrete datasets only; not continuous functions. For continuous functions, the **Resolution** parameter, in the *Series Page* of the Graph Properties dialog (see *Changing Graph Properties*), is the way to control the sampling of the function that is used for drawing.

17.4.2 Autoscale when model/function added

When a model or function is added to a graph (via a *Send to Plot* command or a drag and drop), CurveExpert Professional can automatically autoscale the graph. In this case, you can set the preference to “Never” or “Always”. If the autoscale lock has been set for a graph (see *Graph Toolbar*), the graph will not be autoscaled regardless of this setting; in other words, the graph’s autoscale lock takes precedence over this preference.

17.4.3 Autoscale when data changed

In order to keep the data in view at all times in all graphs, CurveExpert Professional can automatically autoscale the graphs when the underlying dataset changes. The disadvantage of doing this is that, for large datasets, the redraw can be slow. Also, if you have specified your own limits to axes in the graphs, they would be lost if the data changes. If the autoscale lock has been set for a graph (see *Graph Toolbar*), the graph will not be autoscaled regardless of this setting; in other words, the graph's autoscale lock takes precedence over this preference.

To control when autoscaling takes place, select either "Never" or "Always".

17.4.4 Autoscale when model/function removed

When a model or function is removed from a graph (either by removing the result from your document altogether, or by right clicking a result on a plot and selecting "Remove"), CurveExpert Professional can automatically autoscale the graph. In this case, you can set the preference to "Never" or "Always". If the autoscale lock has been set for a graph (see *Graph Toolbar*), the graph will not be autoscaled regardless of this setting; in other words, the graph's autoscale lock takes precedence over this preference.

17.4.5 Middle mouse button for graph properties

Normally, the right mouse button is used to access the context menu on graphs. However, for mouse rotation of 3D graphs, it is sometimes more intuitive to use the right button for that purpose. So, you can set the middle mouse to be the graph context menu activator if desired.

For 2D plots, this setting has no effect. The graph context menu can always be accessed with either a middle or right click of the mouse.

17.4.6 Highlight pickable objects

As you move the mouse over a curve or data, it will highlight, signifying that you can click it to modify its properties. Also, it signifies that, on right-click, you can access the details for that series. However, the redrawing necessary for this feature can make the graph response sluggish. So, deactivate this feature if you don't need to see the highlighting or access details directly from the graph.

17.4.7 Continuous redraw during pan

During a pan, all continuous series must be regenerated on-the-fly if this feature is turned on. This can be expensive, so if it is OK not to see the curve dynamically update during a pan, turn this checkbox off. The series will then only be regenerated each time the mouse button is released during a panning operation.

As a reminder, this setting does not affect discrete datasets on the graph; only results that are continuous.

17.4.8 Automatically plot top # results

When computing batches of nonlinear regressions at once, if all of them were plotted on the “Top Results” graph, it would be quite messy. Here, you can decide how many of the results will be plotted on the graph for each batch that you compute. The default behavior is to take the best three fits for every batch that you compute. Other results that you want to visualize can always be added by right clicking them and selecting *Send to Plot*, or by dragging and dropping the result to the plot.

17.4.9 Preserve aspect ratio when printing

Normally, any graph is fit to the page when it is printed. If you would like to preserve the aspect ratio of your graph when printing, check this box. In other words, if you have a graph that is long and skinny, to keep it long and skinny when printed, check this box.

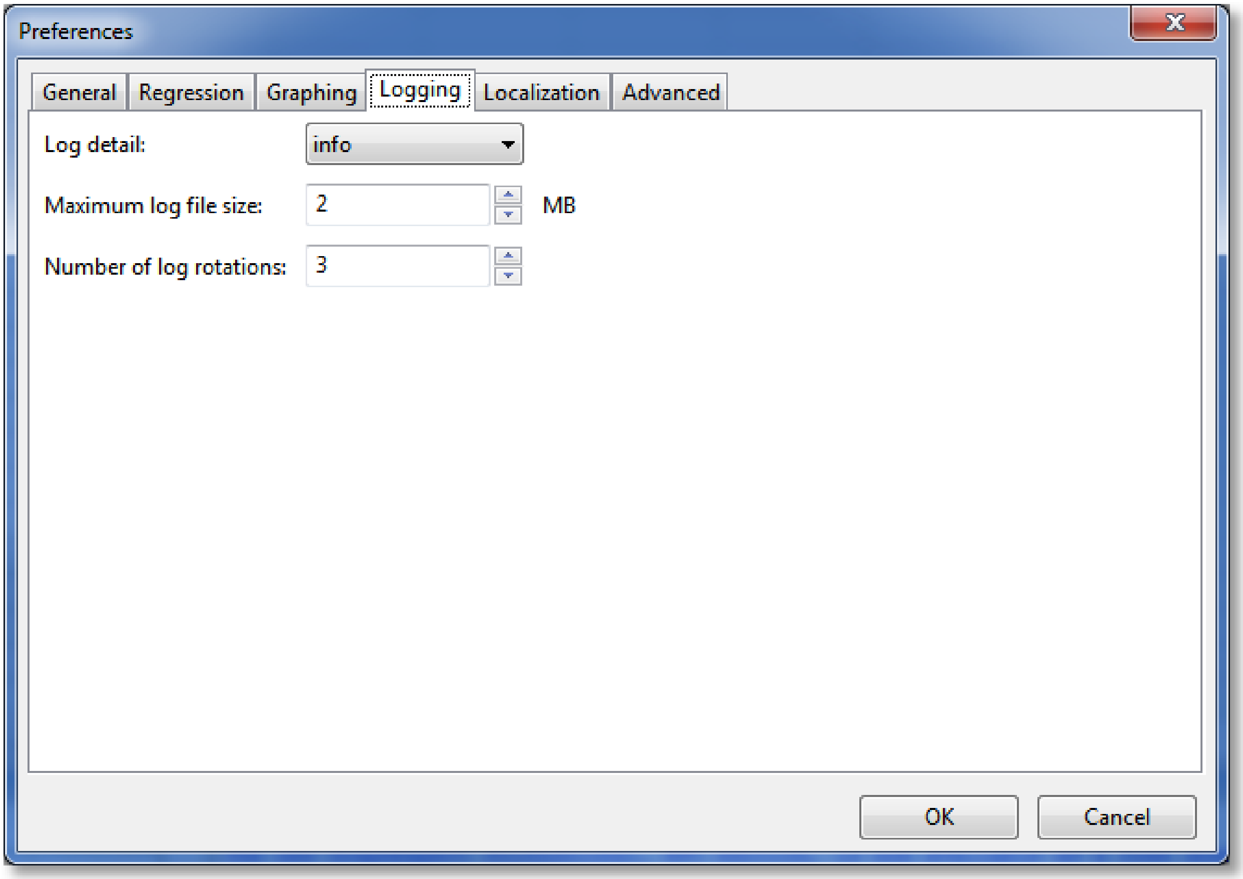
17.4.10 Global mathtext font

The font used to render equations (see mathtext) through the application can be set to Computer Modern, STIX Serif, or STIX Sans-serif. Computer Modern looks very much like TeX equations. STIX Serif looks slightly different, and is meant to blend with surrounding text that is serifed (such as Times Roman). STIX Sans-serif is obviously a non-serifed font, meant to blend with surrounding text that is nonserifed (such as Helvetica or Arial).

17.4.11 Default graph theme

A graph theme is the “look” of the graph, and new themes can be constructed either with the graph theme manager (*Tools->Manage Graph Themes*) or by saving a theme directly from a graph by right-clicking and selecting *Save Theme*. Every graph has a default theme, and you can set what that default theme is here. This way, all plots will be constructed with the look that you desire.

17.5 Logging Preferences



Log detail: info

Maximum log file size: 2 MB

Number of log rotations: 3

17.5.1 Log detail

CurveExpert Professional keeps a log in $HOME/.curveexpert/logs (see *Log Viewer*) ; the log detail setting determines the detail of messages that are stored in the log. The default setting is “info”, which logs all messages that appear in the message window, as well as internal information generated by the application. To have less information logged, move the log detail setting to *warning*, *error*, or *critical*. To log more (for example, if trying to sort out a software problem), move the log detail setting to *debug*.

Note that the logs are easily viewable by selecting *Tools->Log Viewer* in the main menu.

17.5.2 Number of log rotations

When the log reaches a maximum size (configurable below), it will be rotated. This means that if the log file in question is main.log, it will be renamed to main.log.1, and all of the other previously rotated log files will be renamed from main.log.1 to main.log.2, and so on. The number of log rotations determines how many of these files you would like to keep. A number of log rotations of three means that you plan to keep main.log, main.log.1, main.log.2, and main.log.3.

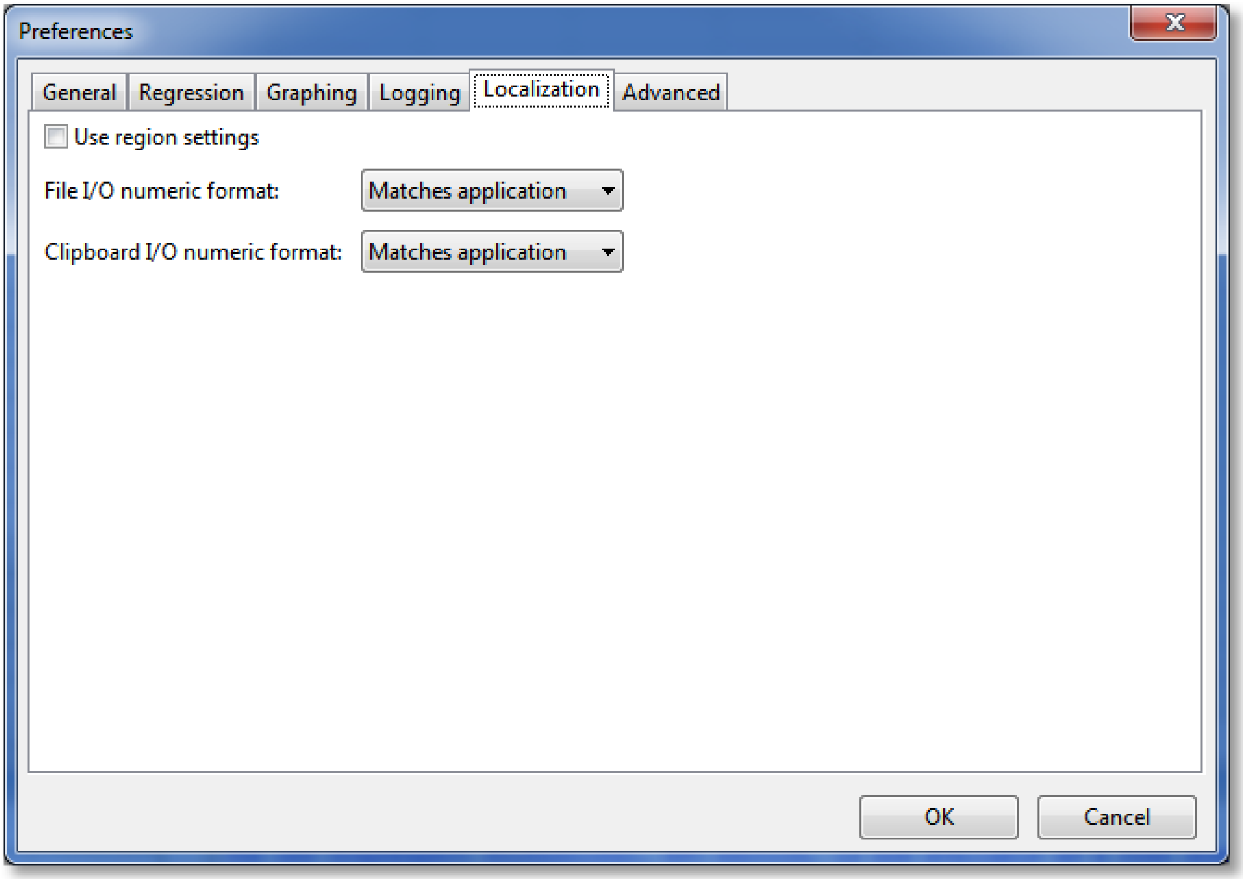
17.5.3 Maximum log file size

Choose the largest file size (in megabytes) that any log file will be allowed to reach before rotation. Note that the log files for multicore workers will be limited to one-fourth of this number. The maximum size on disk that can be taken by the collection log files is therefore

$$space(MB) = s_{max}(n_{rotations} + 1) + s_{max}(n_{rotations} + 1)n_{workers}/4$$

With the default settings of a 2MB log size and 3 rotations, the maximum amount of disk space consumed by the logs is then 16MB, assuming a quad core machine in which four worker processes are utilized.

