GraphExpert Professional Documentation

Release 1.6.0

Daniel G. Hyams

Oct 22, 2020

CONTENTS

CHAPTER

ONE

WHAT IS GRAPHEXPERT PROFESSIONAL?

GraphExpert Professional is a cross-platform solution for publication-quality graphing of both discrete data and continuous functions. XY plots, bar charts, polar plots, scatterplots, contour plots, and 3D scatter/surface plots are supported. Also, graphs can be annotated with a variety of drawing tools such that annotations, arrows, and images can quickly and easily be added to the plot.

GraphExpert Professional takes a unique approach in managing the multiple datasets and functions. Datasets may be derived from other datasets in a multitude of ways (extraction, transformations, histogramming, etc.), and a parent/child relationship is always maintained. This means that you can concentrate on the required operations to process your raw data into a usable form, and if the raw data changes for any reason, all of the operations update automatically. Further, any datasets that are already on a graph update automatically. This means less time performing repetitive work, and more time for the real task at hand.

As a simple example for illustration, imagine that you have a raw dataset in a file, and you would like to plot columns 1 and 3 of the dataset along with a best-fit line. Simply import the raw dataset into GraphExpert Professional, extract the first and third columns, and then run a linear regression on that extraction. Then drag and drop both the extraction and linear regression onto a plot for visualization. GraphExpert Professional keeps track of the operations that you performed, so that if the raw dataset is modified in any way, the extraction, linear regression, and the plot is updated automatically. The procedure is intuitive and easy.

Data analyses such as regression, creation of histogram data, ANOVAs, t-tests, computing integrals/differences, moving averages, processing replicates, and computing Fast Fourier Transforms are supported.

The major features of the software are enumerated below:

- **Multiplatform**: runs on Windows, Mac, and Linux. Saved files are portable between platforms.
- **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. XY plots, polar plots, scatterplots, 3D 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 the ability to move backward to previous views of the graph.
- **Easy-to-use User Interface**: most mathematically-intensive applications are very difficult to use. GraphExpert 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 and display its data on a graph takes only five clicks.
- **Robust file import**: data files come in many shapes and sizes, and GraphExpert Professional makes importing your data files very easy. The smart file reader avoids non-data areas of your file dynamically, and intelligently finds labels for each column of data in your file.
- **File hotlinking**: instead of reading the data file directly, GraphExpert Professional can dynamically draw a dataset from a file, updating the dataset as the file changes.

- **Extract data directly from an Excel(TM) spreadsheet**: GraphExpert Professional can extract and hotlink to data directly from an Excel(TM) file, updating the resulting dataset as the file changes.
- **Powerful transformations**: datasets can be transformed using a suite of very powerful operators that can perform simple operations (such as extracting a column or set of columns) to complex operations (such as an arbitrary formula that generates a new column from others).
- **Miniprograms**: datasets can be directly generated with a Python code called a “miniprogram”. The dataset updates dynamically when/if the program is changed.
- **Functions**: define function formulas yourself, that be used either to generate more datasets, or to plot directly. For more advanced users, a function can be as complex as necessary, as long as it can be expressed in Python code. This allows the creation of extremely complicated functions, instead of the normal one-line-equation function limitation in other software. A very large library of built-in mathematical functions (e.g. trigonometric functions, Bessel functions, etc.) is available for direct use in your formulas.
- **Quality spreadsheet**: the built-in spreadsheet allows you to manually enter data and/or modify it in-place 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. GraphExpert Professional will install for all users (if you are the Administrator), or for you only (if you are not). This allows you to install GraphExpert 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:

```
> GraphExpertPro-1.5.6-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\GraphExpert 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.

GraphExpert 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

GraphExpert 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 GraphExpert 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 GraphExpertPro-1.5.6-Linux-x86_64.run
% ./GraphExpertPro-1.5.6-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: GraphExpert 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 GraphExpert 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.

GraphExpert 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. GraphExpert Professional integrates well with GNOME, offering a desktop icon, appropriate menu entries and icons, file association between .gxp files, and icons for .gxp 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 .gxp files are not yet functional.

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

```
~/GraphExpertPro/gepro.sh
```

2.2 Starting a Trial

If you want to evaluate GraphExpert 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 GraphExpert 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.graphexpert.com/order.

2.3 Purchasing

To purchase GraphExpert Professional, visit http://www.graphexpert.com/order, and choose the product that you would like to purchase.

2.4 After Purchase: licensing

If you have purchased a GraphExpert 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 GraphExpert 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 GraphExpert Professional on the command line as follows:

```
> gepro.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/GraphExpertPro.app --args --license the_license_filename.txt --quit
```

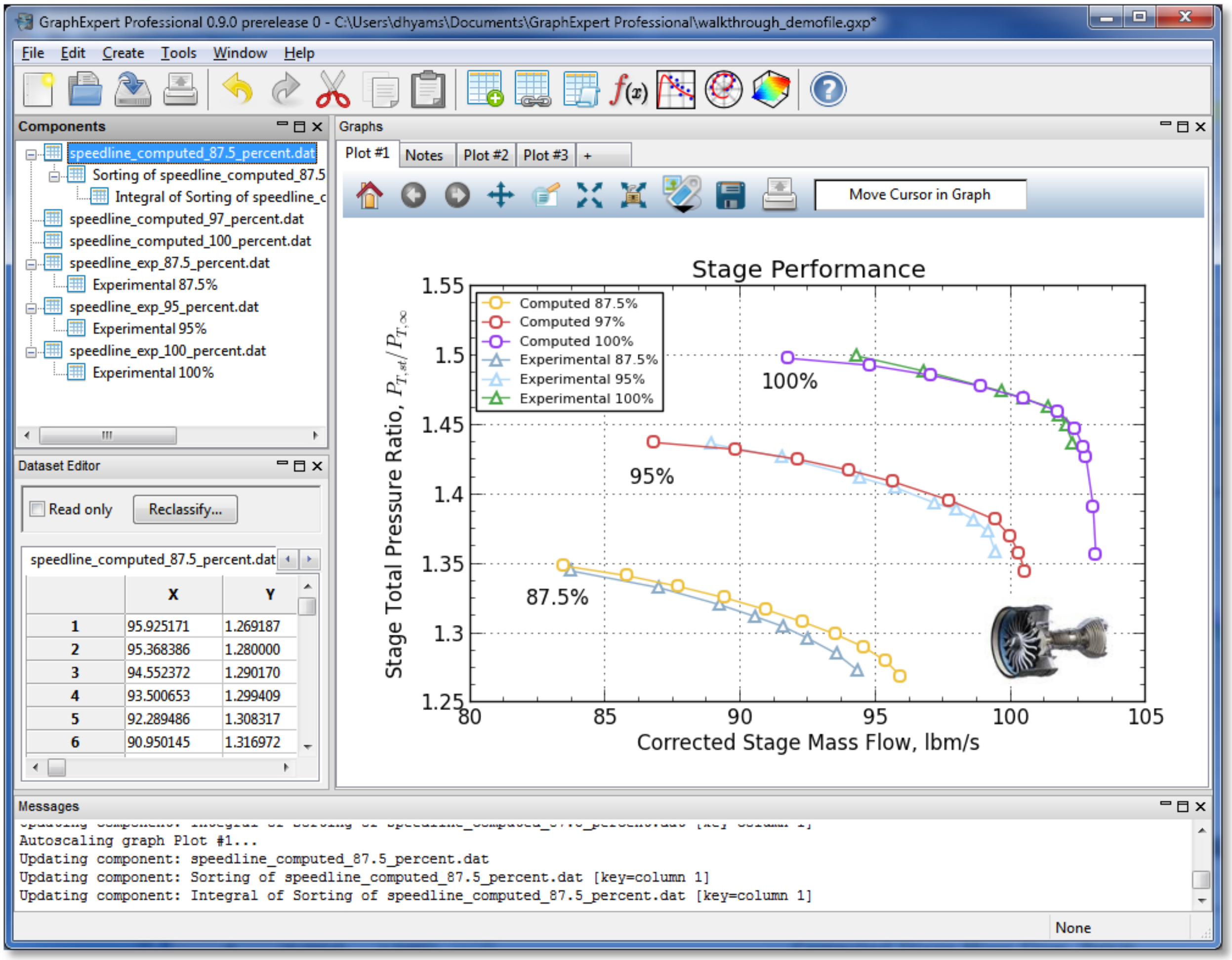
The application will write a file "license.log" that prints the results of the licensing (success or failure).

CHAPTER

THREE

GETTING STARTED

This guide is a walkthrough of the use of GraphExpert Professional. This guide gives you essential information you need to know in order to use the application effectively, and demonstrates just a few of the software's features. See the *User Interface* chapter for a full explanation of the user interface.



speedline_computed_87.5_percent.dat
Sorting of speedline_computed_87.5
Integral of Sorting of speedline_c
speedline_computed_97_percent.dat
speedline_computed_100_percent.dat
speedline_exp_87.5_percent.dat
Experimental 87.5%
speedline_exp_95_percent.dat
Experimental 95%
speedline_exp_100_percent.dat
Experimental 100%

☐ Read only

speedline_computed_87.5_percent.dat

	X	Y
1	95.925171	1.269187
2	95.368386	1.280000
3	94.552372	1.290170
4	93.500653	1.299409
5	92.289486	1.308317
6	90.950145	1.316972

```
Updating component: Integral of Sorting of speedline_computed_87.5_percent.dat [key=column 1]
Autoscaling graph Plot #1...
Updating component: speedline_computed_87.5_percent.dat
Updating component: Sorting of speedline_computed_87.5_percent.dat [key=column 1]
Updating component: Integral of Sorting of speedline_computed_87.5_percent.dat [key=column 1]
```

3.1 Load data into GraphExpert Professional

Read a sample file supplied with GraphExpert Pro. Choose *File->Import Data*, double click on "example_data", and double click the file "speedline_exp_87.5_percent.dat". When the File Import dialog appears, simply click OK (see *Importing from file*). When only one file is read in for import, the File Import dialog will guide you through the process. When multiple files are read in for import, each file is just imported with reasonable default settings. So, read in another five files, by choosing *File->Import Data* again, and select the rest of the speedline files this time (use the shift or ctrl key to multiselect them).

At this point, you should see all of the datasets listed in the *Components Pane*.

3.2 View the raw data

Double click one of the "speedline_computed" datasets to view it in the *Dataset Editor*. Also double click one of the "speedline_exp" datasets. GraphExpert Pro treats each column as a series, when things are plotted; obviously, the experimental data is not exactly oriented as we need it to be.

3.3 Perform an operation on some datasets

Right-click the "speedline_exp_87.5_percent.dat" component, and choose *Operators->Transpose* (see *Transpose*). The transpose is now shown in the *Components Pane* as a child component (see *Managing Datasets and Functions*). Double click the child (labeled "Transpose of…" at this point), to show how the dataset has been rearranged.

You can do the same operation to the other two experimental datasets. Use Shift or Ctrl to multiselect both, right-click, and select *Operators->Transform* as before.

Alternatively, the transpose operations could have been performed in the much more general Transformation facility (see *Transform*). To do this, right click a component, and select *Transform*. Taking the transpose of the entire dataset would use the expression:

```
DSET.T
```

here, DSET means the entire dataset, and .T means to take the transpose of it.

3.4 Rename your components (optional)

We would like to give these newly formed components nice names, so select one, and press F2 (alternatively, right click each and select *Rename*). Rename each transformation to whatever is desired (in this case, all are renamed to "Experimental xx%" or "Computed xx%", where xx is the appropriate percentage).

3.5 Plot the datasets

Use your Ctrl key to multiselect all of the "computed" datasets, and the new transformed datasets that were just built. Right click, and select "Send to Current Plot". Alternatively, you could just drag and drop the bunch of components to the graph.

3.6 Visualize

In order to visualize the datasets and results previously read and/or computed, you can either right-click each one and select "Send to Current Graph", or drag and drop the desired item to a graph (see *Graphing*).

3.7 Apply a theme

Right click on the graph, and select *Apply Theme->Elegant White - Soft*. See *Graph Themes*.

3.8 Modify the plot

The properties of a graph can be extensively modified through the graph properties dialog (see *Changing Graph Properties*).

Double click each of the axis labels to change them to something more appropriate. For the y axis, enter:

```
Stage Total Pressure Ratio, $P_{T,st}/P_{T,\infty}$
```

For the x axis, enter:

```
Corrected Stage Mass Flow, lbm/s
```

This demonstrates how *Mathtext* works, which allows you to display mathematical equations and/or symbology anywhere on the graph.

Double click on the title of the graph, and change the title to "Stage Performance".

Right click on the graph, and select *Graph Properties*. Under *Series*, you can adjust the appearance of each curve, and the label that appears in the legend. Here, if desired, change all of the experimental curves to use triangle-up as their markers, and change the legend entry to read something appropriate for all (we used "Computed 87.5%" and so on).

The legend text is still a little large and covers up part of the data, so double click the legend, and edit the legend labels on the right side of the dialog to shorten the names to something more practical. Alternatively, you can click "Label Font", and decrease its point size such that the legend becomes small enough to fit into the empty upper left area of the graph. The legend can also be dragged and dropped into any desired location.

3.9 Add annotations

We can add annotations (see *Annotations*) to the plot by clicking the *Artistic Elements* button at the top of the graph; this button's icon has a black downward pointing triangle. Click this, and click the blue tag; this will add an annotation to your graph. For the contents, type in "100%", and turn off "Show box". Press OK, and then drag the annotation around to where you want it to be. Place it next to the pair of curves that represent the 100% speedline. Now, you can either copy this annotation twice, or just add two more annotations for the 95% and 87.5% speedlines.

3.10 Add an image

To add an image (see *Images*) , just copy one to the clipboard and then right click the graph; select *Paste*. Alternatively, just press Ctrl+V while the Graphs pane is focused. Just to demonstrate how easy it is to add an image, use your web browser to search the web for "turbomachinery stage", go to the images section of the search engine that you happen to use, right click an image, pick "Copy image", and then retun to GraphExpert Professional and paste in the image.

Click the image to interactively resize and move it to the desired location. As always, right click to modify properties of the image.

3.11 Compute an integral

GraphExpert Professional has a library of computations that can be performed on a dataset. Return to the Components Pane, Right click one of the "computed" datasets, and pick *Operators->Sort*. Then on the newly created, sorted dataset, pick *Operators->Integral*. This creates a new dataset that can be viewed as the others can be in the Dataset Editor.

3.12 Learn how component updating works

GraphExpert Professional updates components automatically if their parent changes (see *Managing Datasets and Functions*). To show how this works, open the "speedline_computed_87.5_percent.dat" component in the Dataset Editor by double-clicking it. Change the value at row 2, column Y to 1.4. Notice that your plot has updated, but furthermore, the sorted component updated, and the integral updated. Double click them to view them and verify.

3.13 Place your graph in a report

This is extremely easy; just right click your graph and select *Copy*. Then go to your report software (i.e., Microsoft Word), and paste.

3.14 Add more plots

Select *Create->Polar Plot* and then *Create->3D Plot*. In this session, we don't do anything with these plots, other than to let you know that they are available.

CHAPTER

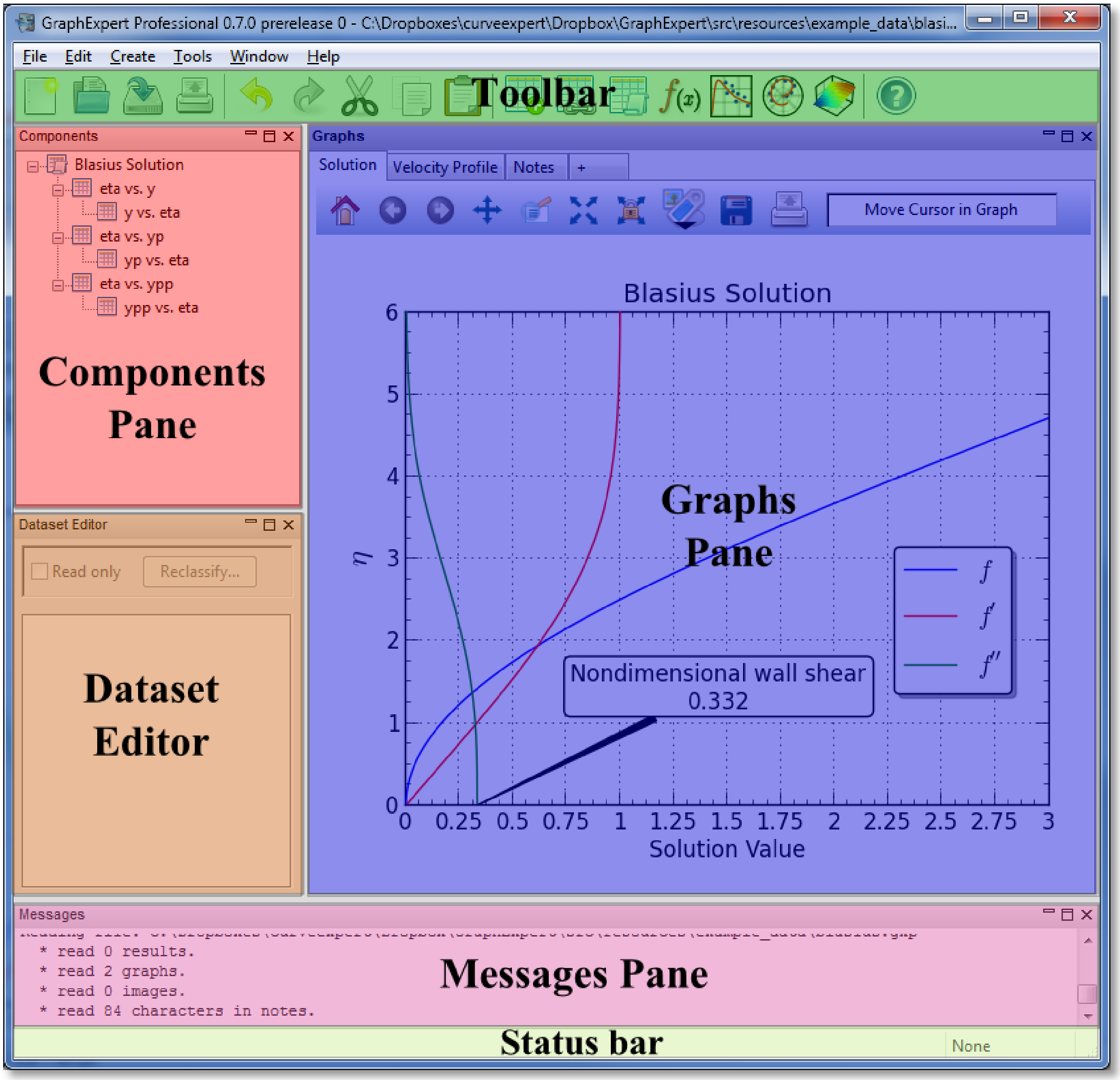
FOUR

USER INTERFACE

The GraphExpert Professional user interface is divided into four panes:

- Components
- Dataset Editor
- Graphs
- Messages

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



