

# Chapter I-1

## Introduction to Igor Pro

Overview of Igor Pro .....	2
Igor Objects .....	2
Waves — The Key Igor Concept .....	2
How Objects Relate .....	4
More Objects.....	5
A Typical Igor Session.....	5
Creating Data.....	5
Displaying Data .....	6
Saving Your Work .....	6
Processing Data.....	6
Generating Output.....	6
Igor's User Interface .....	6
The Command Window .....	7
Menus, Dialogs and Commands .....	8
How to Learn to Use Igor .....	8
Getting Help Online .....	9
Igor Tips (Macintosh only) .....	9
Status Line Help, Tool Tips and Context-Sensitive Help (Windows only).....	10
The Igor Help System.....	10
Using Igor for Heavy-Duty Jobs .....	11
Built-in Functions and Operations .....	11
Igor Extensions.....	11
User-Defined Procedures.....	12

## Chapter I-1 — Introduction to Igor Pro

---

### Overview of Igor Pro

Igor Pro is an integrated program for visualizing, analyzing, transforming and presenting data.

Igor Pro's features include:

- Publication-quality graphics
- High-speed data display
- Ability to handle large data sets
- Curve-fitting, Fourier transforms, smoothing, histograms and other analyses
- Waveform arithmetic
- Image display and processing
- Combination graphical and command-line user interface
- Automation and data processing via a built-in programming environment
- Extensibility through modules written in the C language

Some people use Igor simply to produce high-quality, finely-tuned graphs and page layouts. Others use Igor as an all-purpose workhorse to acquire, analyze and present experimental data using its built-in programming environment. We have tried to write the Igor program and this manual to fulfill the needs of the entire range of Igor users.

### Igor Objects

The basic objects that all Igor users work with are:

- waves
- graphs
- tables
- page layouts

A collection of objects is called an “experiment” and is stored in an experiment file. When you open an experiment, Igor recreates the objects that comprise it.

### Waves — The Key Igor Concept

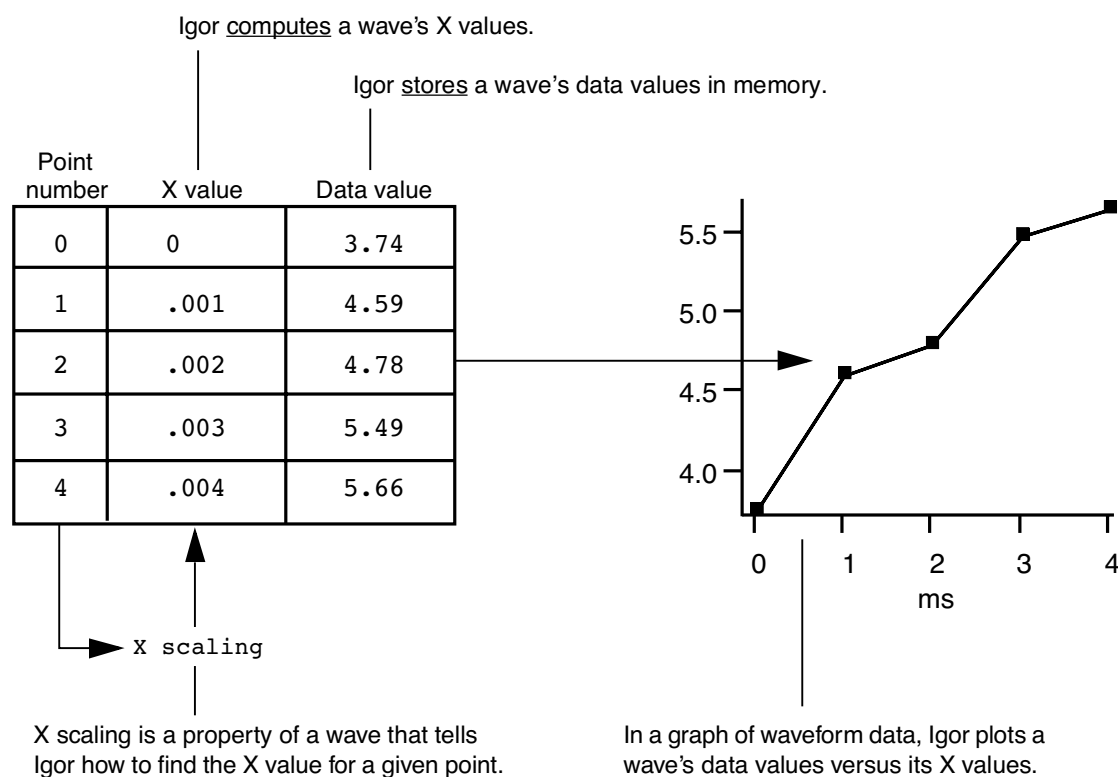
We use the term “wave” to describe the Igor object that contains an array of numbers. Wave is short for “waveform”. The wave is the most important Igor concept.

Igor was originally designed to deal with waveform data. A waveform typically consists of hundreds to thousands of values measured at evenly spaced intervals of time. Such data is usually acquired from a digital oscilloscope, scientific instrument or analog-to-digital converter card.

The distinguishing trait of a waveform is the *uniform spacing* of its values along an axis of time or other quantity. An Igor wave has an important property called “X scaling” which

## Chapter I-1 — Introduction to Igor Pro

you set to tell Igor what the spacing of your data is. Igor *stores* the y component for each point of a wave in memory but it *computes* the x component based on the wave's X scaling.



Waves can have from one to four dimensions and can contain either numeric or text data.

Igor is also capable of dealing with data that does not fit the waveform metaphor. We call this XY data. Igor can treat two waves as an XY pair. In an XY pair, the data values of one wave supply the x component and the data values of another wave supply the y component for each point in the pair.

A few analysis operations, such as Fourier transforms, inherently work only on waveform data. They take a wave's X scaling into account.

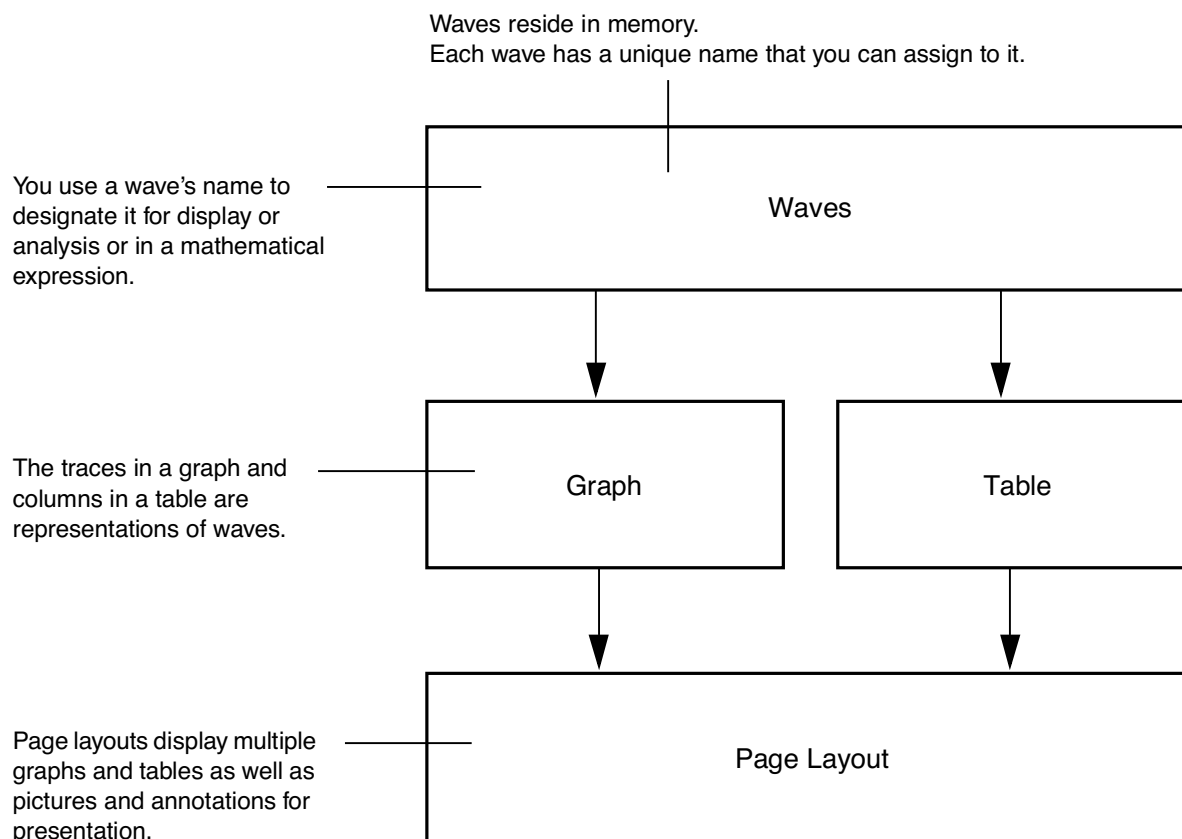
Other operations work equally well on waveform or XY data. Igor can graph either type of data and its powerful curve fitting works on either type.

Most users create waves by loading data from a text file. You can also create waves by typing in a table or by evaluating a mathematical expression.

## Chapter I-1 — Introduction to Igor Pro

---

### How Objects Relate



You refer to a wave by its name. Although you can display waves in a graph or table, a wave does not need to be displayed to exist.

Graphs are used to visualize waves and to generate high-quality printouts for presentation. The traces in a graph are representations of waves. If you modify a wave, Igor automatically updates graphs. Igor labels the axes of a graph intelligently. Tick marks never run into one another and are always “nice” values no matter how you zoom around.

Tables are used to enter, inspect or modify wave data. A table in Igor is not the same as a spreadsheet in other graphing programs. A column in a table is a *representation* of the contents of a wave. The wave continues to exist even if you remove it from the table or close the table entirely.

Page layouts permit you to arrange multiple graphs and tables as well as pictures and annotations for presentation. If you modify a graph or table, either directly or indirectly by changing the contents of a wave, Igor automatically updates its representation in a layout.

Both graphs and layouts include drawing tools for adding lines, arrows, boxes, polygons and pictures to your presentations.

### More Objects

In addition to the basic objects, here are some additional objects that advanced users may find useful:

- numeric and string variables
- data folders
- notebooks
- control panels
- surface plots

A numeric variable stores a single number and a string variable stores an array of text characters. Numeric and string variables are useful for storing bits of data for procedures that you may write.

A data folder can contain waves, numeric variables, string variables and other data folders. Data folders provide a way to keep a set of related data, such as all of the waves from a particular run of an experiment, together and separate from like-named data from other sets.

A notebook is like a text-editor or word-processor document. You can use a notebook to keep a log of results or to produce a report. Notebooks are also handy for looking at Igor technical notes or other text documentation.

A control panel is a window containing buttons, checkboxes and other controls and read-outs. You can create a control panel as an interface to procedures that you have written. This is a technique for advanced users.

A surface plot displays three-dimensional data as a point-cloud, wire frame, filled surface, scatter plot, or path in space.

## A Typical Igor Session

### Creating Data

Usually, you start off by loading data into Igor from a text file. The load operation creates waves and, if you want, can also create a table to display the waves. Igor assigns default names to the waves but gives you a chance to provide meaningful names.

If you have data to enter manually then the sequence is a bit different. You start by creating an empty table. Then you type your data in, point-by-point, or paste your data from the clipboard into the table. In either case, this creates waves in memory that are displayed in the table.