17.6 Localization Preferences



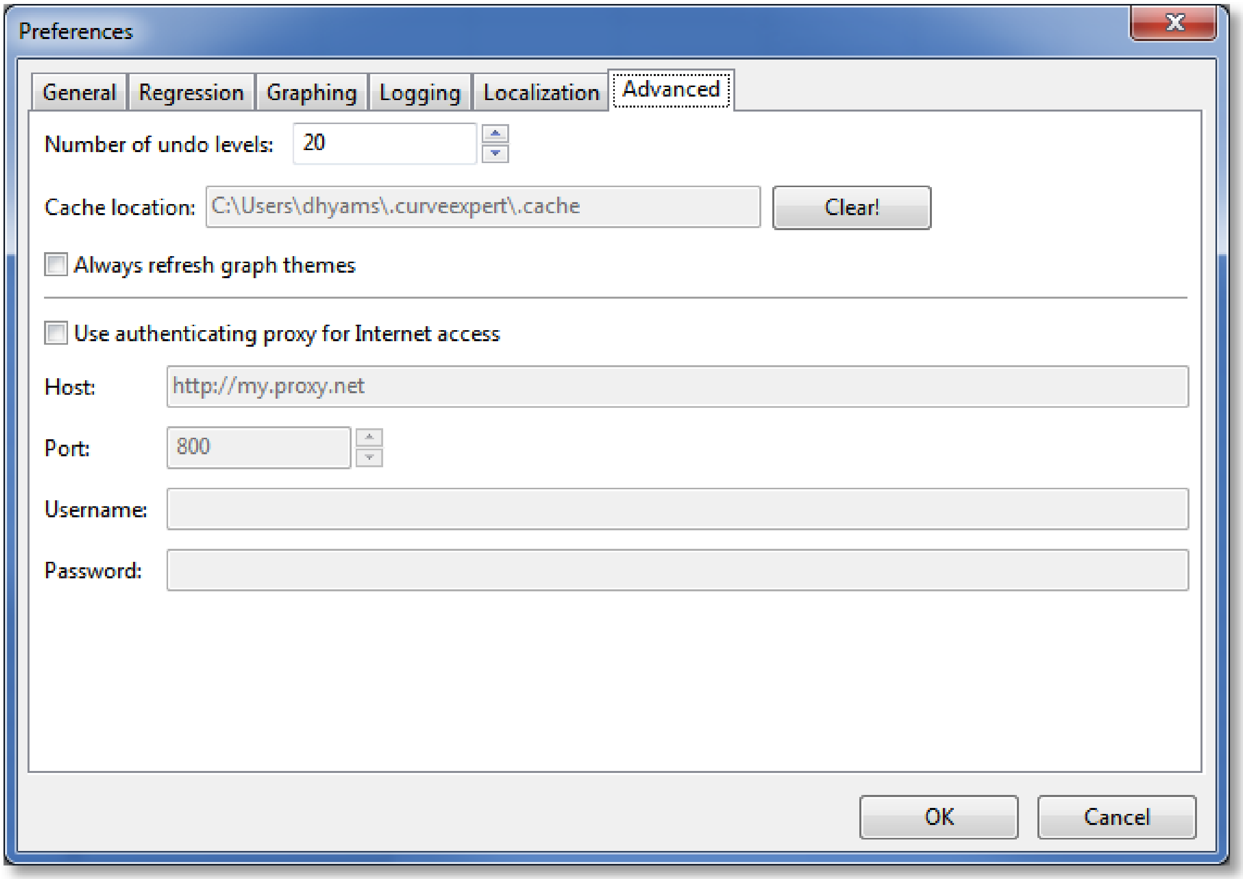
□ Use region settings

File I/O numeric format: Matches application

Clipboard I/O numeric format: Matches application

The localization preferences are documented in the *Localization* chapter, in the *Configuring CurveExpert Professional to match your workflow* section.

17.7 Advanced Preferences



Number of undo levels: 20

Cache location: C:\Users\dhyams\.curveexpert\.cache

□ Always refresh graph themes

□ Use authenticating proxy for Internet access

Host: http://my.proxy.net

Port: 800

Username:

Password:

17.7.1 Number of undo levels

Any undo mechanism consumes some memory to save the previous state of the application; the number of undo levels setting allows you to customize how many previous states are saved that you can undo to. The parts of CurveExpert Professional that support Undo operations are the graphs, data spreadsheet, notes, and messages.

17.7.2 Cache location

The location of a cache that CurveExpert Professional uses. It is not changeable, but it can be cleared if need be. This cache operates very similarly to a web browser's cache. Clearing it will not affect the correct operation of the program, but will force the regeneration of the information that was cached, slowing performance temporarily.

17.7.3 Always refresh graph themes

If set, this option will merge the graph themes from those available with the software (called the stock themes), to the user's private theme directory ($HOME/.curveexpert/graph_themes). This happens whenever CurveExpert Professional starts. The purpose of keeping this option selected is that, when CurveExpert Professional is updated/reinstalled, the user will not get any new themes or updated themes that are shipped with the software. In most cases, merging the stock themes alongside your custom themes in $HOME/.curveexpert/graph_themes is the desired behavior.

It should be noted, however, that if you have deleted any of the stock themes, they will reappear. Also, if you have saved modified themes with the same name as the stock themes (which is not recommended), those themes will be overwritten.

If the behavior noted above is not desirable, uncheck this option. It should be remembered, though, that if left unset, the themes will not be updated when any new version of CurveExpert Professional is installed; the themes in the $HOME/.curveexpert/graph_themes folder will be left untouched.

17.7.4 Proxy settings

CurveExpert Professional uses Internet access to begin your trial period (see *Installation and Activation*), check for latest versions, upload issue reports (if enabled, see *Automatically send issue reports*), and upload logs (see *Log Viewer*). If you do not use a proxy, you can leave *Use authenticating proxy for Internet access* unchecked. However, if you are behind a proxy, please fill in the host, port and port for your proxy server (https:// or http:// in the server name is optional). If your proxy server requires authentication, also fill in the username and password fields with your username/password combination for the proxy server.

CHAPTER EIGHTEEN

LOCALIZATION

18.1 Executive Summary

CurveExpert Professional localizes itself to the same region as the operating system you are running on by default; therefore you should see the appropriate treatment of decimal points (either a comma, or a dot) throughout the application. If you do not want this behavior, but want to operate in English only, uncheck the *Use local region settings* box and restart CurveExpert Professional.

CurveExpert Professional offers the ability, when it is reading/writing data to/from the clipboard or a file, to allow the user to specify what style of decimals to use for this, presuming that it differs from the default locale.

- If you need to read files that use the dot-decimal notation most often, select "English" for *File I/O numeric format*; same comments apply for the clipboard format.
- If you need to read files that use the comma-decimal notation most often, select "European" for "File I/O numeric format"; same comments apply for the clipboard format.

18.2 Configuring CurveExpert Professional to match your workflow

18.2.1 Use local region settings

The *Use local region settings* checkbox is by default checked, and will cause the application to use your native language style for numerical formatting. This is the desired behavior in most situations. However, if you want to operate the software in English only mode (decimals will always be dots), deselect this checkbox. If you change this setting, you must restart CurveExpert Professional for the change to take effect.

18.2.2 File I/O numeric format

The *Edit->Preferences->Localization->File I/O numeric format* choice allows you to override the application-wide locale setting for file input/output, to explicitly define the numeric format for reading and writing files. This affects the file import dialog as well as *File->Export*, and the table generator in the **Result Details** window. With this setting, for example, users that utilize commas for decimals in CurveExpert Professional can choose to read and write files using dots for decimals. And vice versa.

Note: In the file import dialog (see *Raw file import*), there is still a chance to change the numeric format before reading the file. This setting simply sets the default numeric format in that dialog appropriately.

18.2.3 Clipboard I/O numeric format

The *Edit->Preferences->Localization->Clipboard I/O numeric format* choice allows you to override the application-wide locale setting for clipboard input/output, to explicitly define the numeric format for copying and pasting. This affects the copy/paste in the spreadsheet, as well as copying numeric data out any window that might offer it. With this setting, for example, users that utilize commas for decimals in CurveExpert Professional can choose to copy and paste data to/from the clipboard using dots for decimals. And vice versa.

18.2.4 File Import: numeric format

In the file import dialog (*File->Open*), there is a switch to explicitly pick the numerical formatting (English or European style) before reading the file. This gives the user the opportunity, on a file-by-file basis, to specify the numerical format. See *Raw file import*.

18.3 File overrides

If your file contains the following within the first ten lines of the file:

```
#DataFileProperties: locale=EN
```

or:

```
#DataFileProperties: locale=EURO
```

this signals to the CurveExpert Professional file importing mechanism that the the file uses English or European numerical formatting (as appropriate), and the default there (see *Raw file import*) will be set appropriately. The user may still override the default, however, but in most cases this would be a bad idea, because the file itself is stating what the correct locale is.

So, you can use this capability in your own data files if you like. All of the sample data files shipped with CurveExpert Professional contain this line, so that the samples work straightforwardly regardless of the application's current region settings. This small addition to a text datafile lets the file itself express its own numerical formatting.

18.4 CurveExpert Professional files

CurveExpert Professional files (.cxp) are not affected by region settings; therefore, they are portable across localities as well as being portable across platforms (Windows, Mac, Linux).

18.5 NIST verification files

The data files associated with the NIST datasets are explicitly US-locale, and are treated as such when they are read in as a result of *Tools->Run NIST Validation* performed within CurveExpert Professional (see *Validation*). However, if you read these files in yourself, note that you MUST select an English numerical formatting in the file import dialog, if it is not already set.

CHAPTER

NINETEEN

VALIDATION

19.1 Introduction

CurveExpert Professional is validated against the Statistical Reference Datasets Project of the National Institute of Standards and Technology. These datasets can be downloaded directly at http://www.itl.nist.gov/div898/strd/general/dataarchive.html, but are also included verbatim in the CurveExpert Professional distribution for you to use yourself. The datasets validated against are all of the sets in the "Linear Regression" and "Nonlinear Regression" categories.

19.2 Running the Suite

You can validate CurveExpert Professional against these datasets simply by choosing *Tools->Run NIST Validation*. All validation cases, with all "starts", will be run. The NIST test suite provides different starts (i.e., initial guesses) for each nonlinear regression validation. CurveExpert Professional covers all of these starts in the test suite, and the results are output to the message window.

Some competing packages do not include all starts in their validation sets, but CurveExpert Professional shows the results of all of them. At the time of this writing, there is one failure (dataset BoxBOD, Start #1), where, for the given start, the initial correction to parameter β_2 causes underflow in the model, and from this there is no recovery.

19.3 Performing validations yourself

If you would like to perform validations against the test data yourself, all data files are contained in the "example_data/NIST" folder in the initial data directory (see *Where files are located*). Or, you can always download the files from NIST at the link above.

Note: The NIST datasets all list the dependent variable (usually y) **before** the independent variable (usually x). So, when you read the files, select *Rotate Columns* in the file import dialog, or alternatively, rotate the columns after you read in the dataset.

All nonlinear regression validation runs are performed in the automated suite with rtol=ptol=1.0e-14, and maxiter=1000.

CHAPTER

TWENTY

MISCELLANEOUS

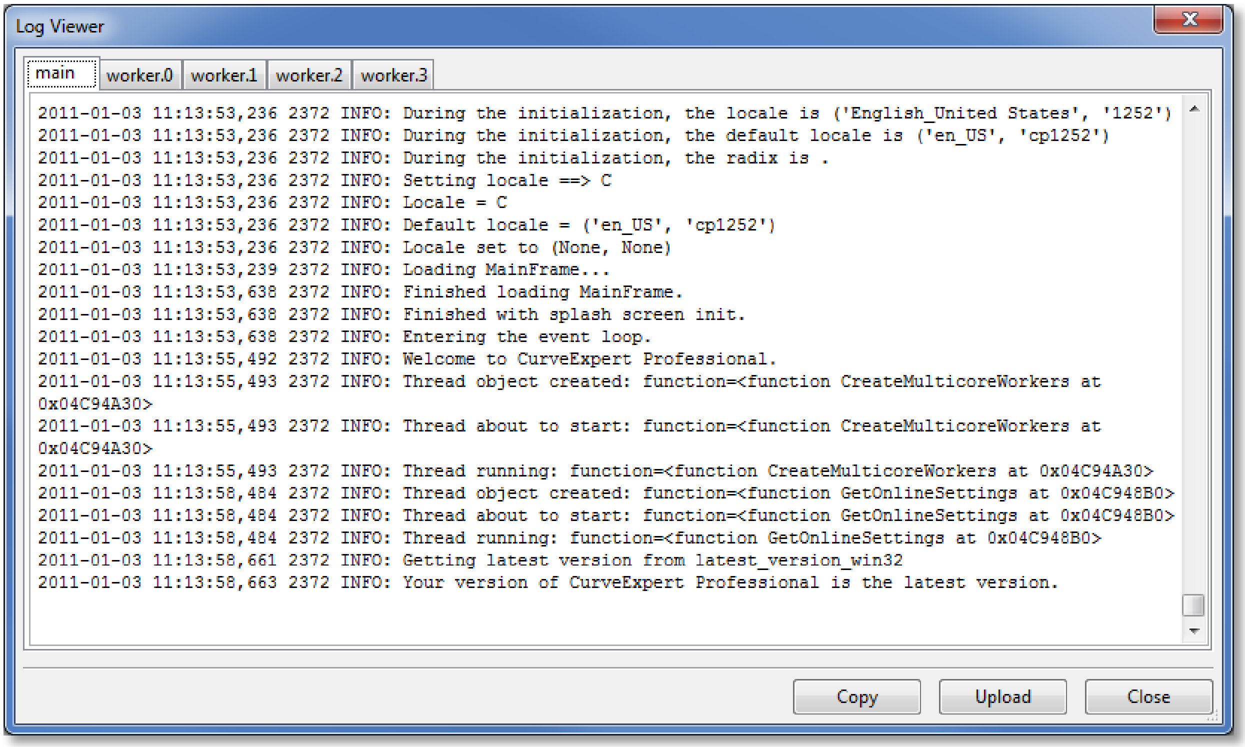
20.1 Log Viewer

To view the log files, select *Tools->View Logs* from the main menu. All logs are stored in the $HOME/.curveexpert/logs directory, and roll according to the preferences that you have set in *Edit->Preferences->Logging* (see *Logging Preferences*). So, you can easily view logs for previous sessions of CurveExpert Professional with this tool.

To copy a portion of the log, select the desired portion, and press the *Copy* button. Likewise, to upload a portion of the log to the CurveExpert Professional server (http://www.curveexpert.net), press the *Upload* button. Uploading of a log is typically used in conjunction with reporting or tracing a software issue. If no part of the log is selected, the entire log will be copied to clipboard or uploaded, as appropriate.