```
Reading file: C:\Dropboxes\curveexpert\Dropbox\GraphExpert\src\resources\example_data\blasius.gxp
  * read 0 results.
  * read 2 graphs.
  * read 0 images.
  * read 84 characters in notes.
```

All of the panes can be shown or hidden via the *Window* menu in the menu bar. Further, panes (except for the Graphs 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 pane) in full size.

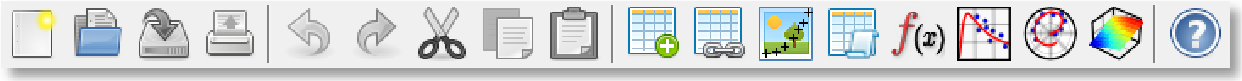
All panes can be hidden by clicking the ‘X’ at the top right of the pane, or by deselecting the corresponding pane name in the *Window* menu. To show a pane after being hidden, select the corresponding pane name in the *Window* menu.

4.1 Menus

The menus at the top of the GraphExpert Professional window give an interface into the main functionality of the application. See *Menu Reference* for a summary 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
- Create
- Help

The buttons are each documented below.

Icon	Meaning
🗋	File->New. Reinitializes the application with no components and no graphs (completely empties the document).
📂	File->Open. Open a GraphExpert Professional file.
💾	File->Save. Save the current components, graphs, and notes into a GraphExpert Professional document.
🖨	File->Print. Prints all graphs.
↶	Edit->Undo. Undo the last appropriate action
↷	Edit->Redo. Redo the last appropriate undone action
✂	Edit->Cut. Cut the selected items(s), placing them on the clipboard
⎘	Edit->Copy. Copy the selected item(s) to the clipboard
📋	Edit->Paste. Paste the contents of the clipboard appropriately into the focused pane
⊞	Create->Dataset. Create a new dataset to work with, having the specified size and fill. See *Creating new datasets*.
🔗	Create->Hot link. Create a new link to a file that acts as a source for a dataset; dataset automatically updates when the file
🖼	Create->Digitizer. Create datasets by extracting data from an image (usually a scanned image of a plot or map). See *Digit*
📜	Create->Miniprogram. Create a Python code that returns a dataset. See *Miniprograms*.
$f(x)$	Create->Function. Create a continuous function that can be used directly for plotting or to derive other datasets. See *Func*
📈	Create->XY Plot. Create a fresh XY plot in the Graphs pane. See *XY Plots*.
◎	Create->Polar Plot. Create a fresh polar plot in the Graphs pane. See *Polar Plots*.
⬡	Tools->3D Plot. Create a fresh three dimensional plot (for plotting items with two independent variables) in the Graphs pa
?	Help->Manual. Open the documentation.

4.3 Components Pane

The Components pane is the heart of data management in GraphExpert Professional. Here, you will see all available datasets and functions that can be visualized, as well as the parent/child relationship between them. Right clicking in the empty area underneath any component items will display a context sensitive menu allowing you to create datasets (see *Creating new datasets*), create functions (see *Simple functions*), create hotlinks (see *Hot linking to a text or Excel file*), or import data from file (see *Importing from file*).

Components can be used to create another components; in essence allowing the user to derive datasets and functions from existing datasets and functions. See *Managing Datasets and Functions* for how to operate on your components that are displayed.

Hovering your mouse over any component will provide a tooltip that displays vital information concerning that component. For example, for a dataset that is a transformation of another (see *Transform*), the tooltip will display the number of rows and columns in the dataset, and the transformation expression. If there has been a problem in evaluating a derived dataset, the tooltip will also indicate the reason for the problem.

A component may be renamed by right clicking it and selecting *Rename*, or, one can click on an already selected component, and the name will become editable. Also, pressing the F2 key is a shortcut to be able to rename a component.

Components may be multiple-selected by holding the Shift or Ctrl key while selecting components. Once a group of components are selected, any action performed by a right click applies to all of the selected components.

Double clicking on a component invokes an appropriate action to view it. For a function, a popup window shows the equation for the function and the parameters of the function (if any). If the double-clicked component is a dataset, the Dataset Editor opens the dataset in a new tab (if necessary).

A component (or multiple components) can be removed by right clicking it and selecting *Remove*; or press the Del key.

4.3.1 Icons

The icons displayed in the Components pane convey information about the type and state of each component.

Icon	Meaning
▦	A dataset stored in memory.
⚠	A dataset stored in memory where the data may not be correct. Example: taking the integral of unsorted data.
⛔	The component is a dataset, but the updating has failed catastrophically. Example: complete failure in evaluating a transform.
🔗	An actively updating dataset hotlink.
⏸	A hotlink that is active and linked to a file, but the updating has been paused by the user.
⚠	A hotlinked dataset where the updating from file failed. Example: the file is unavailable or no permissions to read.
📄	A dataset that is generated by the action of a miniprogram.
⚠	A miniprogram dataset that has failed due to an exception in the Python code, or does not return the correct information.
f(x)	A simple expression function.
f⚠	A function that has not updated properly, so the previous state was retained.
f⛔	A function that has failed catastrophically. Example: divide by zero in the function.
f(x)	A programmed function.
f⚠	A programmed function that has not updated properly, so the previous state was retained.
f⛔	A function that has failed catastrophically, due to an exception in the Python code, or does not return the correct information.

4.4 Dataset Editor

The **Dataset Editor** pane is a notebook where datasets can be edited or viewed. A dataset can be edited/viewed by

- double clicking the dataset in the Components pane
- drag/drop of the dataset from the Components pane
- right click the dataset in the Components pane, and select **View/Edit** from the context menu.

Once your dataset is opened in the Dataset Editor, a new tab will be created for it. This tab will remain until either the dataset is destroyed (in the Components pane), or you close the tab by right clicking it and selecting **Close**.

A dataset can be made read only (protected from editing) by selecting the **Read only** checkbox near the top of the Dataset Editor pane.

Note: Only datasets that are are the top of the Component hierarchy can be edited; other datasets, by definition, derive their values from their parent dataset, and such their values are noneditable. These datasets can be viewed in the Dataset Editor, but the **Read only** checkbox will remain dimmed and selected to indicate the "always read-only" nature of child datasets.

A dataset can be reclassified (see *Reclassifying columns*) from the Dataset Editor by clicking the **Reclassify** button near the top of the editor pane. This is an equivalent operation to right clicking the dataset in the Component pane, and selecting *Reclassify*.

See *The spreadsheet* for documentation concerning how to operate and use the features of the data spreadsheet.

4.5 Graphs Pane

The **Graphs** pane is a notebook where all graphs reside (as well as any notes that you would like to save). Refer to *Graphing* for documentation concerning how to work with graphs.

To add another graph, press the "+" tab; this asks you what type of graph to add, and then adds that graph to the notebook for you to work with as you see fit. You may also rename any of your 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**.

Remember that, for a graph that you have created, you can drag and drop components from the **Components** pane to visualize anything of interest.

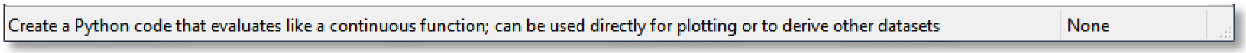
4.6 Messages Pane

The **Messages Pane** is an area where GraphExpert 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



Create a Python code that evaluates like a continuous function; can be used directly for plotting or to derive other datasets

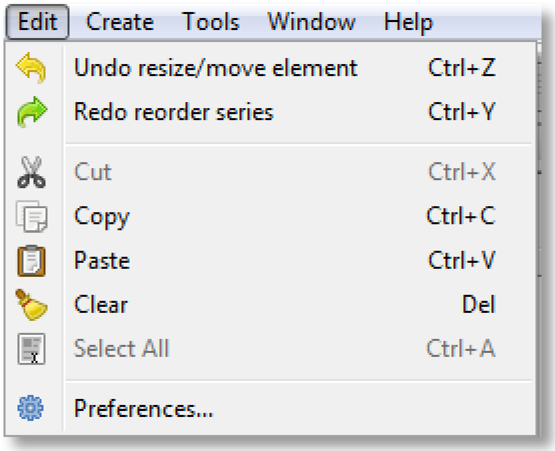
None

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

Section	Displays
1	help for menu items and toolbar buttons
2	current locale

4.8 Undo and Redo

In the *Edit* Menu, the **Undo** and **Redo** items always indicate what will be undone or redone; this useful feature tells you as the user exactly what will happen upon selection of an Undo or Redo.



At the present time, each pane in GraphExpert Professional keeps track of its own undo states. Therefore, the proper pane must be "focused" to perform undo/redo actions on the context in that pane. The "focus" of a pane is shown by its title bar being colored and the pane name being bolded. Focus can be transferred from one pane to another by simply clicking the mouse in the desired pane.

Usually, this requirement is intuitive and transparent to the user. For example, if a user is working in the Component pane (creating datasets, say), the Component pane already has the focus, so undo/redo works naturally. However, if the user is working in the component pane, and decides that an edit in the Dataset Editor should be undone, the user must click in the Dataset Editor pane somewhere before issuing the undo; otherwise, it is the Components pane that services the undo request. Observing the text in the **Undo** and/or **Redo** entries in the *Edit* menu is an easy way of making sure that an undo/redo command does what it is expected to.

A quick way of summarizing the above paragraph is that undo/redo commands apply to the panes separately, and not to the application as a whole.

4.9 Menu Reference

Note: On OSX, some menu items appear both in the *GraphExpertPro* 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 GraphExpert Pro* (documented below as *File->Exit*).

4.9.1 File menu

Menu choice	Purpose
File->New	Reinitializes the application with no components and no graphs (completely empties the document).
File->Open	Opens a new GraphExpert Professional file.
File->Recent files	Opens a previously opened GraphExpert Professional file.
File->Close	Closes the current file.
File->Save	Save the current components, graphs, and notes into a GraphExpert Professional document. See *Saving Data*.
File->Save As	Save the current data as a new GraphExpert Professional document, with a different name. See *Saving Data*.
File->Import Data	Import raw data from a text file into the current document as a component. See *Importing from file*.
File->Import Data from Clipboard	Import raw text from the clipboard into the current document as a component. See *Importing from file*.
File->Print	Prints all graphs
File->Print Preview	Shows a print preview of the entire set of graphs, optionally printing them as well
File->Page Setup	allows setting of preferences for margins, paper orientation, etc., for printing.
File->Exit	quit GraphExpert Professional

4.9.2 Edit menu

Menu choice	Purpose
Edit->Undo	Undo the last appropriate action. See *Undo and Redo*.
Edit->Redo	Redo the last appropriate undone action . See *Undo and Redo*.
Edit->Cut	cut the selected items(s), placing them on the clipboard
Edit->Copy	Copy the selected item(s) to the clipboard
Edit->Paste	Paste the contents of the clipboard appropriately into the focused pane
Edit->Clear	Clear the selected item(s)
Edit->Select All	Select all items in the focused pane
Edit->Preferences	Inspect or change GraphExpert Professional preferences. See *Application Preferences*.

4.9.3 Create menu

Menu choice	Purpose
Create->Dataset	Create a new dataset to work with, having the specified size and fill. See *Creating new datasets*.
Create->Hot link	Create a new link to a file that acts as a source for a dataset; dataset automatically updates when the file changes. See *Hot linking to a text or Excel file*.
Create->Miniprogram	Create a Python code that returns a dataset. See *Miniprograms*.
Create->Function	Create a continuous function that can be used directly for plotting or to derive other datasets. See *Functions*.
Create->Programmed function	Create a Python code that evaluates like a continuous function; can be used directly for plotting or to derive other datasets. See *Programmed functions*.
Create->XY Plot	Create a fresh XY plot in the Graphs pane. See *XY Plots*.
Create->Polar Plot	Create a fresh polar plot in the Graphs pane. See *Polar Plots*.
Create->3D Plot	Create a fresh three dimensional plot (for plotting items with two independent variables) in the Graphs pane. See *3D Plots*.

4.9.4 Tools menu

Menu choice	Purpose
Tools->Manage Graph Themes	Create/delete graph themes. See *Graph Themes* and *Graph Theme Manager*.
Tools->View Logs	View the log files. See *Log Viewer*.

4.9.5 Window menu

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

4.9.6 Help

Menu choice	Purpose
Help->Manual	Displays the GraphExpert Professional manual (HTML version).
Help->Web Site	Displays the GraphExpert Professional web site.
Help->Discussion Forum	Displays the GraphExpert Professional user-to-user discussion forum.
Help->Tutorial Videos	Displays the available video tutorials for GraphExpert Professional
Help->Activate Trial Period	activates a 30 day trial period for GraphExpert 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

MANAGING DATASETS AND FUNCTIONS

5.1 Introduction

GraphExpert Professional is designed to take raw data as generated by your own process (such as experimental data, raw survey results, etc.), manage and transform that data, and then visualize the data. This chapter focuses on the management and manipulation of these data. To most effectively use the application, an understanding of components is necessary.

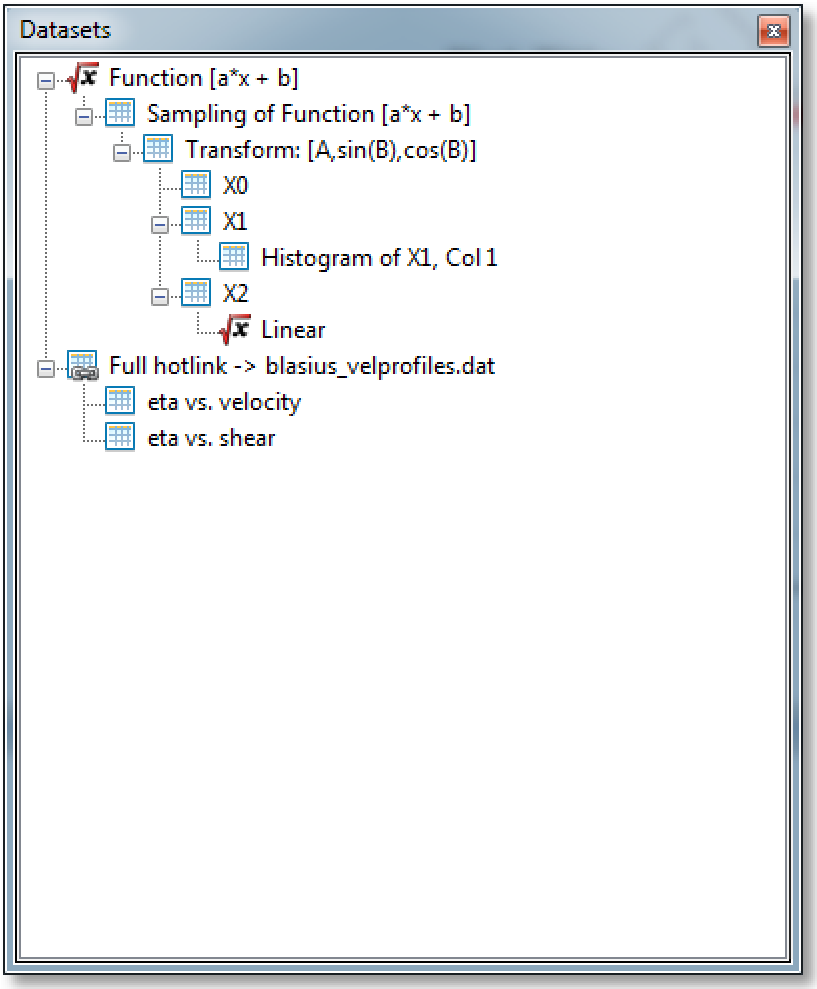
5.2 What is a Component?

By way of definition, a *component* is anything that can be visualized in one or more graphs. A dataset is a component that is discrete in nature; a function is a component that is continuous in nature. Both can be managed and manipulated in GraphExpert Professional. Datasets are typically created as documented in *Creating new datasets*. Functions are created as documented in *Functions*. In the abstract sense, a component is simply a source of data; either directly as is the case with datasets, or indirectly as is the case with functions (a function must be queried in order to return data).

The crux of GraphExpert Professional's management is that a component (function or dataset) can be derived from another. If this is done, there is then a parent-child relationship between these two components, and the child is automatically updated whenever the parent changes. Any changes in components are also automatically reflected on any graphs that happen to be referring to the changed components.

Hovering the mouse over any component will show an informational tooltip relevant to that particular component.

To move a component up or down (in order to reorder them), select the desired component, hold the Alt key, and press the up/down arrows.


Datasets
Function [a*x + b]
Sampling of Function [a*x + b]
Transform: [A,sin(B),cos(B)]
X0
X1
Histogram of X1, Col 1
X2
Linear
Full hotlink -> blasius_velprofiles.dat
eta vs. velocity
eta vs. shear

5.3 Operations on any component

5.3.1 Details

The details for any component can be queried by right clicking it, and selecting *Details* from the resulting menu. The details shown are appropriate for the component selected.

5.3.2 Send to current graph

A component can be visualized on the currently visible graph (in the “Graphs” pane) by right clicking it, and selecting *Send to current graph*. Alternatively, the component can be dragged and dropped onto the visible graph.

5.3.3 Send to new graph

A component can be visualized on a new graph (in the **Graphs Pane**) by right clicking it, and selecting *Send to new graph*.

5.3.4 Renaming

Any component can be renamed by right clicking it, and selecting *Rename* from the resulting menu. Alternatively, the component can be renamed by left clicking on an already-selected component, which makes the name editable in-place. Also, F2 can be pressed to begin the renaming process.

5.3.5 Cloning

Any component can be cloned by right clicking it, and selecting *Clone* from the resulting menu. Cloning a component creates a copy of it, in its current state, and places the copy at the top level of the component hierarchy (meaning that the clone does not depend on any other component). The clone’s name is the same as the component that it was cloned from with “Copy of” prepended.

If the component being cloned has descendants, these are cloned as well and properly placed as descendants of the clone.

5.3.6 Deletion/Removal

Any component can be removed by right clicking it, and selecting *Delete* from the resulting menu. If a component is deleted, any graph that is visualizing that component will update appropriately (the series will be removed).

5.3.7 Moving Up/Down

Any component can be moved up or down in relation to its siblings. One way to perform this operation is to right click the component, and select *Move Up* or *Move Down* from the resulting menu. Alternatively, you can use the Alt-ArrowUp and Alt-ArrowDown keys. The primary purpose of moving your components up and down is simply to organize and order your components as desired.

5.4 Operations on Datasets

5.4.1 Column split

A dataset can be column split, placing each of its columns into a separate dataset for further manipulation. To perform this, right click a dataset, and select **Column split** from the resulting menu. This creates ncol new child datasets, where ncol is the number of columns in the dataset selected. Since these new datasets are children, they are noneditable and update automatically when the parent changes.

5.4.2 Split

A dataset with multiple dependent variables can be split, which means to create n new datasets from the marriage of the independent variables with each of the dependent variables separately. For example, if a five-column dataset is present with two independent variables and three dependent variables, then three new child datasets will be created: dataset #1 with the independent variables and dependent variable #1, dataset #2 with the independent variables and dependent variable #2, and dataset #3 with the independent variables and dependent variable #3.

Since these new datasets are children, they are noneditable and update automatically when the parent changes.

5.4.3 Join

Multiple datasets may be joined into one. To join datasets, first shift-select (or ctrl-select) them, right click, and select **Join into NxM dataset**; N and M will be appropriately filled by the size of the dataset that will be formed from the operation.

GraphExpert Pro automatically detects whether or not the selected datasets are compatible, and in what direction the joining can occur:

- If the selected datasets all have the same number of columns, the **Join** menu option will indicate that the datasets will be stacked vertically in the joining process. For example, a 10x2 and 15x2 dataset will result in a 25x2 dataset (25 rows and 2 columns).
- If the selected datasets all have the same number of rows, the **Join** menu option will indicate that the datasets will be stacked horizontally in the joining process. For example, a 10x2 and a 10x3 dataset will results in a 10x5 dataset (10 rows and 5 columns).
- If the selected datasets match in both rows and columns, both joining directions will appear in the context menu, and you will be able to choose the desired joining process.
- If the selected datasets do not have the same number of columns (or rows), the **Join** context menu option will not be shown at all.

The datasets will be joined in the order that they were selected.

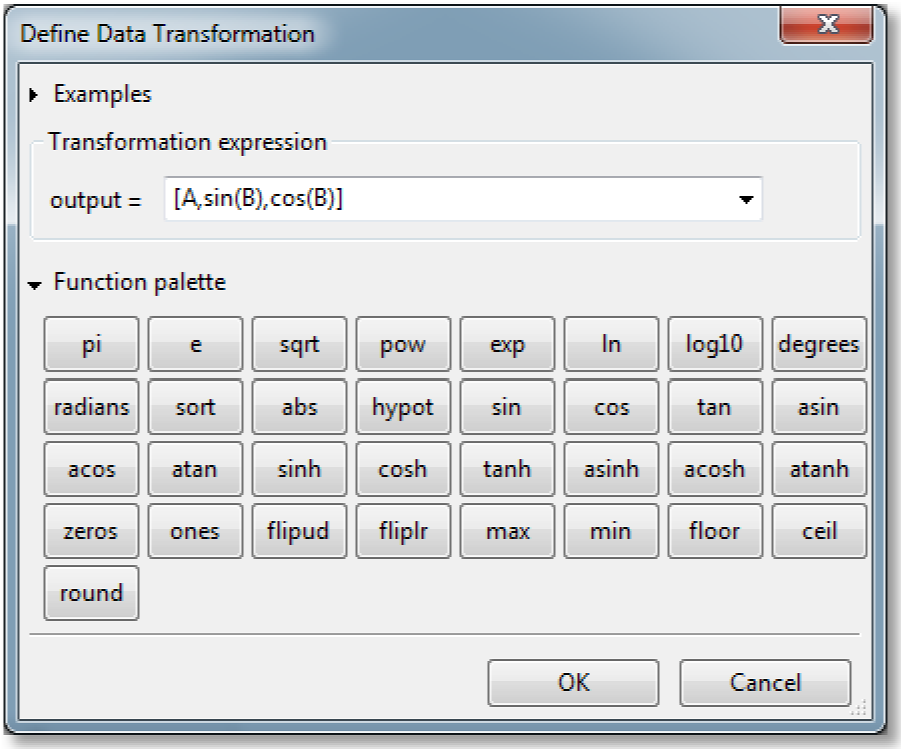
5.4.4 Reclassify

A dataset can have its columns reclassified according their desired use (independent variable, dependent variable, or standard deviation column (see *Datasets*). To perform this, right click a dataset, and select **Reclassify** from the resulting menu. See *Reclassifying columns* for details on how to attain the classification that you desire for your dataset.

If the desire is to rearrange columns, a better tool is the *Transform* capability, documented below.

5.4.5 Transform

The dataset transformation tool in GraphExpert Professional is a very useful and powerful mechanism that can be used to derive other datasets from one that is present. To access the tool, right click a dataset and select **Transform**. The following dialog will be presented:



output = [A,sin(B),cos(B)]

A new expression can be typed directly into the **Transformation expression** field. Here, the full complement functions as documented in *Appendix A: Math Functions* can be utilized. For convenience, however, a function palette can be expanded, which appears as a set of calculator-like buttons. Pressing any of these palette buttons will insert the corresponding text into the **Transformation expression** field, or you are free to type the expression out yourself. Remember that the function palette only has a subset of the available functions in GraphExpert Professional.

A transformation expression is essentially just a list of a columns (more generally, a list of 2D datasets). This list of columns is glued together horizontally in order to form a new dataset. The letters "A", "B", "C", etc. stand for columns of the original dataset (obviously, this limits transformations to dealing with only the first 26 columns of the dataset using this notation; use DSET if it is desired to access more columns; eg., DSET[:,27], which means all rows and the 28th column). The name "DSET" stands for all of the columns of the original dataset.

Slice indexing (e.g., A[5:10], A[-5:], A[0:-1]) can also be used in order to specify a region, as well. Slices are zero based, and the slice runs from the first index up to but not including the last index. With the slicing operators, you can manipulate any subset of the dataset that you like. For example, A[5:10] will extract the 6th through 10th rows of column A. Or, DSET[1:3:5:10] extracts the 6th through 10th rows of columns 1 and 2.

The transpose operator ".T" may be used to take the transpose of any column, dataset, or section of a dataset. For example, if you want to extract the top left section of a dataset and take the transpose of that, the expression would be [DSET[0:5,0:10].T].

Examples of usage of the transformation dialog are below; these examples range from extremely simple to moderately advanced, to demonstrate the types of transformations that can be accomplished.

In all of these examples, let us assume that the original dataset (the one that we selected when choosing **Transform** from the context menu) has four columns.

To extract the first column:

```
[A]
```

To move the last two columns to the front:

```
[C,D,A,B]
```

To create a dataset consisting of the first column and three copies of the last column:

```
[A,D,D,D]
```

To keep the first column and apply the sin() function to the second column (discarding the rest):

```
[A,sin(B)]
```

To create a new dataset keeping the first column, and adding the other three columns of the original dataset to form the second column:

```
[A,B+C+D]
```

To create a new dataset, offsetting column B by a constant to form three new columns:

```
[A,B+1,B+2,B+3]
```

To square all entries in the second column:

```
[A,B^2,C,D]
```

To raise the elements in column B to the power given by the corresponding elements in column C:

```
[A,B^C]
```

To convert column A from degrees to radians, sort B, and reverse the entries in column D, and leave column C alone:

```
[radians(A),sort(B),C,flipud(D)]
```

To add on a new column consisting of all fives:

```
[A,B,C,D,ones_like(A)*5]
```

To make a copy of the input dataset (same action as [A,B,C,D]):

```
[DSET]
```

To prepend column A to the input dataset, creating a 5 column dataset (same action as [A,A,B,C,D]):

```
[A,DSET]
```

To create a new dataset with rows 5-10 of columns A and C:

```
[A[4:10],C[4:10]]
```

To create a new dataset with first column and last two columns (note that DSET is a 2D array, so it requires two slicing indices):

```
[A,DSET[:,-2:]]
```

To take the standard deviation of every column in a dataset (the dataset returned will be a 1x4 array):

```
[standarddev(DSET)]
```

To take the average of the last 3 columns of the dataset (the dataset returned will be a 1x3 array:

```
[average(B),average(C),average(D)]   -- or -- [average(DSET[:,-3:])]
```

To transpose the entire dataset:

```
[DSET.T]
```

To save every 4th row of the entire dataset:

```
[DSET[::4]]
```

Also, previous data transformations that have been defined by the user can be accessed via the small down-arrow pulldown at the right end of the **Transformation Expression** field.

The transformed dataset is a child of the original one, which means that the transformation will be automatically re-executed whenever the parent dataset changes.

In the end, the result of the transformation is a 2D dataset. If the transformation expression that you have entered is "ragged"; meaning that each entry in the list does not have the same number of rows, GraphExpert Professional will zero fill in order to make the number of rows consistent.

Note: The OK button will not enable until the expression given is valid.

5.4.6 Sort

To sort your dataset, right click a dataset component and select **Sort** from the resulting Operator menu. You will be queried for the column to use as the key when sorting, and then a new child dataset will be created that contains the sorted values. Sorting is vital for correct results when splining, integration, and differencing.

5.4.7 Transpose

To transpose your dataset, right click a dataset component and select **Transpose** from the resulting Operator menu. Functionally, the transpose operator is identical to doing a *Transform* with DSET.T as the transformation expression. However, transposing directly with this menu item also has the feature that row names and column names are also flipped, if they are set.

5.4.8 t-test

To perform a t-test for two columns of data, shift-select or control-select the datasets you want to compute a t-test for, right click, and select **t-test** from the resulting menu. The t-test will be computed, using each column in all of the selected datasets (there should always be exactly two total columns in all of the selected datasets) as a separate, independent, population.

The calculation of the t-test will give you a new top-level component, which holds the t-test calculation results. The t and p value are given. Standard practice is to reject the null hypothesis that the two columns of data were sampled from populations with the same mean, if $P < 0.05$ (this is for 95% confidence).

Note that t-tests will NOT update as their parents are changed; there is currently no capability within GraphExpert Professional to keep track of multiple parents; for this reason, a top-level, non-updating component is created. If you want to update the t-test due to its parent datasets changing for some reason, it must be recomputed.

5.4.9 ANOVA (Analysis of Variance)

To perform a one-way ANOVA of a list of datasets, shift-select or control-select the dataset components you want compute an ANOVA for, right click, and select **ANOVA** from the resulting menu. The ANOVA will be computed, using each column in all of the selected datasets as a separate population.

The calculation of the ANOVA will give you a new top-level component, which holds the ANOVA calculation results. The F and p value are given, along with critical F-values for confidences of 85%, 90%, 95%, and 99% respectively. If F < Fcrit, the null hypothesis (that there is no statistical difference between the set of columns) cannot be rejected. Correspondingly, if F > Fcrit, the null hypothesis can be rejected; there is a statistical difference between the set of columns. The P-value is the probability that one or more of the columns are drawn from a different population, and can certainly be used in lieu of comparing F to Fcrit. Standard practice is to reject the null hypothesis that all of the data were sampled from populations with the same mean, if P < 0.05 (this is for 95% confidence).

Note that ANOVAs will NOT update as their parents are changed; there is currently no capability within GraphExpert Professional to keep track of multiple parents; for this reason, a top-level, non-updating component is created. If you want to update the ANOVA due to its parent datasets changing for some reason, it must be recomputed.

5.4.10 Process Replicates

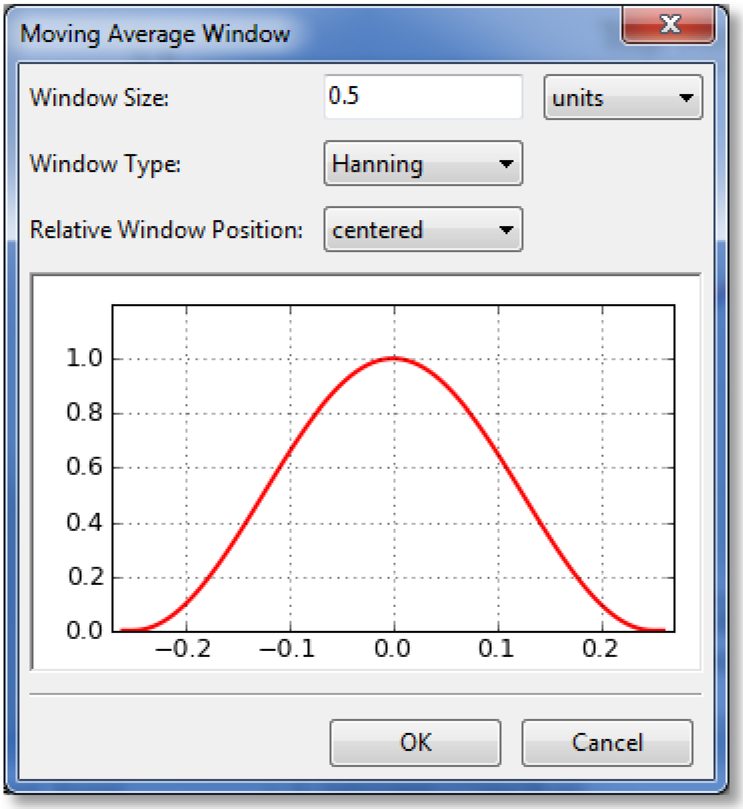
The **Process Replicates** operator works on a dataset with multiple dependent variables. An average and standard deviation (across the rows) is taken for all dependent variables, Then, the dependent variable columns are replaced with the average and standard deviation columns.

For example, let's say that your 50x16 dataset consisted of single x (independent varaible) column, and 15 y (dependent variable) columns. Process replicates will take the average and standard deviation across the rows of the 15 dependent variable columns. The 50x3 output dataset then consists of your original x column, the average column, and the standard deviation column.

This operation is common when the dependent variable columns are repeated measurements for the same dependent variable. This functionality allows you to quickly reduce this situation to a data set that has a column for the independent variable, a column for the dependent variable, and a column for the standard deviation in the dependent variable.

5.4.11 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, right click a dataset in the Components pane, and select **Moving average** from the resulting **Operators** 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 a child component is created in which the moving average is contained.

5.4.12 Compute Differences

The differences of a dataset can be computed by right clicking a dataset component, and selecting **Compute differences** from the resulting menu. The dependent variable of the resulting dataset is the difference between adjacent rows of the dependent variable of the parent dataset, and the independent variable is the average of adjacent rows of the independent variable of the parent dataset. A new child dataset is created for each differencing operation. As such, the differences are automatically updated whenever the parent's data is changed.

Differences can be taken for any dataset with a single independent variable. Note that single column datasets have an implied independent variable, which is the row index.

5.4.13 Compute integral

The integral of a dataset can be computed by right clicking a dataset component, and selecting **Compute integral** from the resulting menu. An integral is computed via Simpson's rule, using the independent variable vs. each dependent variable in the dataset. Note that the result of an integral operation is a 1x1 dataset. A new child dataset is created for each integral operation. As such, the integrals are automatically updated whenever the parent's data is changed.

Integrals can be taken for any dataset with a single independent variable. Note that single column datasets have an implied independent variable, which is the row index.

5.4.14 Compute Histogram

A histogram (single column datasets only) can be computed by right clicking a dataset component, and selecting **Compute Histogram** from the resulting menu. A dialog will appear such that the number of bins in the histogram can be selected.

The resulting dataset contains the bin edges as the independent variable, and the frequency of occurrence as the dependent variable. This dataset is a child of the original dataset, and as such, will update automatically whenever the parent's data is changed.

5.4.15 Compute DFT

A Discrete Fourier Transform (single column datasets only) can be computed by right clicking a dataset component, and selecting **Compute DFT** from the resulting menu. The resulting dataset contains the frequency in as the independent variable, and power (which the square of the real and imaginary components of the computed DFT for the data) as the dependent variable. Only the first half of the power spectrum is valid (due to the Nyquist criterion) and therefore only the first half is returned. When the input data has a length that is a power of 2, the Fast Fourier Transform (FFT) algorithm is used to compute the DFT.

This dataset is a child of the original dataset, and as such, will update automatically whenever the parent's data is changed.

5.4.16 Linear Regression

A linear regression can be computed by right clicking a dataset component, and selecting **Linear Regression** from the resulting menu. The result of performing the linear regression is the function

$$y = a + bx$$

where a and b have been adjusted as to optimally fit the data. This function is a child of the dataset, and as such, will update automatically whenever the parent's data is changed.

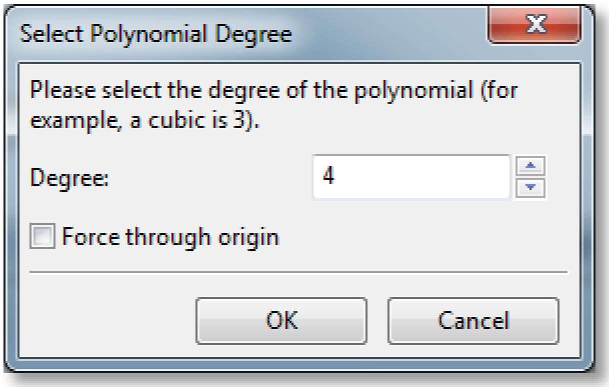
5.4.17 Polynomial Regression

A polynomial regression can be computed by right clicking a dataset component, and selecting **Linear Regression** from the resulting menu. The result of performing the polynomial regression is the function

$$y = a + bx + cx^2 + dx^3....$$

where the parameters a, b, c, etc. have been adjusted as to optimally fit the data. This function is a child of the dataset, and as such, will update automatically whenever the parent's data is changed.

Upon choosing a polynomial regression, the following dialog is presented:



Select Polynomial Degree

Please select the degree of the polynomial (for example, a cubic is 3).

Degree: 4

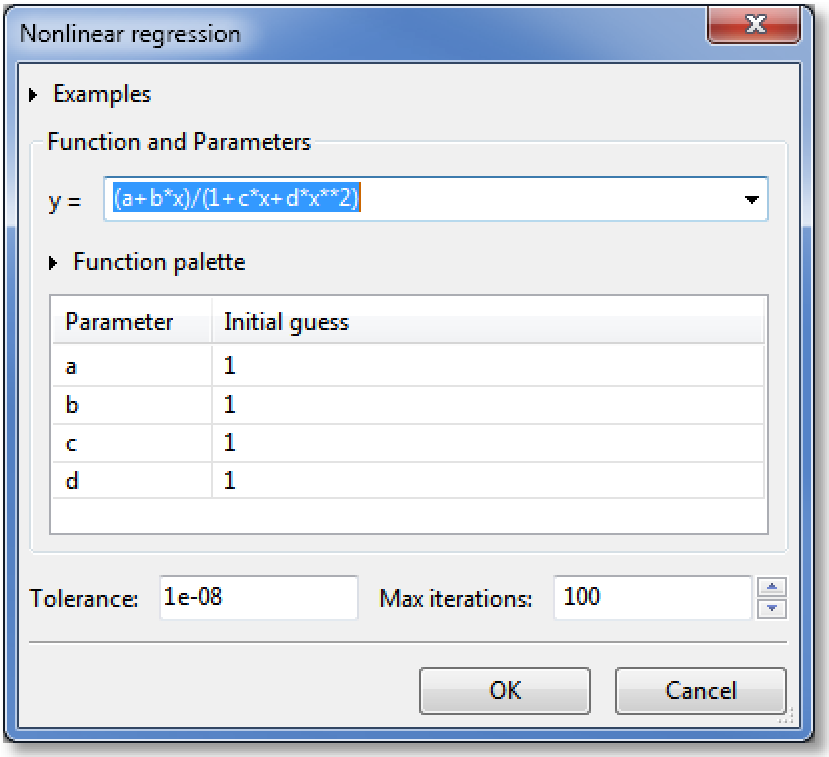
□ Force through origin

Here, the degree of the polynomial regression (up to 19th degree polynomials are supported) can be selected, and the user can choose whether to force the polynomial through the origin; in that case, the constant a (the intercept) is always zero, and only the other parameters are optimized.

This function is a child of the dataset, and as such, will update automatically whenever the parent's data is changed.

5.4.18 Nonlinear Regression (curve fit)

A nonlinear regression can be computed by right clicking a dataset component, and selecting **Nonlinear Regression** from the resulting menu. Upon choosing a nonlinear regression, the following dialog is presented, that allows you to define the model that will be used for the curve fit:



y = (a+b*x)/(1+c*x+d*x**2)

Parameter	Initial guess
a	1
b	1
c	1
d	1

Tolerance: 1e-08 Max iterations: 100

A new expression that defines the model can be typed directly into the **y =** field. Here, the full complement of functions as documented in *Appendix A: Math Functions* can be utilized. For convenience, however, a function palette can be expanded, which appears as a set of calculator-like buttons. Pressing any of these palette buttons will insert the corresponding text into the **y =** field, or you are free to simple type the expression out yourself. Remember that the function palette only has a subset of the available functions in GraphExpert Professional.

Also, previous models that have been defined by the user can be accessed via the small down-arrow pulldown at the right end of the **y =** field.

In the expression, "x" is always interpreted to mean the independent variable. Likewise, if you are forming a 3D function, "x1" and "x2" are interpreted to mean the first and second independent variables, respectively. If there are more independent variables than two, continue with x3, x4, etc.

Any other variables introduced in the expression are taken to be parameters that will be optimized by the nonlinear regression (curve fitting) process. For example, if your expression is "cos(freq*x)", then "freq" is taken to be an adjustable parameter. Parameters are automatically detected as you type.

Most of the time for well-scaled data, the default initial guesses do not need to be changed. However, in some situations, the default initial guesses are inappropriate (leading to an immediate divide by zero; for example, if the model is y=x/(1-a)) or the default initial guesses do not allow the optimization process to converge. In this case, you as the user should supply reasonable initial guesses for the model's parameters. Doing do will greatly enhance the performance of the optimization algorithms in GraphExpert Professional.

The **Tolerance** setting dictates how accurate the optimization is; the setting is used for changes in the parameters, and changes in the residual. Neither can be above the tolerance setting for a successful optimization. The default value is 1.0e-8, which states that the optimization process cannot be terminated until all parameters and the residual change by less than 1.0e-8 during an iteration of the algorithm.

The **Max iterations** setting determines that maximum number of iterations that the optimization algorithm is allowed. In most case, it takes no more than ten iterations for an optimization to converge, but this parameter prevents runaway calculations from occurring. An optimization that exceeds the max number of iterations will return semi-successfully, in that you will have a curve fit to work with. However, use caution in using output from such nonconverged results.

Note: The OK button will not enable until the expression given for the model is valid.

This nonlinear regression is a child of the dataset, and as such, will update automatically whenever the parent's data is changed.

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 `()`.

Reserved Words

In any model, the independent variable is always denoted by `x`. If you are writing a function 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 cannot not be used as a parameter name in your function.

All mathematical functions supported by GraphExpert 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 GraphExpert 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.

5.4.19 Lowess Smoothing

A Lowess smoothing can be computed by right clicking a dataset component, and selecting **Lowess Smoothing** from the resulting 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 GraphExpert Professional 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.

This function is a child of the dataset, and as such, will update automatically whenever the parent's data is changed.

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.

5.4.20 Cubic Spline

A cubic spline can be computed by right clicking a dataset component, and selecting **Cubic Spline** from the resulting menu. The result of performing the cubic spline is a function made up of piecewise cubics

$$y = a + bx + cx^2 + dx^3$$

that are continuous at the knots, as well as the first two derivatives being continuous at the knots. Note that the spline computed is a natural cubic spline, which means that 2nd derivative is zero at the endpoints.

This function is a child of the dataset, and as such, will update automatically whenever the parent's data is changed.

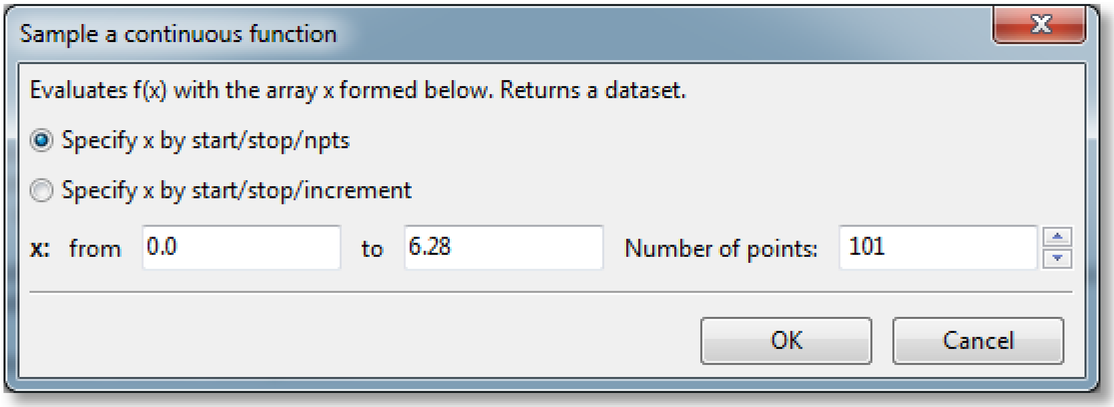
5.5 Operations on Functions

5.5.1 Edit Function

Selecting **Edit Function** will allow you to edit the expression (for *Simple functions*) or Python code (for *Programmed functions*), the parameters, and the default domain for a function.

5.5.2 Sample

Sampling gives the ability to create a dataset from a continuous function. A 2D or 3D function can be sampled by right clicking it, and then selecting **Sampling** from the resulting menu. The following dialog will be shown:



Sample a continuous function

Evaluates f(x) with the array x formed below. Returns a dataset.

● Specify x by start/stop/npts

○ Specify x by start/stop/increment

x: from 0.0 to 6.28 Number of points: 101

The sampling can be specified by either giving the start and stop point along with the desired number of points (samples), or one can give the start and stop point along with the distance (increment) between each sample. Upon pressing OK, the new dataset is created. This new dataset is a child of the function, and as such, will update automatically whenever the parent is changed (for functions, this typically means that the function parameters are changed in some way).

For a 3D function, a similar sampling dialog will appear, but allows the user to sample in the two independent variable directions.

5.5.3 Compute Derivative

The derivative of the function at a particular point can be computed by right clicking a function component, and selecting **Compute derivative** from the resulting menu. The derivative is computed via finite difference, and the result is a new dataset component that contains the result. A new child dataset created for for the derivative; as such, the derivative is automatically updated whenever the parent function is updated.

Derivatives can be taken for any function with a single independent variable.

5.5.4 Compute integral

The integral of a function can be computed by right clicking a dataset component, and selecting **Compute integral** from the resulting menu. A dialog will appear to allow you to type in the limits of integration. An integral is then computed, with the output being a 1x1 dataset. A new child dataset is created for each integral operation; as such, the integral is automatically updated whenever the parent function is updated.

An integral can be taken for any continuous function with a single independent variable.

CHAPTER

SIX

DATASETS

6.1 Introduction: Classifying dataset columns

Datasets in GraphExpert Professional are a two-dimensional table of numbers; a dataset is made up of a number of rows and a number of columns.

Datasets may have an arbitrary number of independent variables and an arbitrary number of dependent variables. Optionally, the dependent variable(s) may each have an associated standard deviation data column. If standard deviation columns are present, they follow each dependent variable column.

The layout of a dataset is:

```
{x1,x2,...,xj,y1,[std1],y2,[std2],...,yk,[stdk]}
```

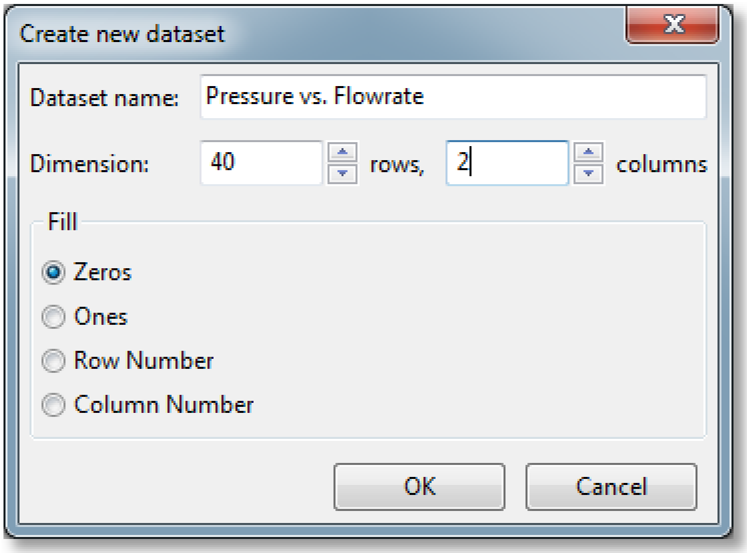
where j is the number of independent variables, and k is the number of dependent variables. The brackets indicate optional columns.

6.2 Creating new datasets

New datasets in GraphExpert Professional can be created via several means, which are listed below.

6.2.1 From scratch

A new dataset can be created directly via the *Create->Dataset* menu choice (or the corresponding toolbar button). The following dialog will then appear:



Create new dataset

Dataset name: Pressure vs. Flowrate

Dimension: 40 rows, 2 columns

Fill

● Zeros

○ Ones

○ Row Number

○ Column Number

Here, you may set the dimension of the dataset (number of rows and number of columns), as well as the initial value of the values in the dataset. Choosing a **Fill** of *Zeros* or *Ones* will obviously fill the new dataset will all zeros or ones as appropriate. Choosing a **Fill** of *Row Number* or *Column Number* will fill your dataset with the one-based row or column number for each entry, respectively.

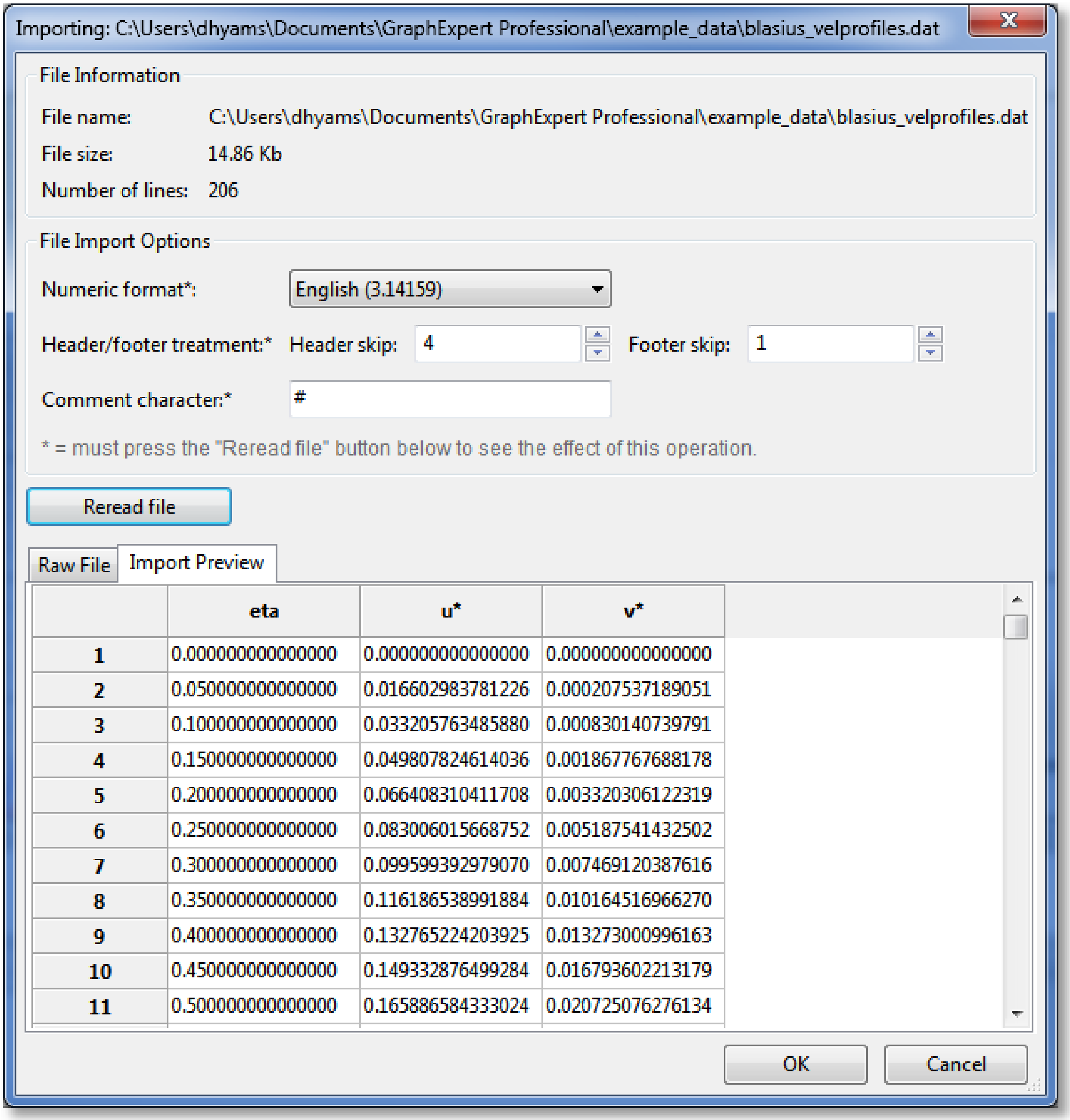
It is perfectly reasonable to create a new dataset with zero rows and zero columns; in this case, the usual workflow is to double-click the dataset in the "Components" panel in order to edit it and add values manually.

6.2.2 Importing from file

A new dataset can be created by reading a file with the raw values inside. To do this, choose *File->Import Data*; GraphExpert Professional provides a robust file import mechanism. The file to be read in may have any extension that you choose (.dat and .txt are common).

Most of the time, the data to be read in is contained within a text file of some sort, with headers and comments interspersed. GraphExpert 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.

After selecting the file to be read in, the file import dialog is displayed:



Importing: C:\Users\dhyams\Documents\GraphExpert Professional\example_data\blasius_velprofiles.dat

File Information

File name: C:\Users\dhyams\Documents\GraphExpert Professional\example_data\blasius_velprofiles.dat

File size: 14.86 Kb

Number of lines: 206

File Import Options

Numeric format*: English (3.14159)

Header/footer treatment:* Header skip: 4 Footer skip: 1

Comment character:* #

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

	eta	u*	v*
1	0.000000000000000	0.000000000000000	0.000000000000000
2	0.050000000000000	0.016602983781226	0.000207537189051
3	0.100000000000000	0.033205763485880	0.000830140739791
4	0.150000000000000	0.049807824614036	0.001867767688178
5	0.200000000000000	0.066408310411708	0.003320306122319
6	0.250000000000000	0.083006015668752	0.005187541432502
7	0.300000000000000	0.099599392979070	0.007469120387616
8	0.350000000000000	0.116186538991884	0.010164516966270
9	0.400000000000000	0.132765224203925	0.013273000996163
10	0.450000000000000	0.149332876499284	0.016793602213179
11	0.500000000000000	0.165886584333024	0.020725076276134

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.

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.

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.

GraphExpert 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 GraphExpert 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.

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 '%'.

Refreshing the preview

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

File import preview and raw file preview

The file import preview shows the result of the reading of the file, augmented by the current import settings discussed above. This preview shows you exactly what GraphExpert Professional will read as a dataset.

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.

Column Labels

GraphExpert 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 punctutation “()+-*”. 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)”.

Other file metadata

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

Finishing

When the file import preview looks like the desired dataset, hit OK; your data will be read and a new dataset created for it in the “Components” pane.

6.2.3 Importing from clipboard

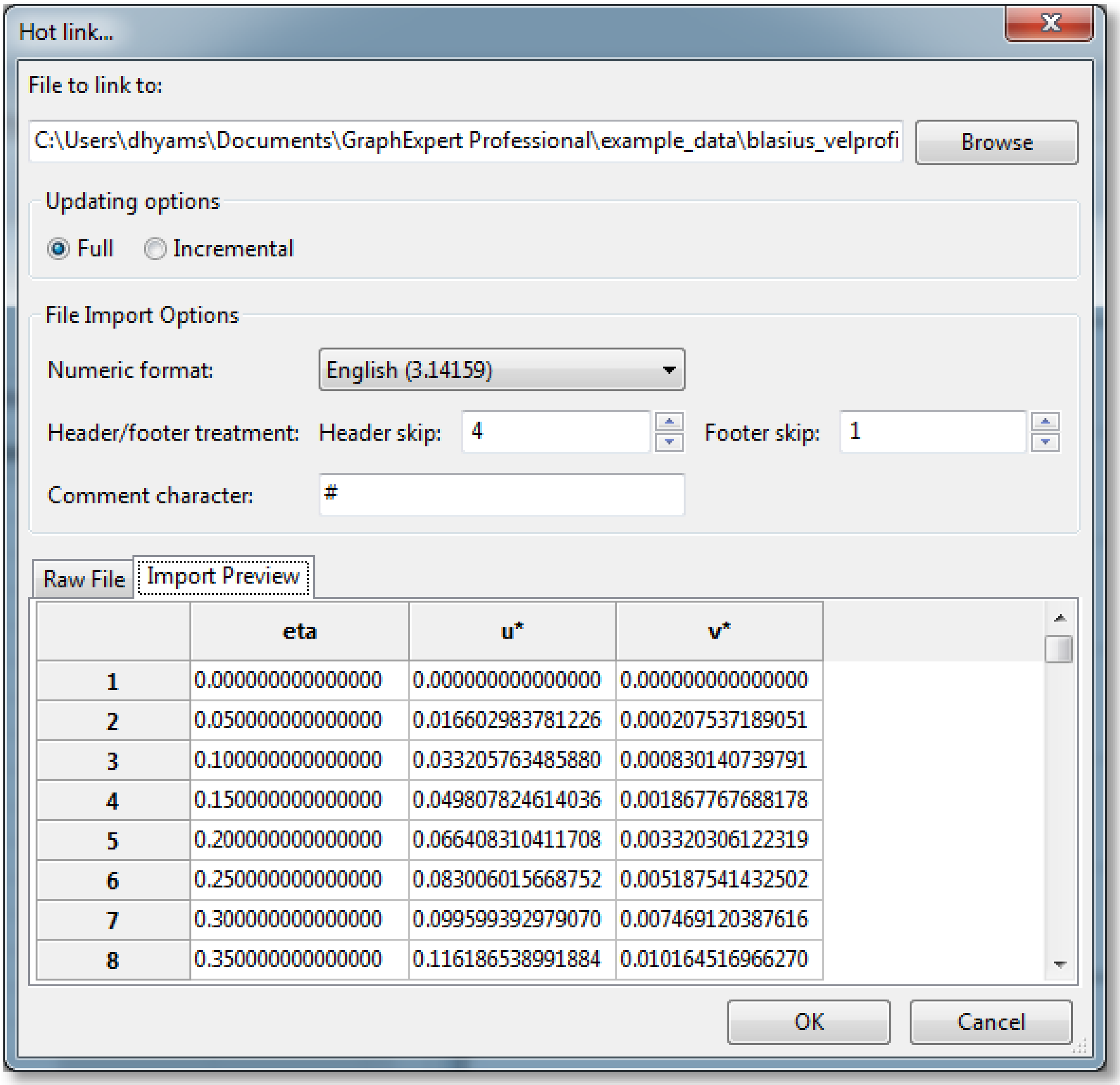
A new dataset can be created from data stored on the clipboard. To do this, choose *File->Import Data from Clipboard*. If there is valid data present, it is immediately imported and made available in the “Components” pane. If valid data is not present, a message will appear in the “Messages” pane.

A typical use of this function is to import data from Excel or any other application that stores its data in a table-like format. Highlight the desired numerical data in the other application, copy it, and then switch to GraphExpert Professional and select *File->Import Data from Clipboard*. Alternatively, you may paste the data directly in the spreadsheet, if you have a dataset already created in GraphExpert Professional ready to receive the data. For this, see *The spreadsheet*.

6.2.4 Hot linking to a text or Excel file

In some cases, data is continually written to a file, or it is desired to automatically update a dataset within GraphExpert Professional when the data inside a particular file changes. Here, the hot linking support in GraphExpert Professional should be used. A hot link automatically updates the dataset whenever the underlying file (called the hot link) is changed. To be more specific, a file is "changed" whenever its modification time, as reported by the operating system, changes. A change in modification time is checked for periodically by GraphExpert Professional. A hot link can point to a regular text file, or to an Excel file (.xls or .xlsx).

To create a hotlink, select *Create->Hot link* from the menu, or click the corresponding toolbar button. The following dialog will appear:



File to link to: C:\Users\dhyams\Documents\GraphExpert Professional\example_data\blasius_velprofi

Updating options

● Full ○ Incremental

File Import Options

Numeric format: English (3.14159)

Header/footer treatment: Header skip: 4 Footer skip: 1

Comment character: #

	eta	u*	v*
1	0.000000000000000	0.000000000000000	0.000000000000000
2	0.050000000000000	0.016602983781226	0.000207537189051
3	0.100000000000000	0.033205763485880	0.000830140739791
4	0.150000000000000	0.049807824614036	0.001867767688178
5	0.200000000000000	0.066408310411708	0.003320306122319
6	0.250000000000000	0.083006015668752	0.005187541432502
7	0.300000000000000	0.099599392979070	0.007469120387616
8	0.350000000000000	0.116186538991884	0.010164516966270

The first step to create a hot link is to set the name of the file to link to; the *Browse* button helps to locate the file if needed. Technically, however, the file does not need to exist at the time the hot link is created.

To a text file

The **Updating options** should then be set to *Full* or *Incremental*. A full hot link means that the entire file is reread each time the file changes. This type of updating is appropriate for files where the entire content might change. An incremental hot link is appropriate for files that have new values appended to them periodically (for example, a log file). Here, GraphExpert Professional only examines the bit of the file that has been added to the end of a file since the last update. The additional data is read and appended to the existing dataset.

Note: The data generated from an incremental hot link may become corrupt or incorrect if the update happens at exactly the same time that a third-party application is writing to the file. In this rare case, the data may be completely refreshed by right clicking the hot link in the “Components” pane, and selecting *Force hot link refresh*.

After the updating options, there are several **File import options** that will be used when reading the underlying file. These options are the same as that discussed above for the regular file importing procedure (see *Importing from file*). First, the **Numeric Format** can be set to English or European, in order to properly recognize the decimal points used within the file. Next, the **Header skip** and **Footer skip** can be set to skip over non-data items in your file. GraphExpert Professional looks at the file itself, and guesses at the appropriate header and footer skip, but obviously you are free to set these values as appropriate. Finally, the *Comment character* can be set, which tells the file importing mechanism within GraphExpert Professional which lines are non-data lines (i.e. comment lines). All comment lines will be skipped automatically as the file is read.

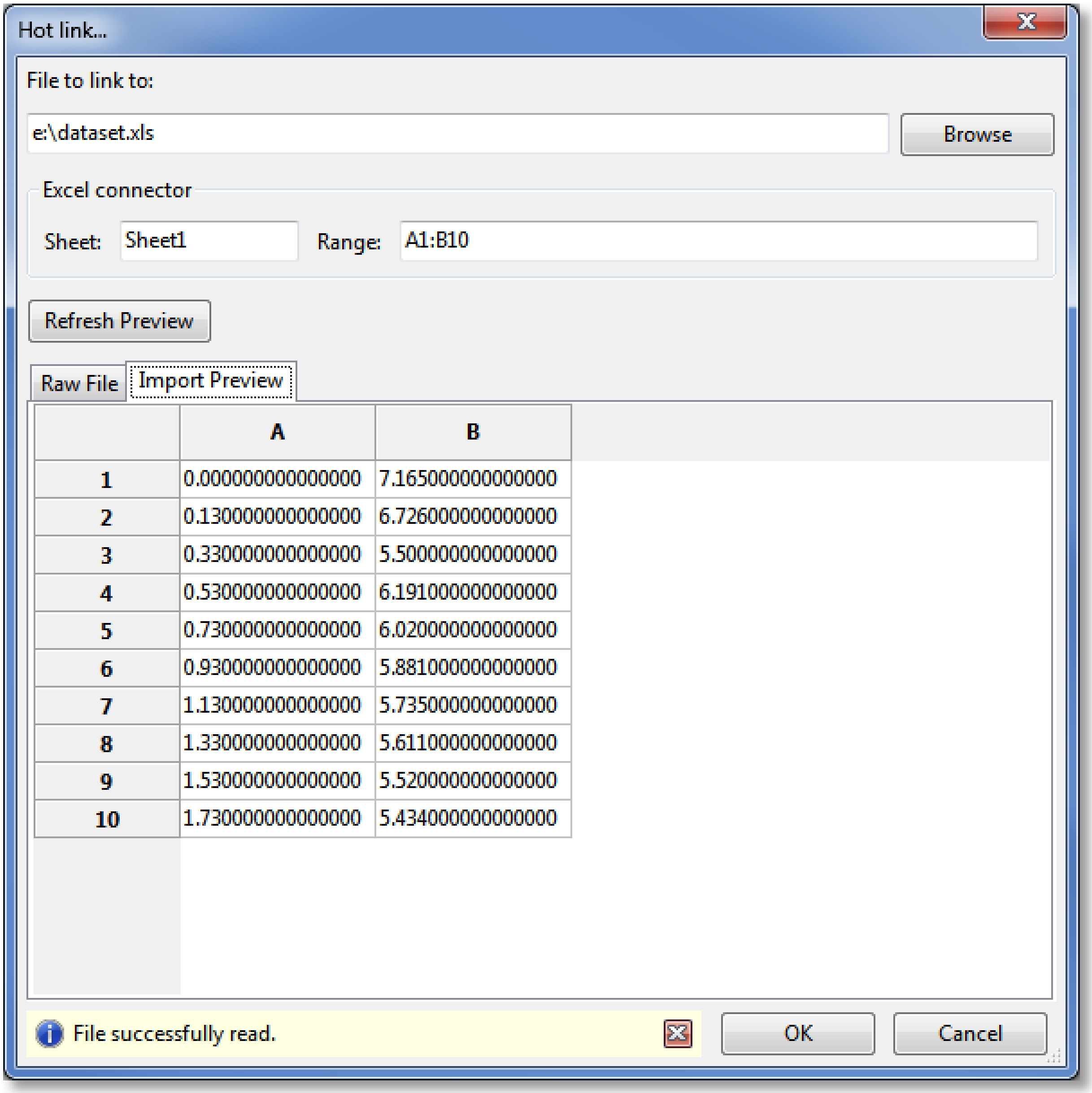
The previewing pages at the bottom of the dialog show the contents of the file itself, as well as what the data in the file looks like after reading, using the current import settings.

Once the OK button in this dialog is pressed, the hot link will appear in the “Components” pane, and will automatically update when the underlying file changes.

To an Excel file

GraphExpert Professional supports hot linking directly to data inside of an Excel(TM) file, in either .xls or .xlsx formats. This is often of great convenience to the practitioner, since often the raw data is contained in a spreadsheet; this Excel connector allows GraphExpert Professional to connect directly to that data source rather than requiring a second copy of the data.

To create this hot link, just point the “File to link to:” entry to an Excel file. As soon as a .xls or .xlsx file extension is detected, the dialog will change its controls such that the location of the data within the spreadsheet may be set as follows:



File to link to:

e:\dataset.xls

Sheet: Sheet1

Range: A1:B10

	A	B
1	0.000000000000000	7.165000000000000
2	0.130000000000000	6.726000000000000
3	0.330000000000000	5.500000000000000
4	0.530000000000000	6.191000000000000
5	0.730000000000000	6.020000000000000
6	0.930000000000000	5.881000000000000
7	1.130000000000000	5.735000000000000
8	1.330000000000000	5.611000000000000
9	1.530000000000000	5.520000000000000
10	1.730000000000000	5.434000000000000

File successfully read.

In the **Excel Connector** section, enter the sheet name where the data resides, as well as the range where the data resides. The range is specified exactly as you see the data within the Excel spreadsheet. A simple example of a range, which extracts two columns and ten rows from the top left corner of a particular sheet, is:

```
A1:B10
```

Also, noncontiguous ranges are supported through the use of the comma notation:

```
A1:B10,D2:D11
```

would result in a 10x3 dataset. If the number of rows between noncontiguous selections does not match, GraphExpert Professional automatically pads each column with zeros such that each column of the dataset has a consistent number of rows. Press the **Refresh Preview** button to see the results of your extraction settings.

The previewing pages at the bottom of the dialog show the results of the Excel(TM) data extraction, using the current *File*, *Sheet*, and *Range* settings. Once the OK button in this dialog is pressed, the hot link will appear in the “Components" pane, and will automatically update when the underlying Excel(TM) file changes. Note that live changes in the Excel(TM) file cannot be tracked; the Excel(TM) must be saved for GraphExpert Professional to "notice" that something has changed.

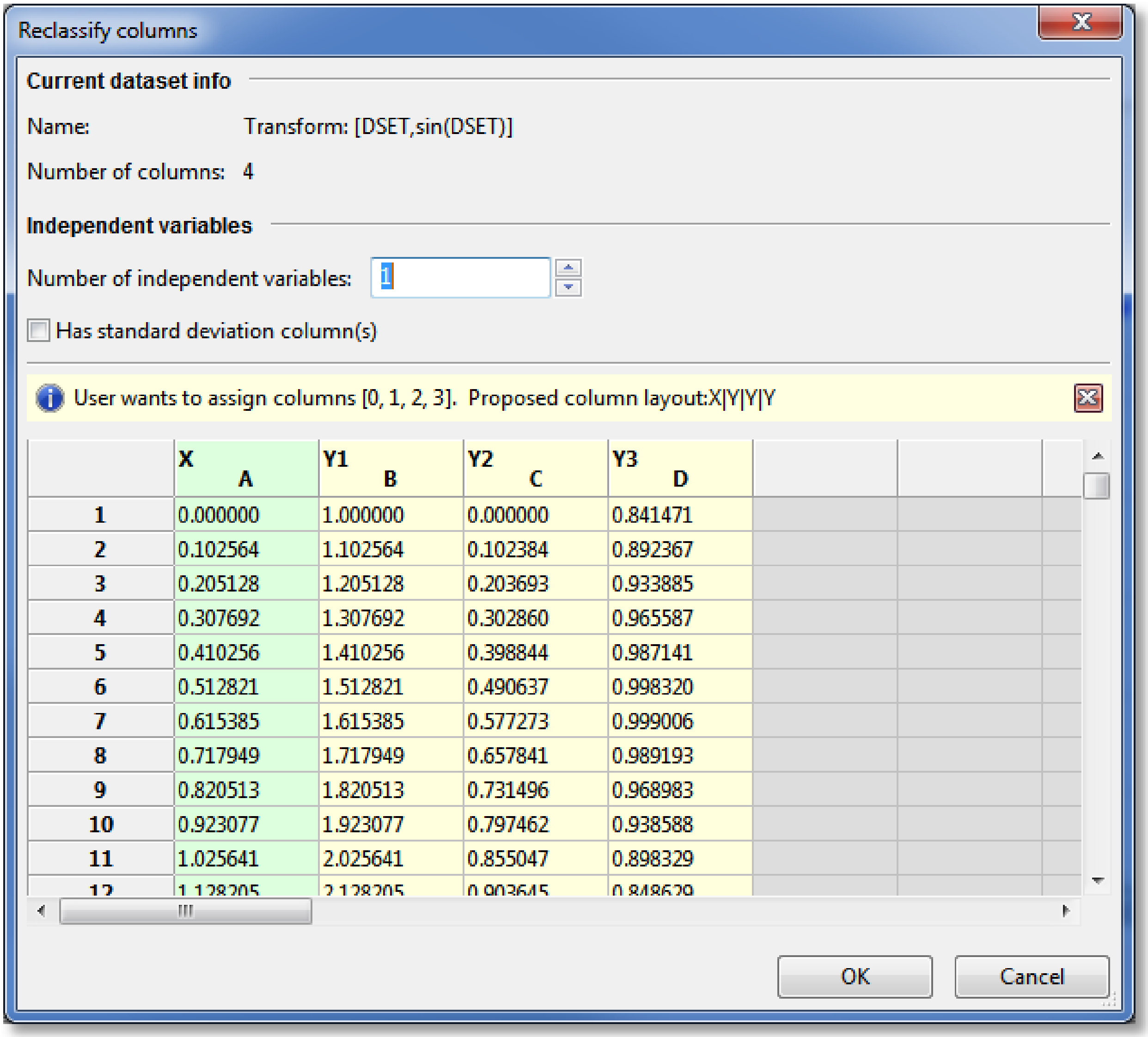
A hot link to an Excel file behaves just as another other hot link does. The modification time of the Excel file is monitored, and if the file is updated, your data in GraphExpert Professional is automatically updated.

6.3 Reclassifying columns

A dataset, since it is just a collection of columns, needs some additional information as to how to interpret the meaning of those columns (i.e. which columns are independent variables, dependent variables, or standard deviation). This "metadata" in GraphExpert Professional is accomplished by explicitly storing the list of columns that make up the independent variables, dependent variables, and standard deviation columns. GraphExpert Professional, by default, guesses that the first column of a dataset is the independent variable and the rest are dependent variables (except in the case of a one-column dataset, in which case it guesses that the column is a dependent variable). If this guess is not correct for your data, you will need to reclassify the columns in order to describe what each column contains.

Note: Even child datasets can have their columns reclassified. Reclassification affects the metadata of the dataset, not the data itself (which updates automatically), and therefore is a legal and logical operation to perform on a child dataset.

To reclassify the columns of a dataset, right click a dataset in the "Components" pane and select **Reclassify**. Doing so will display a dialog similar to the following:



Current dataset info

Name: Transform: [DSET,sin(DSET)]

Number of columns: 4

Independent variables

Number of independent variables: 1

☐ Has standard deviation column(s)

User wants to assign columns [0, 1, 2, 3]. Proposed column layout:X|Y|Y|Y

	X A	Y1 B	Y2 C	Y3 D		
1	0.000000	1.000000	0.000000	0.841471		
2	0.102564	1.102564	0.102384	0.892367		
3	0.205128	1.205128	0.203693	0.933885		
4	0.307692	1.307692	0.302860	0.965587		
5	0.410256	1.410256	0.398844	0.987141		
6	0.512821	1.512821	0.490637	0.998320		
7	0.615385	1.615385	0.577273	0.999006		
8	0.717949	1.717949	0.657841	0.989193		
9	0.820513	1.820513	0.731496	0.968983		
10	0.923077	1.923077	0.797462	0.938588		
11	1.025641	2.025641	0.855047	0.898329		
12	1.128205	2.128205	0.903645	0.848629		

The top section of the dialog gives some information about the dataset being reclassified, such as the name and number of columns in the dataset.

The next section of the dialog, *Independent variables*, allows you to set the number of independent variables in the dataset, and also whether or not standard deviation columns are present. If standard deviation columns are present, there shold be one stddev column following each dependent variable column.

As selections are made in the upper part of the dialog, the lower part shows a preview of the dataset, simultaneously informing the user the effects of the independent variable settings and standard deviation column settings. Independent variable columns are colored green, and dependent variable columns are colored yellow. Standard deviation columns are colored orange. Finally, unused columns are colored red.

It is not erroneous to leave certain columns unused in a dataset. GraphExpert Professional will behave appropriately should you graph the dataset; any unused columns are ignored. The red highlighting and warning message in this dialog is there to inform the user of that fact, however.

Also for identification, each column header is labeled with “X”,”Y”, or “STD” in the upper left area of the header; this means that the column is currently tagged as an independent variable, dependent variable, or standard deviation column, respectively. The same information is shown in the middle information bar as in the following example:

```
User wants to assign columns [0,1].  Proposed column layout: X|Y|S|_
```

Again, “X” means independent variable, “Y” means dependent variable, “S” means a standard deviation column, and “_” means that the column is currently unassigned to any role.

If the reclassification settings are invalid, this fact will be shown also in the information bar area above the dataset preview.

It is perfectly reasonable to set the number of independent variables to zero. In this case, GraphExpert Professional, uses the one-based row index of each data value as the independent variable when needed (i.e. for graphing).

6.4 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*:

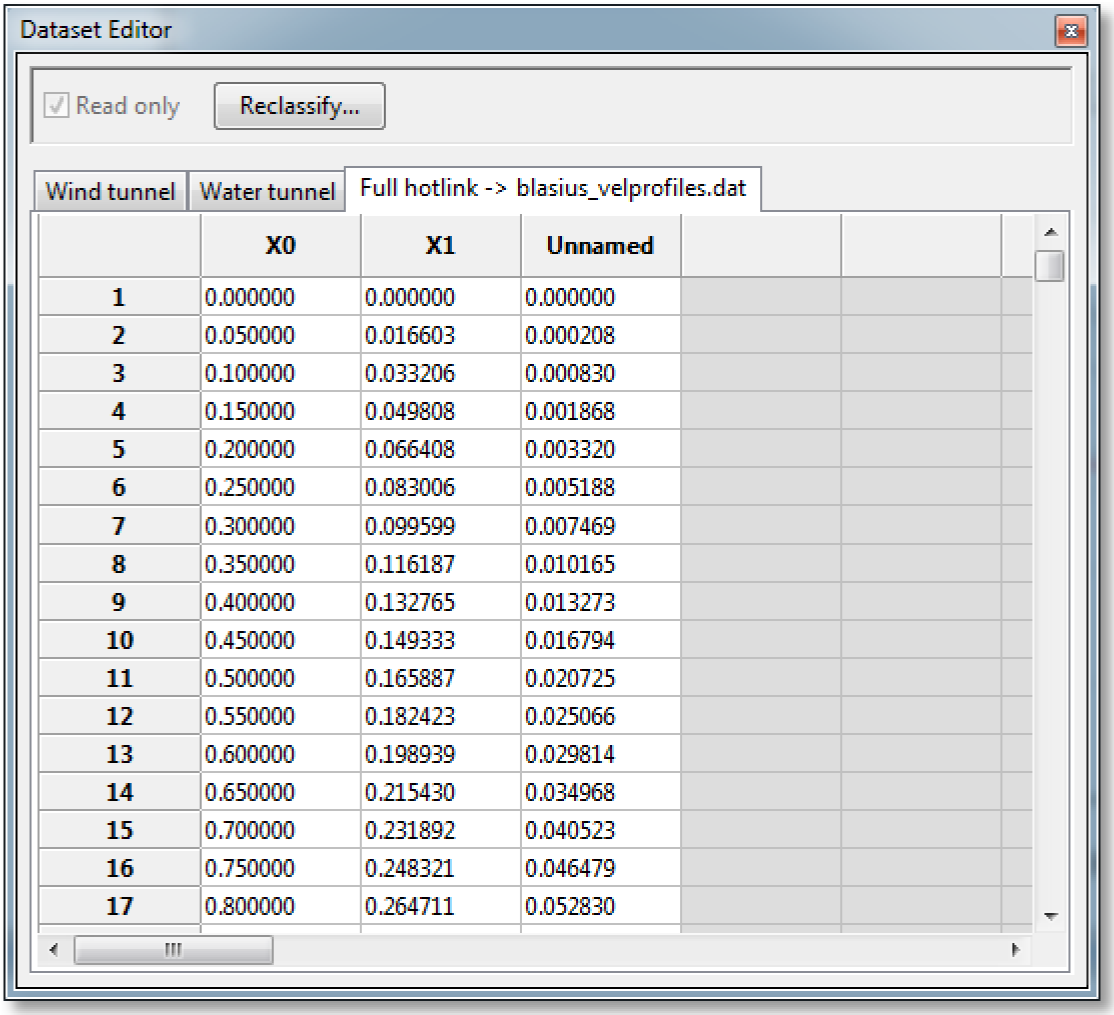
$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

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

6.5 The spreadsheet

The spreadsheet resides in the “Dataset Editor” pane in GraphExpert Professional.

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



	X0	X1	Unnamed		
1	0.000000	0.000000	0.000000		
2	0.050000	0.016603	0.000208		
3	0.100000	0.033206	0.000830		
4	0.150000	0.049808	0.001868		
5	0.200000	0.066408	0.003320		
6	0.250000	0.083006	0.005188		
7	0.300000	0.099599	0.007469		
8	0.350000	0.116187	0.010165		
9	0.400000	0.132765	0.013273		
10	0.450000	0.149333	0.016794		
11	0.500000	0.165887	0.020725		
12	0.550000	0.182423	0.025066		
13	0.600000	0.198939	0.029814		
14	0.650000	0.215430	0.034968		
15	0.700000	0.231892	0.040523		
16	0.750000	0.248321	0.046479		
17	0.800000	0.264711	0.052830		

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.

6.5.1 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.

6.5.2 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 either be a complete set of rows, or a complete set of columns in the dataset. Therefore, the “Cut” item in the data menu is dimmed if your selection does not fit the above criteria. This behavior is due to the fact that a cut can remove complete rows or complete columns from your dataset, but cannot create a “hole” in the dataset.

A paste from the clipboard informs GraphExpert 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.

6.5.3 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”.

6.5.4 Inserting a Column

To insert a column into your dataset, right click on the the spreadsheet and select “Insert Column”. For this option to appear, you must either be right clicking on a column header, or have a complete set of columns selected in the spreadsheet. A column will be inserted at the leftmost selection point, with following columns pushed rightward in order to make room. Note if the dataset is a member of any plot, it will be removed; insertion of a column means that the dataset is implicitly reclassified, which forces its removal from graphs that it appears in.

6.5.5 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.

6.5.6 Removing Columns

To remove columns from a dataset, right click on the spreadsheet and select “Remove Column”. For this option to appear, you must either be right clicking on a column header, or have a complete set of columns selected in the spreadsheet. All selected columns will be removed, with the unremoved columns to the right of the selection sliding left to fill the vacated space. Note if the dataset is a member of any plot, it will be removed; insertion of a column means that the dataset is implicitly reclassified, which forces the removal from graphs that it appears in.

6.5.7 Undo and Redo

So, you can feel comfortable manipulating data knowing that any transformation, clear, or cut/paste operation can be undone.

6.6 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).

6.7 Operating on Data

GraphExpert 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, right click the spreadsheet, and then select one of the data operations from the resulting context menu. If no selection is made, the operation is assumed to apply to the entire dataset (all rows and all columns). Note that any operation performed in this manner is done in-place; i.e., the dataset is being directly modified and no children are created.

6.7.1 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 GraphExpert 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, integrating your data, differencing your data, or other operations that depend on the data being sorted. Remember that the sorting, if performed directly on the data using the spreadsheet, performs the sorting in-place and no child will be created.

6.7.2 Scale

To multiply by or divide the selected data by a factor, right click on the spreadsheeet and pick **Scale**. The scaling is performed in-place on the dataset (i.e. no child is created).

6.7.3 Translate

To add a constant value to the selected data, right click on the spreadsheet and pick **Translate**. The translation is performed in-place on the dataset (i.e. no child is created).

6.7.4 Crop

To drop points from the selected part of your dataset that are above or below a certain threshhold, right click on the spreadsheet and pick **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.

6.7.5 Clamp

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

6.7.6 Rotate

To rotate columns in a dataset, right click on the spreadsheet and select **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.

6.7.7 Transforming Data

Right clicking on the spreadsheet and picking **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.

Transforming data directly in the spreadsheet does the operation in-place, and does not create a child dataset. To create a dataset transformation that does create a child dataset, see *Transform*.

6.7.8 Extracting Data

A subset of the dataset in the spreadsheet can be extracted as a child of the dataset. To do this, simply highlight the desired section of your dataset that you would like to extract, right click the spreadsheet, and select **Extract Selection**. This will create a child of that dataset, which will automatically update its data as the parent is modified.

The selection must be a contiguous section of the dataset.

CHAPTER

SEVEN

FUNCTIONS

7.1 Introduction

Functions in GraphExpert Professional are handled as an “evaluator object” that can generate a single dependent variable, on-demand, from a given independent variable or variables. A function can optionally have parameters that are adjustable either by the user, or by an algorithm such as a regression.

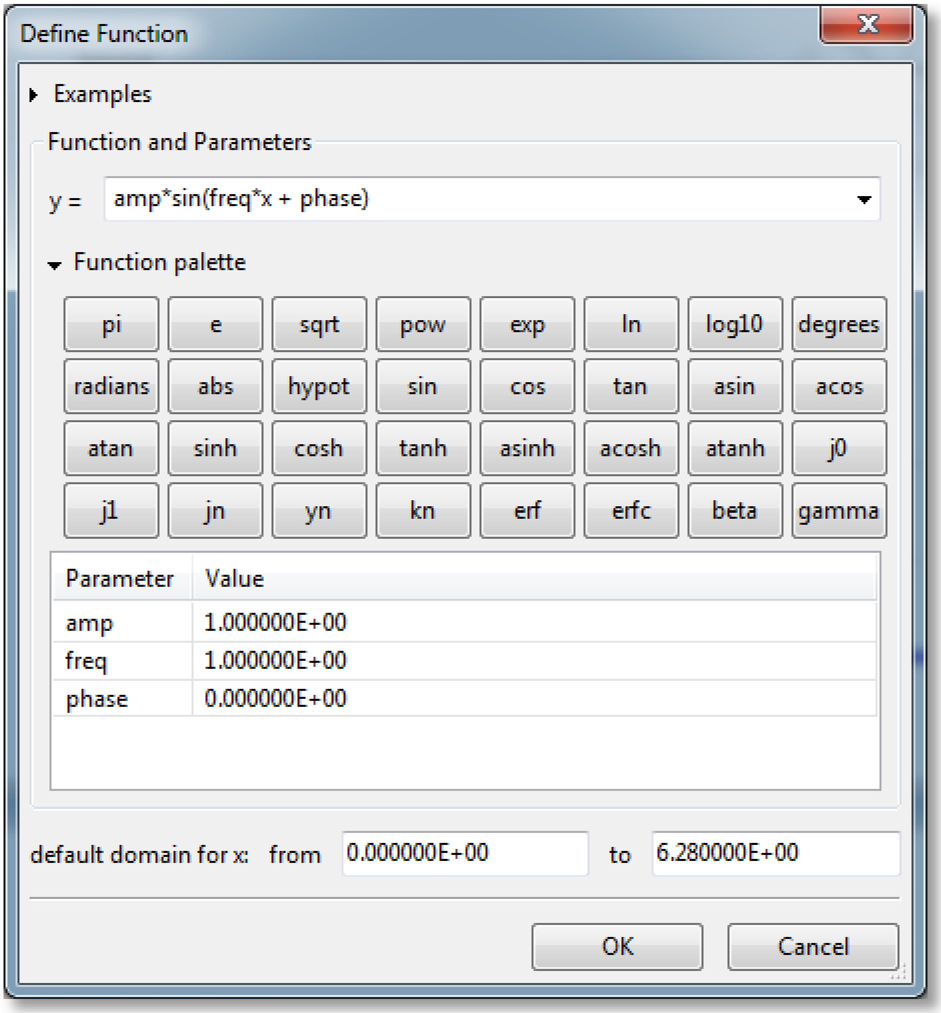
There are two types of functions in GraphExpert Professional: *Simple functions*, and see *Programmed functions*. Simple functions are just a single expression (y = f(x)), and are sufficient for most purposes. However, if your needs are more demanding, a programmed function allows you to write a Python code that can be as complex as required to evaluate a function.

7.2 Simple functions

New functions in GraphExpert Professional can be created via several means, which are listed below.

7.2.1 From scratch

A new function can be created directly via the *Create->Function* main menu choice (or the corresponding toolbar button). The following dialog will then appear:



y = amp*sin(freq*x + phase)

Parameter	Value
amp	1.000000E+00
freq	1.000000E+00
phase	0.000000E+00

default domain for x: from 0.000000E+00 to 6.280000E+00

A new expression can be typed directly into the **y =** field. Here, the full complement of functions as documented in *Appendix A: Math Functions* can be utilized. For convenience, however, a function palette can be expanded, which appears as a set of calculator-like buttons. Pressing any of these palette buttons will insert the corresponding text into the **y =** field, or you are free to simple type the expression out yourself. Remember that the function palette only has a subset of the available functions in GraphExpert Professional.

In the expression, "x" is always interpreted to mean the independent variable. Likewise, if you are forming a 3D function, "x1" and "x2" are interpreted to mean the first and second independent variables, respectively.

Any other variables introduced in the expression are taken to be parameters. This allows the creation of a function with adjustable parameters. For example, if your expression is "cos(freq*x)", then "freq" is taken to be an adjustable parameter, which you can then set in the **Parameter/Value** table. Parameters are automatically detected as you type.

The default domain for your function affects how it is autoscaled on any graph that it is placed on. The autoscaling mechanism will look at the default domain range and take this as a bound for the x-autoscaling, and do a quick evaluation of the function over the default domain to determine the y-autoscaling. The default domain mechanism works analagously for a 3D function; the default domain is set for the x1 and x2 axes and the same procedure applies.

Also, previous functions that have been defined by the user can be accessed via the small down-arrow pulldown at the right end of the **y =** field.

Note: The OK button will not enable until the expression given is valid.

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 `()`.

Reserved Words

In any model, the independent variable is always denoted by `x`. If you are writing a function 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 cannot not be used as a parameter name in your function.

All mathematical functions supported by GraphExpert 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 GraphExpert 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 function'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.

7.2.2 As a child of a dataset

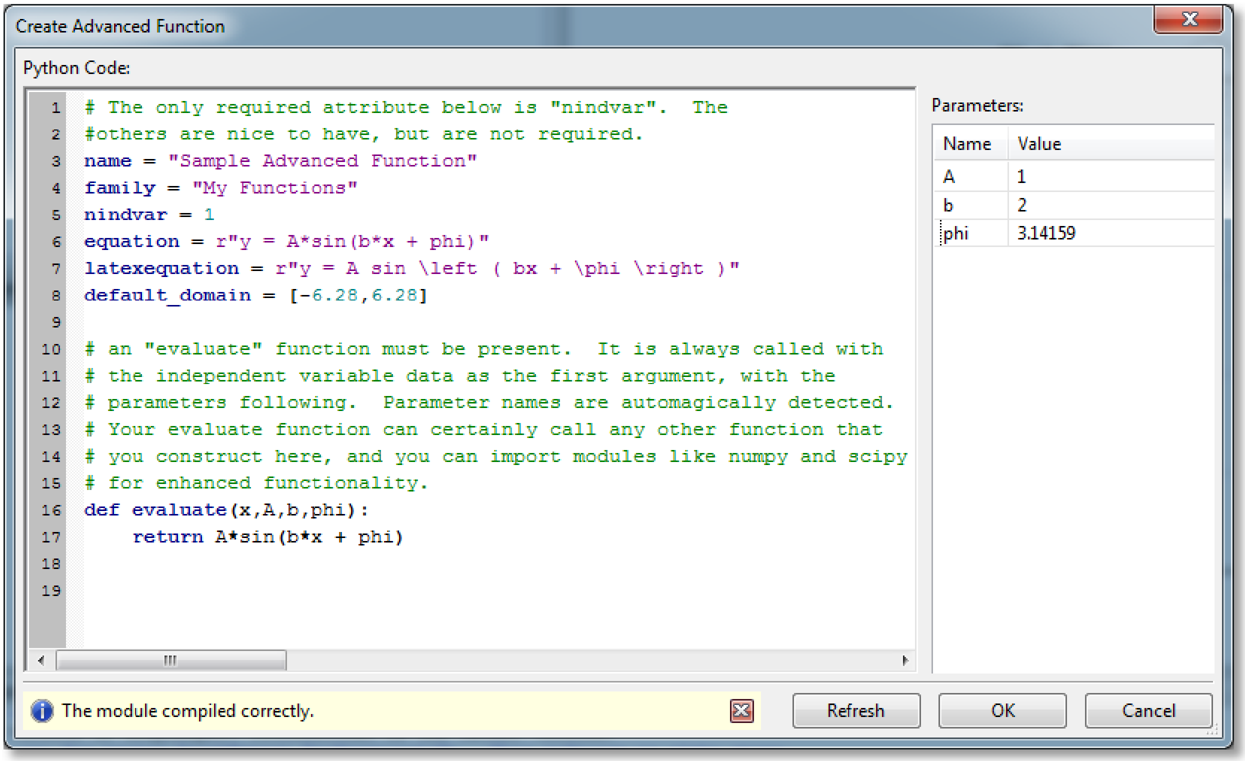
GraphExpert Professional supports various operations on datasets that result in a function. Two examples are linear regression and cubic splines. Creating a function in this manner is documented in *Operations on Datasets*.

7.2.3 Via drag and drop from CurveExpert Professional

Any result computed from CurveExpert Professional can be dragged into the **Components** pane to add it to GraphExpert Professional. If the result is dragged onto the graph itself, it is both added as a component, and added to that graph in one step.

7.3 Programmed functions

To create a new programmed function, choose **Create->Programmed Function** from the main menu. The following dialog will then appear:



```python
# The only required attribute below is "nindvar".  The
#others are nice to have, but are not required.
name = "Sample Advanced Function"
family = "My Functions"
nindvar = 1
equation = r"y = A*sin(b*x + phi)"
latexequation = r"y = A sin \left ( bx + \phi \right )"
default_domain = [-6.28,6.28]

# an "evaluate" function must be present.  It is always called with
# the independent variable data as the first argument, with the
# parameters following.  Parameter names are automagically detected.
# Your evaluate function can certainly call any other function that
# you construct here, and you can import modules like numpy and scipy
# for enhanced functionality.
def evaluate(x,A,b,phi):
    return A*sin(b*x + phi)
```

Name	Value
A	1
b	2
phi	3.14159

The module compiled correctly.

To make sure that your function is valid code, press the *Refresh* button. If the code compiles correctly (this does not imply correctness of the code itself), an information bar will appear at the bottom to indicate this. If the code does not compile correctly, a stack trace will appear in the information bar to help you track down the problem.

The second purpose of the *Refresh* button is to allow a rescan of the code to determine the number of parameters that are passed to the "evaluate" functions. These parameters appear in the *Parameters* list, and you can set their numerical values there as well.

7.3.1 Writing the code

Every programmed function must have one `evaluate` function, and the `nindvar` attribute must be set. The `evaluate` function is what will be called whenever a result is required of your programmed function. `nindvar` informs the application how many independent variables are involved when the function is evaluated.

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

As you enter the Python code for your function, it is syntax highlighted, and syntax errors are detected dynamically. The standard cut/copy/paste operators are also available for use as you write your code.

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 programmed function. Just follow the examples given here and in the programmed function dialog itself.

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.

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

Every programmed function must have an `evaluate` function defined, and a `nindvar` attribute set. Other items that specify metadata (`name`, `equation`, `latexequation`, `default_domain`), are optional.

7.3.2 `nindvar`

This is a variable to define the number of independent variables that this programmed function expects. Most of the time, this value will be 1, but can obviously be higher depending on how many independent variables that you want to be associated with your programmed function. If greater than one, the x variable passed to `evaluate` is a 2D dataset consisting of all of the independent variable columns.

7.3.3 `name`

This is an attribute to define the name of your programmed function, which is the name that is used for the component that is created for your programmed function, by default. The type of this attribute is a string, and should be enclosed in quotes.

7.3.4 `family`

The family attribute is not currently used but is reserved for future use.

7.3.5 `equation`

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

7.3.6 `latexequation`

This is an attribute 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 a graphical rendering of your equation is required in GraphExpert Professional. This attribute is optional, but if not given, renderings will show “no equation”.

7.3.7 `default_domain`

The default domain is used for autoscaling purposes, and should be given as a list with a min and max value that should be used as the limits for autoscaling. The default domain itself determines the limits in the x, and the limits of the function evaluated over the default domain determines the limits in the y. An example is:

```
default_domain = [0.1,1.1]
```

For a 3D function (nindvar = 2), the default domain should be given as a 2D list; for example:

```
default_domain = [[0.1,1.1],[-1.0,1.0]]
```

which gives a domain of 0.1 to 1.1 on the x1 axis, and -1.0 to 1.0 on the x2 axis.

Remember that the default domain only suggests a range to the autoscaling mechanism, and that is its only purpose. It is certainly possible to visualize more of your function as needed, in a different range.

7.3.8 `evaluate(x,param1,param2,param3,...)`

This is the most important part of your programmed function; `evaluate()` is a function that is called whenever your program is evaluated with the independent variables and a set of parameters. This obviously occurs when GraphExpert Professional needs to draw the function on a graph.

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):

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

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

CHAPTER

EIGHT

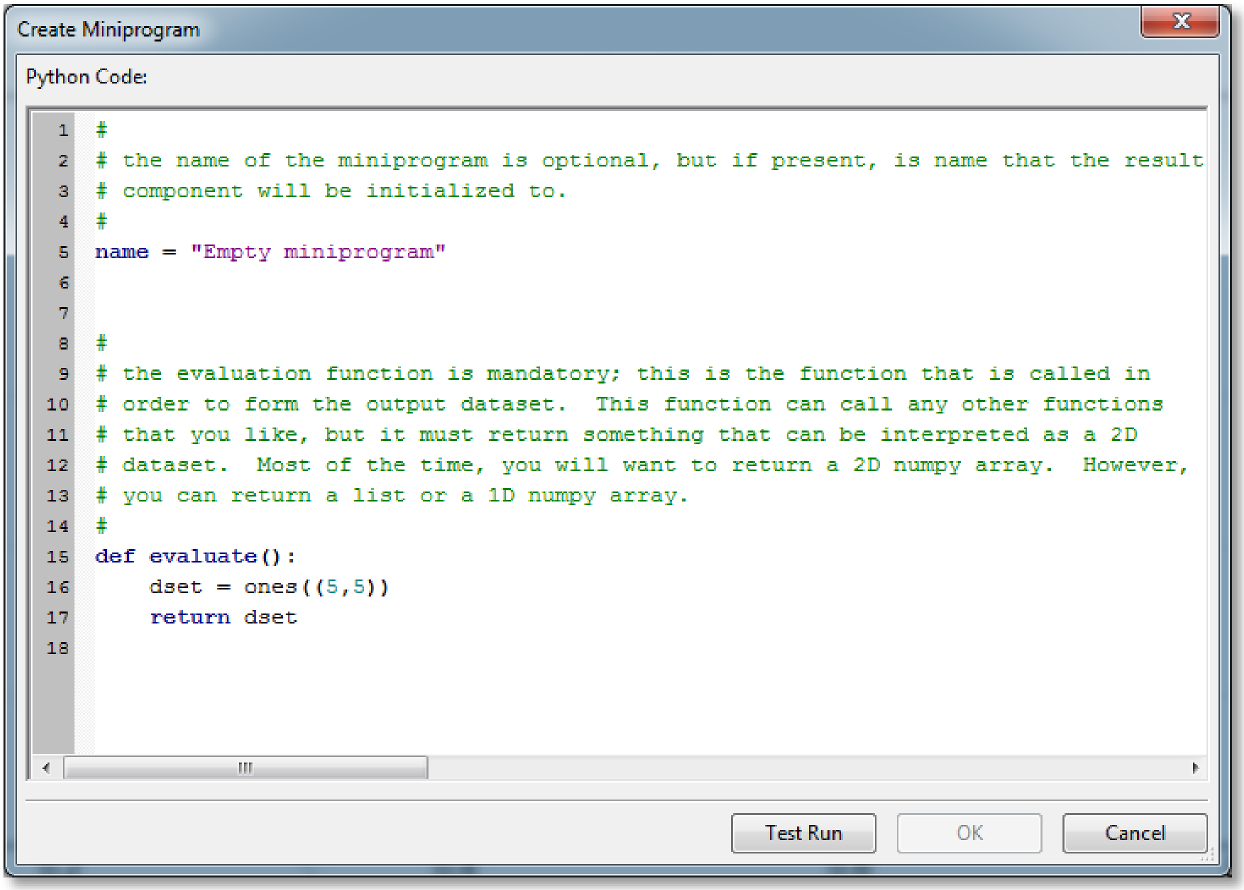
MINIPROGRAMS

8.1 Introduction

Miniprograms in GraphExpert Professional are “small” Python codes that output a dataset. In essence, a miniprogram *is* a dataset, except that the source of the data is a program rather than being a set of numbers in the computer’s memory. A miniprogram can be as simple or as complex as the user desires, and the full power of the Python language and add-on modules is available for use. For convenience, GraphExpert Professional provides all of the functions listed in *Appendix A: Math Functions* in your miniprogram’s namespace for direct use.

8.2 Creating a miniprogram

A new miniprogram can be created directly via the *Create->Miniprogram* main menu choice (or the corresponding toolbar button). The following dialog will then appear:



```
#
# the name of the miniprogram is optional, but if present, is name that the result
# component will be initialized to.
#
name = "Empty miniprogram"


#
# the evaluation function is mandatory; this is the function that is called in
# order to form the output dataset.  This function can call any other functions
# that you like, but it must return something that can be interpreted as a 2D
# dataset.  Most of the time, you will want to return a 2D numpy array.  However,
# you can return a list or a 1D numpy array.
#
def evaluate():
    dset = ones((5,5))
    return dset

```

The operation of this dialog is quite straightforward. Enter your miniprogram, and click the *Test Run* button to test. Once the miniprogram has passed the test, click the *OK* button.

If the code executes its test run correctly (this does not imply correctness of the code itself), an information bar will appear at the bottom to indicate this, and it will also report the dimension of the returned dataset. If the code does not compile correctly, a stack trace will appear in the information bar to help you track down the problem.

Note: The OK button will not enable until the program has been test run.

8.2.1 Writing the code

Every miniprogram must have one `evaluate` function. The `evaluate` function is what will be called whenever GraphExpert Professional determines that the data should be updated.

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

As you enter the Python code for your function, it is syntax highlighted, and syntax errors are dynamically detected. The standard cut/copy/paste operators are also available for use as you write your code.

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 creating a miniprogram. Just follow the examples given here and in the miniprogram dialog itself.

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.

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

Again, every miniprogram must have an `evaluate` function defined, and furthermore, the `evaluate` function should take no (mandatory) arguments. Other items that specify metadata (currently, only `name`), are optional.

8.2.2 `name`

This is an attribute to define the name of your miniprogram, which is the name that is used for the component that is created for your miniprogram, by default. The type of this attribute is a string, and should be enclosed in quotes.

8.2.3 `evaluate()`

This is the most important part of your miniprogram; `evaluate()` is a function that is called whenever a dataset update is required from your miniprogram.

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