## Chapter I-1 — Introduction to Igor Pro

---

### Displaying Data

You now have named waves in memory. The next step is usually to display the waves in one or more graphs. In some cases, you may want to display data in a table to examine or modify it.

### Saving Your Work

At this point you have done enough that it's worthwhile to save your work. You have created a collection of Igor objects — waves, graphs and tables. We call this collection an “experiment”. When you choose Save Experiment from the File menu, Igor saves the entire collection in a file which we call an “experiment file”. You could now quit Igor. Later, you could run Igor and open the experiment file to restore all of the objects.

The collection of all objects that currently exist in Igor is called the “current experiment”. An experiment file is a collection of objects saved in a file on disk.

### Processing Data

Often the next step is to do some kind of analysis. During an analysis, for example curve fitting, you may create more waves and graphs.

### Generating Output

You now have both raw and processed data. You may want to print a graph or to export a graph as a picture to a word processing or page layout program. You may want to combine multiple graphs in an Igor page layout and then print or export the layout.

## Igor's User Interface

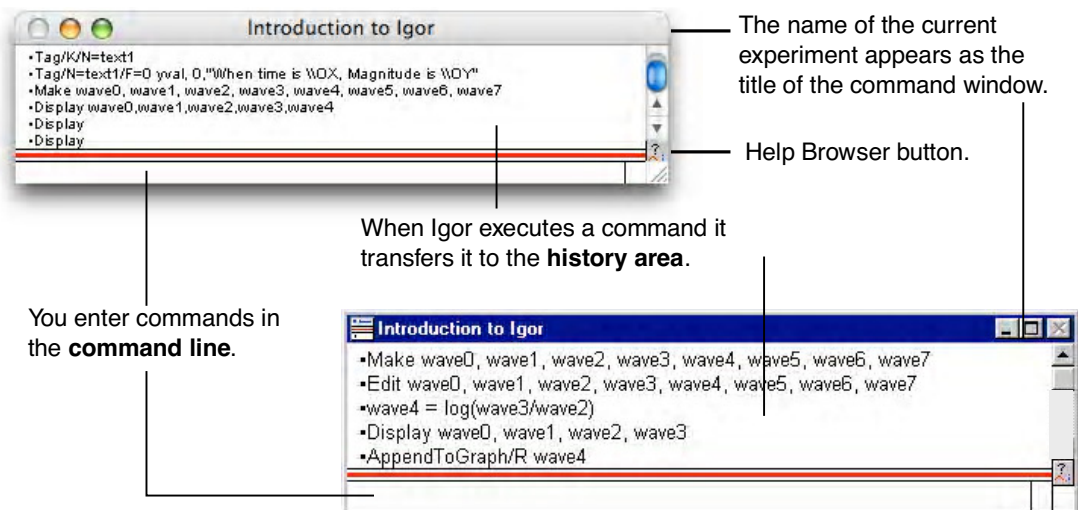
Igor uses a combination of the familiar graphical user interface and a command line interface. This approach gives Igor both ease-of-use and programmability.

The job of the user interface is to allow you to apply Igor's operations and functions to objects that you create. You can do this in three ways:

- via menus and dialogs
- by typing Igor commands directly into the command line
- by writing Igor procedures

### The Command Window

The command window is Igor's control center. It appears at the bottom of the screen.



Igor also uses the history area to report results of analyses like curve-fitting or waveform statistics.

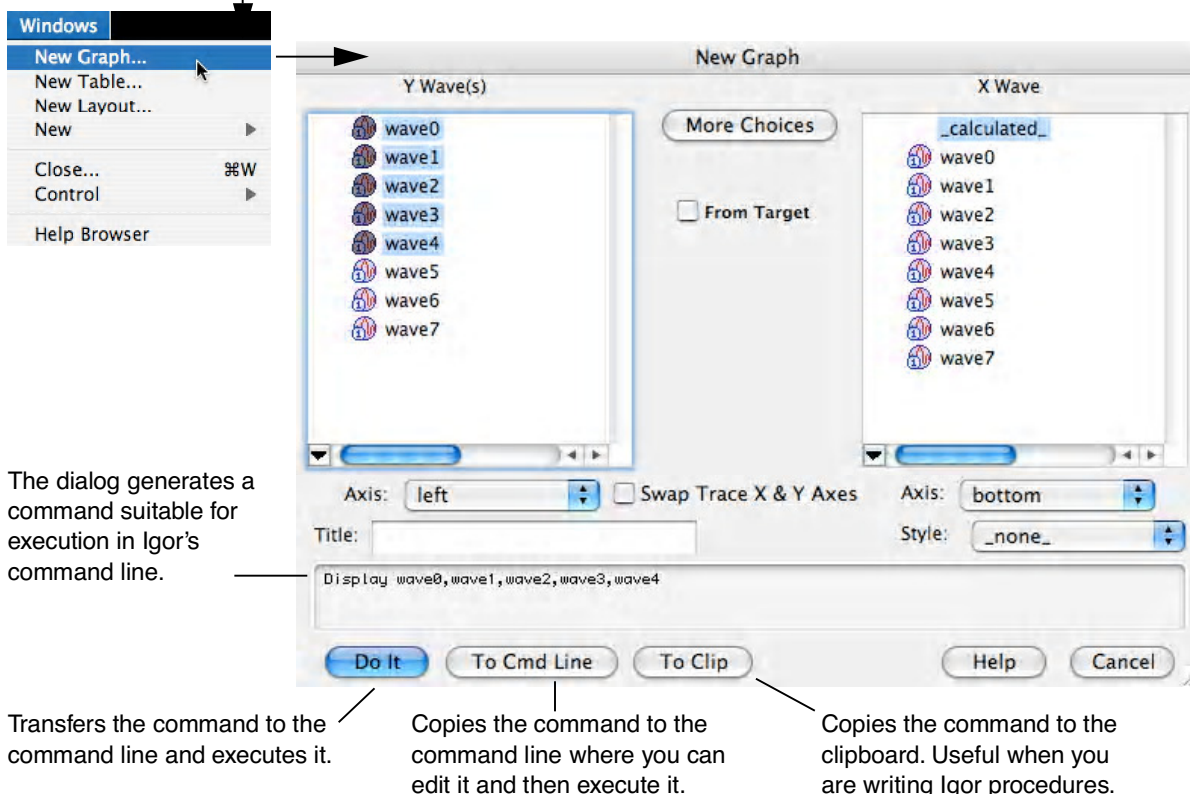
## Chapter I-1 — Introduction to Igor Pro

### Menus, Dialogs and Commands

Menus and dialogs provide easy access to Igor's operations.

When you choose a menu item...

...Igor presents a dialog.



As you click and type in the items in the dialog, Igor generates an appropriate command. The command being generated is displayed in the **command box** near the bottom of the dialog.

As you get to know Igor, you will find that some commands are easier to invoke from a dialog and others are easier to enter directly in the command line.

There are some menus and dialogs that bypass the command line, usually because they perform functions that have no command line equivalents.

## How to Learn to Use Igor

To harness the power of Igor, you need to understand its fundamental concepts. Some of these concepts are familiar to you. However, Igor also uses a few concepts that will be new to you and may seem strange at first. The most important of these are *waves* and *experiments*.

Take a little time to learn how to use Igor. It will save you a lot of time in the long run.

- Go through the **Guided Tour of Igor Pro** in Chapter I-2.

## Chapter I-1 — Introduction to Igor Pro

- Learn how to use Igor's extensive Macintosh Igor Tips and Windows context-sensitive help.
- Familiarize yourself with the Igor Help Browser (choose Igor Help Browser from the Help menu).
- Browse the items in the Learning Aids folder.
- Browse the Igor experiments in the Examples folder.
- Load some of your data into Igor and play around.

Once you start to get the feel for Igor you may want to delve in deeper.

- Learn how to use the online manual. You can open the online manual via the Igor Help Browser.
- Read the *User's Guide* or at least Chapter II-3, **Experiments, Files and Folders**, and Chapter II-5, **Waves**.

Before you start serious programming in Igor, you should read all of Chapter IV-1, **Working with Commands**, Chapter IV-2, **Programming Overview**, and Chapter IV-3, **User-Defined Functions** in the *Programming* manual.

## Getting Help Online

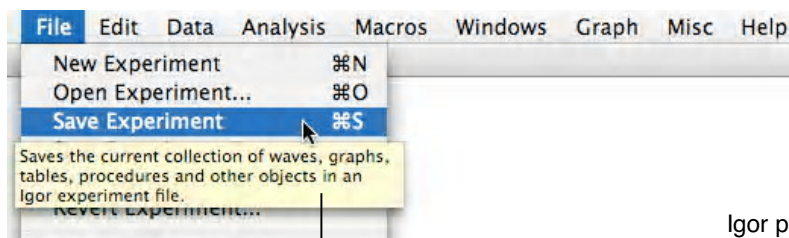
Igor includes an extensive help facility that gives you both brief tips and in-depth descriptions.

There are several kinds of online help available in Igor:

- Igor Tips (*Macintosh only*)
- tool tips, status line help and context-sensitive help (*Windows only*)
- the Igor help system

### Igor Tips (*Macintosh only*)

Igor Tips allows a program to present a brief explanation of menus, dialogs and other user interface items.



Igor Tips provides a bit of extra information on the topic of your choice.



You can turn Igor Tips on using the help menu or by pressing option-help.

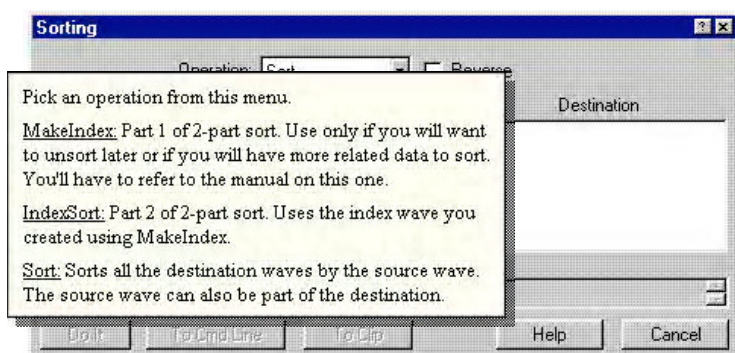
Igor provides extensive Igor Tips for menu items, dialog items and many other user-interface items.

## Chapter I-1 — Introduction to Igor Pro

### Status Line Help, Tool Tips and Context-Sensitive Help (*Windows only*)

Igor's Windows help system provides three ways to get immediate help on a button or tool, a menu item, or a dialog item. Status line help automatically shows brief descriptions of menu items and tools in the status line area at the bottom of the main Igor Pro window. Tool tips provide very brief descriptions of tools or buttons when you point the cursor at the item. Context-sensitive help displays a pop-up window containing information about the menu item, icon or dialog item of interest. Depending upon the situation, the context-sensitive help may be accessed in several ways:

- |                  |  |
|------------------|--|
| Menus and Icons: | Press Shift+F1 to get the question-mark cursor  and click the item of interest.   |
| Dialogs:         | Click the question-mark button in the upper-right corner of dialog to get the question-mark cursor  , then click a dialog item. |



### The Igor Help System

Igor has an online help system that provides help for the following things:

- dialogs
- built-in and external functions and operations
- general topics

The help provided is an abridged version of what is in the Igor manuals.

**Getting Help Online** on page I-9 explains the Igor Help system in detail. In brief, you should know the following:

- Use the Help menu or press the Help key (*Macintosh*) or F1 key (*Windows*) to display the Igor Help Browser.
- Use the Igor Help Browser Shortcuts tab to get a list of handy shortcuts and techniques.
- Use the Igor Help Browser Search tab to search Igor help, procedure and example files.