There is one log file per process that CurveExpert Professional runs. If you are using multicore capability (CurveExpert Pro only), there will be several log files (named worker.n, where n is a number from 0 to the number of helper processes minus one) in addition to the main log file. These worker log files document the actions that each worker has performed, and can be copied or uploaded in identical fashion to the main log.

See *Logging Preferences* for more detail on configuring the logging options.



```
2011-01-03 11:13:53,236 2372 INFO: During the initialization, the locale is ('English_United States', '1252')
2011-01-03 11:13:53,236 2372 INFO: During the initialization, the default locale is ('en_US', 'cp1252')
2011-01-03 11:13:53,236 2372 INFO: During the initialization, the radix is .
2011-01-03 11:13:53,236 2372 INFO: Setting locale ==> C
2011-01-03 11:13:53,236 2372 INFO: Locale = C
2011-01-03 11:13:53,236 2372 INFO: Default locale = ('en_US', 'cp1252')
2011-01-03 11:13:53,236 2372 INFO: Locale set to (None, None)
2011-01-03 11:13:53,239 2372 INFO: Loading MainFrame...
2011-01-03 11:13:53,638 2372 INFO: Finished loading MainFrame.
2011-01-03 11:13:53,638 2372 INFO: Finished with splash screen init.
2011-01-03 11:13:53,638 2372 INFO: Entering the event loop.
2011-01-03 11:13:55,492 2372 INFO: Welcome to CurveExpert Professional.
2011-01-03 11:13:55,493 2372 INFO: Thread object created: function=<function CreateMulticoreWorkers at
0x04C94A30>
2011-01-03 11:13:55,493 2372 INFO: Thread about to start: function=<function CreateMulticoreWorkers at
0x04C94A30>
2011-01-03 11:13:55,493 2372 INFO: Thread running: function=<function CreateMulticoreWorkers at 0x04C94A30>
2011-01-03 11:13:58,484 2372 INFO: Thread object created: function=<function GetOnlineSettings at 0x04C948B0>
2011-01-03 11:13:58,484 2372 INFO: Thread about to start: function=<function GetOnlineSettings at 0x04C948B0>
2011-01-03 11:13:58,484 2372 INFO: Thread running: function=<function GetOnlineSettings at 0x04C948B0>
2011-01-03 11:13:58,661 2372 INFO: Getting latest version from latest_version_win32
2011-01-03 11:13:58,663 2372 INFO: Your version of CurveExpert Professional is the latest version.
```

20.2 The Python Sandbox

As the name implies, the Python sandbox is a place for you to "play" and learn about Python. To access the Python sandbox, select *Tools->Python sandbox* from the main menu.

The sandbox gives you access to a interactive Python session, with some powerful addon modules (numpy, scipy, PIL, and wxPython) available for your use. What you do in the sandbox is up to you; you can use it to learn about Python or the addon modules, or you can use it to do interactive work from the command line. For example:

```
>>> import numpy
>>> x = numpy.linspace(0,2,100)
>>> y = numpy.sin(x)
>>> y
>>> numpy.savetxt('my_file.txt',y)
```

will write out a file containing the sine of x for 100 points sampled between 0 and 2. Note that typing 'y' by itself did nothing more than just display the contents of that variable).

Use Ctrl+Up and Ctrl+Down to cycle through your history of commands. The notebook in the lower part of the sandbox shows the last documentation page that was accessed (context sensitive), and the history of commands that you have typed. Closing the sandbox does not close your session; you may resume it by opening the sandbox again. Sessions are not, however, preserved across different invocations of CurveExpert Professional.

20.3 Running Multiple Instances

Running multiple instances of CurveExpert Professional is supported; the only side effect is that, because all instances share the same log files, logging might not function properly.

20.4 Where files are located

CurveExpert Professional places files in several locations on disk; the locations are documented below.

20.4.1 Program Files

The default locations are listed below, assuming that the default is accepted during the installation process. If you have changed the location of the program files, obviously they will be placed where you specified.

Platform	Program Files Location
Windows (64 bit)	c:\Program Files (x86)\CurveExpert Professional
Windows (32 bit)	c:\Program Files\CurveExpert Professional
Mac (OSX)	/Applications/CurveExpertPro.app
Linux (installed as user)	$HOME/CurveExpertPro
Linux (installed as root)	/usr/local/CurveExpertPro

20.4.2 Application Settings

Platform	Application Settings Location
Windows	in registry: HKEY_CURRENT_USER\Software\CurveExpert Professional
Mac (OSX)	$HOME/Library/Preferences/CurveExpert Professional Preferences
Linux	$HOME/.CurveExpert Professional

20.4.3 Working/Cache Area

An area is reserved for various caches and items that are needed for normal application operation. This area is used for the cache, logs, storage of graph themes, and storage of custom user models/functions.

Platform	Working/Cache Area Location
Windows	$HOME\.curveexpert
Mac (OSX)	$HOME/.curveexpert
Linux	$HOME/.curveexpert

20.4.4 CurveExpert Professional Data Files

The data files directory can be customized using *Edit->Preferences->General*, but the default locations are below. This directory is also where the example data that ships with CurveExpert resides.

Platform	Documents Location
Windows	$MYDOCUMENTS\CurveExpert Professional
Mac (OSX)	$HOME/Documents/CurveExpert Professional
Linux	$HOME/Documents/CurveExpert Professional

CHAPTER

TWENTYONE

END USER LICENSE AGREEMENT

IMPORTANT – READ CAREFULLY: This End-User License Agreement (“EULA”) is a legal agreement between you (either an individual or single entity) and Mr. Daniel Hyams of 115 Cresthaven Drive, Madison, AL 35758(“Mr. Hyams”) for the software product “CurveExpert Professional”, which will be designated by the term “Software Product”. The Software Product consists computer software and accompanying electronic documentation. By installing, copying, or otherwise using the Software Product, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the Software Product.

1. **Grant of License.** Mr. Hyams grants to you as an individual, a personal, nonexclusive license to make and use copies of the Software Product. Copies must be made only for backup or archival purposes.
2. **Shareware version.** You are hereby granted a non-exclusive license to use the Software Product for an evaluation period of 30 days, after which you must pay the registration fee to continue using the Software Product. However, use of the Software Product during the 30-day evaluation period is free of charge. Unregistered use of the Software Product after the 30-day evaluation period is in violation of Federal copyright laws and the terms of this EULA. Violators will be prosecuted to the maximum extent under the law. You may make exact copies and distribute the shareware version of the Software Product and its documentation so long as there is no charge applied to distribution of these unmodified copies of the Software Product. You are specifically prohibited from charging or requesting donations for ANY and ALL copies of this Software Product or its documentation. You are prohibited from distributing the Software Product or any portion of the Software Product or its documentation as part of another product (commercial or otherwise) without PRIOR expressed written consent from Mr. Hyams.
3. **Registered Version** If you have registered the Software Product with Mr. Hyams (meaning to remit payment for purchase of the software), the 30-day evaluation period in Section 2 is waived. Mr. Hyams agrees to provide you with valid license key for the Software Product, copied onto media deemed suitable, within 30 days of receipt of payment.
4. **Reverse engineering.** You may not reverse engineer, decompile, or disassemble the Software Product.
5. **Upgrades.** All subsequent versions of the software product fall under the terms of this EULA. Upgrades are designated as software packages of the same name but with an incremented version number.
6. **Copyright.** All title and copyrights in and to the Software Product are owned by Mr. Hyams. The Software Product is protected by copyright laws and international treaty provisions.
7. **Warranty Disclaimer.** The Software Product and the accompanying files are sold “as is” without warranty of any kind. All warranties, either express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose, are expressly and specifically disclaimed.
8. **Liability.** Neither Mr. Hyams or anyone else who has been involved in the creation, production, or delivery of this Software Product shall be liable for any direct, indirect, consequential, special, incidental, or similar damages arising out of the use or inability to use such software even if Mr. Hyams has been advised of the possibility of such damages or claims. In no event shall Mr. Hyams’s liability for any damages ever exceed the price paid for the license to the Software Product, regardless of the form of the claim. The individual using the Software Product bears all risk as to the quality and performance of the Software Product.

9. **Governing Law.** This agreement shall be governed by the laws of the State of Alabama and shall inure to the benefit of Mr. Hyams, his heirs, successors, and assigns. Any action or proceeding brought by either party against the other arising out of or relating to this agreement shall be brought only in a state or federal court of competent jurisdiction located in Madison, Alabama. The parties hereby consent to in personam jurisdiction of said courts.

CHAPTER

TWENTYTWO

THIRD-PARTY LICENSING

CurveExpert Professional owes its existence to several excellent open source packages. This section is in the documentation in order to comply with licensing requirements of these packages.

22.1 Python

PSF LICENSE AGREEMENT FOR PYTHON 2.6.6

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 2.6.6 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 2.6.6 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2010 Python Software Foundation; All Rights Reserved" are retained in Python 2.6.6 alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 2.6.6 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 2.6.6.
4. PSF is making Python 2.6.6 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 2.6.6 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 2.6.6 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 2.6.6, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python 2.6.6, Licensee agrees to be bound by the terms and conditions of this License Agreement.

22.2 WxWidgets and WxPython

22.2.1 Preamble

The licensing of the wxWidgets library is intended to protect the wxWidgets library, its developers, and its users, so that the considerable investment it represents is not abused.

Under the terms of the wxWindows licence, you as a user are not obliged to distribute wxWidgets source code with your products, if you distribute these products in binary form. However, you are prevented from restricting use of the library in source code form, or denying others the rights to use or distribute wxWidgets library source code in the way intended.

The wxWindows licence establishes the copyright for the code and related material, and it gives you legal permission to copy, distribute and/or modify the library. It also asserts that no warranty is given by the authors for this or derived code.

The core distribution of the wxWidgets library contains files under two different licences:

- Most files are distributed under the GNU Library General Public Licence, version 2, with the special exception that you may create and distribute object code versions built from the source code or modified versions of it (even if these modified versions include code under a different licence), and distribute such binaries under your own terms.
- Most core wxWidgets manuals are made available under the "wxWidgets Free Documentation Licence", which allows you to distribute modified versions of the manuals, such as versions documenting any modifications made by you in your version of the library. However, you may not restrict any third party from reincorporating your changes into the original manuals.

Other relevant files:

- licence.txt: a statement that the wxWidgets library is covered by the GNU Library General Public Licence, with an exception notice for binary distribution.
- licendoc.txt: the wxWidgets Documentation Licence.
- lgpl.txt: the text of the GNU Library General Public Licence.
- gpl.txt: the text of the GNU General Public Licence, which is referenced by the LGPL.

22.2.2 wxWindows Library License, Version 3.1

> Copyright (c) 1998-2005 Julian Smart, Robert Roebling et al
>
> Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.
>
> WXWINDOWS LIBRARY LICENCE
>
> TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
>
> This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.
>
> This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public Licence for more details.

You should have received a copy of the GNU Library General Public Licence along with this software, usually in a file named COPYING.LIB. If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

EXCEPTION NOTICE

1. As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3.1 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3.1 of the Licence document.

2. The exception is that you may use, copy, link, modify and distribute under your own terms, binary object code versions of works based on the Library.

3. If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.

4. If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

22.3 Numpy

Copyright (c) 2005-2009, NumPy Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

22.4 Scipy

22.5 Matplotlib

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between JDH and Licensee. This License Agreement does not grant permission to use JDH trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using matplotlib 0.91, Licensee agrees to be bound by the terms and conditions of this License Agreement.

22.6 xlrd

CurveExpert Professional utilizes Excel reading capabilities from the xlrd library. This product includes software developed by Roman V. Kiseliov <roman@kiseliov.ru>.

22.7 BioPython

CurveExpert Professional utilizes the Lowess algorithm originating from the Biopython package.

Biopython License Agreement

22.8 winpaths

CurveExpert Professional utilizes a single function from winpaths in order to find the “My Documents” directory under Windows. The required MIT license follows:

The MIT License

22.9 Python Imaging Library (PIL)

The Python Imaging Library (PIL) is

22.10 Icons and Artwork

Some icons are attributed to http://www.fatcow.com/free-icons/. Also, Tango icons http://tango.freedesktop.org/ are used in this project.

22.11 JPEG library

This software is based in part on the work of the Independent JPEG Group.

CHAPTER

TWENTYTHREE

APPENDIX A: MATH FUNCTIONS

This section documents the math functions that are available for use in CurveExpert Professional custom models, functions, and data transformations.

All functions below, unless otherwise stated, work on either scalars (a single number) or an array (a vector). If an array is passed to any of these functions, the function is applied elementwise to each member of the array.

23.1 Constants

23.1.1 pi

The mathematical constant 3.14159. . . .

23.1.2 exp1

The mathematical constant 2.71828. . . ..

23.2 Transcendental Functions

23.2.1 acos(x)

Trigonometric inverse cosine. The inverse of *cos* so that, if `y = cos(x)`, then `x = acos(y)`. The domain is [-1, 1].

acos is a multivalued function: for each *x* there are infinitely many numbers *z* such that *cos(z) = x*. The convention is to return the angle *z* whose real part lies in *[0, pi]*.

The inverse *cos* is also known as *arccos* or cos^-1.

23.2.2 acosh(x)

Inverse hyperbolic cosine.

acosh is a multivalued function: for each *x* there are infinitely many numbers *z* such that *cosh(z) = x*. The convention is to return the *z* whose imaginary part lies in *[-pi, pi]* and the real part in `[0, inf]`.

23.2.3 asin(x)

Inverse sine. The inverse sine of each element in *x*, in radians and in the closed interval `[-pi/2, pi/2]`.

arcsin is a multivalued function: for each *x* there are infinitely many numbers *z* such that *sin*(*z*) = *x*. The convention is to return the angle *z* whose real part lies in [-pi/2, pi/2].