```python
name = "My miniprogram"
def evaluate():
    dset = ones((10,5)) # create a dataset that contains all ones, 10 rows, 5 columns
    return dset
```

The `evaluate` function returns anything that can be coerced into a 2D dataset. This can be a float, a list of floats, a list of list of floats, or (most commonly) a numpy array.

Alternatively, a dictionary may also be returned from the miniprogram. This method is used when the user desires to return some metadata about the dataset along with the dataset itself. If a dictionary is returned, it is required to have the `dset` key, which should have the dataset to return as its value. Metadata currently recognized by GraphExpert Professional is the `colnames` key, which maps to a list of column names, and the `rownames` key, which maps to a list of row names. An example of this is below:

```python
name = "My miniprogram"
def evaluate():
    dset = ones((10,3))
    colnames = ["Distance","Velocity","Acceleration"]
    retval = {'dset':dset,'colnames':colnames}
    return retval
```

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

8.2.4 Reserved Words

All mathematical functions supported by GraphExpert 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.

8.3 Miniprogram examples

8.3.1 A simple function

Certainly, the example below is more appropriately implemented via a Function (**Create->Function**). However, this example is useful to show how simple a miniprogram can be, in order to get you starting using them. This example just computes the sin function between 0 and 2*pi, and returns the dataset:

```
name = "sin function"

def evaluate():
   n = 101
   x = linspace(0,2*pi,n)
   y = sin(x)
   dset = column_stack((x,y))
   return dset
```

The "linspace" call creates a vector from 0 to 2*pi, made up of n points; then we take the sine of every entry in that array, returning the result in y. At this point, we have two separate 1D arrays, of dimension 1x101. We would like to return the x data in the first column of a dataset, and the y data as the second column. So, we first "column stack" the two arrays to get a 101x2 2D dataset, Now, just return that dataset, we are are done.

8.3.2 Blasius solution

The following miniprogram calculates the Blasius solution (describes the fluid boundary layer that forms near a flat plate under 2D, laminar, and steady conditions and a uniform oncoming flow) and returns it as a dataset:

```
import scipy
import scipy.integrate

name = "Blasius Solution"

def blasius(y, eta):
    return [y[1], y[2], -0.5*y[0]*y[2]]

def evaluate():
   n = 201
   initial_condition = [0,0,0.33206]
   eta = linspace(0, 10, n)
   y = scipy.integrate.odeint(blasius, initial_condition, eta)
   eta = eta.reshape((n,1))
   dset = hstack((eta,y))

   colnames = ["eta","y","yp","ypp"]

   return {'dset':dset,'colnames':colnames}
```

In this example, you can see that the evaluate() function uses the ordinary differential equation solver available in scipy. All of the functionality in scipy is available, which includes numerical integration, optimization, and many other very useful algorithms.

8.3.3 Ellipse

The following miniprogram calculates the points for an ellipse:

```python
name = "Ellipse"

def evaluate():
    a = 2.0 # major axis
    b = 1.0 # minor axis
    x = linspace(-a,+a,201) # create 201 points between -a and +a
    y1 = b*sqrt(1-x**2/a**2) # evaluate the ellipse equation
    dset = vstack((x,y1,-y1)).T # stack the datasets vertically, take transpose

    # the output dataset is 200x3.
    return dset
```

8.3.4 Moody diagram

The Moody diagram example (see the examples shipped with GraphExpert Professional) demonstrates using a miniprogram in order to solve the Colebrook equation in order to find a family of constant roughness curves. This particular example uses the scipy optimizer in order to do so. Please see the example for further details; you can see each miniprogram by right-clicking on the appropriate component, and choosing **Edit Miniprogram**.

CHAPTER

NINE

GRAPHING

9.1 Introduction

Graphs are also themeable in order to save the user time in the production of a large number of similar plots.

In GraphExpert Professional, 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. XY plots, polar plots, scatterplots, 3D 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 the ability to move backward to previous views of the graph.

9.2 Graph Types

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

9.2.1 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.

9.2.2 Polar Plots

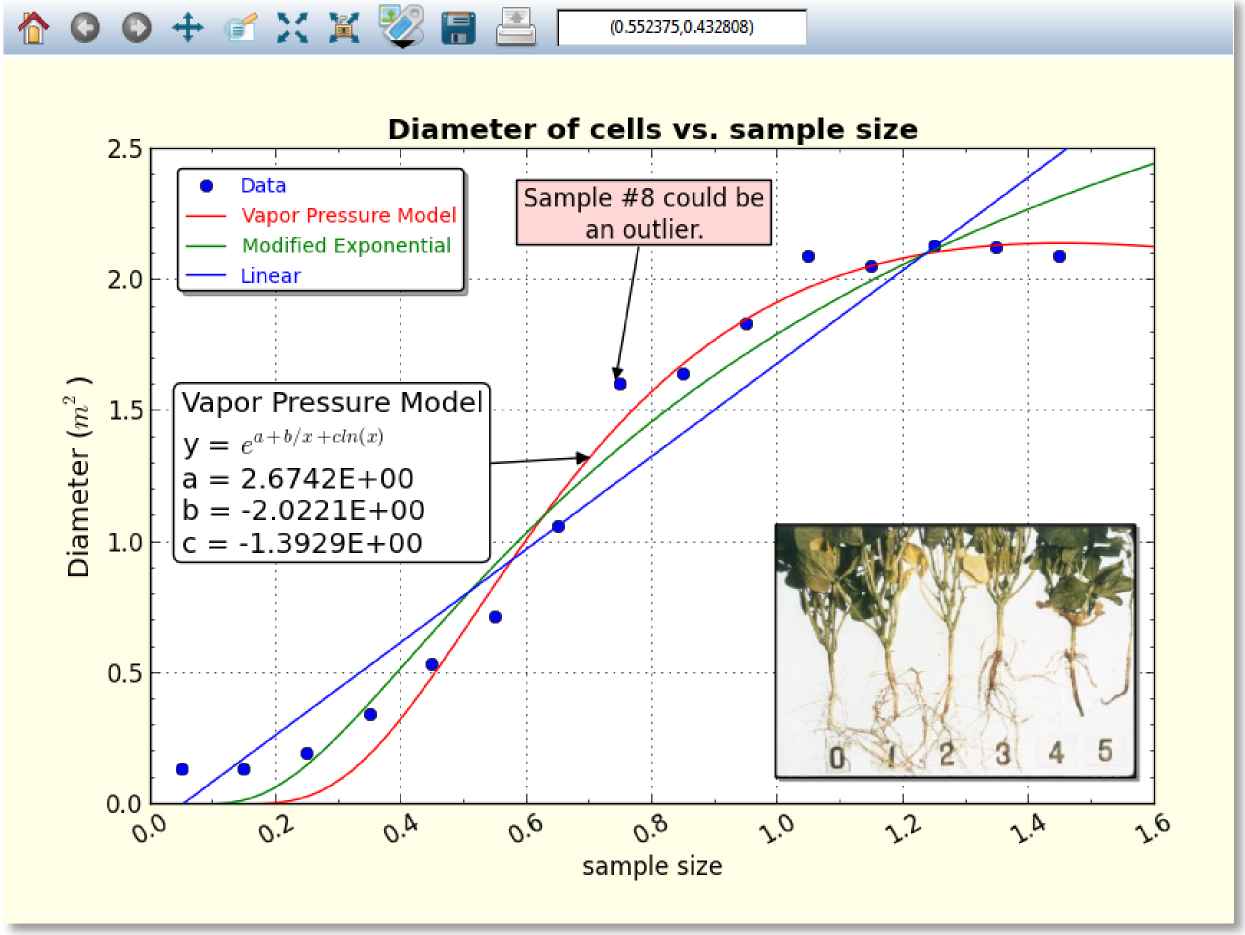
Polar plots have one radial axis and one angular axis. Polar plots always treat the angular axis in radians, although the numbers shown on the angular axis can easily be formatted as degrees. In other words, a dataset that ranges between $[0, 2\pi]$ will completely wrap around a polar plot.

9.2.3 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).

9.3 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:



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*.

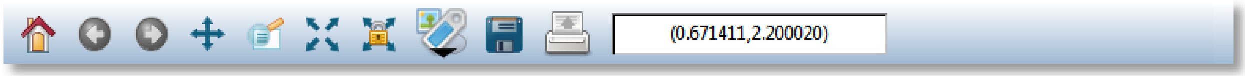
9.4 Interacting with a graph

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.

9.4.1 Graph Toolbar

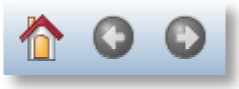
The first style of interaction with a graph is to use the toolbar. A sample of a toolbar is shown below:



(0.671411,2.200020)

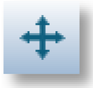
All buttons have tooltip help, so if you can’t remember what a button does, either watch the status bar at the bottom of the main window as you point to the buttons, or hover your mouse over a button to see its function.

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

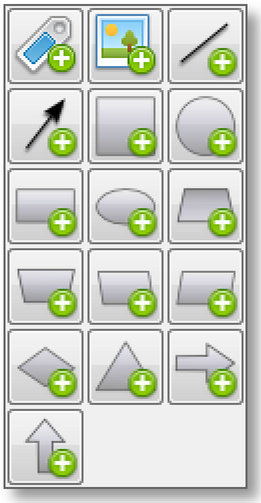
The autoscale button chooses new limits and new tick intervals for your graph that fit the current dataset. See also *Autoscaling*.

Autoscale lock

GraphExpert 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 GraphExpert 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.

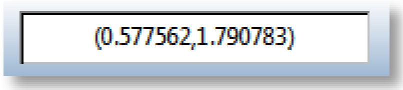
Save

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.

9.4.2 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.

9.4.3 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.

9.4.4 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 *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 GraphExpert 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.

9.4.5 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.

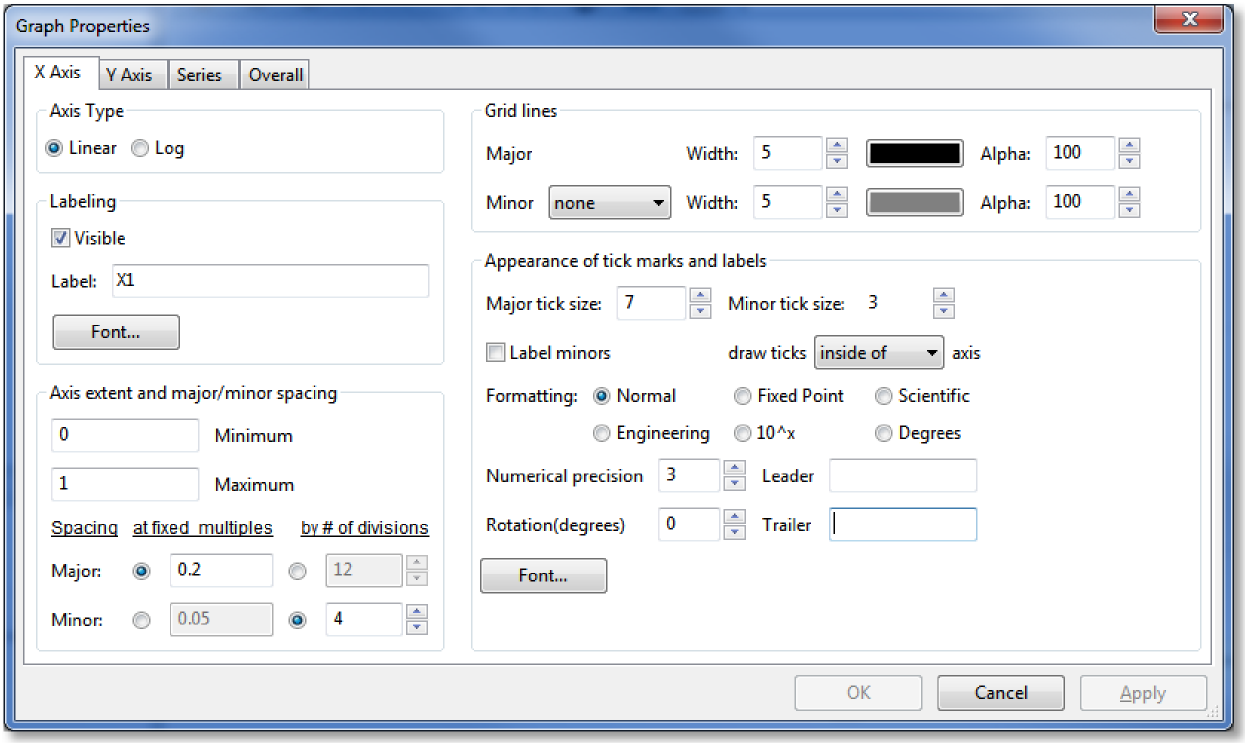
If text is on the clipboard, you can press Ctrl+V (Cmd+V on a Mac) on a graph, or right-click and select *Paste*. The text on the clipboard automatically becomes an annotation. You can also drag and drop text from another source for the same effect.

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.

9.5 Changing Graph Properties

The graph properties window is divided by functionality into pages.

9.5.1 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).

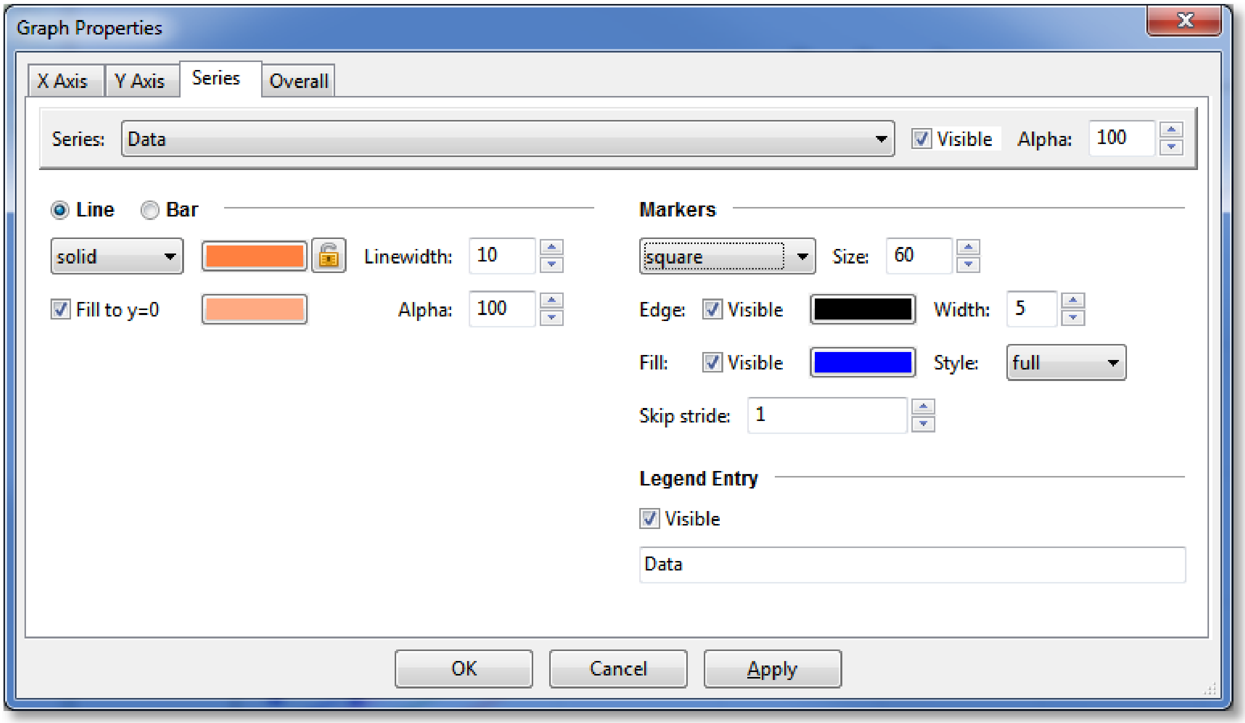
The format of the tick mark labels is determined by the combination of the **Formatting** setting and the **Numerical precision** setting. **Formatting** can be set to normal for a generic style of numerical formatting, fixed point, scientific, engineering, 10^x, or degrees. In all cases except for normal 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.

9.5.2 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.

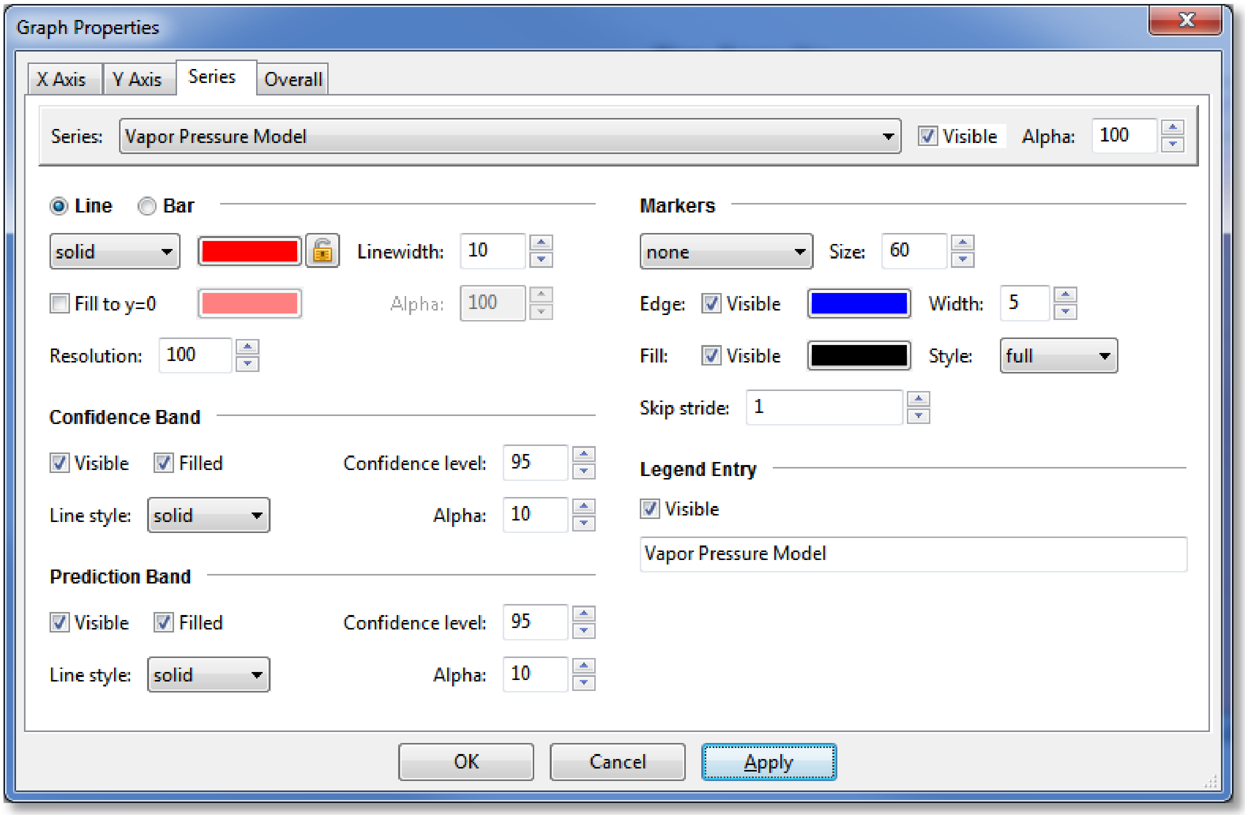
The lock button adjacent to the line color selector determines whether or not the marker edge color and fill color (see below) remain the same as the line color. If left unlocked, the marker edge color and fill color may be adjusted independently from line color. If locked, the marker edge color and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



Graph Properties

X Axis | Y Axis | Series | Overall

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

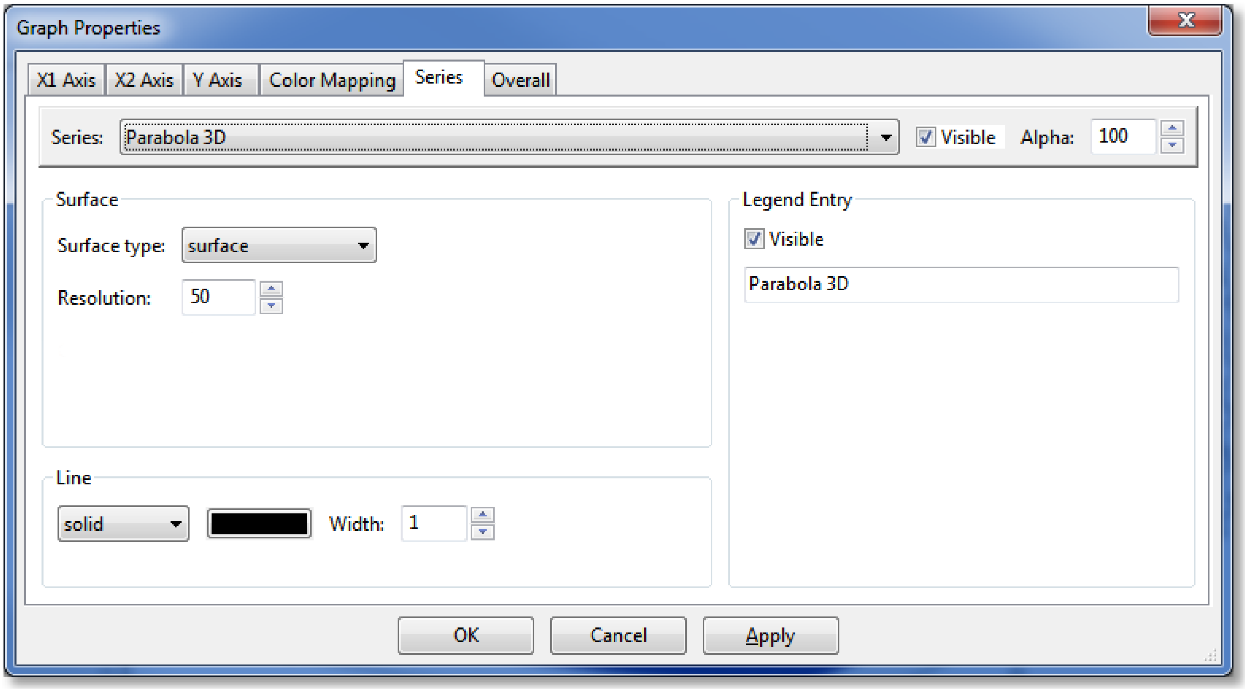
Legend Entry

☑ 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

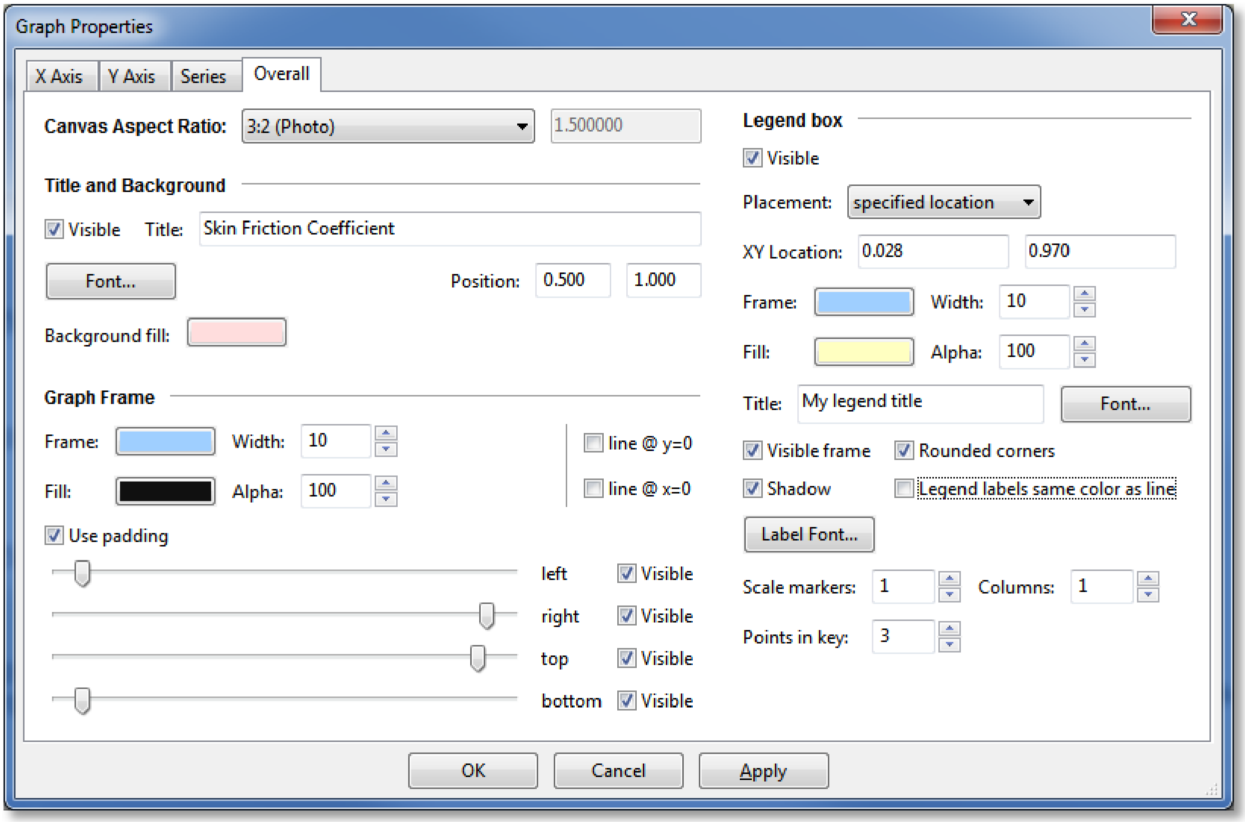
☑ Visible

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.

9.5.3 Overall Page



Canvas Aspect Ratio: 3:2 (Photo) 1.500000

Title and Background

☑ Visible Title: Skin Friction Coefficient

Font... Position: 0.500 1.000

Background fill: 🟥

Graph Frame

Frame: 🟦 Width: 10 □ line @ y=0

Fill: ⬛ Alpha: 100 □ 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 Font...

☑ Visible frame ☑ Rounded corners

☑ Shadow □ Legend labels same color as line

Label Font...

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.

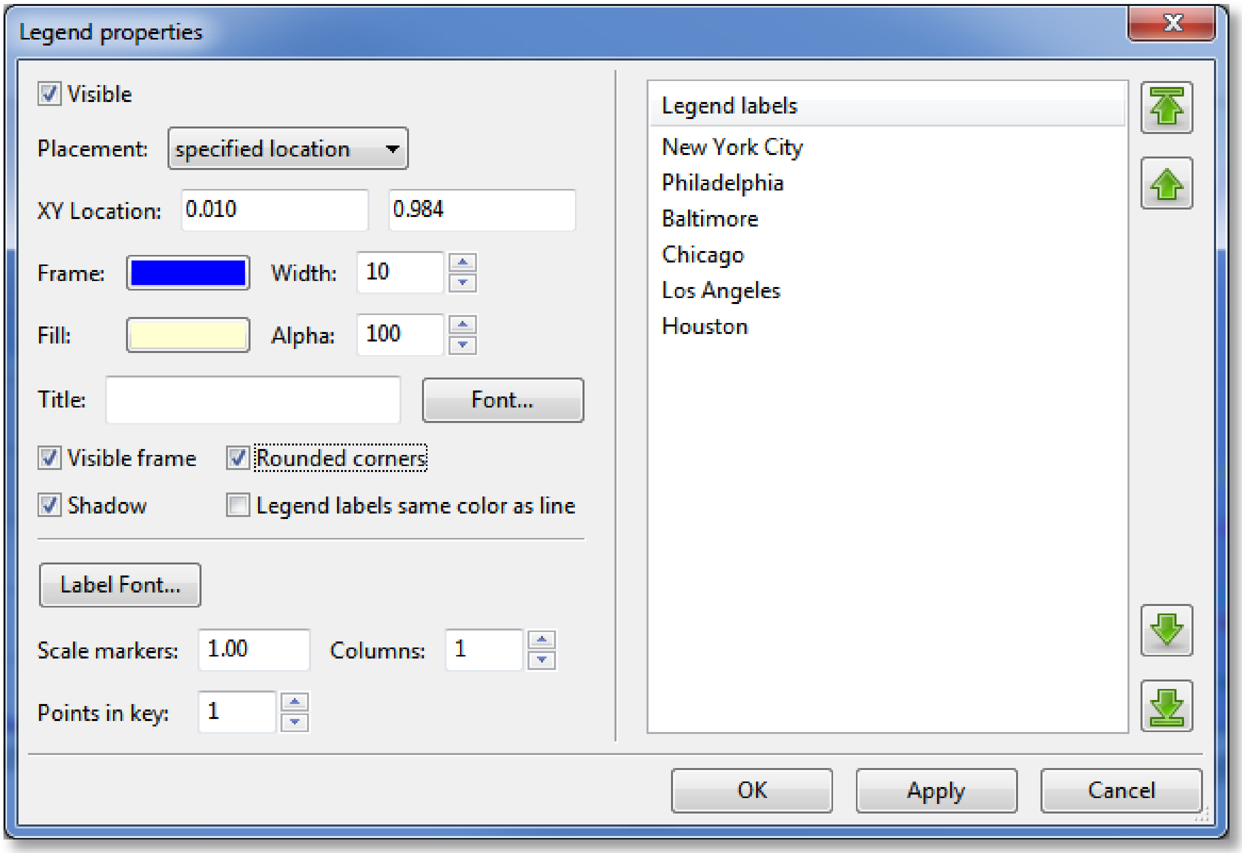
The **Legend Box** section is located in the right side of the *Overall* page. See *Legend Properties* for details.

9.6 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.

9.6.1 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.

9.6.2 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**.

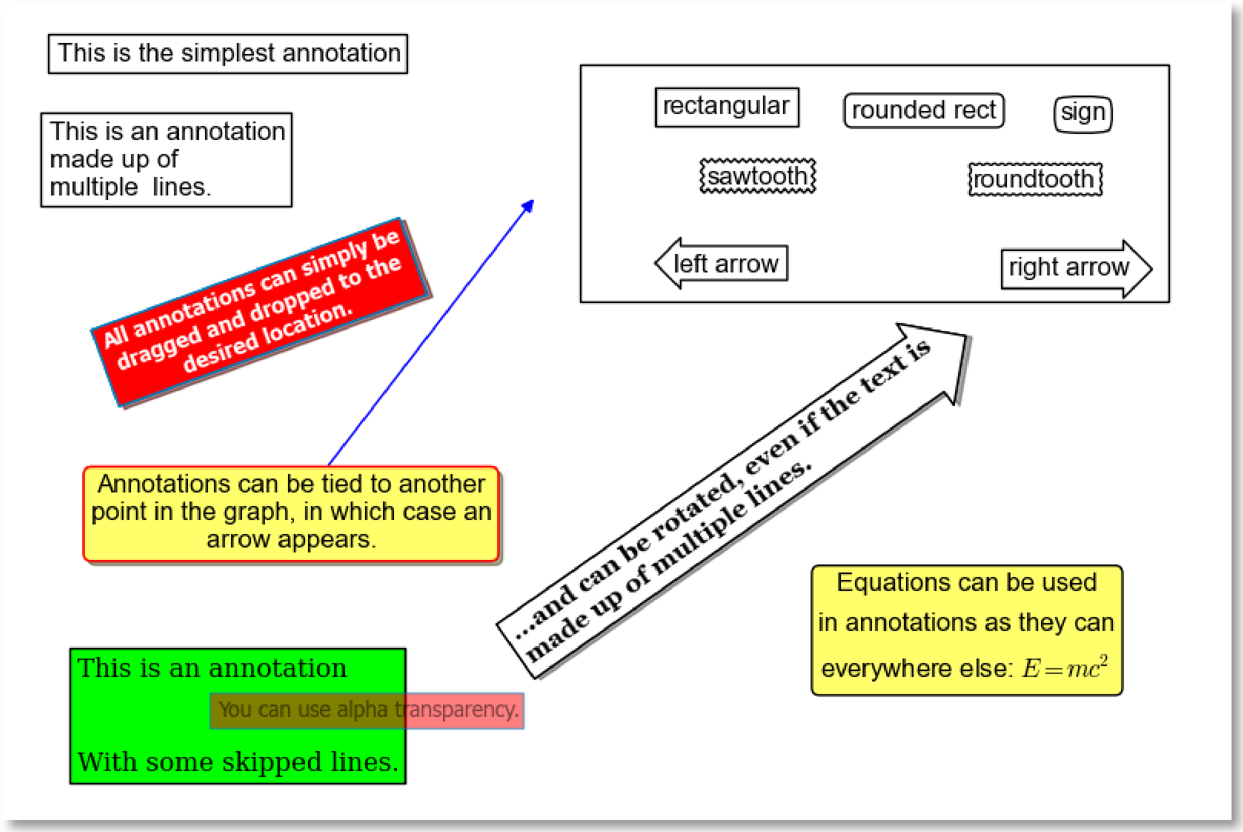
9.6.3 Legend Ordering

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.

9.7 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.
Annotations can be tied to another point in the graph, in which case an arrow appears.
This is an annotation
You can use alpha transparency.
With some skipped lines.
rectangular
rounded rect
sign
sawtooth
roundtooth
left arrow
right arrow
...and can be rotated, even if the text is made up of multiple lines.
Equations can be used in annotations as they can everywhere else: $E = mc^2$

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.

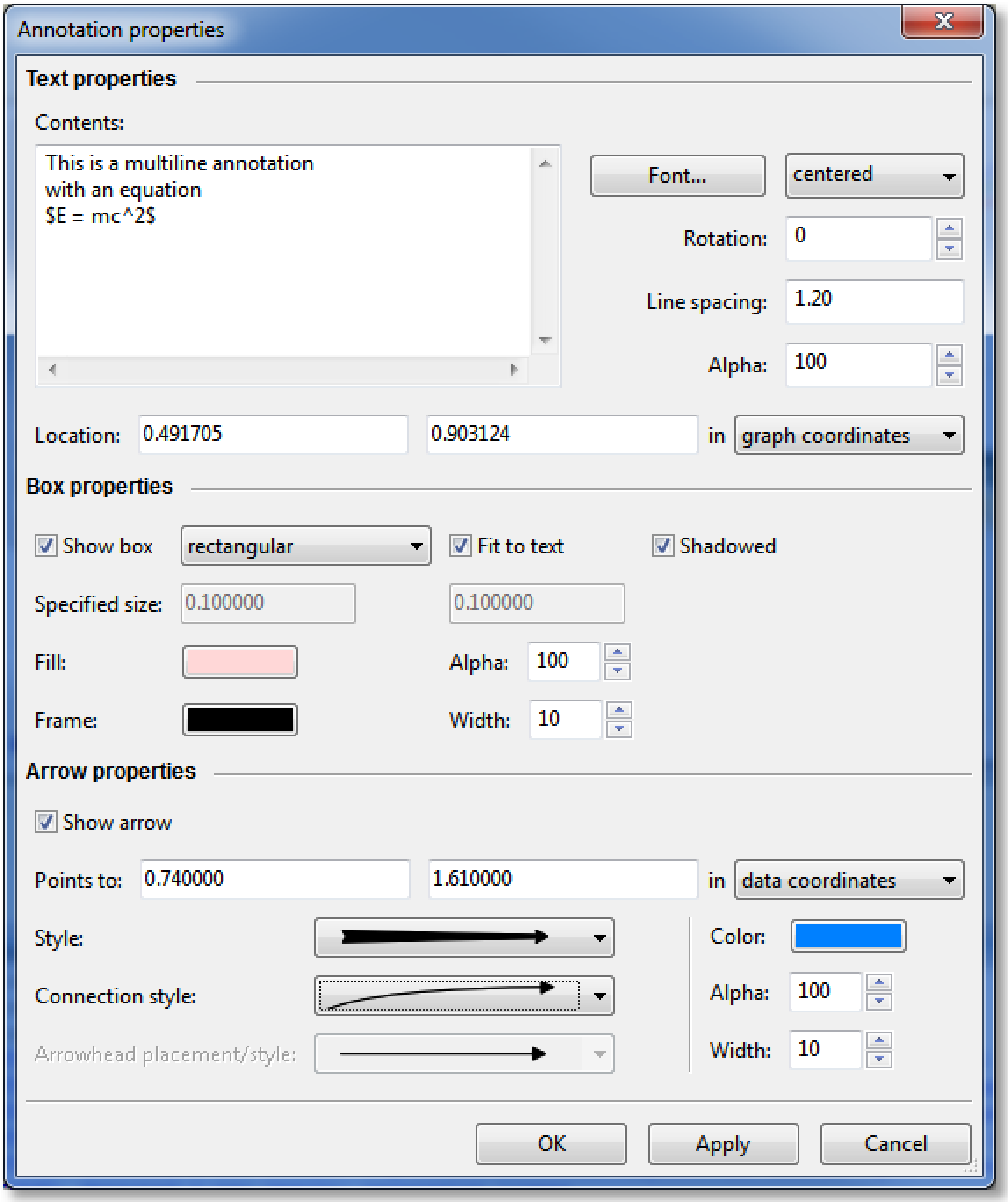
9.7.1 Cutting, Copying, and Pasting Annotations

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 GraphExpert 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 GraphExpert 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.

9.7.2 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
$E = mc^2$
```