- Use the Igor Help Browser Command Help tab to get reference information on functions (routines like mean, area and interp, that return a direct result) and operations (routines like CurveFit, Display and FFT, that return no direct result).
- Use the Help button in a dialog to get an orientation to the dialog.

## Using Igor for Heavy-Duty Jobs

If you generate a lot of raw data or need to do custom technical computing, you will find it worthwhile to learn how to put Igor to heavy-duty use. It is possible to automate some or all of the steps involved in processing and presenting your data. To do this, you must learn how to write Igor procedures.

Igor includes a built-in programming environment that lets you manipulate waves, graphs, tables and all other Igor objects. You can invoke built-in operations from your own procedures to build higher-level operations customized for your work.

Learning to write Igor procedures is easier than learning a lower-level language such as FORTRAN or C. The Igor programming environment is interactive so you can write and test small routines and then put them together piece-by-piece. It allows you to deal with very high level objects such as waves and graphs but also gives you fine control over your data. Nonetheless, it is still programming. To master it requires an effort on your part.

The Igor programming environment is described in detail in the *Programming* manual. You can also learn about Igor programming by examining the WaveMetrics Procedures and example experiments that were installed on your hard disk.

Igor provides three kinds of routines, described in the next sections, that you can use to create and modify objects.

### Built-in Functions and Operations

A built-in function is an Igor routine, such as sin, exp or ln, that directly returns a result. A built-in operation is a routine, such as Display, FFT or Integrate, that acts on an object and may create new objects but does not directly return a result.

### Igor Extensions

An extension is a piece of external (not built-in to Igor) code that adds functionality to Igor. We use the terms “external operation” or “XOP” and “external function” or “XFUNC” for operations and functions added by extensions. An extension can add menus, dialogs and windows to Igor as well as operations and functions.

To write an extension, you must be a programmer and you need the optional (extra cost) **Igor External Operations Toolkit**. See **Creating Igor Extensions** on page IV-161.

Although *writing* an extension is a job for a programmer, anyone can *use* an extension. The Igor installer automatically installs commonly used extensions in the Igor Extensions folder. These extensions are available for immediate use. Less commonly used extensions

## Chapter I-1 — Introduction to Igor Pro

are installed in the More Extensions folder. You can activate one of these extensions by placing an alias (*Macintosh*) or shortcut (*Windows*) to it in the Igor Extensions folder.

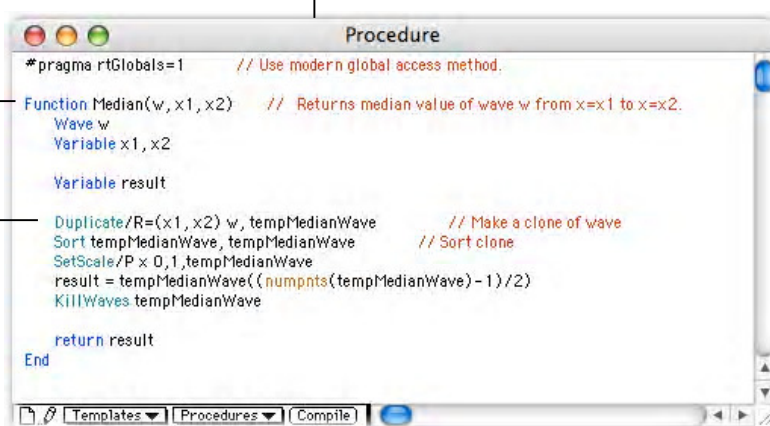
### User-Defined Procedures

A user-defined procedure is a routine written in Igor's built-in programming language by entering text in a procedure window. It can call upon built-in or external functions and operations as well as other user-defined procedures to manipulate Igor objects. Sets of procedures are stored in procedure files.

You can create Igor procedures by entering text in a procedure window.

Each procedure has a name which you use to invoke it.

Procedures can call operations, functions or other procedures. They can also perform waveform arithmetic.



## Chapter I-2

# Guided Tour of Igor Pro

Overview .....	15
Assumptions .....	15
Terminology .....	15
About the Tour .....	15
General Tour .....	16
Launching Igor Pro .....	16
Entering Data .....	16
Making a Graph .....	18
Touching up a Graph .....	18
Adding a Legend .....	21
Adding a Tag .....	21
Using Preferences .....	23
Making a Page Layout .....	24
Saving Your Work .....	26
Loading Data .....	26
Appending to a Graph .....	28
Offsetting a Trace .....	29
Un-Offsetting a Trace .....	29
Drawing in a Graph .....	30
Making a Window Recreation Macro .....	32
Recreating the Graph .....	33
Saving Your Work .....	33
Graphically Editing Data .....	33
Making a Category Plot (Optional) .....	34
The Command Window .....	38
Browsing Waves .....	40
Synthesizing Data .....	41
Zooming & Panning .....	42
Making a Graph with Multiple Axes .....	43
Saving Your Work .....	45
Using Cursors .....	45

## Chapter I-2 — Guided Tour of Igor Pro

---

Removing a Trace and Axis.....	46
Creating a Graph with Stacked Axes .....	46
Appending to a Layout.....	48
Saving Your Work .....	48
Creating Controls.....	49
Creating a Dependency.....	51
Saving Your Work .....	52
End of the General Tour .....	52
Analysis Tour #1 .....	53
Launching Igor Pro.....	53
Creating Synthetic Data .....	53
Curve Fitting to a Gaussian.....	54
Sorting .....	55
Fitting to a Sub-Range .....	55
Extrapolating a Fit .....	57
Appending a Fit .....	58
Analysis Tour #2.....	59
Launching Igor Pro.....	59
Creating Synthetic Data .....	59
Histogram of White Noise.....	59
Histogram of Gaussian Noise .....	60
Curve Fit of Histogram Data.....	61
Curve Fit Residuals (Optional) .....	62
Writing a Procedure (Optional).....	64
Saving a Procedure File (Optional) .....	68
Including a Procedure File (Optional) .....	69
For Further Exploration .....	70

### Overview

In this chapter we take a look at the main functions of Igor by stepping through some typical operations. Our goal is to orient you so that you can comfortably read the rest of the manual or explore the program on your own. You will benefit most from this tour if you actually do the instructed operations on your computer as you read this chapter. Screen shots are provided to keep you synchronized with the tour.

### Assumptions

This guided tour as well as the rest of this manual assumes that you are a reasonably experienced computer user. If not, please consult your computer documentation before proceeding.

We also assume that you have already installed the complete Igor Pro package. Installation instructions are displayed when you run the installer program.

### Terminology

If you have read Chapter I-1, **Introduction to Igor Pro**, you already know these terms.

Experiment:	The entire collection of data, graphs and other windows, program text and whatnot that make up the current Igor environment or workspace.
Wave:	Short for waveform, this is basically a named array of data with optional extra information.
Name:	Since Igor contains a built-in programming and data transformation language, each object must have a unique name.
Command:	This is a line of text that instructs Igor to do some task. Igor is command-driven so that it can be easily programmed.

### About the Tour

This tour consists of three sections: General Tour, Analysis Tour #1 and Analysis Tour #2.

The General Tour is a rambling exploration intended to introduce you to the way things work in Igor and give you a general orientation. This will touch on nearly every aspect of Igor Pro except data analysis, programming and multidimensional data. To sample these other Igor features, investigate some of the sample experiments in the Examples folder. This tour takes 2 to 4 hours but does not have to be performed in one sitting.

The Analysis Tours guide you through Igor's data analysis facilities including simple curve fitting.

When you've completed the first tour you may want to explore freely on your own before starting the second tour which assumes that you are comfortable with the basics of Igor.

### General Tour

In this exercise, we will generate data in three ways (typing, loading and synthesizing) and will generate graph, table and page layout windows. We will jazz up a graph and a page layout with a little drawing and some text annotation. At the end we will explore some of the more advanced features of Igor Pro.

### Launching Igor Pro

1. **Double-click the Igor Pro application file on your hard disk.**  
If Igor was already running, choose New Experiment from the File menu.
2. **Use the Misc menu to turn preferences off.**  
Turning preferences off ensures that the tour works the same for everyone.

### Entering Data

1. **If a table window is showing, click in it to bring it to the front.**  
When Igor starts up, it normally creates a new blank table unless you have turned this feature off in the Miscellaneous Settings dialog. If the table is not showing, perform the following two steps:
  - 1a. **Choose the New Table item from the Windows menu.**  
The New Table dialog will appear.
  - 1b. **Click the Do It button.**  
A new blank table will be created.
2. **Type “0.1” and then press return (Macintosh) or enter (both platforms).**  
A wave named “wave0” was created with 0.1 as the value of the first point. Entering a value in the first row (point 0) of the first blank column automatically creates a new wave.

## Chapter I-2 — Guided Tour of Igor Pro

3. Type the following numbers, pressing return or enter after each one:

1.2

1.9

2.6

4.5

5.1

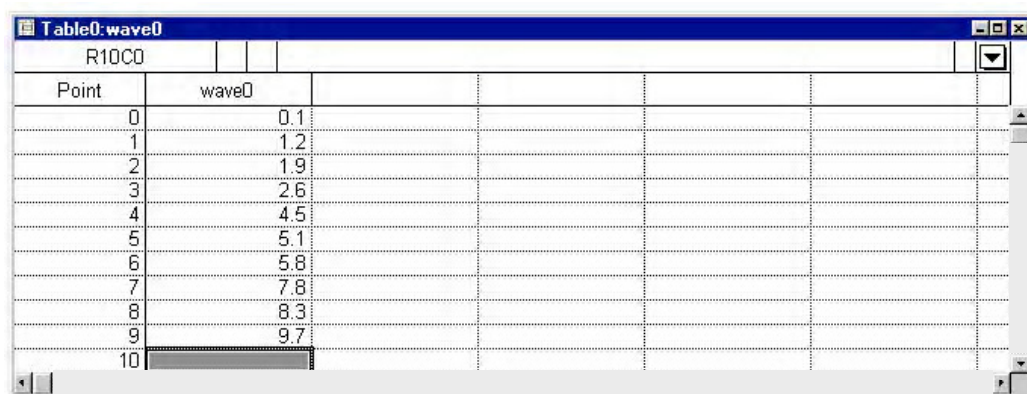
5.8

7.8

8.3

9.7

Your table should look like this:



Point	wave0
0	0.1
1	1.2
2	1.9
3	2.6
4	4.5
5	5.1
6	5.8
7	7.8
8	8.3
9	9.7
10	

4. Click in the first cell of the first blank column.

5. Enter the following numbers in the same way:

-0.12

-0.08

1.3

1

0.54

0.47

0.44

0.2

0.24

0.13

6. Choose Rename from the Data menu.

## Chapter I-2 — Guided Tour of Igor Pro

---

7. Click “wave0” in the list, click the arrow icon, and then select all of the characters in the New Name column of the list.
8. Replace “wave0” with “time”.  
Notice Igor wouldn’t let you use the name “time” because it is a built-in string function. We apologize for usurping such a common name.
9. Change the name to “timeval”.
10. Select “wave1” from the list, click the arrow icon, press tab until the name in the New Name box is highlighted, and type “yval”.
11. Click Do It.  
Notice the column headers in the table have changed to reflect the name changes.