The inverse sine is also known as *arcsin* or sin^{-1}.

23.2.4 asinh(x)

Inverse hyperbolic sine.

arcsinh is a multivalued function: for each *x* there are infinitely many numbers *z* such that *sinh(z)* = *x*. The convention is to return the *z* whose imaginary part lies in *[-pi/2, pi/2]*.

The inverse hyperbolic sine is also known as *arcsinh* or `sinh^-1`.

23.2.5 atan(x)

Trigonometric inverse tangent. The inverse of tan, so that if `y = tan(x)` then `x = atan(y)`.

atan is a multi-valued function: for each *x* there are infinitely many numbers *z* such that tan(*z*) = *x*. The convention is to return the angle *z* whose real part lies in [-pi/2, pi/2].

The inverse tangent is also known as *arctan* or tan^{-1}.

23.2.6 atan2(y,x)

Arc tangent of `y/x` choosing the quadrant correctly.

The quadrant (i.e., branch) is chosen so that `arctan2(y, x)` is the signed angle in radians between the ray ending at the origin and passing through the point (1,0), and the ray ending at the origin and passing through the point (*x2*, *x1*). By IEEE convention, this function is defined for *x* = +/-0 and for either or both of *y* and *x* = +/-inf (see Notes for specific values).

Returns angle in radians, in the range `[-pi, pi]`.

23.2.7 atanh(x)

Inverse hyperbolic tangent.

arctanh is a multivalued function: for each *x* there are infinitely many numbers *z* such that *tanh(z)* = *x*. The convention is to return the *z* whose imaginary part lies in *[-pi/2, pi/2]*.

The inverse hyperbolic tangent is also known as *arctanh* or `tanh^-1`.

23.2.8 cos(x)

Trigonometric cosine.

23.2.9 cosh(x)

Hyperbolic cosine. Equivalent to `1/2 * (exp(x) + exp(-x))`.

23.2.10 erf(x)

Error function of x defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

23.2.11 erfc(x)

Returns

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

23.2.12 exp(x)

Calculate the exponential.

The irrational number `e` is also known as Euler's number. It is approximately 2.718281, and is the base of the natural logarithm, `ln` (this means that, if $x = \ln y = \log_e y$, then $e^x = y$. For real input, `exp(x)` is always positive.

23.2.13 ln(x)

Natural logarithm (synonym for log). The natural logarithm *ln* is the inverse of the exponential function, so that *ln(exp(x))* = *x*. The natural logarithm is logarithm in base *e*.

23.2.14 log(x)

Natural logarithm. The natural logarithm *log* is the inverse of the exponential function, so that *log(exp(x))* = *x*. The natural logarithm is logarithm in base *e*.

23.2.15 log10(x)

Return the base 10 logarithm. NaNs is returned if x is negative.

23.2.16 sin(x)

Trigonometric sine.

23.2.17 sinh(x)

Hyperbolic sine, element-wise. Equivalent to `1/2 * (exp(x) - exp(-x))`.

23.2.18 sqrt(x)

Return the positive square-root. Returns NaN if x is negative.

23.2.19 tan(x)

Compute tangent. Equivalent to `sin(x)/cos(x)`.

23.2.20 tanh(x)

Compute hyperbolic tangent. Equivalent to `sinh(x)/cosh(x)`.

23.3 Bessel Functions

23.3.1 bessel_j0(x)

The Bessel function of order 0 at x.

23.3.2 bessel_j1(x)

The Bessel function of order 1 at x.

23.3.3 bessel_jn(n,x)

Bessel function of integer or real order n.

23.3.4 bessel_y0(x)

The Bessel function of the second kind of order 0 at x.

23.3.5 bessel_y1(x)

The Bessel function of the second kind of order 1 at x.

23.3.6 bessel_yn(n,x)

Bessel function of the second kind of integer or real order n at x.

23.3.7 bessel_i0(x)

Modified Bessel function of the first kind, order 0.

23.3.8 bessel_i1(x)

Modified Bessel function of order 1 at x.

23.3.9 bessel_k0(x)

Modified Bessel function of the second kind (sometimes called the third kind) of order 0 at x.

23.3.10 bessel_k1(x)

Modified Bessel function of the second kind (sometimes called the third kind) of order 1 at x.

23.3.11 bessel_kn(n,x)

Modified Bessel function of the second kind (sometimes called the third kind) for integer or real order n at x.

23.3.12 bessel_i0e(x)

Exponentially scaled modified Bessel function of order 0 at x. bessel_i0e(x) = exp(-|x|) * bessel_i0(x).

23.3.13 bessel_i1e(x)

Exponentially scaled modified Bessel function of order 0 at x. bessel_i1e(x) = exp(-|x|) * bessel_i1(x).

23.3.14 bessel_k0e(x)

Exponentially scaled modified Bessel function of the second kind (sometimes called the third kind) of order 0 at x. bessel_k0e(x) = exp(x) * bessel_k0(x).

23.3.15 bessel_k1e(x)

Exponentially scaled modified Bessel function of the second kind (sometimes called the third kind) of order 1 at x. bessel_k1e(x) = exp(x) * bessel_k1(x).

23.3.16 bessel_jvp(n,x,i)

Return the ith derivative of bessel_jn(n,x).

23.3.17 bessel_yvp(n,x,i)

Return the nth derivative of bessel_yn(n,x).

23.3.18 bessel_kvp(n,x,i)

Return the nth derivative of bessel_kn(n,x).

23.3.19 bessel_ivp(n,x,i)

Return the nth derivative of bessel_in(n,x).

23.4 Comparison Functions

23.4.1 maximum(a,b)

Element-wise maximum of array elements; compares two arrays and returns a new array containing the elementwise maxima. If one of the elements is a NaN, that element is returned.

23.4.2 minimum(a,b)

Element-wise minimum of array elements; compares two arrays and returns a new array containing the elementwise minima. If one of the elements is a NaN, that element is returned.

23.4.3 max(array)

Return the maximum of an array (returns a scalar). Optionally, the keyword 'axis' can be given to specify, for a 2D dataset, if the max should be taken down the columns (axis=0) or across the rows(axis=1).

23.4.4 min(array)

Return the minimum of an array (returns a scalar). Optionally, the keyword 'axis' can be given to specify, for a 2D dataset, if the min should be taken down the columns (axis=0) or across the rows(axis=1).

23.5 Miscellaneous Functions

23.5.1 abs(x)

Calculate the absolute value.

23.5.2 ceil(x)

Returns the ceiling of the input. The ceil of the scalar *x* is the smallest integer *i*, such that *i >= x*.

23.5.3 copysign(x1,x2)

Change the sign of x1 to that of x2.

23.5.4 degrees(x)

Convert angles from radians to degrees.

23.5.5 factorial(n)

Take the factorial of an integer n.

23.5.6 floor(x)

Return the floor of the input. The floor of the scalar *x* is the largest integer *i*, such that *i <= x*.

Some spreadsheet programs calculate the "floor-towards-zero", in other words `floor(-2.5) == -2`. Here, however, we use the a definition of *floor* such that *floor(-2.5) == -3*.

23.5.7 fmod(x1,x2)

Return the remainder of division. Returns the remainder of the division of *x1* by *x2*.

The result of the modulo operation for negative dividend and divisors is bound by conventions. In *fmod*, the sign of the remainder is the sign of the dividend. In *remainder*, the sign of the divisor does not affect the sign of the result.

23.5.8 hypot(x1,x2)

Given the "legs" of a right triangle, return its hypotenuse. Equivalent to `sqrt(x1**2 + x2**2)`.

23.5.9 isinf(x)

Test for positive or negative infinity.

23.5.10 isnan(x)

Test for Not a Number (NaN).

23.5.11 legendre(n,x)

Legendre function of order n evaluated at x. Orthogonal over [1,1] with weighting function 1.

23.5.12 lstsq_solve(A,B)

Solve a least squares problem, where A is a rectangular matrix. Returns a tuple of four quantities: x (the least squares solution), residuals (array ncol long), rank (integer indicating the effective rank of A), and s (the singular values of a). Common usage is as follows:

```
(soln,resid,rank,s) = solve_lstsq(A,B)
```

Can throw an exception if the computation is too ill-conditioned.

23.5.13 sh_legendre(n,x)

Shifted Legendre function of order n evaluated at x. Orthogonal over [0,1] with weighting function 1.

23.5.14 radians(x)

Convert angles from degrees to radians.

23.5.15 round(a,number_of_decimals)

Round an array to a given number of decimals. number_of_decimals is an integer denoting the number of decimal places to retain.

23.5.16 special_beta(p,q)

Returns

$$\beta(p,q) = \Gamma(p)\Gamma(q)/\Gamma(p+q)$$

23.5.17 special_lambda(nu,x)

Returns

$$\Lambda_\nu(x) = \frac{J_\nu(x)}{\left(\frac{1}{2}x\right)^\nu}$$

The Bessel function used here is the same as special_jn(n,x).

23.5.18 special_gamma(x)

Gamma function. The gamma function is often referred to as the generalized factorial since x*gamma(x) = gamma(x+1) and gamma(n+1) = n! for natural number n.

Note that the domain is limited to [1.0e-6,10].

23.5.19 trunc(x)

Return the truncated value of the input. The truncated value of the scalar x is the nearest integer i which is closer to zero than x is. In short, the fractional part of the signed number x is discarded.

23.6 Array Specific Functions

23.6.1 append(array)

Append values to the end of an array.

23.6.2 array(object)

Create an array from an array_like object, like a Python list or another array.

23.6.3 asarray(object)

Convert the input to an array. a is an array_like object, like a Python list.

23.6.4 arange(start,stop,step)

Return evenly spaced values within a given interval.

Values are generated within the half-open interval `[start, stop)` (in other words, the interval including *start* but excluding *stop*).

23.6.5 average(a)

Compute the arithmetic average (identical to the mean) of an array (returns a scalar). Optionally, the keyword ‘axis’ can be given to specify, for a 2D dataset, if the average should be taken down the columns (axis=0) or across the rows(axis=1).

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

23.6.6 column_stack((a,b,c,d,. . .))

Stack 1D arrays as columns into a 2D array.

23.6.7 copy(object)

Return an array copy of the given object.

23.6.8 concatenate((a1,a2,a3,. . .))

Join a sequence of arrays together.

23.6.9 dot(a,b)

Dot product of two arrays a and b (returns a scalar)

23.6.10 empty(shape)

Return a new array of given shape, without initializing entries. If the new array should be one dimensional, just an integer will suffice for the shape. If the new array should be multidimensional, then my_array = empty((n,3)) is an appropriate example.

23.6.11 eye(n)

Return a 2-D array with ones on the diagonal and zeros elsewhere.

23.6.12 fliplr(a)

Flip (reverse) an array in the left/right direction (columnwise).

23.6.13 flipud(a)

Flip (reverse) an array in the up/down direction (rowwise).

23.6.14 hstack((a,b,c,d,. . . .))

Stack arrays in sequence horizontally (columnwise)

23.6.15 identity(n)

Return the identity array. The identity array is a square array with ones on the main diagonal.

23.6.16 linspace(start,stop,n)

Return evenly spaced numbers over a specified interval.

Returns *n* evenly spaced samples, calculated over the interval [*start*, *stop*].

23.6.17 mean(a)

Compute the arithmetic mean (identical to the average) of an array (returns a scalar). Optionally, the keyword 'axis' can be given to specify, for a 2D dataset, if the mean should be taken down the columns (axis=0) or across the rows(axis=1).

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

23.6.18 median(a)

Compute the median of an array along the specified column. Returns the median of the array elements. Optionally, the keyword 'axis' can be given to specify, for a 2D dataset, if the median should be taken down the columns (axis=0) or across the rows(axis=1).

Given a vector V of length N, the median of V is the middle value of a sorted copy of V, `V_sorted` - i.e., `V_sorted[(N-1)/2]`, when N is odd. When N is even, it is the average of the two middle values of `V_sorted`.

23.6.19 nonzero(a)

Returns as an integer array, the indices of the elements are are nonzero. This can be used for indexing into another array.

23.6.20 ones(shape)

Return a new array of given shape, filled with ones. For example, my_array = ones(n) If the new array should be one dimensional, just an integer will suffice for the shape. If the new array should be multidimensional, then my_array = ones((n,3)) is an appropriate example.

23.6.21 ones_like(a)

Returns an array of ones with the same dimension as the given array.

23.6.22 roll(a)

Roll array elements rowwise; rows that roll beyond the last position are reintroduced at the first.

23.6.23 rand(nrow,ncol)

Returns an array of the given dimension (nrow x ncol) and fill it with random samples from a uniform distribution over [0,1).

23.6.24 sort(a)

Return an ascending sorted copy of the array.

23.6.25 standarddev(a)

Return the standard deviation of an array. Optionally, the keyword 'axis' can be given to specify, for a 2D dataset, if the standard deviation should be taken down the columns (axis=0) or across the rows(axis=1). Also optionally, the keyword 'ddof' can be given to determine the divisor used in the calculations. By default, ddof is 1.

$$\sigma = \sqrt{\frac{1}{n - DOF}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

23.6.26 unique(a)

Find the unique elements of an array; returns the sorted unique elements of an array. Works on only 1D arrays; the array will be flattened if it is not already 1D.

23.6.27 variance(a)

Return the variance of the array. Optionally, the keyword 'axis' can be given to specify, for a 2D dataset, if the variance should be taken down the columns (axis=0) or across the rows(axis=1).

23.6.28 vstack((a,b,c,d,. . . .))

Stack arrays in sequence vertically (rowwise).