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.

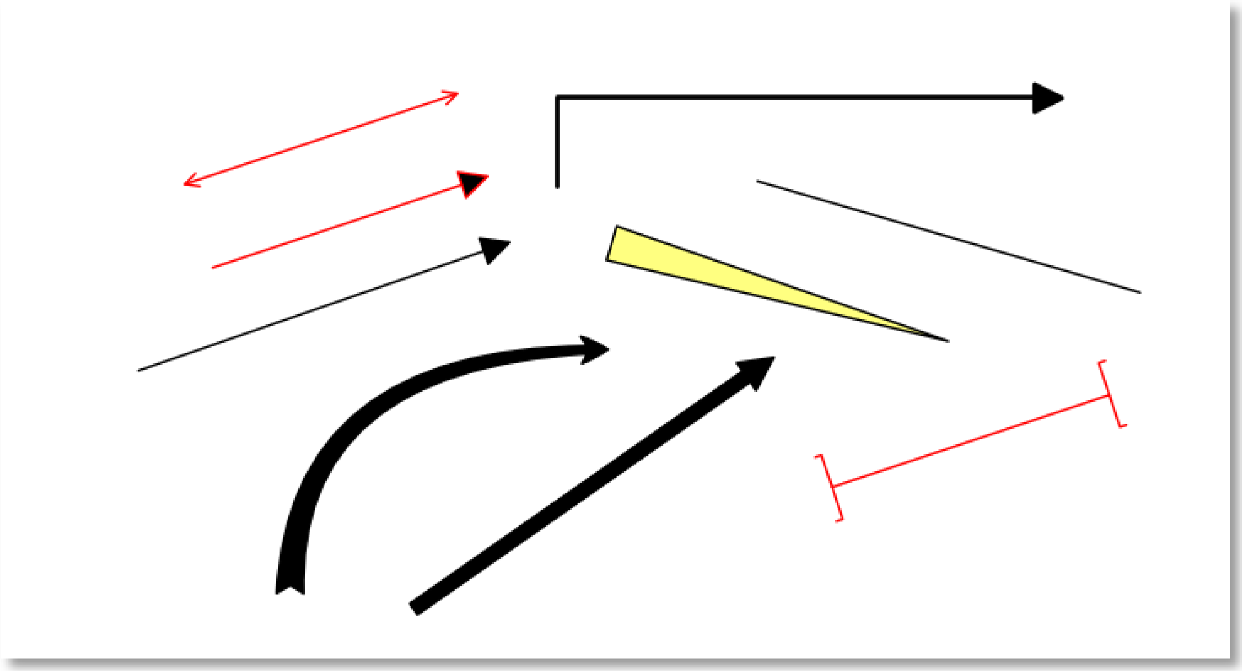
9.8 Automatic annotations

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**.

9.9 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.

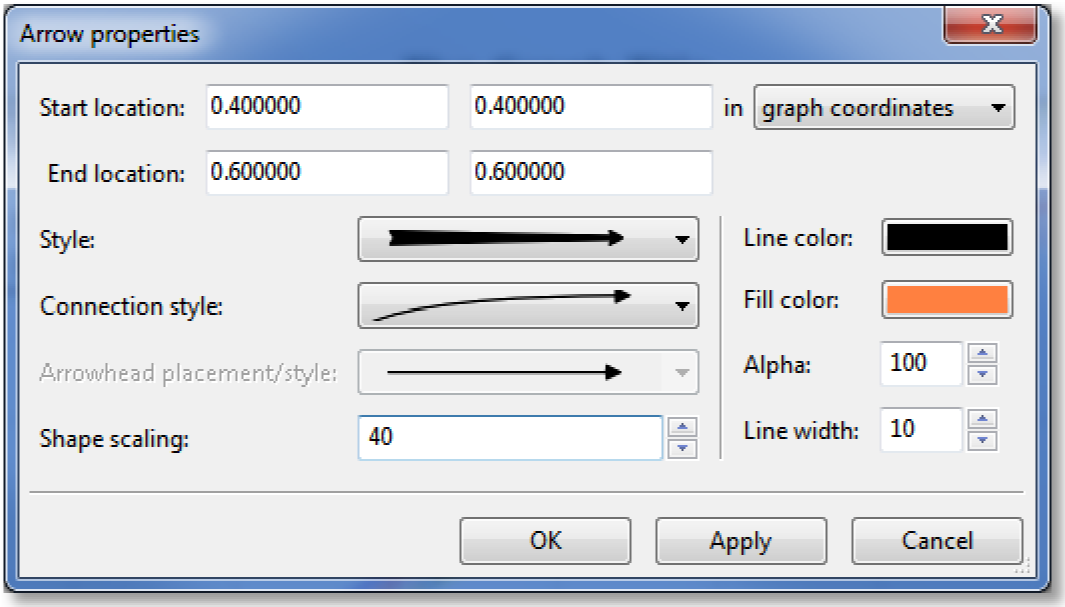
To edit an arrow, right click it and select *Arrow Properties*. To remove an arrow, right click it and select *Remove* or press the DEL key after selecting the desired arrow.

9.9.1 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 GraphExpert Professional.

9.9.2 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.

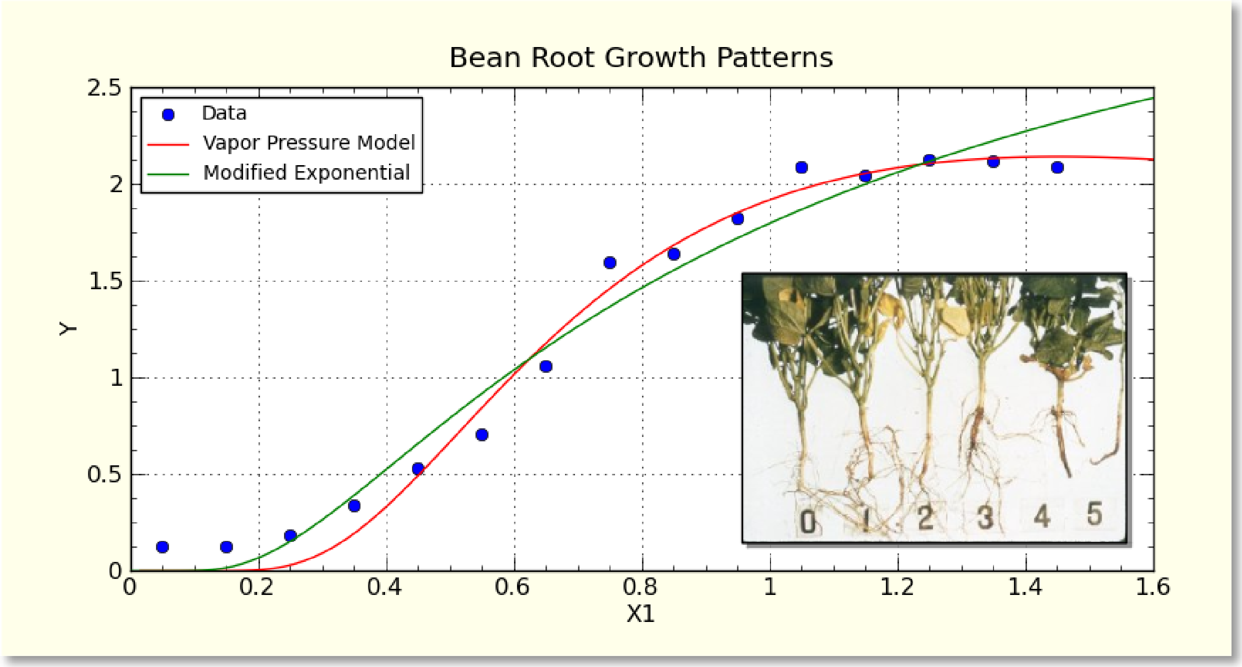
Note that arrowhead placement/styles are only valid for “plain” arrow styles. Also, the XY connection style is only valid for the “plain” arrow style. Finally, the connection style for the “thick” arrow style is always straight. The graphical interface will prevent you from selecting invalid combinations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

9.10 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.

9.10.1 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 GraphExpert Professional.

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

9.10.2 Image properties

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

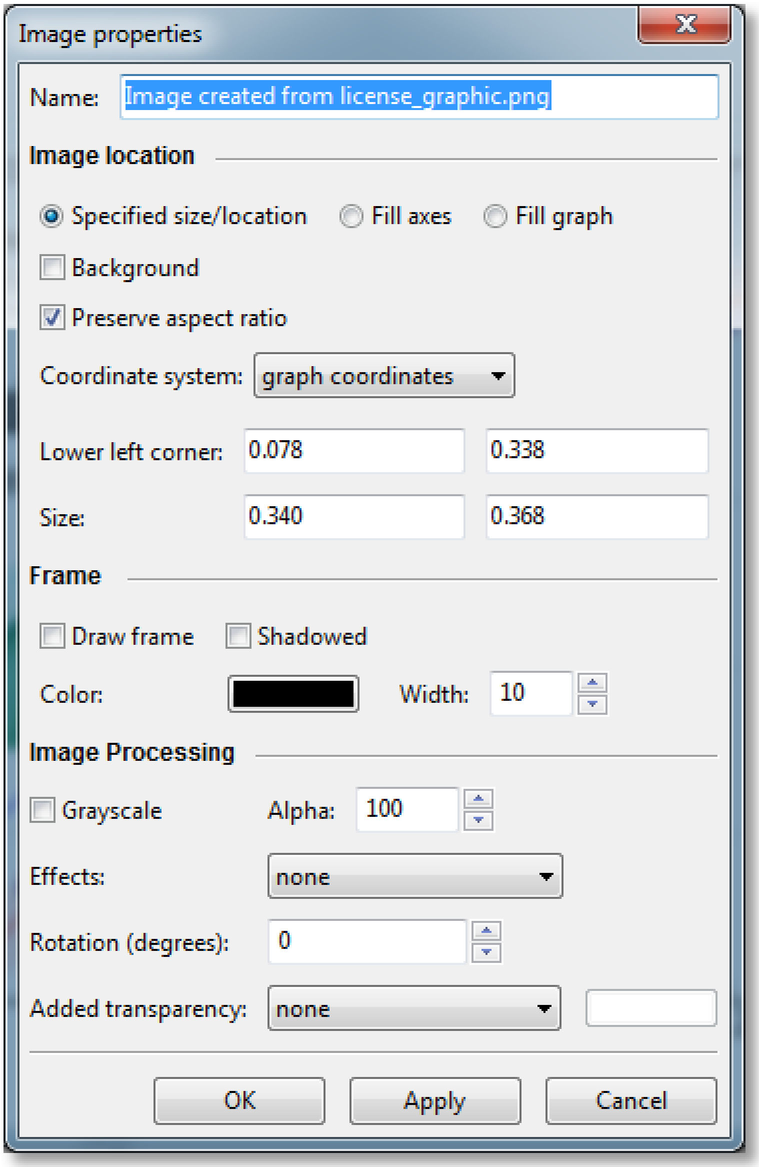


Image properties

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 GraphExpert 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

GraphExpert 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).

9.11 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 GraphExpert Professional retains all object information, so you can add or remove elements from the graph after-the-fact with an illustration package, if desired.

9.12 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.

9.13 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.

9.14 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.

9.14.1 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 GraphExpert 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*.

9.14.2 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.

9.14.3 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.

9.15 2D Contour Graphing

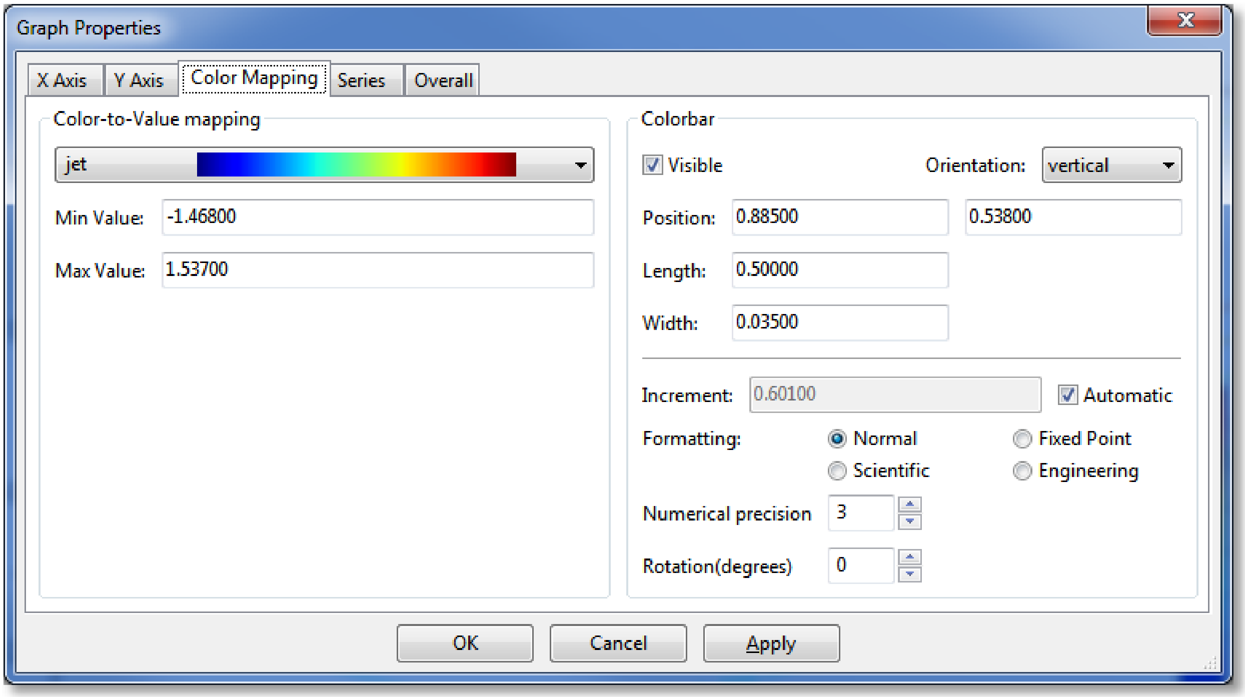
To create a contour plot, add a 3D dataset or function (one with two independent variables) on an XY plot.

9.15.1 Interactive Viewing

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

9.15.2 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:


Graph Properties

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

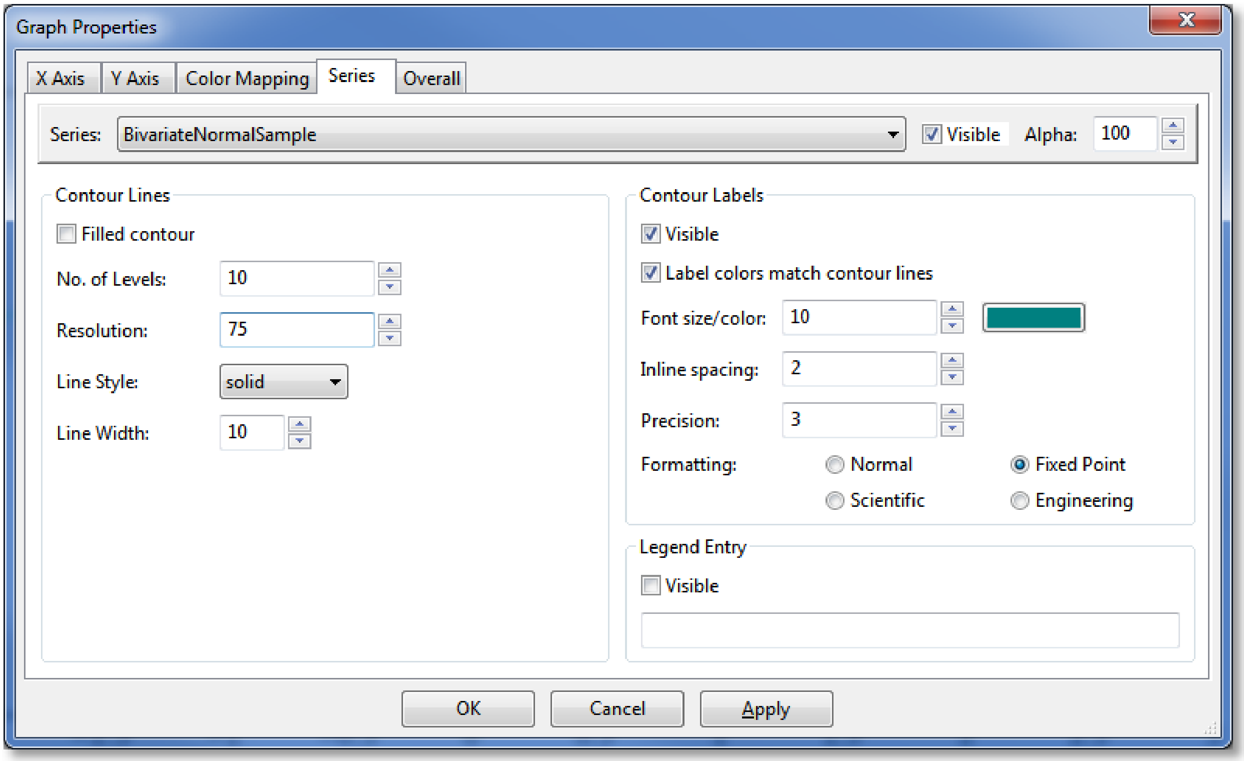
Numerical precision 3

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).

9.16 3D Graphing

A 3D graph can be created by the user by selecting *Create->3D plot* from the GraphExpert Professional main menu, or by clicking the appropriate toolbar button. One can also create a new 3D plot by clicking the "+" tab in the graph notebook.

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.

9.16.1 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.

9.16.2 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.

9.17 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 Details Window 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

9.18 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.

9.18.1 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:

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

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

9.18.2 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:

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

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

9.18.3 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:

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

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

9.18.4 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}`	$\mathcal{CALLIGRAPHY}$

9.18.5 Global Fonts

In GraphExpert 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.