### Making a Graph

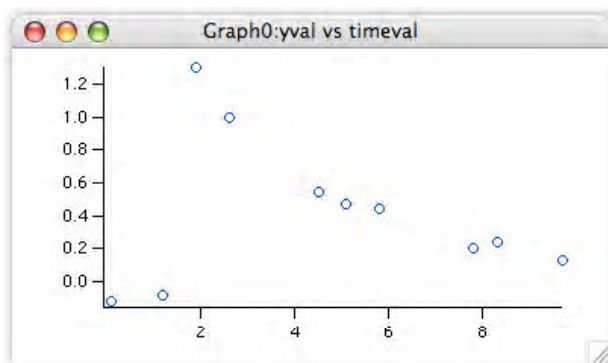
1. Choose New Graph from the Windows menu.  
The New Graph dialog will appear. This dialog comes in a simple form that most people will use and a more complex form that allows you to create complex multi-axis graphs in one step.
2. If you see a button labeled Fewer Choices, click it.  
The button is initially labeled More Choices because the simpler form of the dialog is the default.
3. In the Y Wave(s) list, select “yval”.
4. In the X Wave list, select “timeval”.
5. Click Do It.  
A simple graph will be created.

### Touching up a Graph


1. Position the cursor directly over the trace in the graph and double-click.  
The Modify Trace Appearance dialog will appear. You could also have chosen the corresponding menu item from the Graph menu. (*Note:* The Graph menu will only appear when a graph is the target window.)
2. Choose Markers from the Mode pop-up menu.
3. Select the open circle from the pop-up menu of markers.
4. Set the marker color to blue.

5. **Click Do It.**


Your graph should now look like this:



6. **Position the cursor over the bottom axis line.**

The cursor changes to this shape: . This indicates the cursor is over the axis and also that you can offset the axis (and the corresponding plot area edge) to a new position.

7. **Double-click directly on the axis.**

The Modify Axis dialog will appear. (If another dialog appears, click cancel and try again, making sure the  cursor is showing.)

8. **If it is not already showing, click the Axis tab.**

9. **Choose On from the Mirror Axis pop-up.**

10. **Click the Auto/Man Ticks tab.**

11. **Select the Minor Ticks checkbox.**

12. **Click the Ticks and Grids tab.**

13. **Choose Inside from the Location pop-up.**

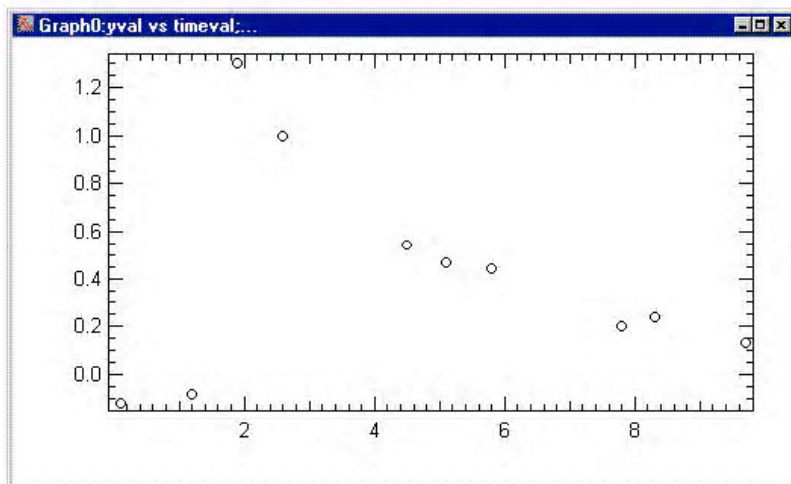
14. **Choose the left axis from the Axis pop-up menu in the top/left corner of the dialog and then repeat steps 8 through 13.**

## Chapter I-2 — Guided Tour of Igor Pro

---

15. Click Do It.

Your graph should now look like this:



16. Again double-click the bottom axis.

The Modify Axis dialog appears again.

17. Click the Axis tab.

18. Deselect the Axis Standoff box.

19. Choose the left axis from the Axis pop-up menu and repeat step 18.

20. Click Do It.

Notice that some of the markers now overlap the axes. The axis standoff setting pushes out the axis so that markers and traces do not overlap the axis. You can use Igor's preferences to ensure this and other settings default to your liking.

21. Double-click one of the tick mark labels (e.g., "6") on the bottom axis.

The Modify Axis dialog reappears, this time with the Axis Range tab showing. If another dialog or tab appears, cancel and try again, making sure to double click one of the tick mark labels on the bottom axis.

22. Choose "Round to nice values" from the pop-up menu that initially reads "Use data limits".

23. Choose the left axis from the Axis pop-up menu and repeat step 22.

24. Click Do It.

Notice that the limits of the axes now fall on "nice" values.

### Adding a Legend

1. **Choose Add Annotation from the Graph menu.**  
The Add Annotation dialog will appear.
2. **Choose Legend from the pop-up menu in the top/left corner.**  
Igor puts text to create a legend into the Annotation text entry area. The Preview area shows what the annotation will look like. Note that the text “\s(yval)” generates the symbol for the yval wave. This is called an “escape sequence” and is used for special effects such as this.
3. **In the Annotation area, change the second instance of “yval” to “Magnitude”.**
4. **Select the Position tab and choose Right Top from the Anchor pop-up menu.**  
Specifying an anchor point helps Igor keep the annotation in the best location as you make the graph bigger or smaller.
5. **Click Do It.**

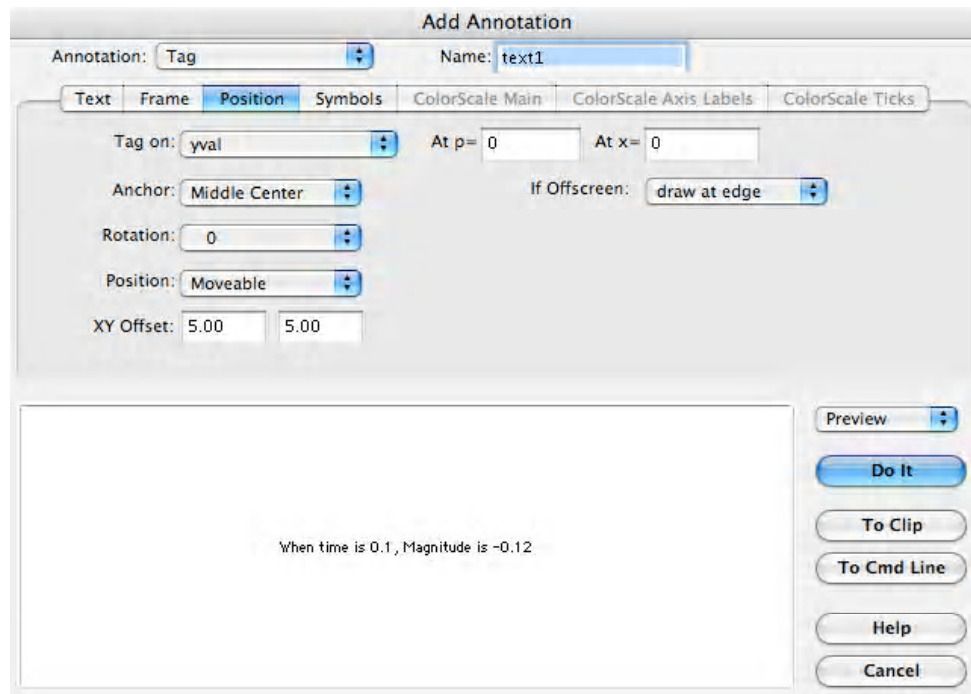
### Adding a Tag

1. **Choose Add Annotation from the Graph menu.**
2. **Choose Tag from the pop-up menu in the top/left corner.**
3. **In the Annotation area of the Text tab, type “when time is ”.**
4. **Choose “Attach point X value” from the Dynamic pop-up menu.**  
Igor inserted the “\0X” escape code into the Annotation text entry area.
5. **In the Annotation area, add “, Magnitude is ”.**
6. **Choose “Attach point Y value” from the Dynamic pop-up menu.**
7. **Switch to the Frame tab and choose None from the Annotation Frame pop-up menu. Now choose Arrow from the Connect Tag to Wave With pop-up menu.**

## Chapter I-2 — Guided Tour of Igor Pro

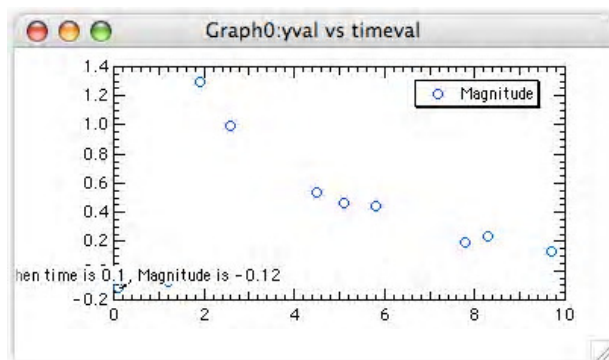
8. Select the Position tab and choose “Middle center” from the Anchor pop-up menu.

The dialog should now look like this:



9. Click Do It.


Your graph should now look like this:



The tag is attached to the first point. An arrow is drawn from the center of the tag to the data point but you can't see it because it is hidden by the tag itself.

10. Position the cursor over the text of the tag.

The cursor changes to a hand. This indicates you can reposition the tag relative to the data point it is attached to.

11. **Drag the tag up and to the right about 1 cm.**  
You can now see the arrow.
12. **With the cursor over the text of the tag, press the option key (*Macintosh*) or Alt key (*Windows*).**  
The cursor changes to this shape: . (You may need to move the cursor slightly to get it to change.)
13. **With the option (*Macintosh*) or Alt (*Windows*) key still down, drag the box cursor to a different data point.**  
The tag jumps to the new data point and the text is updated to show the new X and Y values. Option-drag (*Macintosh*) or Alt-drag (*Windows*) the tag to different data points to see their X and Y values.  
Notice that the tip of the arrow touches the marker. This doesn't look good, so let's change it:
14. **Double-click the text part of the tag.**  
The Modify Annotation dialog appears.
15. **Select the Frame tab and change the Line/Arrow Standoff from "Auto" to "10".**
16. **Click the Change button.**  
The point of the arrow no longer touches the marker.

### Using Preferences

If you have already set preferences to your liking and do not want to disturb them, you can skip this section.

1. **Use the Misc menu to turn preferences on.**
2. **Choose Capture Graph Prefs from the Graph menu.**  
A dialog will appear.
3. **Click the checkboxes for XY plot axes and for XY plot wave styles.**
4. **Click Capture Preferences.**
5. **From the Windows menu, choose New Graph.**
6. **Choose "yval" as the y wave and "timeval" as the x wave.**
7. **Click Do It.**  
The new graph will have a style similar to the model graph.
8. **Press the option (*Macintosh*) or Alt (*Windows*) key while clicking the close box of the new graph.**  
The new graph is killed without presenting a dialog.
9. **Choose Capture Graph Prefs from the Graph menu.**

## Chapter I-2 — Guided Tour of Igor Pro

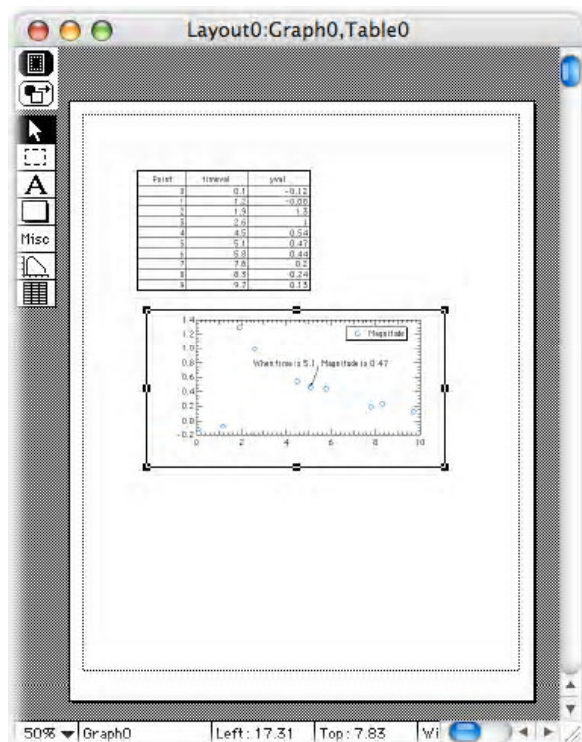
---


10. Click the checkboxes for axes and for wave styles.
11. Click Revert to Defaults.
12. Use the Misc menu to turn preferences off.