23.6.29 zeros(shape)

Return a new array of given shape, filled with zeros. For example, my_array = zeros(n). If the new array should be one dimensional, just an integer will suffice for the shape. If the new array should be multidimensional, then my_array = zeros((n,3)) is an appropriate example.

23.6.30 zeros_like(other_array)

Return an array of zeros with the same dimension as a given array. For example, my_array = zeros_like(x)

23.7 Logical Functions

23.7.1 array_and(b1,b2)

Applies the logical and operator, elementwise, to two boolean arrays b1 and b2. Returns another boolean array.

23.7.2 array_or(b1,b2)

Applies the logical or operator, elementwise, to two boolean arrays b1 and b2. Returns another boolean array.

23.7.3 array_xor(b1,b2)

Applies the logical xor operator, elementwise, to two boolean arrays b1 and b2. Returns another boolean array.

23.7.4 array_not(b1)

Applies the logical not operator, elementwise, to a boolean array b1. Returns another boolean array.

23.7.5 array_gt(x1,x2)

Returns a boolean array that tests x1 > x2 elementwise.

23.7.6 array_lt(x1,x2)

Returns a boolean array that tests x1 < x2 elementwise.

23.7.7 array_le(x1,x2)

Returns a boolean array that tests x1 <= x2 elementwise.

23.7.8 array_ge(x1,x2)

Returns a boolean array that tests x1 >= x2 elementwise.

23.7.9 array_eq(x1,x2)

Returns a boolean array that tests x1 == x2 elementwise.

23.7.10 array_neq(x1,x2)

Returns a boolean array that tests x1 != x2 elementwise.

23.7.11 array_almost_eq(x1,x2, rtol, atol)

Tests two arrays x1 and x2, elementwise, to see if they are nearly equal within the tolerances specified in rtol and atol. Returns a boolean array. rtol is a relative tolerance, and atol is an absolute tolerance. To determine closeness, the relative difference rtol*abs(b) and the absolute difference atol are added together to compare against the absolute difference between the elements of a and b.

CHAPTER

TWENTYFOUR

APPENDIX B: MATH SYMBOL REFERENCE

24.1 Subscripts and superscripts

Command	Result
$alpha_i > beta_i$	$\alpha_i > \beta_i$
$alpha^2 > beta_i^a$	$\alpha^2 > \beta_i^a$

24.2 Fractions, binomials and stacked numbers

```
$\frac{3}{4}$
```

$$\frac{3}{4}$$

```
$\binom{3}{4}$
```

$$\binom{3}{4}$$

```
$\stackrel{3}{4}$
```

$$\stackrel{3}{4}$$

```
$\frac{5 - \frac{1}{x}}{4}$
```

$$\frac{5 - \frac{1}{x}}{4}$$

24.3 Using parenthesis properly

```
$(\frac{5 - \frac{1}{x}}{4})$
```

$$(\frac{5 - \frac{1}{x}}{4})$$

Notice how the parenthesis do not properly wrap the object inside. To address this, use the \left and \right qualifiers in front of your parenthesis.

```
$\left(\frac{5 - \frac{1}{x}}{4}\right)$
```

$$\left(\frac{5 - \frac{1}{x}}{4}\right)$$

24.4 Radicals

```
$\sqrt{2}$
```

$$\sqrt{2}$$

```
$\sqrt[3]{x}$
```

$$\sqrt[3]{x}$$

24.5 Accents

Command	Result
\acute a or \'a	$\acute{a}$
\bar a	$\bar{a}$
\breve a	$\breve{a}$
\ddot a or \"a	$\ddot{a}$
\dot a or \.a	$\dot{a}$
\grave a or \`a	$\grave{a}$
\hat a or \^a	$\hat{a}$
\tilde a or \~a	$\tilde{a}$
\vec a	$\vec{a}$
\widehat{xyz}	$\widehat{xyz}$
\widetilde{xyz}	$\widetilde{xyz}$

24.6 Symbols

Lower-case Greek

α \alpha	β \beta	χ \chi	δ \delta	$\digamma$ \digamma
ϵ \epsilon	η \eta	γ \gamma	ι \iota	κ \kappa
λ \lambda	μ \mu	ν \nu	ω \omega	ϕ \phi
π \pi	ψ \psi	ρ \rho	σ \sigma	τ \tau
θ \theta	υ \upsilon	ε \varepsilon	$\varkappa$ \varkappa	φ \varphi
ϖ \varpi	ϱ \varrho	ς \varsigma	ϑ \vartheta	ξ \xi
ζ \zeta				

Upper-case Greek

Δ \Delta	Γ \Gamma	Λ \Lambda	Ω \Omega	Φ \Phi	Π \Pi
Ψ \Psi	Σ \Sigma	Θ \Theta	ϒ \Upsilon	Ξ \Xi	℧ \mho
∇ \nabla					

Hebrew

ℵ \aleph	ℶ \beth	ℸ \daleth	ℷ \gimel

Delimiters

/ /	[[	⇓ \Downarrow	⇑ \Uparrow	‖ \Vert	\ \backslash
↓ \downarrow	⟨ \langle	⌈ \lceil	⌊ \lfloor	⌞ \llcorner	⌟ \lrcorner
⟩ \rangle	⌉ \rceil	⌋ \rfloor	⌜ \ulcorner	↑ \uparrow	⌝ \urcorner
| \vert	{ \{	‖ \\|	} \}	]]	| |

Big symbols

⋂ \bigcap	⋃ \bigcup	⨀ \bigodot	⨁ \bigoplus	⨂ \bigotimes
⨄ \biguplus	⋁ \bigvee	⋀ \bigwedge	∐ \coprod	∫ \int
∮ \oint	∏ \prod	∑ \sum		

Standard function names

Pr \Pr	arccos \arccos	arcsin \arcsin	arctan \arctan
arg \arg	cos \cos	cosh \cosh	cot \cot
coth \coth	csc \csc	deg \deg	det \det
dim \dim	exp \exp	gcd \gcd	hom \hom
inf \inf	ker \ker	lg \lg	lim \lim
lim inf \liminf	lim sup \limsup	ln \ln	log \log
max \max	min \min	sec \sec	sin \sin
sinh \sinh	sup \sup	tan \tan	tanh \tanh

Binary operation and relation symbols

≎ \Bumpeq	⋒ \Cap	⋓ \Cup
≑ \Doteq	⋈ \Join	⋐ \Subset
⋑ \Supset	⊩ \Vdash	⊪ \Vvdash
≈ \approx	≊ \approxeq	∗ \ast
≍ \asymp	϶ \backepsilon	∽ \backsim
⋍ \backsimeq	⊼ \barwedge	∵ \because
≬ \between	◯ \bigcirc	▽ \bigtriangledown
△ \bigtriangleup	◀ \blacktriangleleft	▶ \blacktriangleright
⊥ \bot	⋈ \bowtie	⊡ \boxdot
⊟ \boxminus	⊞ \boxplus	⊠ \boxtimes
• \bullet	≏ \bumpeq	∩ \cap
· \cdot	∘ \circ	≗ \circeq
≔ \coloneq	≅ \cong	∪ \cup
⋞ \curlyeqprec	⋟ \curlyeqsucc	⋎ \curlyvee
⋏ \curlywedge	† \dag	⊣ \dashv
‡ \ddag	⋄ \diamond	÷ \div
⋇ \divideontimes	≐ \doteq	≑ \doteqdot
∔ \dotplus	⩞ \doublebarwedge	≖ \eqcirc
≕ \eqcolon	≂ \eqsim	⪖ \eqslantgtr
⪕ \eqslantless	≡ \equiv	≒ \fallingdotseq

⌢ \frown	≥ \geq	≧ \geqq
⩾ \geqslant	≫ \gg	⋙ \ggg
⪊ \gnapprox	≩ \gneqq	⋧ \gnsim
⪆ \gtrapprox	⋗ \gtrdot	⋛ \gtreqless
⪌ \gtreqqless	≷ \gtrless	≳ \gtrsim
∈ \in	⊺ \intercal	⋋ \leftthreetimes
≤ \leq	≦ \leqq	⩽ \leqslant
⪅ \lessapprox	⋖ \lessdot	⋚ \lesseqgtr
⪋ \lesseqqgtr	≶ \lessgtr	≲ \lesssim
≪ \ll	⋘ \lll	⪉ \lnapprox
≨ \lneqq	⋦ \lnsim	⋉ \ltimes
∣ \mid	⊧ \models	∓ \mp
⊯ \nVDash	⊮ \nVdash	≉ \napprox
≇ \ncong	≠ \ne	≠ \neq
≠ \neq	≢ \nequiv	≱ \ngeq
≯ \ngtr	∋ \ni	≰ \nleq
≮ \nless	∤ \nmid	∉ \notin
∦ \nparallel	⊀ \nprec	≁ \nsim
⊄ \nsubset	⊈ \nsubseteq	⊁ \nsucc
⊅ \nsupset	⊉ \nsupseteq	⋪ \ntriangleleft

⋬ \ntrianglelefteq	⋫ \ntriangleright	⋭ \ntrianglerighteq
⊭ \nvDash	⊬ \nvdash	⊙ \odot
⊖ \ominus	⊕ \oplus	⊘ \oslash
⊗ \otimes	∥ \parallel	⊥ \perp
⋔ \pitchfork	± \pm	≺ \prec
⪷ \precapprox	≼ \preccurlyeq	⪯ \preceq
⪹ \precnapprox	⋨ \precnsim	≾ \precsim
∝ \propto	⋌ \rightthreetimes	≓ \risingdotseq
⋊ \rtimes	∼ \sim	≃ \simeq
/ \slash	⌣ \smile	⊓ \sqcap
⊔ \sqcup	⊏ \sqsubset	⊏ \sqsubset
⊑ \sqsubseteq	⊐ \sqsupset	⊐ \sqsupset
⊒ \sqsupseteq	⋆ \star	⊂ \subset
⊆ \subseteq	⫅ \subseteqq	⊊ \subsetneq
⫋ \subsetneqq	≻ \succ	⪸ \succapprox
≽ \succcurlyeq	⪰ \succeq	⪺ \succnapprox
⋩ \succnsim	≿ \succsim	⊃ \supset
⊇ \supseteq	⫆ \supseteqq	⊋ \supsetneq
⫌ \supsetneqq	∴ \therefore	× \times
⊤ \top	◂ \triangleleft	⊴ \trianglelefteq

≜ \triangleq	▸ \triangleright	⊵ \trianglerighteq
⊎ \uplus	⊨ \vDash	∝ \varpropto
⊲ \vartriangleleft	⊳ \vartriangleright	⊢ \vdash
∨ \vee	⊻ \veebar	∧ \wedge
≀ \wr		

Arrow symbols

⇓ \Downarrow	⇐ \Leftarrow
⇔ \Leftrightarrow	⇚ \Lleftarrow
⟸ \Longleftarrow	⟺ \Longleftrightarrow
⟹ \Longrightarrow	↰ \Lsh
⇗ \Nearrow	⇖ \Nwarrow
⇒ \Rightarrow	⇛ \Rrightarrow
↱ \Rsh	⇘ \Searrow
⇙ \Swarrow	⇑ \Uparrow
⇕ \Updownarrow	↺ \circlearrowleft
↻ \circlearrowright	↶ \curvearrowleft
↷ \curvearrowright	⇠ \dashleftarrow
⇢ \dashrightarrow	↓ \downarrow
⇊ \downdownarrows	⇃ \downharpoonleft
⇂ \downharpoonright	↩ \hookleftarrow
↪ \hookrightarrow	⇝ \leadsto
← \leftarrow	↢ \leftarrowtail
↽ \leftharpoondown	↼ \leftharpoonup
⇇ \leftleftarrows	↔ \leftrightarrow
⇆ \leftrightarrows	⇋ \leftrightharpoons
↭ \leftrightsquigarrow	⇜ \leftsquigarrow

⟵ \longleftarrow	⟷ \longleftrightarrow
⟼ \longmapsto	⟶ \longrightarrow
↫ \looparrowleft	↬ \looparrowright
↦ \mapsto	⊸ \multimap
⇍ \nLeftarrow	⇎ \nLeftrightarrow
⇏ \nRightarrow	↗ \nearrow
↚ \nleftarrow	↮ \nleftrightarrow
↛ \nrightarrow	↖ \nwarrow
→ \rightarrow	↣ \rightarrowtail
⇁ \rightharpoondown	⇀ \rightharpoonup
⇄ \rightleftarrows	⇄ \rightleftarrows
⇌ \rightleftharpoons	⇌ \rightleftharpoons
⇉ \rightrightarrows	⇉ \rightrightarrows
⇝ \rightsquigarrow	↘ \searrow
↙ \swarrow	→ \to
↞ \twoheadleftarrow	↠ \twoheadrightarrow
↑ \uparrow	↕ \updownarrow
↕ \updownarrow	↿ \upharpoonleft
↾ \upharpoonright	⇈ \upuparrows

Miscellaneous symbols

$ \$	Å \AA	Ⅎ \Finv
⅁ \Game	ℑ \Im	¶ \P
ℜ \Re	§ \S	∠ \angle
‵ \backprime	★ \bigstar	■ \blacksquare
▲ \blacktriangle	▼ \blacktriangledown	⋯ \cdots
✓ \checkmark	® \circledR	Ⓢ \circledS
♣ \clubsuit	∁ \complement	© \copyright
⋱ \ddots	◇ \diamondsuit	ℓ \ell
∅ \emptyset	ð \eth	∃ \exists
♭ \flat	∀ \forall	ℏ \hbar
♡ \heartsuit	ℏ \hslash	∭ \iiint
∬ \iint	∬ \iint	ı \imath
∞ \infty	ȷ \jmath	… \ldots
∡ \measuredangle	♮ \natural	¬ \neg
∄ \nexists	∰ \oiiint	∂ \partial
′ \prime	♯ \sharp	♠ \spadesuit
∢ \sphericalangle	\ss	▽ \triangledown
∅ \varnothing	△ \vartriangle	⋮ \vdots
℘ \wp	¥ \yen	

CHAPTER

TWENTYFIVE

APPENDIX C: THEORY

25.1 Linear Regression

Models which consist of a linear combination of a particular set of functions $\mathrm{X_k}$ are called linear models, and linear regression can be used to minimize the difference between the model and data. The general form of this kind of model is

$$y(x) = \sum a_k X_k(x)$$

where $\mathrm{X_k(x)}$ are fixed functions of x that are called the basis functions, and $\mathrm{a_k}$ are the free parameters. Note that "linear" refers only to dependence of the model on the parameters $\mathrm{a_k}$; the functions $\mathrm{X_k(x)}$ may be nonlinear. Minimization of the above linear model is performed with respect to the merit function

$$S(a) = \sum_{i=1}^{n} \left[y_i - \sum_{k=1}^{np} a_k X_k(x_i) \right]^2$$

The minimum of the above equation occurs where the derivative of S with respect to the parameters disappears. Substituting the linear model into this function, taking the first derivative, and setting this equal to zero gives the normal equations that can be solved directly for the parameters $\mathrm{a_k}$.