9.18.6 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"$\hat i\ \ \hat \imath$"
```

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

9.18.7 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	$\mho$ \mho
∇ \nabla					

Hebrew

$\aleph$ \aleph	$\beth$ \beth	$\daleth$ \daleth	$\gimel$ \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

$\bigcap$ \bigcap	$\bigcup$ \bigcup	$\bigodot$ \bigodot	$\bigoplus$ \bigoplus	$\bigotimes$ \bigotimes
$\biguplus$ \biguplus	$\bigvee$ \bigvee	$\bigwedge$ \bigwedge	$\coprod$ \coprod	$\int$ \int
$\oint$ \oint	$\prod$ \prod	$\sum$ \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

TEN

DIGITIZING

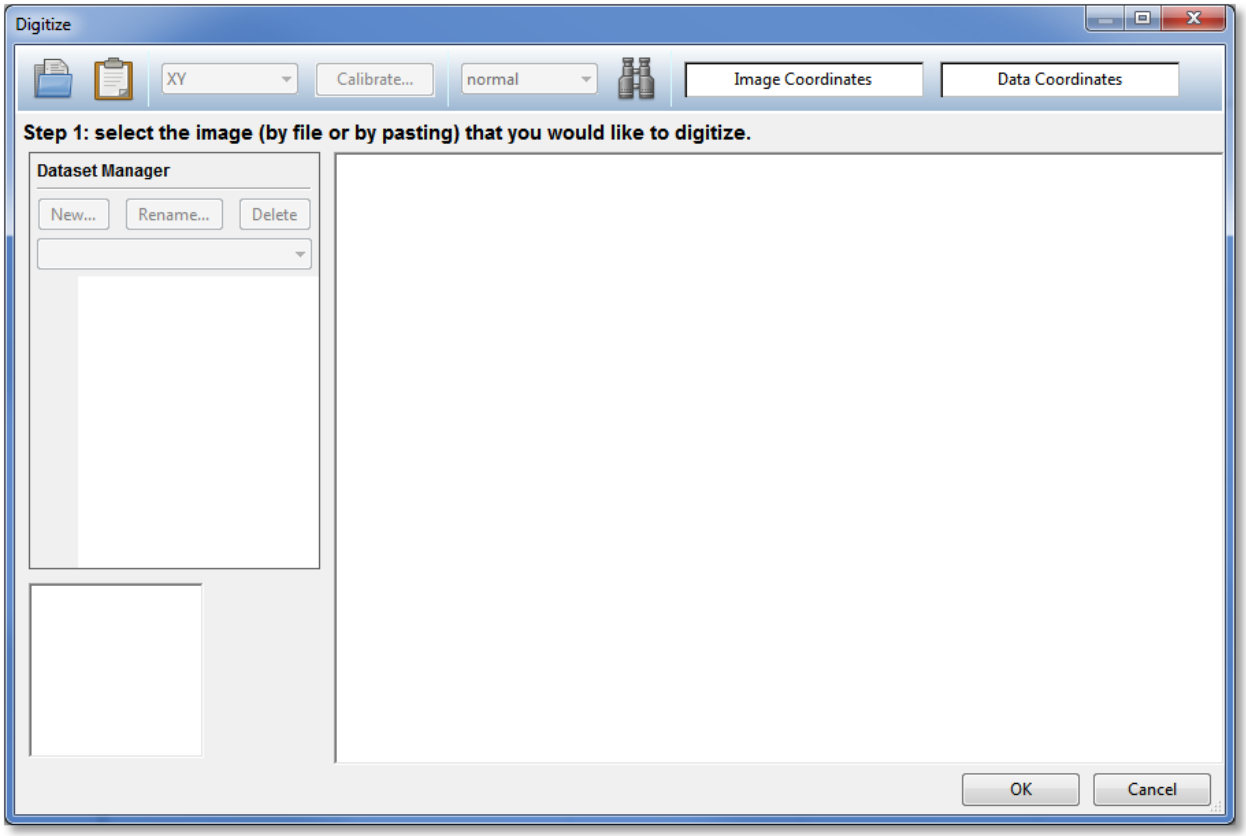
10.1 Introduction

In some cases, the data to be analyzed is in the form of a picture or image. In GraphExpert 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 *Create->Digitizer* 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:



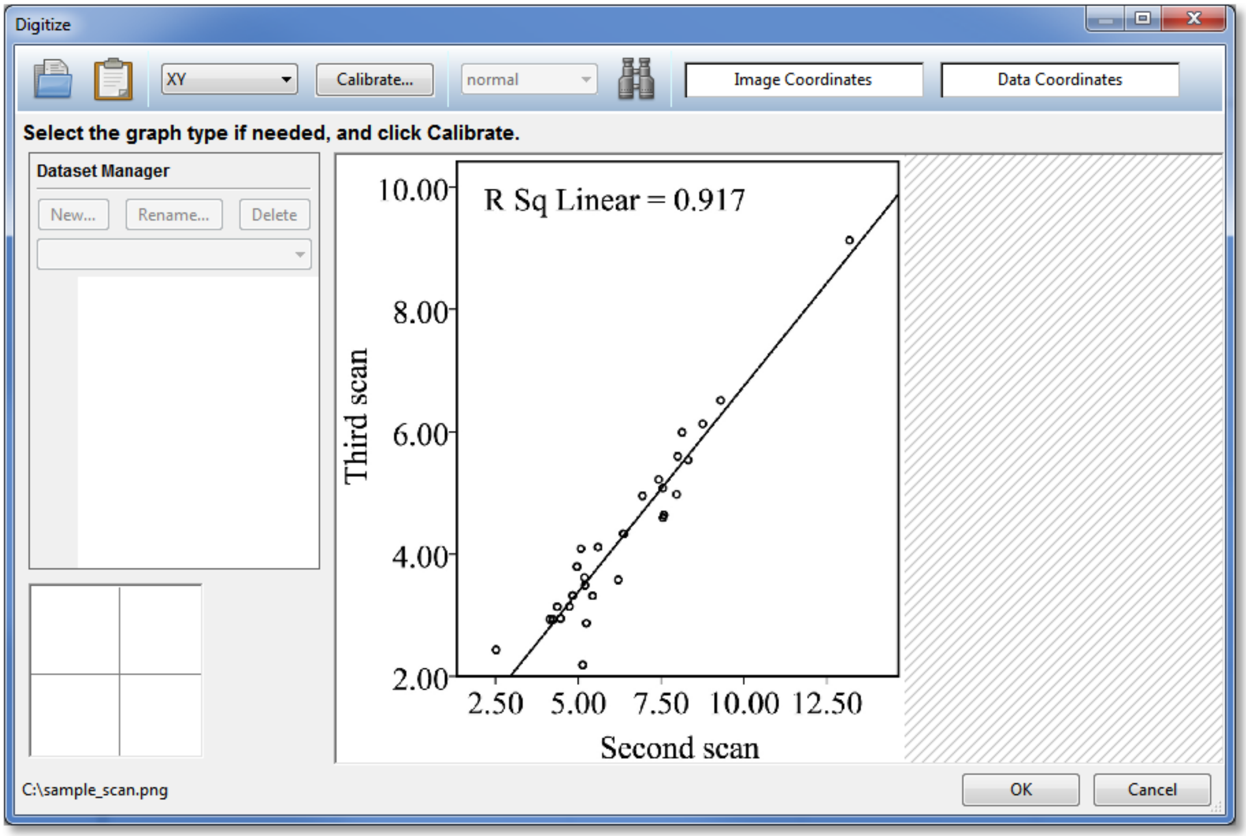
Step 1: select the image (by file or by pasting) that you would like to digitize.

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.

10.2 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.png

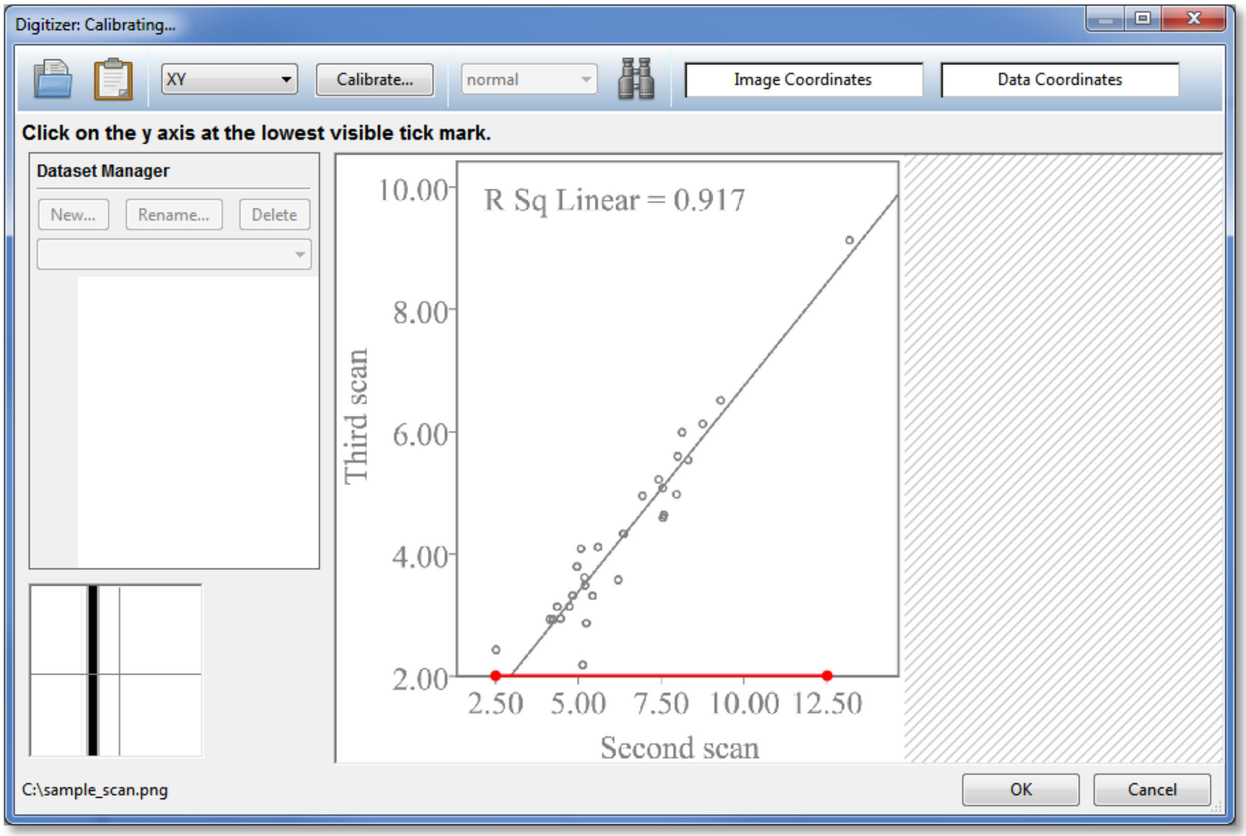
Note that once the image is imported into the digitizer, the file that it was imported from (if it was not pasted) does not need to remain on disk. The image data will be saved as part of the digitizer, and is also persistent in the GraphExpert Professional document.

10.3 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, GraphExpert 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.

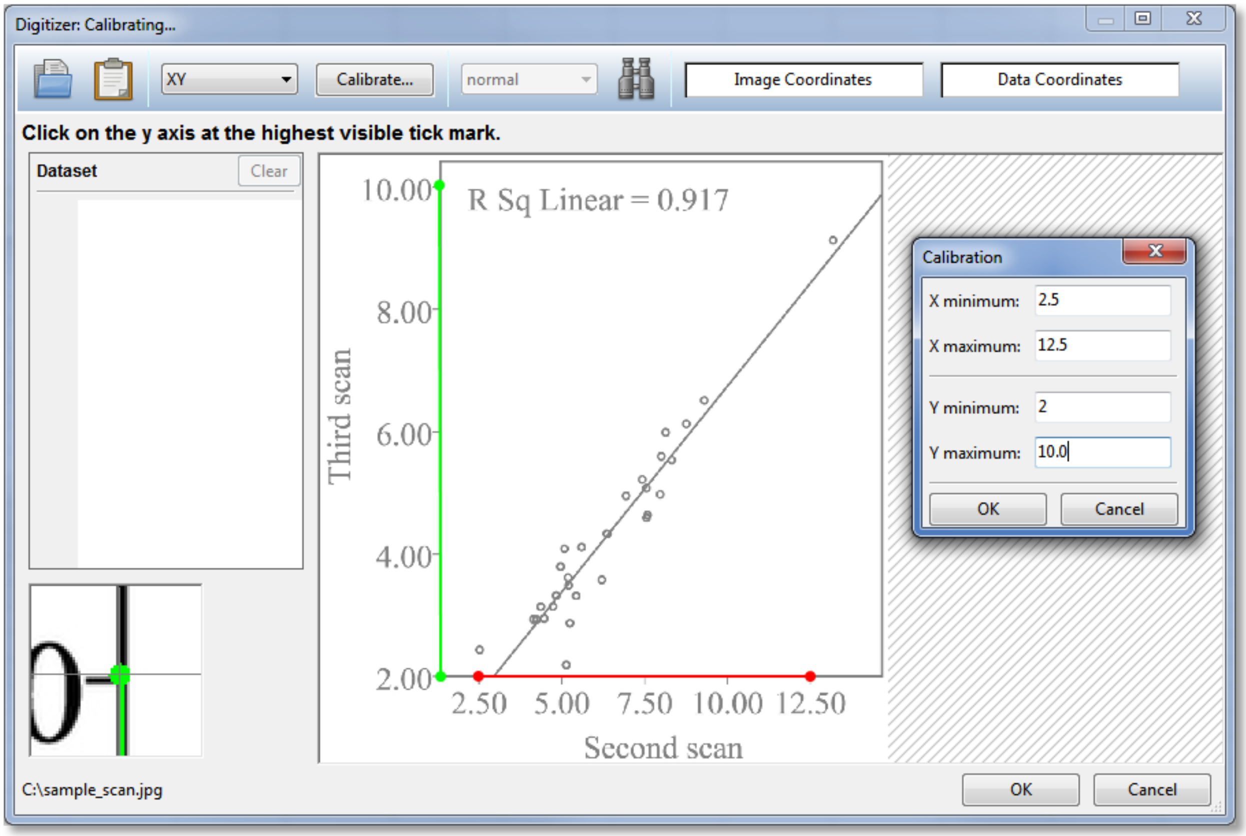


Click on the y axis at the lowest visible tick mark.

C:\sample_scan.png

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.

X minimum: 2.5

X maximum: 12.5

Y minimum: 2

Y maximum: 10.0

C:\sample_scan.jpg

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.

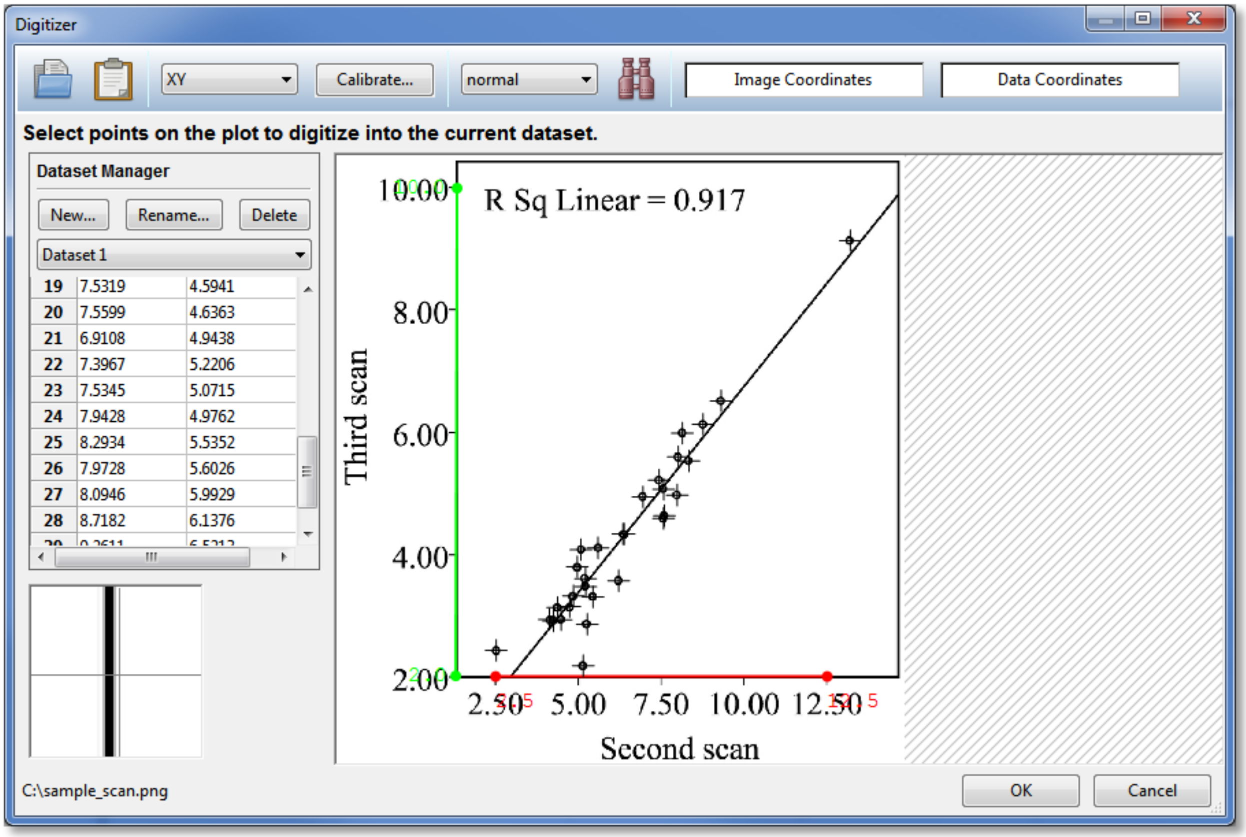
As you move the mouse over the image, you will notice that the *Image Coordinates* and *Data Coordinates* information bars will begin to show information concerning the current mouse location. The image coordinate is the pixel location within the image. The data coordinate is the transformed data value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

10.3.1 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.

10.4 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 *Dataset Manager* 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.

19	7.5319	4.5941
20	7.5599	4.6363
21	6.9108	4.9438
22	7.3967	5.2206
23	7.5345	5.0715
24	7.9428	4.9762
25	8.2934	5.5352
26	7.9728	5.6026
27	8.0946	5.9929
28	8.7182	6.1376

C:\sample_scan.png

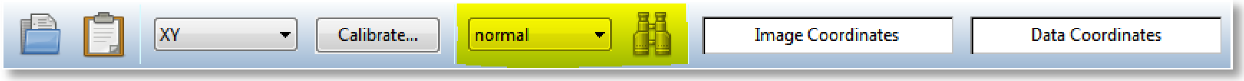
As the mouse passes over a marker, both the marker and its corresponding data value in the dataset are highlighted in magenta.

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.

10.4.1 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.

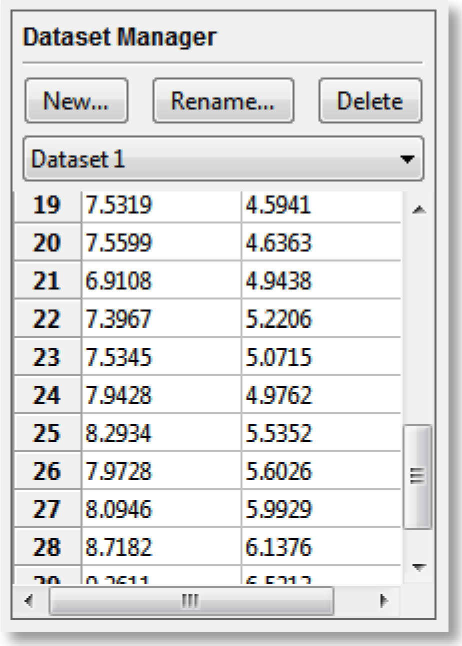
During the point autodetection, the toolbar will turn red.

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.

10.5 Dataset management

Multiple datasets can be digitized from one image, and the Dataset Manager area facilitates the creation, naming, selection, and deletion of datasets. All datasets created in the digitization will be child datasets that you can manipulate in GraphExpert Professional just like any other dataset in the Components pane. The dataset management area is shown below:



19	7.5319	4.5941
20	7.5599	4.6363
21	6.9108	4.9438
22	7.3967	5.2206
23	7.5345	5.0715
24	7.9428	4.9762
25	8.2934	5.5352
26	7.9728	5.6026
27	8.0946	5.9929
28	8.7182	6.1376

Any points added via direct clicking on the canvas, or by point autodetection, are always added to the currently selected dataset. The currently selected dataset is indicated by the chooser directly above the spreadsheet (in the screenshot below, the currently selected dataset is "Dataset 1"). In order to make a different dataset current, simply select another dataset in the chooser.

To create a new dataset, click the *New...* button. You will be prompted to enter the name of the new dataset, and a new, empty dataset will be created and automatically selected as current for you.

To rename the currently selected dataset, click the *Rename…* button. You will be prompted to enter the new name of the currently selected dataset. Note that the names of the child datasets in GraphExpert Professional's Components pane, when you are finished with the digitizer, will reflect the names that you have given here in the digitizer.

To delete the currently selected dataset, click the *Delete* button. GraphExpert Professional will confirm the deletion, and switch the currently selected dataset to the one immediately prior in the dataset chooser.

10.6 Step 4: Complete the digitization

Simply press *OK* in order to import the digitization results into the GraphExpert Pro component manager; the digitizer object is the parent, and each digitized dataset generated in the digitizer dialog is a child; note that the datasets are not editable via the GraphExpert Professional spreadsheet, but can be modified by right clicking the digitizer object in the Components pane, and selecting *Edit* . At any time, you can modify the digitization by selecting *Edit* from the digitization component in the Components pane.

10.7 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

ELEVEN

GRAPH THEME MANAGER

11.1 Introduction

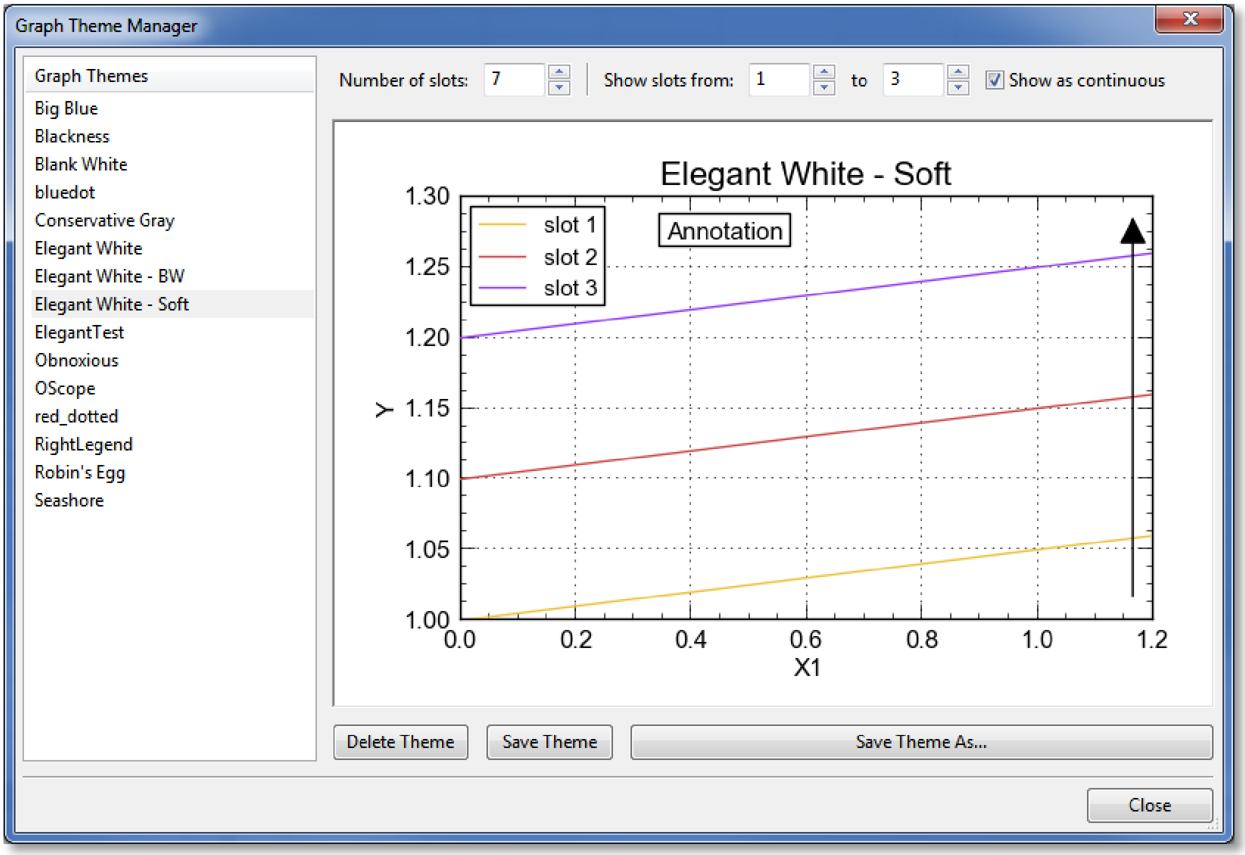
Graph themes allow you to create a custom “look” for your graph, that can be reused on other graphs as appropriate. This can save a lot of time when it comes to setting the properties for your graphs.

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.



Number of slots: 7

Show slots from: 1 to 3

☑ Show as continuous

11.2 Series properties

11.2.1 What are slots?

GraphExpert 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.

11.2.2 Slots and themes

So, the graph theme saves the properties for each slot. The themes that ship with GraphExpert 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. GraphExpert 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. GraphExpert 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.

11.3 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.

11.4 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

TWELVE

SAVING DATA

12.1 Introduction

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

12.2 Writing a GraphExpert Professional (GXP) file

If you want to save all of your components, graphs, and notes to disk, you can choose *File->Save* or *File->Save As* in order to write a GXP file to disk. These files can be read in later using *File->Open*, obviously. GXP files are portable across platforms (Windows, Mac, Linux), so that you can seamlessly read a file on Windows that was created on a Mac, for example. GXP files save all possible information about your current session.

12.3 Exporting a dataset to text file

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

12.4 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.

A graph can also be copied to the clipboard by right clicking and selecting *Copy*.

12.5 Printing

All graphs in the Graphs 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, these can be set in the **File->Page Setup** dialog.

CHAPTER

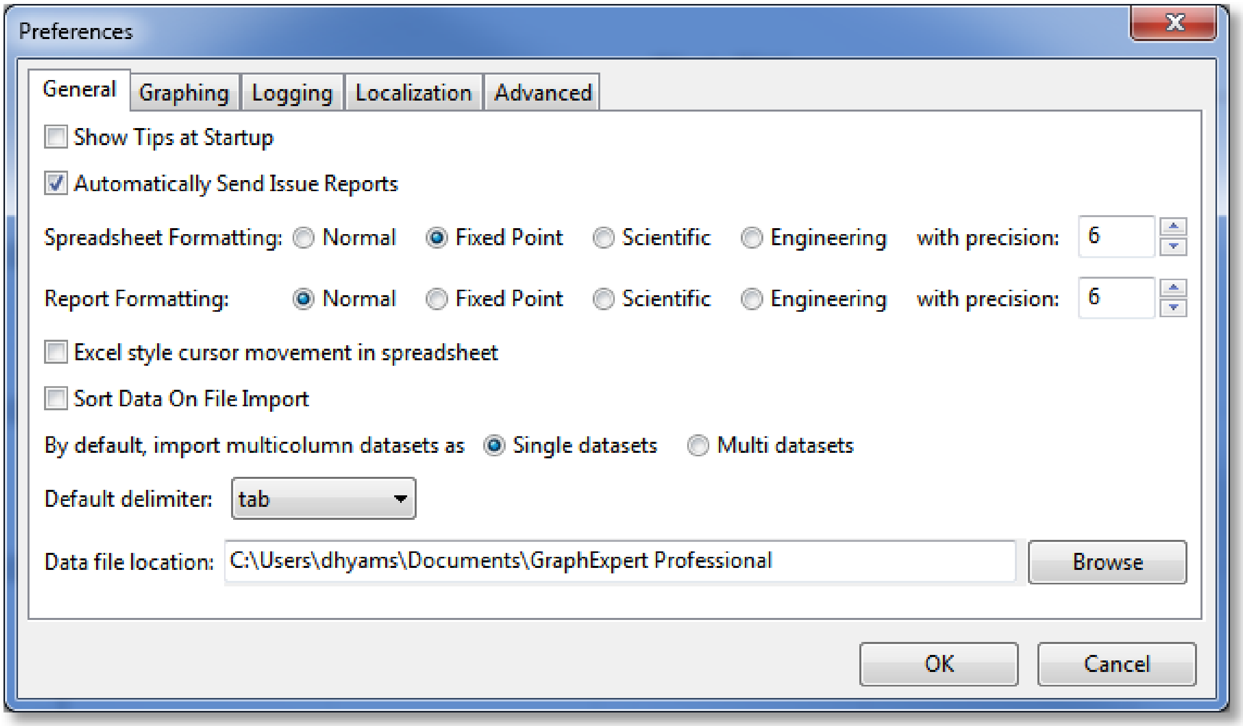
THIRTEEN

APPLICATION PREFERENCES

13.1 Introduction

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

13.2 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

□ Excel style cursor movement in spreadsheet

□ Sort Data On File Import

By default, import multicolumn datasets as ● Single datasets ○ Multi datasets

Default delimiter: tab

Data file location: C:\Users\dhyams\Documents\GraphExpert Professional

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

13.2.1 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.

13.2.2 Automatically send issue reports

GraphExpert 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.

13.2.3 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.

13.2.4 Excel style cursor movement in spreadsheet

By default, both the TAB and ENTER keys are treated as carraige returns in GraphExpert 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 GraphExpert Professional will still keep your cursor within the bounds of the dataset.

13.2.5 Sort Data On File Import

Normally, GraphExpert 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.

13.2.6 Import multicolumn datasets as

When a dataset with more than two columns is imported, it can be, by default, treated as a single dataset (many independent variables, one dependent variable) or a multi dataset (many independent variables, many dependent variables). Note that if treated as a multi dataset, GraphExpert Professional has no choice but to guess that there is one independent variable in the first column, and the rest of the columns are dependent variables. Conversely, if a single dataset is chosen here, GraphExpert Professional guesses that the dependent variable is in the last column, and the rest of the columns comprise the independent variables. Here, just choose the option that matches your workflow in most circumstances.

13.2.7 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.

13.2.8 Data file location

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

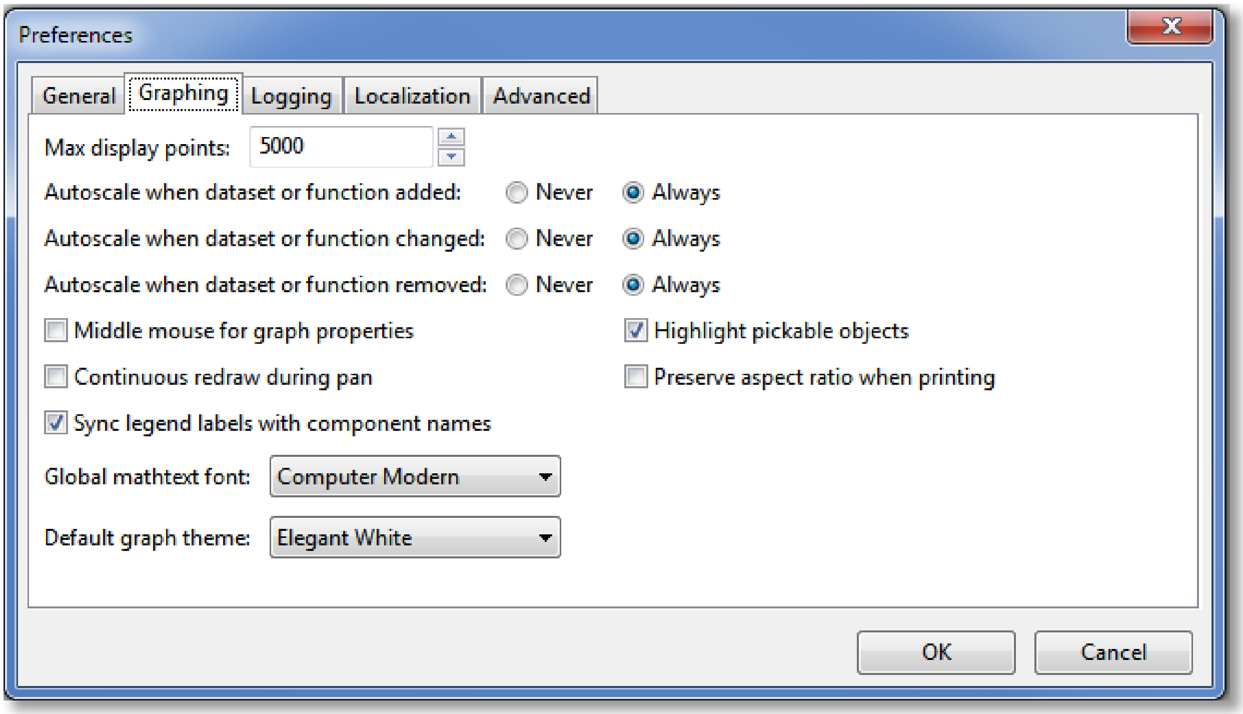
13.2.9 Max number of cores to utilize

GraphExpert 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.

13.3 Graphing Preferences


Preferences

Max display points: 5000

Autoscale when dataset or function added: ○ Never ● Always

Autoscale when dataset or function changed: ○ Never ● Always

Autoscale when dataset or function removed: ○ Never ● Always

□ Middle mouse for graph properties

☑ Highlight pickable objects

□ Continuous redraw during pan

□ Preserve aspect ratio when printing

☑ Sync legend labels with component names

Global mathtext font: Computer Modern

Default graph theme: Elegant White

13.3.1 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.

13.3.2 Autoscale when dataset or function added

When a dataset or function is added to a graph (via a *Send to Plot* command or a drag and drop), GraphExpert 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.

13.3.3 Autoscale when dataset or function changed

In order to keep the data in view at all times in all graphs, GraphExpert Professional can automatically autoscale the graphs when a dataset or function that is plotted on that graph 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”.

13.3.4 Autoscale when dataset or function removed

When a dataset 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”), GraphExpert 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.

13.3.5 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.

13.3.6 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.

13.3.7 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.

13.3.8 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.

13.3.9 Sync legend labels with component names

This option controls how the legend in a plot behaves when its associated component name is changed. For convenience, GraphExpert Professional updates the legend name to match the component name. This way, when you rename a component in the Components pane, all graphs that are displaying that component are automatically updated.

To disable this default behavior, uncheck this box.

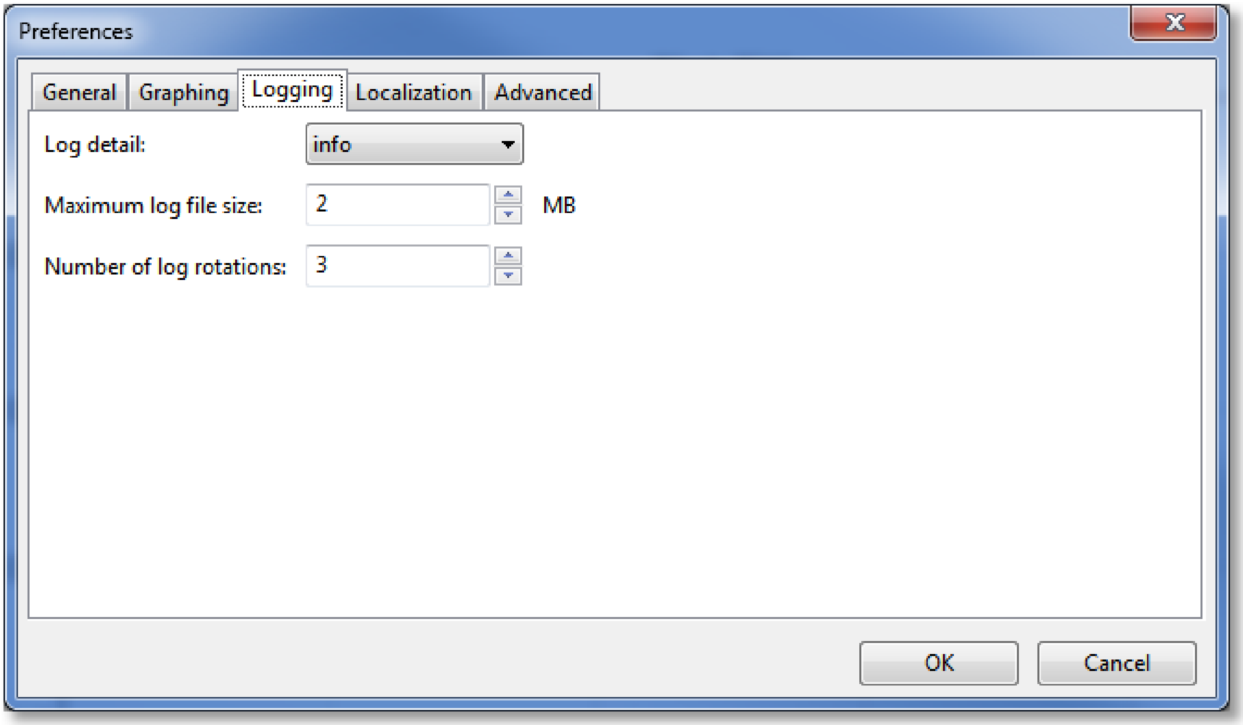
13.3.10 Global mathtext font

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).

13.3.11 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.

13.4 Logging Preferences



Log detail: info

Maximum log file size: 2 MB

Number of log rotations: 3

13.4.1 Log detail

GraphExpert Professional keeps a log in $HOME/.graphexpert/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.

13.4.2 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.

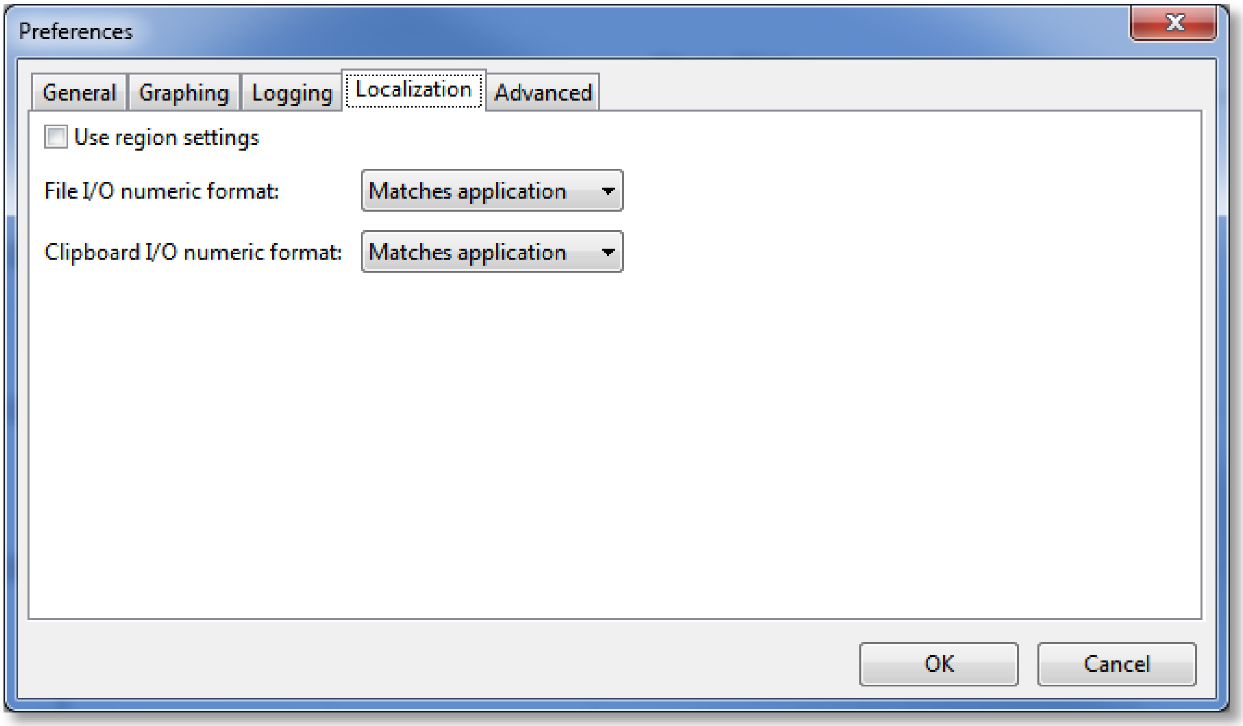
13.4.3 Maximum log file size

Choose the largest file size (in megabytes) that any log file will be allowed to reach before rotation. The maximum size on disk that can be taken by the collection of log files is therefore

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

With the default settings of a 2MB log size and 3 rotations, the maximum amount of disk space consumed by the logs is then 8MB.

13.5 Localization Preferences



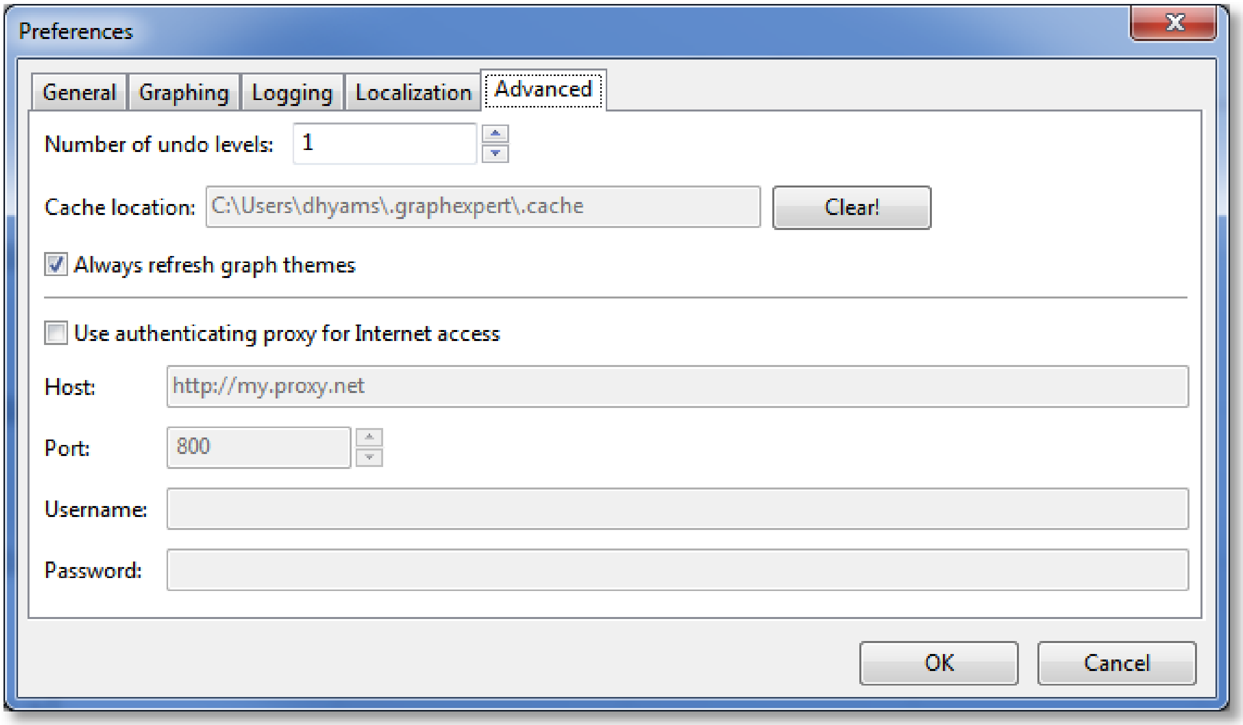
□ Use region settings

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 GraphExpert Professional to match your workflow* section.

13.6 Advanced Preferences



Number of undo levels: 1

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

☑ Always refresh graph themes

☐ Use authenticating proxy for Internet access

Host: http://my.proxy.net

Port: 800

Username:

Password:

13.6.1 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 GraphExpert Professional that support Undo operations are the graphs, data spreadsheets, notes, and messages.

13.6.2 Cache location

The location of a cache that GraphExpert 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.

13.6.3 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/.graphexpert/graph_themes). This happens whenever GraphExpert Professional starts. The purpose of keeping this option selected is that, when GraphExpert 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/.graphexpert/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 GraphExpert Professional is installed; the themes in the $HOME/.graphexpert/graph_themes folder will be left untouched.

13.6.4 Proxy settings

GraphExpert 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

FOURTEEN

LOCALIZATION

14.1 Executive Summary

GraphExpert 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 GraphExpert Professional.

GraphExpert 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.

14.2 Configuring GraphExpert Professional to match your workflow

14.2.1 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 GraphExpert Professional for the change to take effect.

14.2.2 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*, With this setting, for example, users that utilize commas for decimals in GraphExpert Professional can choose to read and write files using dots for decimals. And vice versa.

Note: In the file import dialog (see *Importing from file*), 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.

14.2.3 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 GraphExpert Professional can choose to copy and paste data to/from the clipboard using dots for decimals. And vice versa.

14.2.4 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 *Importing from file*.

14.3 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 GraphExpert Professional file importing mechanism that the the file uses English or European numerical formatting (as appropriate), and the default there (see *Importing from file*) 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 GraphExpert 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.

14.4 GraphExpert Professional files

GraphExpert Professional files (.gxp) are not affected by region settings; therefore, they are portable across localities as well as being portable across platforms (Windows, Mac, Linux).

CHAPTER FIFTEEN

MISCELLANEOUS

15.1 Fonts

Several fonts ship with GraphExpert Professional for portability reasons. You are, of course, welcome to use any font installed on your system on a graph; however, if you plan to share GraphExpert Professional files with others, care should be taken concerning the fonts used. If you use a font that is not installed on your colleague's machine, it will not display properly; it will default to Bitstream Vera Sans.

The following list of fonts are guaranteed to be present on every platform and on every system:

- Arial (the default font used by GraphExpert Professional)
- Bitstream Vera Sans
- Arimo
- Lato
- Playfire Display
- Vollkorn
- Old Standard TT

All of the above fonts are high quality, scalable, Truetype fonts.

Please see the font demonstrator (font_demo.gxp) included with GraphExpert Professional in to see samples of each of these fonts.

15.2 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).

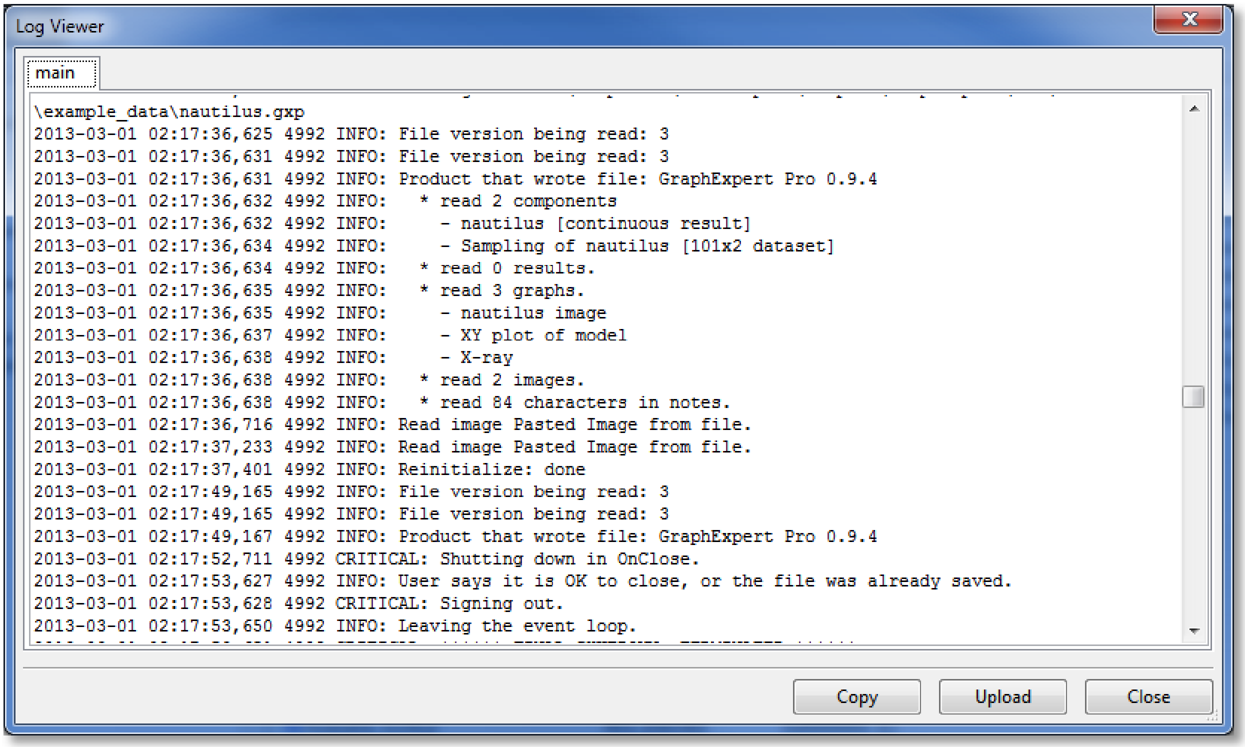
Use Ctrl+Up and Ctrl+Down to cycle through your history of commands. The notebook in the lower part of the sandbox shows the last documentation page that was accessed (context sensitive), and the history of commands that you have typed. Closing the sandbox does not close your session; you may resume it by opening the sandbox again. Sessions are not, however, preserved across different invocations of GraphExpert Professional.

15.3 Log Viewer

To view the log files, select *Tools->View Logs* from the main menu. All logs are stored in the $HOME/.graphexpert/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 GraphExpert 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 GraphExpert Professional server (http://www.graphexpert.com), 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.

See *Logging Preferences* for more detail on configuring the logging options.