### Making a Page Layout


1. **Choose New Layout from the Windows menu.**  
The New Page Layout dialog will appear. The names of all tables and graphs will be shown in the list.
2. **In the Objects to Lay Out list, select Graph0.**
3. **Shift-click (*Macintosh*) or Ctrl-click (*Windows*) on Table0.**
4. **Click Do It.**  
A page layout window will appear with Table0 on top of Graph0.  
The layout initially shows objects at 50% but you may prefer to work at 100%. You can use the pop-up menu at the lower left of the window to change magnification.
5. **Click in Table0.**  
The table becomes selected, resize handles are drawn around the edge and the cursor becomes a hand when over the table.
6. **Click in the middle of the table and drag it so you can see the right edge of the table.**
7. **Position the cursor over the small black square (handle) in the middle of the right side of the table.**  
The cursor changes to a two headed arrow indicating you can drag in the direction of the arrows.
8. **Drag the edge of the table to the left until it is close to the edge of the third column of numbers.**  
You need only get close — Igor snaps to the nearest grid line.
9. **In a similar fashion, adjust the bottom of the table to show all the data but without any blank lines.**

10. Drag the table and graph to match the picture:



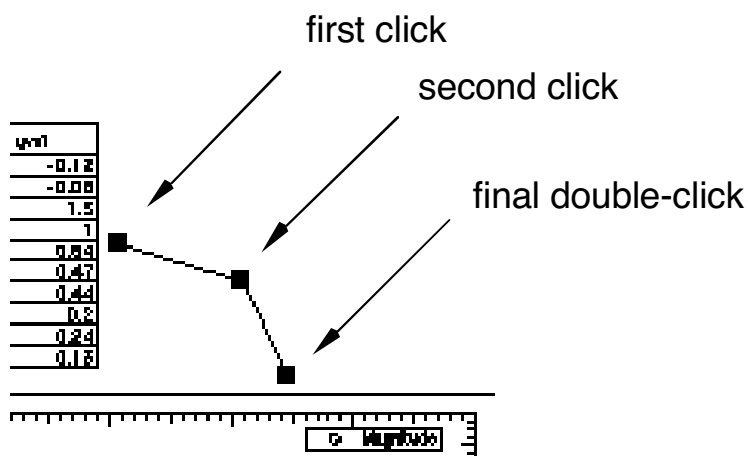
11. Click this icon in the tool palette: 

This activates the drawing tools.

12. Click this icon in the drawing tool palette: 

This is the polygon tool.

13. Click once just to the right of the table, click again about 2 cm right and 1 cm down and finally double-click a bit to the right of the last click and just above the graph.



## Chapter I-2 — Guided Tour of Igor Pro


---


The double-click exits the “draw polygon” mode and enters “edit polygon mode”. If you wish to touch up the defining vertices of the polygon, do so now by dragging the handles (the square boxes at the vertices).

14. **Click the arrow tool in the palette.**

This exits polygon edit mode.

15. **Click the polygon to select it.**

16. **Click the draw environment pop-up icon, , and choose At End from the Line Arrow item.**

17. **Click this icon in the tool palette: **

This is the operate icon. The drawing tools are replaced by the normal layout tools.

We are finished with the page layout for now.

18. **Choose Send Behind from the Control item in the Windows menu.**

### Saving Your Work

1. **Choose Save Experiment from the File menu.**

The save file dialog will appear.

2. **Make sure that Packed Experiment File is selected as the file format.**

3. **Type “Tour #1 a” in the name box.**

4. **Navigate to a folder on your disk where you want to keep your tour files.**

It is best to save your own files outside of the Igor Pro folder to make upgrading in the future simpler.

5. **Click Save.**

If you want to take a break, you can quit from Igor now.

### Loading Data

Before loading data we will use a Notebook window to look at the data file.

0. **If you are returning from a break, launch Igor and open your “Tour #1 a” experiment file. Then turn off preferences using the Misc menu.**

Opening the “Tour #1 a” experiment file restores the Igor workspace to the state it was in when you saved the file. You can open the experiment file by using the Open Experiment item in the File menu or by double-clicking the experiment file.

1. **Choose Notebook from the Open File item in the File menu.**

2. **Navigate to the Igor Pro Folder: Learning Aids: Sample Data folder and open Tutorial Data #1.**

*Windows note:* You will need to select All Files in the Files of type pop-up menu to see the data file in the Select File dialog.

A Notebook window showing the contents of the file will appear. If desired, we could edit the data and then save it. For now we just observe that the file appears to be tab-delimited (tabs separate the columns) and contains names for the columns. Note that the name of the first column will conflict with the data we just entered and the other names have spaces in them.

3. **Click the close box or type command-W (Macintosh) or Ctrl+W (Windows).**

A dialog will appear asking what you want to do with the window.

4. **Click the Kill button.**

The term Kill means to “completely remove from the experiment”. The file will not be affected.

Now we will actually load the data.

5. **Choose Load Delimited Text from the Load Waves item in the Data menu.**

An Open File dialog appears. Windows users will again have to select All Files in the Files of type pop-up menu.

6. **Again choose Tutorial Data #1 and click Open.**

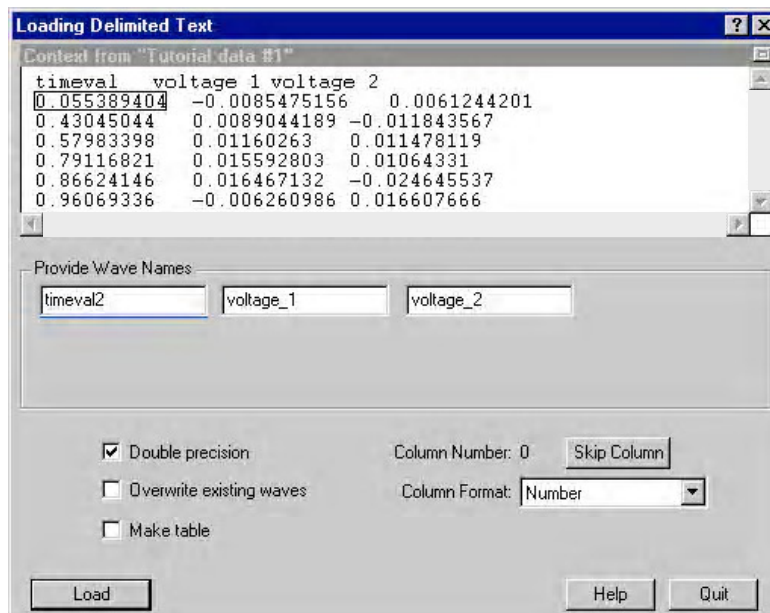
A Loading Delimited Text dialog will appear. Note that the name “timeval” is highlighted with a zebra pattern and an error message is shown. Observe that the names of the other two columns have been fixed up by replacing the spaces with underscore characters.

## Chapter I-2 — Guided Tour of Igor Pro

---

7. Change “timeval” to “timeval2”.

The dialog should now look like this:



8. Click the Make Table box to select it and then click Load.

The data is loaded and a new table is created to show the data.

9. Click the close box of the new table window.

A dialog will be presented asking if you want to create a recreation macro.

10. Click the No Save button.

The data we just loaded is still available in Igor. A table is just a way of viewing data and is not necessary for the data to exist.

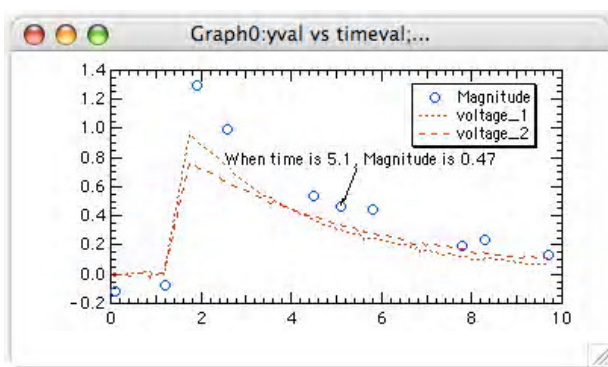
## Appending to a Graph

0. If necessary, click in Graph0 to bring it to the front.
1. Choose Append Traces to Graph from the Graph menu.  
A dialog very similar to the New Graph dialog appears.
2. In the Y Wave(s) list, select voltage\_1 and voltage\_2.
3. In the X Wave list, select timeval2.
4. Click Do It.

Two additional traces will be appended to the graph. Notice that they are also appended to the Legend.

5. **Position the cursor over one of the traces in the graph and double-click.**  
The Modify Trace Appearance dialog will appear with the trace you clicked on already selected.
6. **If necessary, select voltage\_1 in the list of traces.**
7. **Choose dashed line #2 from the Line Style pop-up menu.**
8. **Select voltage\_2 in the list of traces.**
9. **Choose dashed line #3 from the Line Style pop-up menu.**
10. **Click Do It.**

Your graph should now look like this:



### Offsetting a Trace

1. **Position the cursor directly over the voltage\_2 trace.**  
The voltage\_2 trace has the longest dash pattern.
2. **Press and hold the mouse button for about 1 second.**  
The cursor and annotations will disappear and the trace will move with the mouse.
3. **With the mouse button still down, press the shift key and drag the trace up about 1 cm and release.**  
The shift key constrains movement to vertical or horizontal directions.  
You have added an offset to the trace. If desired, you could add a tag to the trace indicating that it has been offset and by how much.

### Un-Offsetting a Trace



1. **Choose Undo Trace Drag from the Edit menu.**  
You can undo many of the interactive operations on Igor windows if you do so before performing the next interactive operation.

## Chapter I-2 — Guided Tour of Igor Pro

---

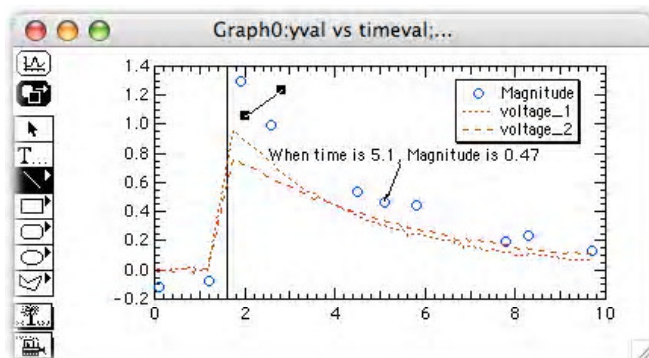
2. **Choose Redo Trace Drag from the Edit menu.**  
The following steps show how to remove an offset after it is no longer undoable.
3. **Double-click the voltage\_2 trace.**  
The Modify Trace Appearance dialog will appear with voltage\_2 selected. (If voltage\_2 is not selected, click it to select it.) The Offset checkbox will be selected.
4. **Deselect the Offset checkbox.**  
This cancels the offset.
5. **Select the Offset checkbox again.**  
A Waves Offset dialog will appear showing the offset value you introduced by dragging.
6. **Click the Cancel button or hit the escape key on your keyboard.**  
The Offset checkbox should still be deselected.
7. **Click Do It.**  
The voltage\_2 trace will be returned to its original position.