25.2 Nonlinear Regression

25.2.1 Introduction

This program uses the Levenberg-Marquardt method to solve nonlinear regressions. This method combines the steepest-descent method and a Taylor series based method to obtain a fast, reliable technique for nonlinear optimization. This statement will be repeated later, but can bear stating at the beginning of this discussion: neither of the above optimization methods are ideal all of the time; the steepest descent method works best far away from the minimum, and the Taylor series method works best close to the minimum. The Levenberg-Marquardt (LM) algorithm allows for a smooth transition between these two methods as the iteration proceeds.

25.2.2 Development

In general, the data modeling equation (with any number of independent variables) can be written as follows:

$$y = y(\vec{x}; \vec{a})$$

The above expression simply states that the dependent variable y can be expressed as a function of the independent variables $\vec{x}$ and vector of parameters $\vec{a}$ of arbitrary length. Note that using the ML method, any nonlinear equation with an arbitrary number of parameters can be used as the data modeling equation. Then, the merit function we are trying to minimize is

$$\chi^2(\vec{a}) = \sum_{i=1}^{n} \left(\frac{y_i - y(\vec{x}_i; \vec{a})}{\sigma_i} \right)^2$$

where N is the number of data points, $\vec{x}_i$ denotes the x data points, y_i denotes the y data points, σ_i is the standard deviation (uncertainty) at point i, and $y(x_i; a)$ is an arbitrary nonlinear model evaluated at the ith data point. This merit function simply measures the agreement between the data points and the parametric model; a smaller value for the merit function denotes better agreement. Commonly, this merit function is called the chi-square.

Now, we will take a step back to provide a framework of optimization methods. From the area of pure optimization, which will not be explained here, two basic ways of finding a function minimum are a Taylor series based method and the steepest-descent method. The Taylor series method states that sufficiently close to the minimum, the function can be approximated as a quadratic. Without detailed explanation of that method, a step from the current parameters a to the best parameters amin can be written as

$$\vec{a}_{min} = \vec{a}_{cur} + H^{-1} \left[-\Delta\chi^2(\vec{a}_{cur}) \right]$$

where H is the Hessian matrix (a matrix of second derivatives). If the approximation of the function as a quadratic is a poor one, then we might instead use the steepest-descent method, where a step to the best parameters from the current parameters is

$$\vec{a}_{min} = \vec{a}_{cur} - c\Delta\chi^2(\vec{a}_{cur})$$

This equation simply states that the next guess for the parameters is a step down the gradient of the merit function. The constant c is forced to be small enough that a small step is taken and the gradient is accurate in the region that the step is taken.

Since we know the chi-square function, we can directly differentiate to obtain the gradient vector and the Hessian matrix. Taking the partial derivatives of the merit function with respect to a gives

$$\frac{\partial\chi^2}{\partial a_k} = -2 \sum_{i=1}^{n} \frac{y_i - y(\vec{x}_i; \vec{a})}{\sigma_i^2} \frac{\partial y(\vec{x}_i; \vec{a})}{\partial a_k}$$

To obtain the Hessian matrix, take the gradient of the gradient above (so that we have a matrix of partial second derivatives):

$$\frac{\partial^2\chi^2}{\partial a_k \partial a_l} = -2 \sum_{i=1}^{n} \left[\frac{1}{\sigma_i^2} \frac{\partial y(\vec{x}_i; \vec{a})}{\partial a_k} \frac{\partial y(\vec{x}_i; \vec{a})}{\partial a_l} - \frac{y_i - y(\vec{x}_i; \vec{a})}{\sigma_i^2} \frac{\partial^2 y(\vec{x}_i; \vec{a})}{\partial a_l \partial a_k} \right]$$

Now, for convenience, define the gradient vector and the curvature matrix as

$$G_k = -\frac{1}{2} \frac{\partial\chi^2}{\partial a_k} = \sum_{i=1}^{n} \left[\frac{y_i - y(\vec{x}_i; \vec{a})}{\sigma_i^2} \frac{\partial y(\vec{x}_i; \vec{a})}{\partial a_k} \right]$$

$$C_{kl} = \frac{\partial^2\chi^2}{\partial a_k \partial a_l} = \sum_{i=1}^{n} \left[\frac{1}{\sigma_i^2} \frac{\partial y(\vec{x}_i; \vec{a})}{\partial a_k} \frac{\partial y(\vec{x}_i; \vec{a})}{\partial a_l} \right]$$

[Note that the second derivative term in C will be ignored because of two reasons: it tends to be small because it is multiplied by $(y - y_i)$, and it tends to destabilize the algorithm for badly fitting models or data sets contaminated with outliers. This action in no way affects the minimum found by the algorithm; it only affects the route in getting there.] So, the Taylor series method (inverse Hessian method) can be written as the following set of linear equations:

$$\sum_{k=1}^{np} C_{kl}\delta a_l = G_k[1]$$

where NP is the number of parameters in the model that is being optimized. This linear matrix will be our workhorse for this method after some modification; it can be solved for the increments da that, when added to the current approximation for the parameters, gives the next approximation. Likewise, we can substitute our “convenient” definitions into the steepest descent formula to obtain

$$\delta a_l = cG_l[2]$$

Neither of the aforementioned optimization methods are ideal all of the time; the steepest descent method works best far away from the minimum, and the Taylor series method works best close to the minimum. The Levenberg-Marquardt (LM) algorithm allows for a smooth transition between these two methods as the iteration proceeds.

The first issue in deriving the LM method is to attach some sort of scale to the constant c in the steepest-gradient method (equation 2). Typically, there is no obvious way to determine this number, even within an order of magnitude. However, in this case, we have access to the Hessian matrix; examining its members, we see that the scale on this constant must be $1/C_{ll}$. But, that still may be too large, so let’s divide that scale by a nondimensional factor (l) and plan on setting this much larger than one so that the step will be reduced (for safety and stability).

The second issue to formulate the LM method is noting that the steepest-descent and Taylor series methods may be combined if we define a new matrix Mij by the following:

$$M_{ii} = C_{ii}(1 + \lambda), i = j$$

$$M_{ij} = C_{ij}, i \neq j$$

This matrix combines equations [1] and [2] into a convenient and compact form. So finally, we have a means of calculating the step da in the parameters by the following system of linear equations:

$$\sum_{k=1}^{np} M_{kl}\delta a_l = G_k[3]$$

Note that when l is large, the matrix M is forced to be diagonally dominant; consequently, the above equation is equivalent to the steepest descent method (equation 2). Conversely, when the parameter l goes to zero, the above equation is equivalent to the Taylor series method (equation 1). So, we vary l to switch between the two methods, continually calculating a parameter correction da that we apply to our most recent guess for the parameter vector.

25.2.3 Procedure

The steps that are taken in the LM algorithm are as follows:

1. Compute $\chi^2(a)$
2. Pick a conservative value for λ (0.001 in CurveExpert)
3. Solve the linear equations (equation 3) for δa
4. Evaluate $\chi^2(a + \delta a)$
5. If $\chi^2(a + \delta a) >= \chi^2(a)$, increase λ by a factor (10 in CurveExpert) and go back to step [3].
6. If $\chi^2(a + \delta a) < \chi^2(a)$, decrease λ by a factor (10 in CurveExpert), correct the parameter vector by $a = a + \delta a$, and go back to step [3].

Iteration is stopped when $|\chi^2(a + \delta a) - \chi^2(a)| < \epsilon$. This tolerance ϵ is the value specified by the user.

CHAPTER

TWENTYSIX

APPENDIX D: REFERENCES

1. Bevington, Philip R., Data Reduction and Error Analysis for the Physical Sciences, 1969, McGraw-Hill, Inc., Library of Congress QA278. B48.
2. Cleveland, William S.: "Robust locally weighted regression and smoothing scatterplots", Journal of the American Statistical Association, December 1979, volume 74, number 368, pp. 829-836.
3. Cleveland, William S., and Devlin, Susan J.: "Locally weighted regression: An approach to regression analysis by local fitting", Journal of the American Statistical Association, September 1988, volume 83, number 403, pp. 596-610.
4. Daniel and Wood, Fitting Equations to Data, 1971. Wiley & Sons, New York. Library of Congress QA297.D35.
5. Neter, Wasserman, and Kutner, Applied Linear Regression Models, 2nd ed. 1989, Irwin Press, Boston, MA. Library of Congress QA278.2 .N46.
6. Press, Flannery, Teukolsky, and Vetterling, Numerical Recipes in Pascal, 1989. Cambridge University Press, New York. Library of Congress QA76.73 .P2N87. Seber and Wild, Nonlinear Regression, 1989. Wiley & Sons, New York. Library of Congress QA278.2 .S425.
7. Ratkowsky, David A., Handbook of Nonlinear Regression Models. 1990. Marcel Dekker, Inc., New York. Library of Congress QA278.2 .R369.
8. Ratkowsky, David A., Nonlinear Regression Modeling: A Unified Practical Approach. 1983. Marcel Dekker, Inc., New York. Library of Congress QA278.2 .R37.

CHAPTER

TWENTYSEVEN

APPENDIX E: PLUGIN API

27.1 Basic API

`api.MainWin()`

Returns a handle to the main window. Can be useful if using graphical widgets from the plugin.

Return type wx.Window

`api.calculate_function(`*abspath*, *relpath*, *data=None*, *add_to_ui=True*`)`

Compute a function from the model given by the abspath/relpath pair, and with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **abspath** (*string*) – can be the strings "$BUILTIN$", "$USERDIR$", or a fully qualified path the location of the function library.
- **relpath** (*string*) – "libf/2D/Bessel/Bessel_j0.py", for example, which is the rest of the path to the actual file that implements the function.
- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.

Return type result object

`api.calculate_linear_regression(`*degree*, *data=None*, *add_to_ui=True*, *weighting=0*`)`

Compute a linear regression of the given degree, and with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **degree** (*integer*) – Degree of the polynomial to compute a linear regression for.
- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.
- **weighting** (*integer*) – weighting parameter

Return type result object

The return value is a result object, which contains all of the relevant information about that result.

`api.calculate_lowess_smoothing(`*data=None*, *add_to_ui=True*`)`

Compute a Lowess smoothing with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.

Return type result object

The return value is a result object, which contains all of the relevant information about that result.

api.**calculate_movingaverage**(*windowsize*, *windowtype='rect'*, *windowpos='centered'*, *windowqual=0*, *data=None*, *add_to_ui=True*)

Compute a moving average with the given parameters and with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **windowsize** (*integer*) – The total extent of the window, from left to right. The extent can be given as a number of points or a distance along the x axis. See windowqual below.
- **windowqual** (*integer*) – If the windowsize is specified as a distance along the x axis, windowqual should be set to 0. If the windowsize is given as a number of points, windowqual should be set to 1.
- **windowtype** (*integer*) – The function that is used as the weighting function. Should be one of the following: "rect", "bartlett", "blackman", "hamming", or "hanning".
- **windowpos** (*integer*) – The position of the window in relation to zero (the centerpoint of the averaging). Should be one of "centered", "leading", or "lagging".
- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.

Return type result object

The return value is a result object, which contains all of the relevant information about that result; returns None on error.

api.**calculate_nonlinear_regression**(*abspath*, *relpath*, *data=None*, *add_to_ui=True*, *weighting=0*)

Compute a nonlinear regression from the model given by the abspath/relpath pair, and with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **abspath** (*string*) – can be the strings "$BUILTIN$", "$USERDIR$", or a fully qualified path the location of the model library.
- **relpath** (*string*) – "lib/2D/Sigmoidal Models/Weibull.py", for example, which is the rest of the path to the actual file that implements the model.
- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.
- **weighting** (*integer*) – weighting parameter

Return type result object

api.**calculate_polynomial_spline**(*degree*, *data=None*, *add_to_ui=True*)

Compute a polynomial spline of the given degree, and with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **degree** (*integer*) – Degree of the polynomials used for the splining.
- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.

Return type result object

The return value is a result object, which contains all of the relevant information about that result.

api.**calculate_savitzky_golay_smoothing**(*data=None*, *window_size=15*, *degree=4*, *add_to_ui=True*)

Compute a Savitzky-Golay smoothing with the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **window_size** (*integer*) – the number of points in the averaging window, in number of points; must be odd.
- **degree** (*integer*) – the degree of the polynomial that will be used to fit the data within the window. Must be less than the number of points in the window.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.

Return type result object

The return value is a result object, which contains all of the relevant information about that result.

api.**calculate_tensionspline**(*tension*, *data=None*, *add_to_ui=True*)

Compute a Tension smoothing with the given tension parameter and the given data. If data is not given, the data currently in the CurveExpert Pro spreadsheet will be used.