```
\example_data\nautilus.gxp
2013-03-01 02:17:36,625 4992 INFO: File version being read: 3
2013-03-01 02:17:36,631 4992 INFO: File version being read: 3
2013-03-01 02:17:36,631 4992 INFO: Product that wrote file: GraphExpert Pro 0.9.4
2013-03-01 02:17:36,632 4992 INFO:   * read 2 components
2013-03-01 02:17:36,632 4992 INFO:     - nautilus [continuous result]
2013-03-01 02:17:36,634 4992 INFO:     - Sampling of nautilus [101x2 dataset]
2013-03-01 02:17:36,634 4992 INFO:   * read 0 results.
2013-03-01 02:17:36,635 4992 INFO:   * read 3 graphs.
2013-03-01 02:17:36,635 4992 INFO:     - nautilus image
2013-03-01 02:17:36,637 4992 INFO:     - XY plot of model
2013-03-01 02:17:36,638 4992 INFO:     - X-ray
2013-03-01 02:17:36,638 4992 INFO:   * read 2 images.
2013-03-01 02:17:36,638 4992 INFO:   * read 84 characters in notes.
2013-03-01 02:17:36,716 4992 INFO: Read image Pasted Image from file.
2013-03-01 02:17:37,233 4992 INFO: Read image Pasted Image from file.
2013-03-01 02:17:37,401 4992 INFO: Reinitialize: done
2013-03-01 02:17:49,165 4992 INFO: File version being read: 3
2013-03-01 02:17:49,165 4992 INFO: File version being read: 3
2013-03-01 02:17:49,167 4992 INFO: Product that wrote file: GraphExpert Pro 0.9.4
2013-03-01 02:17:52,711 4992 CRITICAL: Shutting down in OnClose.
2013-03-01 02:17:53,627 4992 INFO: User says it is OK to close, or the file was already saved.
2013-03-01 02:17:53,628 4992 CRITICAL: Signing out.
2013-03-01 02:17:53,650 4992 INFO: Leaving the event loop.
```

15.4 Running Multiple Instances

Running multiple instances of GraphExpert Professional is supported; the only side effect is that, because all instances share the same log files, logging might not function properly.

15.5 Where files are located

GraphExpert Professional places files in several locations on disk; the locations are documented below.

15.5.1 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)\GraphExpert Professional
Windows (32 bit)	c:\Program Files\GraphExpert Professional
Mac (OSX)	/Applications/GraphExpertPro.app
Linux (installed as user)	$HOME/GraphExpertPro
Linux (installed as root)	/usr/local/GraphExpertPro

15.5.2 Application Settings

Platform	Application Settings Location
Windows	in registry: HKEY_CURRENT_USER\Software\GraphExpert Professional
Mac (OSX)	$HOME/Library/Preferences/GraphExpert Professional Preferences
Linux	$HOME/.GraphExpert Professional

15.5.3 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\.graphexpert
Mac (OSX)	$HOME/.graphexpert
Linux	$HOME/.graphexpert

15.5.4 GraphExpert Pro 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 GraphExpert resides.

Platform	Documents Location
Windows	$MYDOCUMENTS\GraphExpert Professional
Mac (OSX)	$HOME/Documents/GraphExpert Professional
Linux	$HOME/Documents/GraphExpert Professional

CHAPTER

SIXTEEN

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 "GraphExpert 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

SEVENTEEN

THIRD-PARTY LICENSING

GraphExpert 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.

17.1 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.

17.2 WxWidgets and WxPython

17.2.1 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.

17.2.2 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.

17.3 Numpy

17.4 Scipy

17.5 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.

17.6 xlrd

GraphExpert Professional utilizes Excel reading capabilities from the xlrd library. This product includes software developed by Roman V. Kiseliov <roman@kiseliov.ru>.

Portions copyright (C) 2007, Stephen John Machin, Lingfo Pty Ltd All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. 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.

3. None of the names of Stephen John Machin, Lingfo Pty Ltd and 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.

Copyright (C) 2005 Roman V. Kiseliov All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. 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.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by Roman V. Kiseliov <roman@kiseliov.ru>.
4. Redistributions of any form whatsoever must retain the following acknowledgment: This product includes software developed by Roman V. Kiseliov <roman@kiseliov.ru>.

THIS SOFTWARE IS PROVIDED BY Roman V. Kiseliov ``AS IS” AND ANY EXPRESSED 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 Roman V. Kiseliov OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,

17.7 BioPython

GraphExpert Professional utilizes the Lowess algorithm originating from the Biopython package.

Biopython License Agreement

17.8 winpaths

GraphExpert 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

17.9 Python Imaging Library (PIL)

The Python Imaging Library (PIL) is

17.10 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.

17.11 JPEG library

This software is based in part on the work of the Independent JPEG Group.

CHAPTER

EIGHTEEN

APPENDIX A: MATH FUNCTIONS

This section documents the math functions that are available for use in GraphExpert Professional 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.

18.1 Constants

18.1.1 pi

The mathematical constant 3.14159. . . .

18.1.2 exp1

The mathematical constant 2.71828. . . ..

18.2 Transcendental Functions

18.2.1 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.

18.2.2 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]`.

18.2.3 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}.

18.2.4 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`.

18.2.5 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}.

18.2.6 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]`.

18.2.7 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`.

18.2.8 cos(x)

Trigonometric cosine.

18.2.9 cosh(x)

Hyperbolic cosine. Equivalent to `1/2 * (exp(x) + exp(-x))`.

18.2.10 erf(x)

Error function of x defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

18.2.11 erfc(x)

Returns

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

18.2.12 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.

18.2.13 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*.

18.2.14 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*.

18.2.15 log10(x)

Return the base 10 logarithm. NaNs is returned if x is negative.

18.2.16 sin(x)

Trigonometric sine.

18.2.17 sinh(x)

Hyperbolic sine, element-wise. Equivalent to `1/2 * (exp(x) - exp(-x))`.

18.2.18 sqrt(x)

Return the positive square-root. Returns NaN if x is negative.

18.2.19 tan(x)

Compute tangent. Equivalent to `sin(x)/cos(x)`.

18.2.20 tanh(x)

Compute hyperbolic tangent. Equivalent to `sinh(x)/cosh(x)`.

18.3 Bessel Functions

18.3.1 bessel_j0(x)

The Bessel function of order 0 at x.

18.3.2 bessel_j1(x)

The Bessel function of order 1 at x.

18.3.3 bessel_jn(n,x)

Bessel function of integer or real order n.

18.3.4 bessel_y0(x)

The Bessel function of the second kind of order 0 at x.

18.3.5 bessel_y1(x)

The Bessel function of the second kind of order 1 at x.

18.3.6 bessel_yn(n,x)

Bessel function of the second kind of integer or real order n at x.

18.3.7 bessel_i0(x)

Modified Bessel function of the first kind, order 0.

18.3.8 bessel_i1(x)

Modified Bessel function of order 1 at x.

18.3.9 bessel_k0(x)

Modified Bessel function of the second kind (sometimes called the third kind) of order 0 at x.

18.3.10 bessel_k1(x)

Modified Bessel function of the second kind (sometimes called the third kind) of order 1 at x.

18.3.11 bessel_kn(n,x)

Modified Bessel function of the second kind (sometimes called the third kind) for integer or real order n at x.

18.3.12 bessel_i0e(x)

Exponentially scaled modified Bessel function of order 0 at x. bessel_i0e(x) = exp(-|x|) * bessel_i0(x).

18.3.13 bessel_i1e(x)

Exponentially scaled modified Bessel function of order 0 at x. bessel_i1e(x) = exp(-|x|) * bessel_i1(x).

18.3.14 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).

18.3.15 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).

18.3.16 bessel_jvp(n,x,i)

Return the ith derivative of bessel_jn(n,x).

18.3.17 bessel_yvp(n,x,i)

Return the nth derivative of bessel_yn(n,x).

18.3.18 bessel_kvp(n,x,i)

Return the nth derivative of bessel_kn(n,x).

18.3.19 bessel_ivp(n,x,i)

Return the nth derivative of bessel_in(n,x).

18.4 Comparison Functions

18.4.1 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.

18.4.2 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.

18.4.3 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).

18.4.4 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).

18.5 Miscellaneous Functions

18.5.1 abs(x)

Calculate the absolute value.

18.5.2 ceil(x)

Returns the ceiling of the input. The ceil of the scalar *x* is the smallest integer *i*, such that *i >= x*.

18.5.3 copysign(x1,x2)

Change the sign of x1 to that of x2.

18.5.4 degrees(x)

Convert angles from radians to degrees.

18.5.5 factorial(n)

Take the factorial of an integer n.

18.5.6 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*.

18.5.7 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.

18.5.8 hypot(x1,x2)

Given the “legs” of a right triangle, return its hypotenuse. Equivalent to `sqrt(x1**2 + x2**2)`.

18.5.9 isinf(x)

Test for positive or negative infinity.

18.5.10 isnan(x)

Test for Not a Number (NaN).

18.5.11 legendre(n,x)

Legendre function of order n evaluated at x. Orthogonal over [1,1] with weighting function 1.

18.5.12 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.

18.5.13 sh_legendre(n,x)

Shifted Legendre function of order n evaluated at x. Orthogonal over [0,1] with weighting function 1.

18.5.14 radians(x)

Convert angles from degrees to radians.

18.5.15 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.

18.5.16 special_beta(p,q)

Returns

$$\beta(p, q) = \Gamma(p)\Gamma(q)/\Gamma(p + q)$$

18.5.17 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).

18.5.18 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].

18.5.19 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.

18.6 Array Specific Functions

18.6.1 append(array)

Append values to the end of an array.

18.6.2 array(object)

Create an array from an array_like object, like a Python list or another array.

18.6.3 asarray(object)

Convert the input to an array. a is an array_like object, like a Python list.

18.6.4 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*).

18.6.5 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).

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

18.6.6 column_stack((a,b,c,d,. . .))

Stack 1D arrays as columns into a 2D array.

18.6.7 copy(object)

Return an array copy of the given object.

18.6.8 concatenate((a1,a2,a3,. . .))

Join a sequence of arrays together.

18.6.9 dot(a,b)

Dot product of two arrays a and b (returns a scalar)

18.6.10 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.

18.6.11 eye(n)

Return a 2-D array with ones on the diagonal and zeros elsewhere.

18.6.12 fliplr(a)

Flip (reverse) an array in the left/right direction (columnwise).

18.6.13 flipud(a)

Flip (reverse) an array in the up/down direction (rowwise).

18.6.14 hstack((a,b,c,d,. . . .))

Stack arrays in sequence horizontally (columnwise)

18.6.15 identity(n)

Return the identity array. The identity array is a square array with ones on the main diagonal.

18.6.16 linspace(start,stop,n)

Return evenly spaced numbers over a specified interval.

Returns *n* evenly spaced samples, calculated over the interval [*start*, *stop*].

18.6.17 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).

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

18.6.18 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`.

18.6.19 nonzero(a)

Returns as an integer array, the indices of the elements are are nonzero. This can be used for indexing into another array.

18.6.20 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.

18.6.21 ones_like(a)

Returns an array of ones with the same dimension as the given array.

18.6.22 roll(a)

Roll array elements rowwise; rows that roll beyond the last position are reintroduced at the first.

18.6.23 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).

18.6.24 sort(a)

Return an ascending sorted copy of the array.

18.6.25 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.

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

18.6.26 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.

18.6.27 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).

18.6.28 vstack((a,b,c,d,. . . .))

Stack arrays in sequence vertically (rowwise).

18.6.29 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.

18.6.30 zeros_like(other_array)

Return an array of zeros with the same dimension as a given array. For example, my_array = zeros_like(x)

18.7 Logical Functions

18.7.1 array_and(b1,b2)

Applies the logical and operator, elementwise, to two boolean arrays b1 and b2. Returns another boolean array.

18.7.2 array_or(b1,b2)

Applies the logical or operator, elementwise, to two boolean arrays b1 and b2. Returns another boolean array.

18.7.3 array_xor(b1,b2)

Applies the logical xor operator, elementwise, to two boolean arrays b1 and b2. Returns another boolean array.

18.7.4 array_not(b1)

Applies the logical not operator, elementwise, to a boolean array b1. Returns another boolean array.

18.7.5 array_gt(x1,x2)

Returns a boolean array that tests x1 > x2 elementwise.

18.7.6 array_lt(x1,x2)

Returns a boolean array that tests x1 < x2 elementwise.

18.7.7 array_le(x1,x2)

Returns a boolean array that tests x1 <= x2 elementwise.

18.7.8 array_ge(x1,x2)

Returns a boolean array that tests x1 >= x2 elementwise.

18.7.9 array_eq(x1,x2)

Returns a boolean array that tests x1 == x2 elementwise.

18.7.10 array_neq(x1,x2)

Returns a boolean array that tests x1 != x2 elementwise.

18.7.11 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 NINETEEN

APPENDIX B: MATH SYMBOL REFERENCE

19.1 Subscripts and superscripts

Command	Result
$alpha_i > beta_i$	$\alpha_i > \beta_i$
$alpha^2 > beta_i^a$	$\alpha^2 > \beta_i^a$

19.2 Fractions, binomials and stacked numbers

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

$$\frac{3}{4}$$

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

$$\binom{3}{4}$$

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

$$\stackrel{3}{4}$$

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

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

19.3 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.

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

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

19.4 Radicals

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

$$\sqrt{2}$$

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

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

19.5 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}`	$\widetilde{xyz}$

19.6 Symbols

Lower-case Greek

α `\alpha`	β `\beta`	χ `\chi`	δ `\delta`	Ϝ `\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

TWENTY

INDICES AND TABLES

- genindex
- search

INDEX