### Drawing in a Graph

1. **If necessary click in Graph0 to bring it to the front.**
2. **Choose Show Tools from the Graph menu or type command-T (Macintosh) or Ctrl+T (Windows).**  
A toolbar with just two icons showing will be added to the graph. The top icon will be selected indicating that the graph is in the normal (or “operate”) mode.
3. **Click this icon in the tool palette:**   
This puts the graph in drawing mode and activates the drawing tools.
4. **Press option (Macintosh) or Alt (Windows) and press and hold down the mouse button while the cursor is in the draw environment icon**  **(tree and grass).**  
A pop-up menu showing the available drawing layers and their relationship to the graph elements appears (the items in the menu are listed in back-to-front order).
5. **Choose UserBack from the menu.**  
We will be drawing behind the axes, traces and all other graph elements.
6. **Click the rectangle tool and drag out a rectangle starting at the upper-left corner of the plot area (y= 1.4, x=0 on the axes) and ending at the bottom of the plot area and about 1.5 cm in width (y= -0.2, x= 1.8).**

## Chapter I-2 — Guided Tour of Igor Pro

- Click the line tool and draw a line as shown, starting at the left (near the peak of the top trace) and ending at the right:



- Click in the draw environment icon and choose At Start from the Line Arrow item.
- Click in the Text tool icon .
- Click just to the right of the line you have just drawn.  
A dialog will appear.
- Type "Precharge".
- From the Anchor pop-up menu, choose "Left center".
- Click Do It.
- Click the graph's zoom box (Macintosh) or maximize button (Windows).  
Notice how the rectangle and line expand with the graph. Their coordinates are measured relative to the plot area (rectangle enclosed by the axes).
- Click the graph's zoom box (Macintosh) or restore button (Windows).
- Click the Arrow tool and then double-click the rectangle.  
A dialog showing the properties of the rectangle will appear.
- Enter 0 in the Line Thickness box.  
This turns off the frame of the rectangle.
- Choose Light Gray from the Fill Mode pop-up menu.
- Choose black from the Fore Color pop-up menu (the one under the Fill Mode pop-up).
- Click Do It.  
Observe that the rectangle forms a gray area behind the traces and axes.
- Again, double-click the rectangle.

## Chapter I-2 — Guided Tour of Igor Pro

---

22. **From the X Coordinate pop-up menu, choose Axis Bottom.**

The X coordinates of the rectangle will be measured in terms of the bottom axis — as if they were data values.

23. **Press tab until the X0 box is selected and type 0.**

24. **Tab to the Y0 box and type 0.**

25. **Tab to the X1 box and type 1.6.**

26. **Tab to Y1 and type 1.**

The X coordinates of the rectangle are measured in terms of the bottom axis and the left side will be at zero while the right side will be at 1.6. The Y coordinates are still measured relative to the plot area. Since we entered zero and one for the Y coordinates, the rectangle will span the entire height of the plot area.

27. **Click Do It.**

Notice the rectangle is nicely aligned with the axis and the plot area.

28. **Click the operate icon, , to exit the drawing mode.**

29. **Press option (Macintosh) or Alt (Windows), click in the middle of the plot area and drag about 2 cm to the right.**

The axes are rescaled. Notice that the rectangle moved to align itself with the bottom axis.

30. **Choose Undo from the Edit menu.**

### Making a Window Recreation Macro

1. **Click the graph's close box.**

Igor presents a dialog which asks if you want to save a window recreation macro. The graph's name is "Graph0" so Igor suggests "Graph0" as the macro name.

2. **Click Save.**

Igor generated a window recreation macro in the currently hidden procedure window. A window recreation macro contains the commands necessary to recreate a graph, table or page layout. You can invoke this macro to recreate the graph you just closed.

3. **Choose Procedure Window from the Windows menu.**

The procedure window is always present but is usually hidden to keep it out of the way. The window now contains the recreation macro for Graph0. You may need to scroll up to see the start of the macro. Because of the way it is declared, "Window Graph0() : Graph", this macro will be available from the Graph Macros submenu of the Windows main menu.

4. Click the procedure window's close box.

This hides the procedure window. Most other windows will put up a dialog asking if you want to kill or hide the window, but the built-in procedure window and the help windows simply hide themselves.

### Recreating the Graph

1. Choose Graph0 from the Graph Macros item in the Windows menu.  
The Graph0 macro was executed which created a graph of the same name.
2. Repeat step #1.  
The Graph0 macro was executed again but this time Igor gave the graph a slightly different name because a graph named Graph0 already existed.
3. Hold down the option (Macintosh) or Alt (Windows) key and click the close box of Graph0\_1.



The window is killed without presenting a dialog.

### Saving Your Work

1. Choose Save Experiment As from the File menu.
2. Navigate back to the folder where you saved the first time.
3. Change the name to "Tour #1 b" and click Save.

If you want to take a break, you can quit from Igor now.

### Graphically Editing Data


0. If you are returning from a break, open your "Tour #1 b" experiment and turn off preferences.
1. Adjust the positions of the graph and table so you can see both.  
If you have a small monitor or low monitor resolution, just make sure you can see the columns of data when the graph is the front window.
2. If necessary click in the Graph0 window to bring it to the front.
3. Click the drawing mode icon, , to activate the drawing tools.
4. Press option (Macintosh) or Alt (Windows), press and hold down the mouse button while the cursor is in the polygon icon, .

A pop-up menu will appear.

5. Choose Edit Wave from the menu.

## Chapter I-2 — Guided Tour of Igor Pro

---

6. **Click one of the open circles of the yval trace.**  
The trace is redrawn using lines and squares to show the location of the data points.
7. **Click the second square from the left and drag it 1 cm up and to the right.**  
Notice point 1 of yval and timeval changes in the table.
8. **Type command-Z (*Macintosh*) or Ctrl+Z (*Windows*) or choose Undo from the Edit menu.**
9. **Click midway between the first and second point and drag up 1 cm.**  
Notice a new point 1 of yval and timeval appears in the table.
10. **Press option (*Macintosh*) or Alt (*Windows*) and click the new data point with the tip of the lightning bolt.**  
You could also have typed command-Z (*Macintosh*) or Ctrl+Z (*Windows*) to undo the insertion.
11. **Press command (*Macintosh*) or Ctrl (*Windows*), click the line segment between the second and third point and drag a few cm to the right.**  
The line segment is moved and two points of yval and timeval are changed in the table.
12. **Type command-Z (*Macintosh*) or Ctrl+Z (*Windows*) or choose Undo from the Edit menu.**
13. **Click in the operate icon, , to exit drawing mode.**
14. **Choose Revert Experiment from the File menu and answer yes to the resulting dialog.**  
This is only necessary if you strayed from the above instructions.

### Making a Category Plot (Optional)

Category plots show continuous numeric data plotted against non-numeric text categories.

1. **Choose the New Table item from the Windows menu.**
2. **Click in the Do It button.**  
A new blank table will be created. We could have used the existing table but it is best to keep groups of related data together.
3. **Type “Monday” and then press enter.**  
A wave named “textWave0” was created with the text Monday as the value of the first point. Entering a non-numeric value in the first row of the first blank column automatically creates a new text wave.

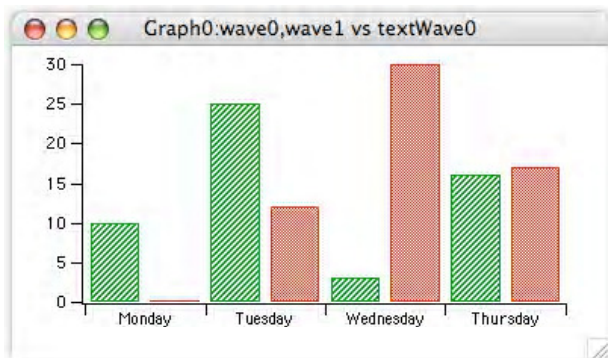
4. **Type the following, pressing enter after each one:**  
Tuesday  
Wednesday  
Thursday
5. **Click in the first cell of the next column and enter the following values:**  
10  
25  
3  
16
6. **Click in the first cell of the next column and enter the following values:**  
0  
12  
30  
17
7. **From the Windows menu, choose New and then Category Plot**  
A dialog similar to the New Graph dialog will appear. This dialog shows only text waves in the right-hand list.
8. **Click the From Target checkbox to select it.**  
This limits the list of waves to those in the target window. The target window is the table we just made.
9. **In the Y Wave(s) list, select both items.**  
Our text wave is already selected in the X Wave list.
10. **Click Do It.**  
A category plot will be created.
11. **Double-click one of the bars.**  
The Modify Trace Appearance dialog will appear.
12. **Using the “+Fill Type” pop-up menu, change the fills of each trace to any desired pattern.**  
You might also want to change the colors.

## Chapter I-2 — Guided Tour of Igor Pro

---

### 13. Click Do It.

Your graph should now look like this:



The remaining steps will explore various options. If you are not particularly interested in category plots, you can quit now, or at any point in the following steps, by closing the graph and table and skipping forward to the next section.

#### 1. **Again double-click one of the bars and, if necessary, select the top trace in the list.**

#### 2. **From the Grouping pop-up menu, choose Stack on next.**

You can use Igor Tips (*Macintosh*) or context-sensitive help (*Windows*) to see what each item does. On Macintosh turn tips on by pressing option-help, or if your keyboard has no help key, through the Help menu. On Windows click the question-mark icon in the top-right corner of the dialog.

#### 3. **Click Do It.**

The left bar in each group is now stacked on top of the right bar.

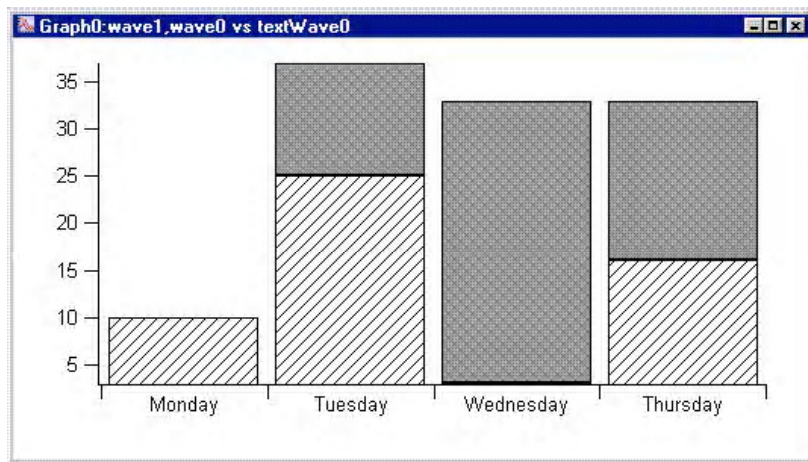
#### 4. **From the Graph menu, choose Reorder Traces.**

#### 5. **Reverse the order of the items in the list by clicking the top item to select it, then clicking it again and dragging it down. Then click Do It.**

The bars are no longer stacked and the bars that used to be on the left are now on the right. The reason the bars are not stacked is that the trace that we set to Stack on next mode is now last and there is no next trace.