Parameters

- **tension** (*float*) – tension parameter for the spline.
- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **add_to_ui** (*boolean*) – whether or not to add the result to the CurveExpert Pro result list.

Return type result object

The return value is a result object, which contains all of the relevant information about that result.

api.**clear**()

Clears all results and the spreadsheet. Functionally the same as the user selecting File->New.

Return type None

api.**clear_message_window**()

Clears all text from the message window

api.**create_fontproperties**(*font='Arial'*, *size=12*, *weight='normal'*, *style='normal'*)

This API creates a FontProperties object for some particular use (for example, a font for the text in the graph's add_annotation API).

To create a FontProperties object, simply do, for example:

fp = api.create_fontproperties(font='TheFontName',weight='normal',style='normal',size=12)

Parameters

- **font** (*string*) – The font given can be any font on your system that you desire to use. The default is "Arial".
- **size** (*integer*) – The size of the font, in points. The default is 12.
- **weight** (*string*) – The weight of the font; can be one of "ultralight", "light", "normal", "regular, "book", "medium", "roman", "semibold", "demibold", "demi", "bold", "heavy", "extra bold", or "black". The default is "normal".
- **style** (*string*) – The style ofthe font; can be one of "normal", "italic", or "oblique". The default is "normal".

api.**curvefinder**(*data=None*, *nonlinreg_builtin=True*, *nonlinreg_custom=True*, *linear=True*, *poly=4, 8*, *lowess=False*, *savgol=False*, *parent_window=None*, *progress_caption=''*, *silent=False*, *do_multicore=True*, *weighting=0*)

Run a batch of regressions against a particular dataset.

Parameters

- **data** (*2D numpy array*) – numpy array, or None to use the data currently in the spreadsheet.
- **nonlinreg_builtin** (*boolean*) – Search through all built-in nonlinear regressions.
- **nonlinreg_custom** (*boolean*) – Search through all custom (user defined) nonlinear regressions.
- **linear** (*boolean*) – Include a linear regression in the search (straight line, y=ax+b)
- **poly** (*boolean or 2-tuple, as appropriate*) – Search a range of polynomial regressions. If polynomials are desired, this parameter should be set to a tuple indicating the min and max degree of the polynomials to calculate. Otherwise False.
- **lowess** (*boolean*) – Include a Lowess smoothing in the search
- **savgol** (*boolean*) – Include a Savitzky-Golay smoothing in the search; uses 5% of the available points as the averaging window, and a quartic polynomial for fitting within the windows.
- **parent_window** (*wxWindow object*) – If a progress dialog is desired, the parent_window for the progress dialog is given here.
- **progress_caption** (*string*) – Title of the progress dialog that is shown.
- **silent** (*boolean*) – If set to True, will not print any intermediate information to the CurveExpert Pro messages window.
- **do_multicore** (*boolean*) – If set to True, multicore capability will be used if available.
- **weighting** (*integer*) – Choice of default weighting (api.WEIGHTING.DEFAULT), no weighting (api.WEIGHTING.NONE), by y (api.WEIGHTING.Y) by y^2 (api.WEIGHTING.Y2), by x (api.WEIGHTING.X), or by x^2 (api.WEIGHTING.X2). Default weighting means to use no weighting unless there is a standard deviation column, in which case the weighting is the square of the standard deviation.

Return type list of result objects, sorted in order of their score.

The return value is a list of result objects, which contains all of the relevant information about that result.

api.**extract_from_excel**(*file_name*, *sheet_name*, *range_spec*, *stdcol=False*)

Reads a data from an Excel file (xls or xlsx).

Parameters

- **file_name** (*string*) – file to read
- **sheet_name** (*string*) – the name of the sheet where the data resides.
- **range_spec** (*string*) – the range of cells in Excel where the data is located. For example, "A1:B10", or "A1:B10,D1:D10".
- **stdcol** (*boolean*) – True if the last column of the data in the spreadsheet represents the standard deviation column.

Return type numpy array

On error, this routine will return None.

api.**get_all_graphs**()

Returns a list of all graphs available (a list of Graph objects).

api.**get_all_results**()

Returns a list of all of the currently computed results; these are the same results as shown in the Results Pane.

Return type list of result objects

api.**get_current_graph**()

Returns a Graph object for the currently displayed graph. If there is no graph currently showing (e.g. the spreadsheet page or notes page is active), this call will return None.

api.**get_data**()

Returns a reference to the dataset shown in the spreadsheet. One can both read and write to this dataset, but its size should not be changed. If the dataset is changed (written to), the API refresh_spreadsheet() must be called once all desired data has been modified.

Return type numpy array

api.**get_main_window**()

Returns a handle to the main window. Can be useful if using graphical widgets from the plugin.

Return type wx.Window

api.**get_spreadsheet_selection**()

Gets the currently selected rectangle in the spreadsheet. If there is no selection, all four members of the tuple will be -1. Note that the numbers are all zero based, so a top left of 0,0 means row 1, column 1.

Return type tuple (top,left,bottom,right)

api.**import_from_clipboard**()

Reads data from the clipboard, importing it with default settings.

Return type None.

api.**import_from_file**(*fname*)

Identical to read_file().

api.**message**(*msg*, *importance=0*)

Print a message to the CurveExpert Pro message pane.

Parameters

- **msg** (*string*) – string that contains the text to send to the message pane
- **importance** (*integer*) – the importance level of the string. Can be set to 0, 1, or 2. Currently, an importance of 0 prints in black, 1 in red, and 2 in blue.

Return type None

api.**messages**(*msgs*, *importance=0*)

Print a batch of messages to the CurveExpert Pro message pane.

Parameters

- **msg** (*list of strings*) – list of strings that contains the text to send to the message pane
- **importance** (*integer*) – the importance level of the string. Can be set to 0, 1, or 2. Currently, an importance of 0 prints in black, 1 in red, and 2 in blue.

Return type None

api.**new_graph**(*name=None*, *style=None*)

Creates a new graph and makes it current. Returns a Graph object.

Parameters

- **name** (*string*) – name of the plot; used for the notebook tab that the graph appears in.
- **style** (*integer*) – The style of the plot. XY, 3D, or polar. If not given, an XY plot is created.

api.**quit**()

Shut down CurveExpert Pro.

Return type None

api.**read_file**(*fname*)

Reads a file from disk. The file can be either a cxp file or a text/data file. If a text/data file, it will be imported with default settings. The data read is placed in the CurveExpert Pro spreadsheet; if the data is needed for manipulation in the plugin, see get_data().

Parameters **fname** (*string*) – file to read

Return type None

api.**refresh_spreadsheet**()

If the data is changed (via calling get_data() and modifying the data), the plugin should call refresh_spreadsheet() in order to notify CurveExpert Pro about the change.

Return type None

api.**remove_graph**(*graph*)

Removes a graph from the stack of graphs.

Parameters **graph** – a Graph object that is the one to remove.

api.**reset_with_data**(*data*, *stddev=False*)

Reinitializes the CurveExpert application with a new dataset. Functionally the same as reading a data file, except that the dataset can be generated inside of a plugin by the user. As such, all current results will be lost.

Parameters

- **data** (*numpy array*) – array of data, which must have at least two columns.
- **stddev** (*bool*) – flag to state whether or not the last columns is standard deviation data (False by default)

api.**result_picker**()

Shows a dialog box from which a result can be selected.

Return type list of selected result objects.

api.**set_current_graph**(*graph*)

Causes a graph be made current; i.e.; made visible in the notebook.

Parameters **graph** – a Graph object that is the one to make current (bring to the front).

`api.`**`set_data`**(*data*)

Sets a new dataset into the CurveExpert application with a new dataset. The new dataset must have the same number of columns as the current dataset for this call to succeed.

Returns zero on success, nonzero on error.

Parameters **`data`** (*`numpy array`*) – array of data, which must have the same number of columns as the current dataset

Return type integer

`api.`**`text_to_clipboard`**(*txt*)

Copy any text string that you have to the clipboard.

Parameters **`txt`** (*`string`*) – string that contains the text to copy to the clipboard

Return type None

`api.`**`update_ui`**()

Allows the user interface to update itself. Useful for flushing text out to the messages window after calls to api.message.

Return type None

27.2 Result Object

`class CommonLib.Result.`**`Result`**(*name='Unnamed'*, *kind='Regression'*, *family='NoFamily'*, *id=0*)

The Result object holds all information concerning any result calculated in CurveExpert/GraphExpert Professional. As such, there are some fields that are not applicable for all results. As a user of the API, you will not be creating result objects directly, but you will receive them as a return value of the result calculation API's, such as calculate_nonlinear_regression.

`name`

The name of the result

`kind`

The kind of the result. Currently, can be Regression, Interpolation, or Smoothing

`family`

The family of the result. For example, "Sigmoidal Models".

`spec`

The specification of the result, which is a dotted string used to uniquely identify the result. For example, regression.nonlinear.Exponential Models.Reciprocal Logarithm

`nindvar`

The number of independent variables that the result is applicable for.

`nparam`

The number of parameters. For example, for a regression model a*x^2 + b*x + c, nparam is 3.

`param`

The optimized values of the parameters.

`param_names`

A list of the parameter names, in the same order as the value listed in the param attribute.

`valid`

Flag to denote whether or not the result is clean; meaning that the data has not change since the result has been computed.

`score`
The score compute for this result, usually shown in the "Score" column in the Results pane.

`corr_coeff`
The coefficient of determination, if applicable.

`real_corr_coeff`
The correlation coefficient, if applicable.

`covariance_matrix`
The covariance matrix of the result as a 2D array, if applicable.

`log`
A list of informational strings generated during the calculation of the result that can be useful in some circumstances.

`errlog`
A list of strings generated during the calculation of the result that, if present, indicate that something has gone awry with the result's calculation.

`covariance_matrix`
The covariance matrix of the result as a 2D array, if applicable.

`status`
Flag to denote the status of the result, as far as its success or failure during computation.

status = -1: uninitialized

status = 0: computed and valid

status = 1: unable to compute (i.e. not enough data points for the model)

status = 2: unable to get an initial guess

status = 3: diverged

status = 4: did not converge; use results with caution.

`eval_function`(*x*, **params*)
evaluates the result using the independent variable(s) and the set of parameters given in the argument list. Note that functions have no parameters. x can be either a single floating point number, or a numpy array. It must have the appropriate number of columns that matches the number of independent variables appropriate for the result.

Parameters

- **x** (*float or array*) – independent variable with which to evaluate the result
- **params** (*float*) – argument list for parameters

Return type the result evaluation. The type and dimension matches the type passed for x.

For example, to evaluate a model that has three parameters,

r.eval_function(x,a,b,c)

where a, b, and c are the values of the three parameters. To evaluate a result with its current set of parameters,

r.eval_function(x,*r.param)

`arclength`(*a*, *b*)
Find the arclength of a result from limits a to b. Supports only functions with one independent variable. If b < a, returns the negative of the arclength.

Parameters

- **a** (*float*) – starting point for the arclength calculation
- **b** (*float*) – ending point for the arclength calculation

Return type the arclength from a to b (float)

`differentiate`(*x*)

Differentiate a result at x. Supports only functions with one independent variable.

Parameters **x** (*float or numpy array*) – location at which to differentiate

Return type the result of the differentiation, as the same type as x.

`integrate`(*a*, *b*)

Integrate a result from limits a to b. Supports only functions with one independent variable.

Parameters

- **a** (*float*) – lower limit of the integration
- **b** (*float*) – upper limit of the integration

Return type the result of the integration (float)

`inverse_eval`(*y*, *x0*)

Find x at which a result is equal to y. Supports only functions with one independent variable.

Parameters

- **y** (*float*) – location at which to find the corresponding x.
- **x0** (*float*) – initial guess for the corresponding x.

Return type the result of the inverse evaluation (float)

Note that this function can raise a RuntimeError exception error if the algorithm does not converge.

`parameters_to_csv`(*delimiter=','*)

Generates a CSV representation of the result’s parameters.

Parameters **delimiter** (*string*) – Delimiter to place between parameter name and value.

Return type string

Example usage (if r is a result object, with a space delimiter):

txt = r.parameters_to_csv(delimiter=’ ‘)

`report`(*confidence=95*, *delimiter=' '*)

Generates a text report for the result, returning that text report as a single string.

Parameters **confidence** (*integer*) – In cases where the confidence level is required (for example, computing the uncertainty in a parameter of a nonlinear regression)

Return type string

Example usage (if r is a result object):

txt = r.report()

27.3 Graph Object

class CommonLib.UGraph.**Graph**(*g*)

The Graph object holds information about a graph and has methods that can be used to manipulate that graph.

name

The name of the graph

style

The style of the graph

style = 1: XY plot

style = 3: 3D plot

style = 4: polar plot

add_annotation(*s*, *pos=None*, ***kwargs*)

Adds an annotation to the plot.

Parameters

- **s** (*string*) – the text for the annotation
- **pos** (*tuple of two floats*) – the position of the annotation (a tuple). If not given, the annotation is positioned at graph coordinate location 0.5,0.5 (the center of the graph).

The add_annotation function can take keyword arguments to modify the properties of the annotation as follows. Any unknown keyword arguments are ignored.

Parameters

- **coord_system** (*integer*) – 0 = figure coordinates, 1 = data coordinates. Default is figure coordinates.
- **font** (*FontProperties object*) – A font properties object. Default is Arial, normal weight, normal style, and 12 point.
- **font_color** (*string*) – The color of the annotation text, which is a string in the following format: "#FFAABB". The default color is black "#000000".
- **multi_alignment** – A string describing the horizontal alignment (justification) of the text within the box. Can be "center", "left", or "right". Default is "center".
- **rotation** (*integer*) – An integer giving the amount of text rotation, in degrees. Can range from -360 to +360.
- **linespacing** (*float*) – A floating point number describing the amount of space to take per line; must be greater then 0.0. Default is 1.2, which very closely resembles single-spaced lines.
- **alpha** (*float*) – alpha transparency of the annotation text. This is a floating point number ranging from 0.0 (fully transparent) to 1.0 (full opaque). Default is 1.0.
- **box_show** (*bool*) – whether or not to show the box around the text.
- **box_style** (*string*) – the style of the box. Can be "square", "real_square", "ellipse", "circle", "diamond", "round", "round4", "sawtooth", "roundtooth", "rarrow", "larrow", "triangle", "parallelogram-right", "parallelogram-left", "trapezoid", or "itrapezoid". The default is "square".
- **box_fit_to_text** (*bool*) – whether or not to fit the box to the size of the text. If False, the parameter box_size is used as the size of the box.

- **box_size** (*tuple of two floats*) – the size of the box, given in the same coordinate system as the annotation's coord_system. Ignored if box_fit_to_text is True.
- **box_fillcolor** (*string*) – the color of the fill inside the box, which is a string in the following format: "#FFAABB". The default color is black "#000000".
- **box_edgecolor** (*string*) – the color of the edge of the box, which is a string in the following format: "#FFAABB". The default color is black "#000000".
- **box_alpha** (*float*) – the transparency setting of the box (which includes its edge). This is a floating point number ranging from 0.0 (fully transparent) to 1.0 (full opaque). Default is 1.0.
- **boxframe_width** (*integer*) – an integer describing the width of the box edge. The default is 1.
- **box_shadow** (*bool*) – whether or not to draw a shadow behind the box.
- **arrow_show** (*bool*) – whether or not to show an arrow in conjunction with the annotation. The arrow always points from the center of the annotation to the point given by (arrow_x, arrow_y). Default: False
- **arrow_coord_system** – 0 = figure coordinates, 1 = data coordinates. Default is data coordinates.
- **arrow_x** (*float*) – The x location of where the arrow should point to, in the coordinate system given by arrow_coord_system.
- **arrow_y** (*float*) – The y location of where the arrow should point to, in the coordinate system given by arrow_coord_system.
- **arrow_alpha** (*float*) – the transparency setting of the arrow . This is a floating point number ranging from 0.0 (fully transparent) to 1.0 (full opaque). Default is 1.0.
- **arrow_color** (*string*) – the color of the arrow, which is a string in the following format: "#FFAABB". The default color is black "#000000".
- **arrow_width** – an integer describing the width of the arrow edge. The default is 1.
- **arrow_style** (*string*) – a string indicating the style of the arrow. Can be "line", "thick", "wedge", or "fancy". The default is "line".
- **arrow_head** (*string*) – a string indicating the style of the arrow's head. Can be -", "->", "<-", "<->", "-|>", "<|-", "<|-|>", "-[", "]-", "]-[", or "|-|". Note that not every arrow head style is compatible with the arrow style. See the CurveExpert Pro GUI for valid combinations. The default is "-|>".
- **arrow_connection_style** (*string*) – a string indicating the style of the connection line between the annotation and the pointed-to-point. Can be one of "straight", "xy", or "arc". The default is "straight". Note that not every arrow connection style is compatible with the arrow style. See the CurveExpert Pro GUI for valid combinations.

add_arrow(*startpos*, *endpos*, ***kwargs*)

Adds an arrow to the plot.

Parameters

- **startpos** (*tuple of two floats*) – start position of the arrow, in graph coordinates.
- **endpos** (*tuple of two floats*) – end position of the arrow, in graph coordinates Should be a tuple.
- **coord_system** (*integer*) – 0 = figure coordinates, 1 = data coordinates. Default is figure coordinates.

- **alpha** (*float*) – alpha transparency of the arrow. This is a floating point number ranging from 0.0 (fully transparent) to 1.0 (full opaque). Default is 1.0.
- **color** (*string*) – The color of the arrow’s edge, which is a string in the following format: “#FFAABB”. The default color is black “#000000”.
- **fillcolor** (*string*) – The color of the arrow’s fill, which is a string in the following format: “#FFAABB”. The default color is black “#000000”.
- **width** (*integer*) – an integer describing the width of the arrow’s line. The default is 1.
- **style** (*string*) – a string indicating the style of the arrow. Can be “line”, “thick”, “wedge”, or “fancy”. The default is “line”.
- **head** (*string*) – a string indicating the style of the arrow’s head. Can be -“, “->”, “<-“, “<->”, “-|>”, “<|-“, “<|-|>”, “-[“, “]-“, “]-[“, or “|-|”. Note that not every arrow head style is compatible with the arrow style. See the CurveExpert Pro GUI for valid combinations. The default is “-|>”.
- **connection_style** (*string*) – a string indicating the style of the connection line between start and end point of the arrow. Can be one of “straight”, “xy”, or “arc”. The default is “straight”. Note that not every arrow connection style is compatible with the arrow style. See the CurveExpert Pro GUI for valid combinations.
- **shape_scaling** (*float*) – A floating point number that determines the “fatness” of the arrow shape. Lower scalings lead to thinner arrow bodies (in cases where the arrow body is a shape, and not just a line), and smaller arrow heads. Higher scalings conversely lead to thicker arrow bodies and correspondingly larger arrow heads.

add_result(*r*)

Adds a computed result to the graph.

Parameters **r** – the result object to add.

Note that adding a result to a graph means that the result must be managed by the application. If it is not, it automatically becomes managed.

apply_theme(*theme_name*)

applies a particular theme to the graph. The theme must be present on the system.

Parameters **theme_name** (*string*) – A string that gives the name of the theme to apply. Example: “Blackness”

autoscale()

autoscales the graph

clear()

clears all series off of the graph and resets the scaling to its default state.

Return type True if successful, False if not.

draw_circle(*xy*, *radius*, *coord_system=0*, *redraw=True*, ***kwargs*)

Draws a circle on the plot, with center xy and the given radius.

At this time, the circle is not persistent, which means that it is not saved along with the graph to file. If a document is read from disk, the circle will not be present; the plugin must be rerun in order to redraw the circle.

Parameters

- **xy** (*a tuple containing two floats*) – location of the centerpoint.
- **radius** (*float*) – the radius of the circle

The draw_circle function can take keyword arguments to modify the properties of the circle as follows. Any unknown keyword arguments are ignored.

Parameters

- **coord_system** (*integer*) – 0 = figure coordinates, 1 = data coordinates. Default is figure coordinates.
- **redraw** (*boolean*) – whether or not to redraw the figure after adding the artist to it. Useful when adding many artists at a time, to reduce screen flicker.
- **edgecolor** – color of the edge, given as a hex code, as in "#00FF00". Common color names may also be given; for example, "red".
- **fillcolor** (*string*) – color of the fill inside the circle, given as a hex code, as in "#00FF00". Common color names may also be given; for example, "red". Can be 'None' for no fill.
- **linewidth** (*integer*) – width of the line in points (1/72 of an inch)
- **linestyle** (*string*) – style of the line. Can be one of 'solid','dashed','dash_dot','dotted', or 'None', which means not to draw the line.
- **alpha** (*float*) – transparency of the circle. Can be 0.0 to 1.0, with zero being transparent, and 1 being opaque.

Return type an artist that represents the circle.

An artist can always be removed from the graph by calling its remove() method.

draw_line(*xarray*, *yarray*, *redraw=True*, *coord_system=0*, ***kwargs*)

Draws a (potentially) multisegmented line on the plot.

At this time, the line is not persistent, which means that it is not saved along with the graph to file. If a document is read from disk, the line will not be present; the plugin must be rerun in order to redraw the line.

Parameters

- **xarray** (*any iterable object returning floats*) – sequence of points for the x coordinates
- **yarray** (*any iterable object returning floats*) – sequence of points for the y coordinates

The draw_line function can take keyword arguments to modify the properties of the line as follows. Any unknown keyword arguments are ignored.

Parameters

- **coord_system** (*integer*) – 0 = figure coordinates, 1 = data coordinates. Default is figure coordinates.
- **redraw** (*boolean*) – whether or not to redraw the figure after adding the artist to it. Useful when adding many artists at a time, to reduce screen flicker.
- **color** (*string*) – color of the line, given as a hex code, as in "#00FF00". Common color names may also be given; for example, "red".
- **linewidth** (*integer*) – width of the line in points (1/72 of an inch)
- **linestyle** (*string*) – style of the line. Can be one of 'solid','dashed','dash_dot','dotted', or 'None', which means not to draw the line.

- **alpha** (*float*) – transparency of the line. Can be 0.0 to 1.0, with zero being transparent, and 1 being opaque.
- **marker** (*string*) – marker for each node of the line. Can be any one of [‘o’ | ‘D’ | ‘h’ | ‘H’ | ‘_’ | ‘’ | ‘None’ | ‘ ‘ | None | ‘8’ | ‘p’ | ‘,’ | ‘+’ | ‘.’ | ‘s’ | ‘*’ | ‘d’ | ‘1’ | ‘3’ | ‘4’ | ‘2’ | ‘v’ | ‘<’ | ‘>’ | ‘^’ | ‘|’ | ‘x’ |
- **markeredgecolor** (*string*) – color of the marker edge, given as a hex code, as in “#00FF00”. Common color names may also be given; for example, “red”. Can be ‘None’ for no marker edge.
- **markerfillcolor** (*string*) – color of the marker fill, given as a hex code, as in “#00FF00”. Common color names may also be given; for example, “red”. Can be ‘None’ for no fill.
- **markeredgewidth** (*integer*) – width of the marker edge, given in points.

Return type an artist that represents the line.

An artist can always be removed from the graph by calling its remove() method.

draw_polygon(*xarray*, *yarray*, *closed=True*, *coord_system=0*, *redraw=True*, ***kwargs*)

Draws a polygon on the plot, with the points xy

At this time, the polygon is not persistent, which means that it is not saved along with the graph to file. If a document is read from disk, the polygon will not be present; the plugin must be rerun in order to redraw the polygon.

Parameters

- **xarray** (*any iterable object returning floats*) – sequence of points for the x coordinates
- **yarray** (*any iterable object returning floats*) – sequence of points for the y coordinates
- **closed** (*boolean*) – if True, the polygon will be closed so that the starting and ending points are the same.

The draw_polygon function can take keyword arguments to modify the properties of the polygon as follows. Any unknown keyword arguments are ignored.

Parameters

- **coord_system** (*integer*) – 0 = figure coordinates, 1 = data coordinates. Default is figure coordinates.
- **redraw** (*boolean*) – whether or not to redraw the figure after adding the artist to it. Useful when adding many artists at a time, to reduce screen flicker.
- **edgecolor** (*string*) – color of the edge, given as a hex code, as in “#00FF00”. Common color names may also be given; for example, “red”.
- **fillcolor** (*string*) – color of the fill inside the polygon, given as a hex code, as in “#00FF00”. Common color names may also be given; for example, “red”. Can be ‘None’ for no fill.
- **linewidth** (*integer*) – width of the line in points (1/72 of an inch)
- **linestyle** (*string*) – style of the line. Can be one of ‘solid’,’dashed’,’dash_dot’,’dotted’, or ‘None’, which means not to draw the line.
- **alpha** (*float*) – transparency of the polygon. Can be 0.0 to 1.0, with zero being transparent, and 1 being opaque.

Return type an artist that represents the polygon.

An artist can always be removed from the graph by calling its remove() method.

`get_all_results()`

Gets a list of all results plotted on a graph.

Return type a list of Result objects.

`get_properties()`

gets the full property object for the graph. All properties are here, but are not documented, unfortunately. If there is some property of the graph that you would like to change that is not accessible through the regular API's, contact the author. Alternatively, use Python's introspection capability to look into the properties object.

`get_title()`

gets the title of the graph

Return type the title of the graph (string)

`hardcopy`(*interactive=False*)

prints the graph to the printer.

Parameters **`interactive`** (*`bool`*) – If set to True, the user will be presented with a dialog to select the printer. Otherwise, the printout goes directly to the default printer.

`redraw()`

Redraws the graph.

`remove_result`(*r*)

Removes a computed result from a graph.

Parameters **`r`** – the result object to remove.

`save_as`(*filename*)

saves the graph as an external image or vector format file (PNG, JPG, TIFF, BMP, SVG, etc.). The format is deduced from the name of the file.

Parameters **`filename`** (*`string`*) – The name of the file to save.

`set_axis_extents`(*axis*, *extents*)

sets the extent of a particular axis.

Parameters

- **`axis`** (*`string`*) – which axis to set. Can be 'x1', 'x2', or 'y'. For convenience, 'x' can be used in place of 'x1', and 'z' may be used in the place of 'y'.
- **`extents`** (*`tuple of two floats`*) – the min and max of the axis extent. Should be a tuple of two floating point numbers.

Return type True if successful, False if not.

`set_axis_label`(*axis*, *label*)

sets the axis label

Parameters

- **`axis`** (*`string`*) – which axis to label. Can be 'x1', 'x2', or 'y'. For convenience, 'x' can be used in place of 'x1', and 'z' may be used in the place of 'y'.
- **`label`** (*`string`*) – a text string to change the label to; mathtext in the string is also valid.

Return type True if successful, False if not.

`set_properties`(*p*)

sets the full property object for the graph. This sets all properties.

Parameters `p` – a full graph property object

`set_title`(*title*)

changes the title of the graph

Parameters `title` (*string*) – text string to change the title to; mathtext in the string is also valid.

Return type True if successful, False if not.

`property title`

gets the title of the graph

Return type the title of the graph (string)

CHAPTER

TWENTYEIGHT

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

INDEX