6. Again using the Modify Trace Appearance dialog, set the top trace to Stack on next. Click Do It.

We could have avoided this extra trip to the modify traces dialog if we had initially set both traces to stack on next.



7. Enter the following values in the next blank column in the table:

7  
10  
15  
9

This creates a new wave named wave2.

8. Click in the graph to bring it to the front.
9. From the Graph menu, choose Append to Graph and then Category Plot.
10. In the Y list, select wave2 and click Do It.

The new trace is appended after the previous two. Since the 2nd trace was in Stack on next mode, the new trace is on the bottom of each set of three stacked bars.

11. Using the Modify Trace Appearance dialog, change the grouping mode of the middle trace to none.

Now the new bars are to the right of a group of two stacked bars. You can create any combination of stacked and side-by-side bars.

12. Double-click directly on the bottom axis.

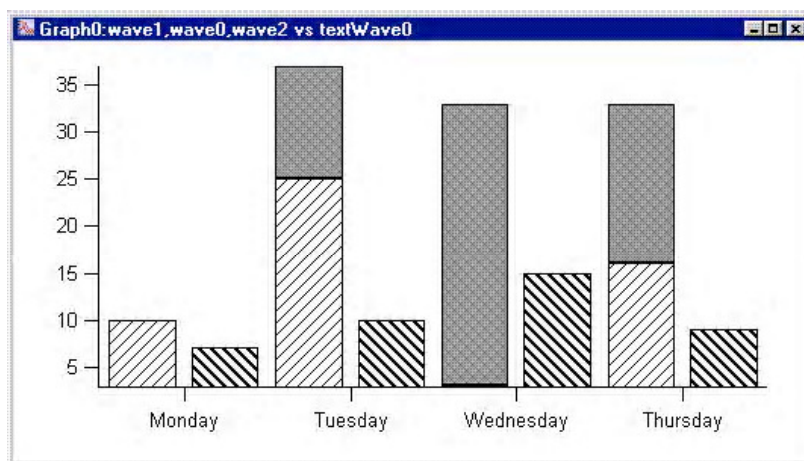
The Modify Axis dialog will appear with the bottom axis selected.

13. Click the Auto/Man Ticks tab.

## Chapter I-2 — Guided Tour of Igor Pro

14. Select the Tick In Center checkbox and then click Do It.

Notice the new positions of the tick marks.



15. Again double-click directly on the bottom axis.
16. Click the Axis tab.
17. Change the value of Bar Gap to zero and then click Do It.  
Notice that the bars within a group are now touching.
18. Use the Modify Axis dialog to set the Category Gap to 50%.  
The widths of the bars shrink to 50% of the category width.
19. Choose Modify Graph from the Graph menu.
20. Select the "Swap X & Y Axes" checkbox and then click Do It.  
This is how you create a horizontal bar plot.
21. Close both the graph and table windows without saving recreation macros.

### The Command Window

Parts of this tour make use of Igor's command line to enter mathematical formulae. Let's get some practice now.

Your command window should look something like this:



The command line is the space below the separator whereas the space above the separator is called the history area.

1. **Click in the command line, type the following line and press return or enter.**

```
Print 2+2
```

The result of the Print command as well as the command itself is placed in the history area.

2. **Tap the up arrow key on your keyboard.**

The line containing the print command is selected, skipping over the result printout line.

3. **Press return or enter.**

The selected line in the history is copied to the command line.

4. **Edit the command line so it matches the following and press return or enter.**

```
Print "The result is ",2+2
```

The Print command takes a list of numeric or string expressions, evaluates them and prints the results into the history. Let's see how you might have discovered that for yourself:

5. **Choose Igor Help Browser from the Help menu.**

The Igor Help Browser appears.

You can also display the help browser by pressing the help key (*Macintosh*) or F1 key (*Windows*), or by clicking the question-mark icon near the right edge of the command window.

6. **Click the Command Help tab in the Igor Help Browser.**

7. **Deselect the Functions and Programming checkboxes and select the Operations checkbox.**

A list of operations appears.

8. **From the pop-up menu next to the Operations checkbox, choose Printing.**

Those operations related to printing appear, including "Print".

9. **Click "Print".**

Reference material appears in the Help windoid.

10. **If necessary, resize the Igor Help Browser so you can see all of the help.**

11. **In the pop-up menu next to the Operations checkbox, choose About Waves.**

12. **Select PlaySound in the list.**

Tip: Click in the list to activate it and then type *p* to jump to PlaySound.

13. **Click the Help windoid, scroll down to the Examples section, and select the first four lines of example text (starting with "Make", ending with "sine-Sound").**

14. **Choose Copy from the Edit menu to copy the selection.**

15. **Close the Igor Help Browser.**

## Chapter I-2 — Guided Tour of Igor Pro

---

**16. Choose Paste from the Edit menu.**

All four lines are pasted into the command line area. You can view the lines using the miniature scroll arrows that appear at the right-hand edge of the command line.

**17. Hit the return or enter key.**

The four lines are executed and a short tone is heard. (*On Windows*, you may see an error message if your computer is not setup for sound.)

**18. Click once on the last line in the history area (PlaySound sineSound).**

The entire line (less the bullet) is selected just as if you pressed the arrow key.

**19. Press return or enter once to transfer the command to the command line and a second time to execute it.**

The tone is heard again as the line executes.

We are finished with the “sineSound” wave that was created in this exercise so let’s kill the wave to prevent it from cluttering up our wave lists.

**20. Choose Kill Waves from the Data menu.**

The Kill Waves dialog appears.

**21. Select “sineSound” and click Do It.**

The sineSound wave is removed from memory.

**22. Again click once on the history line “PlaySound sineSound”.**

**23. Press return or enter twice.**

An error dialog is presented because the sineSound wave no longer exists.

**24. In the error dialog, click Explain Operation**

A help window appears showing the reference text for the PlaySound operation.

**25. Click the close box of the help window.**

**26. Click OK to close the error dialog.**

**27. From the Edit menu, choose Clear Command Buffer or type command-K (*Macintosh*) or Ctrl+K (*Windows*).**

## Browsing Waves

**1. Choose Browse Waves from the Data menu.**

The Browse Waves dialog will appear. You can view the properties of the waves that are in memory and available for use in the current experiment and can also examine waves that are stored in individual binary files on disk.

**2. Click “timeval” in the list.**

The dialog shows the properties of the timeval wave.


3. **Click in the Wave Note area of the dialog.**

A wave note is text you can associate with a wave. You can not only view the note in this dialog but can also enter it. The other fields in this dialog are read-only.

4. **Type the following:**

This wave was created by typing data into a table.

5. **Turn Igor Tips on (Macintosh) or use context-sensitive help (Windows) to examine the dialog.**

*On the Macintosh*, Press option-help. *Under Windows*, click the ? icon in the top right corner of the dialog to get the  cursor and click an item to get help.

6. **Click the other waves in the list while observing their properties.**

7. **Turn Igor Tips off (Macintosh) and then click the done button to exit the dialog.**

8. **Type the following in the command line and press return or enter:**

```
Print "points=", numpts(timeval), ", note=", note(timeval)
```

The Print command takes a list of numeric or string expressions and prints the values into the history. numpts and note are built-in functions. Let's take a side-trip to see what they do:

9. **Press the help key (Macintosh) or the F1 key (Windows) to open the Igor Help Browser and then click the Command Help tab.**

10. **Deselect the Operations and Programming checkboxes and select the Functions checkbox. From the pop-up menu next to the Functions checkbox, choose About Waves.**

11. **In the list, find and click "numpts" and then "note".**

Descriptive text will be shown in the Help area.

12. **Close the Igor Help Browser.**

### Synthesizing Data

1. **Choose Make Waves from the Data menu.**

The Make Waves dialog will appear.

2. **Type "spiralY", tab and then "spiralX".**

3. **Change Rows to 1000.**

4. **Click Do It.**

Two 1000 point waves have been created. They are now part of the experiment but are not visible because we haven't put them in a table or graph.

5. **Choose Change Wave Scaling from the Data menu.**

6. **If a button labeled More Options is showing, click it.**

## Chapter I-2 — Guided Tour of Igor Pro

---

7. In the Wave(s) list, click spiralY and then shift-click (*Macintosh*) or Ctrl-click (*Windows*) spiralX.
8. Choose Start and Right for the SetScale Mode pop-up menu.
9. Enter 0 for Start and 50 for Right.
10. Click Do It.

A SetScale command is executed that tells Igor that the X values of the spiralX and spiralY waves go from 0 to 50 as the point number goes from 0 to 1000.

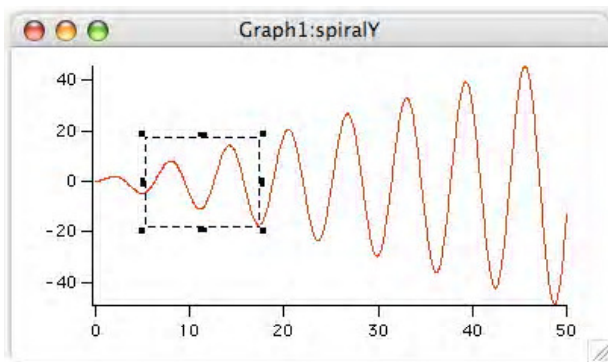
11. If necessary, click in the command window to bring it to the front.
12. Type the following in the command line and then press return or enter:

```
spiralY= x*sin(x); spiralX= x*cos(x)
```

The  $x$  above takes on 1000 values ranging from 0 to 50 as each expression is evaluated.


### Zooming & Panning

1. Choose New Graph from the Windows menu.
2. If necessary, deselect the From Target checkbox.
3. In the Y Wave(s) list, select "spiralY".
4. In the X Wave list, select "\_calculated\_".
5. Click Do It.
6. Position the cursor in the interior of the graph.  
The cursor changes to a dotted cross-hair shape.
7. Drag out a marquee as shown:



You can resize the marquee with the handles (black squares) and can move the marquee by dragging when the hand cursor is showing.

8. Position the cursor inside the marquee.

The mouse pointer changes to this shape:  , indicating that a pop-up menu is available.

9. **Click and choose Expand from the pop-up menu.**  
The axes are rescaled so that the area enclosed by the marquee fills the graph.
10. **Choose Undo Scale Change from the Edit menu or type command-Z (Macintosh) or Ctrl+Z (Windows).**
11. **Choose Redo Scale Change or type command-Z (Macintosh) or Ctrl+Z (Windows).**
12. **Press option (Macintosh) or Alt (Windows) and position the cursor in the middle of the graph.**  
The cursor changes to a hand shape.
13. **With the hand cursor showing, drag about 2 cm to the left.**
14. **Choose Autoscale from the Graph menu or type command-A (Macintosh) or Ctrl+A (Windows).**  
Continue experimenting with zooming and panning as desired.
15. **Type command-option-W (Macintosh) or Ctrl+Alt+W (Windows).**  
The graph is killed. The option (Macintosh) or Alt (Windows) key avoided the normal dialog asking whether to save the graph.

### Making a Graph with Multiple Axes

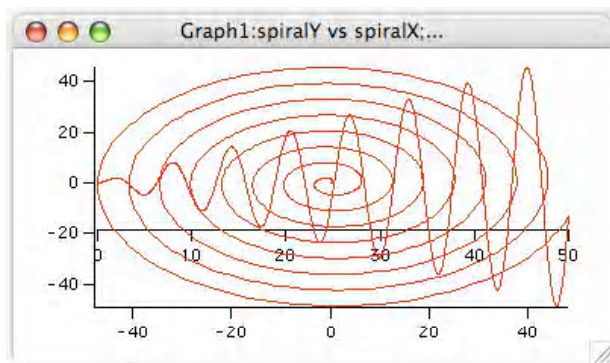
1. **Choose New Graph from the Windows menu.**
2. **If you see a button labeled More Choices, click it.**  
We will be using the more complex form of the dialog which allows you to create multi-axis graphs in one step.
3. **In the Y Wave(s) list, select "spiralY".**
4. **In the X Wave list, select "spiralX".**
5. **Click Add.**  
The above selections are inserted into the list in the center of the dialog.
6. **In the Y Wave(s) list, again select "spiralY".**
7. **In the X Wave list, select "\_calculated\_".**
8. **Choose New from the Axis pop-up menu under the X Wave(s) list.**
9. **Enter "B2" in the name box.**
10. **Click OK.**

## Chapter I-2 — Guided Tour of Igor Pro

---

**11. Click Do It.**

The following graph is created.



The interior axis is called a “free” axis because it can be moved relative to the plot rectangle. We will be moving it outside of the plot area but first we must make room by adjusting the plot area margins.

**12. Press the option (Macintosh) or Alt (Windows) key and position the cursor over the bottom axis until the cursor changes to this shape: .**

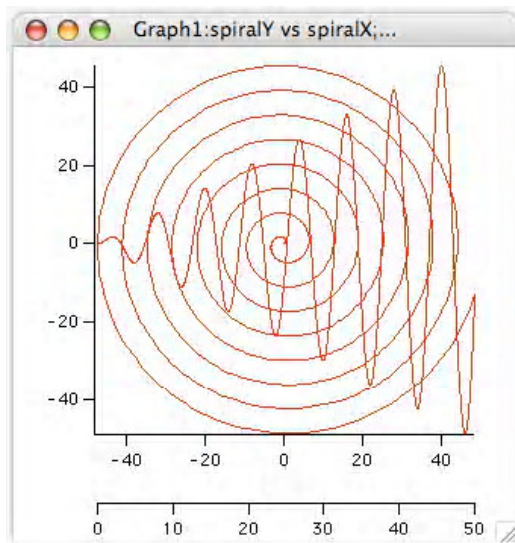
This shape indicates you are over an edge of the plot area rectangle and that you can drag that edge to adjust the margin.

**13. Drag the margin up about 2 cm. Release the option (Macintosh) or Alt (Windows) key.**

**14. Drag the interior axis down into the margin space you just created.**

**15. Resize the graph so the spiral is nearly circular.**




Your graph should now look like this:



### Saving Your Work

1. Choose **Save Experiment As** from the **File** menu.
2. Type **"Tour #1 c"** in the name box and click **Save**.  
If you want to take a break, you can quit from Igor now.

### Using Cursors

0. If you are returning from a break, open your **"Tour #1 c"** experiment and turn off preferences.
1. Click in the graph and choose **Show Info** from the **Graph** menu.  
A cursor info panel will appear at the bottom of the graph. If necessary, increase the width of the graph until you can see X: and Y: in the info panel.
2. Turn on **Igor Tips** (*Macintosh*) or use **context-sensitive help** (*Windows*) to examine the info panel.  
Macintosh shortcut: Press option-help. Windows: Press Shift-F1 to get the  cursor and click an item to get help.
3. Position the cursor over the name area for graph cursor A (the round one).  
The mouse pointer changes to this shape: , indicating that a pop-up menu is available.
4. Click and choose **"spiralY"** from the pop-up menu.  
The A cursor is placed on point zero of spiralY.
5. Repeat for cursor B but choose **"spiralY#1"** from the pop-up menu.  
The wave spiralY is graphed twice. The #1 suffix is used to distinguish the second copy from the first. It is #1 rather than #2 because in Igor, indices start from zero.
6. Position the mouse pointer over the center of the slide control bar .
7. Gently drag the slide bar to the right.  
Both cursors move to increasing point numbers. They stop when one or both get to the end.
8. Practice moving the slide bar to the left and right.  
Notice that the cursors move with increasing speed as the bar is displaced farther from the center.
9. Click once on the dock for cursor A (the round black circle).  
The circle turns white.
10. Move the slide bar to the left and right.  
Notice that only cursor B moves.

## Chapter I-2 — Guided Tour of Igor Pro

---

11. Click cursor B in the graph and drag it to another position on either trace.  
You can also drag cursors from their docks to the graph.
12. Click cursor A in the graph and drag it completely outside the graph.  
The cursor is removed from the graph and returns to its dock.
13. Choose Hide Info from the Graph menu.
14. Click in the command window, type the following and press return or enter.  

```
Print vcsr(B)
```

The y value at cursor B is printed into the history area. There are many functions available for obtaining information about cursors.
15. Click in the graph and then drag cursor B off of the graph.

### Removing a Trace and Axis

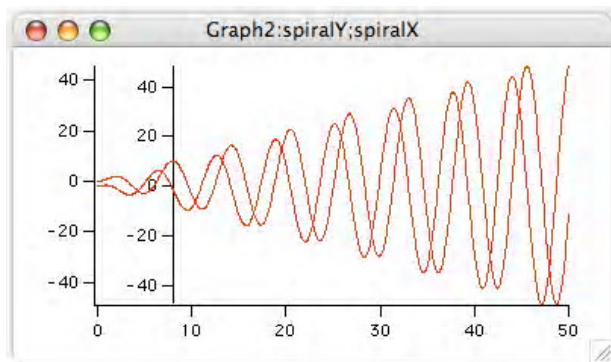
1. Choose Remove from Graph in the Graph menu.  
A dialog appears with spiralY listed twice. When we created the graph we used spiralY twice, first versus spiralX to create the spiral and second versus calculated X values to show the sine wave.
2. Click the second instance of spiralY and click Do It.  
The sine wave and the bottom-most (free) axis are removed. An axis is removed when its last trace is removed.
3. Drag the horizontal axis as far as it will go toward the bottom of the window.  
This returns the margin setting to auto. We had set it to a fixed position when we option-dragged (*Macintosh*) or Alt-dragged (*Windows*) the margin in a previous step.

### Creating a Graph with Stacked Axes

1. Choose New Graph from the Windows menu.
2. If you see a button labeled More Choices, click it.
3. In the Y Wave(s) list, select "spiralY".
4. In the X Wave list, select "\_calculated\_".
5. Click Add.
6. In the Y Wave(s) list, select "spiralX".
7. In the X Wave list, select "\_calculated\_".
8. Choose New from the Axis pop-up menu under the Y Wave(s) list.
9. Enter "L2" in the name box.
10. Click OK.

**11. Click Do It.**

The following graph is created.



In the following steps we will stack the interior axis on top of the left axis.

**12. Double-click the far left axis.**

The Modify Axis dialog will appear. If any other dialog appears, cancel and try again making sure the cursor is over the axis.

**13. Click the Axis tab.**

The Left axis should already be selected in the pop-up menu in the upper-left corner.

**14. Set the Left axis to draw between 0 and 45% of normal.**

**15. Choose L2 from the Axis pop-up menu.**

**16. Set the L2 axis to draw between 55 and 100% of normal.**

**17. Enter 0 in the Free Axis Position Distance box.**

This moves the L2 axis so it is in line with the Left axis.

**18. Choose Bottom from the Axis pop-up menu.**

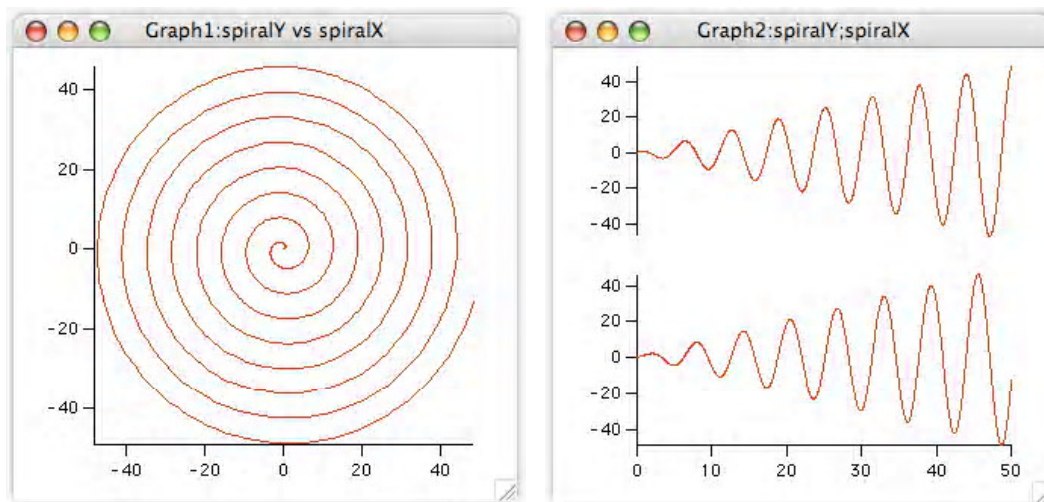
**19. Click the Axis Standoff checkbox to turn standoff off.**

**20. Click Do It.**



**21. Resize and reposition the top two graph windows so they are side-by-side and roughly square.**

## Chapter I-2 — Guided Tour of Igor Pro

22. The graphs should look like this:



### Appending to a Layout

1. Choose Layout0 from the Layouts item in the Windows menu.
2. Adjust the layout window size and scrolling so you can see the blank area below the graph that is already in the layout.
3. Click in the graph icon, , and choose "Graph1".  
Graph1 is added to the layout.
4. Again, click in the graph icon and choose "Graph2".  
Graph2 is added to the layout.
5. Click in the icon of a dashed box .
6. Drag out a marquee that fills the printable space under the original graph.
7. From the Layout menu, choose Arrange Objects.  
The Arrange Objects dialog appears.
8. Select both Graph1 and Graph2. Leave the Use Marquee box selected.
9. Click Do It.  
The two graphs are tiled inside the area defined by the marquee.
10. Choose Send Behind from the Control item in the Windows menu or press **cmd-E** (Macintosh) or **Ctrl+E** (Windows).

### Saving Your Work

1. Choose Save Experiment As from the File menu.

2. **Type “Tour #1 d” in the name box and click Save.**

If you want to take a break, you can quit from Igor now.

### Creating Controls

This section illustrates adding controls to an Igor graph — the type of thing a programmer might want to do. If you are not interested in programming, you can skip to the **End of the General Tour** on page I-52.

0. **If you are returning from a break, open your “Tour #1 d” experiment and turn off preferences.**
1. **Click in the graph with the spiral (Graph1) to bring it to the front.**
2. **Choose Show Tools from the Graph menu or type command-T (Macintosh) or Ctrl+T (Windows).**

A toolbar with just two icons showing is added to the graph. The top icon is selected indicating that the graph is in the normal (or operate) mode.

3. **Click the second icon.**  
This puts the graph in drawing mode and activates the drawing tools and the selector tool (arrow). The selector tool is used to select, move and resize controls.
4. **Choose Control Bar from the Add Controls item in the Graph menu.**  
The Control Bar dialog appears.
5. **Enter a height of 30 points and click Do It.**  
This reserves a space at the top of the graph for controls.
6. **Click in the command line, type the following and press return or enter.**

```
Variable ymult=1, xmult=1
```

This creates two numeric variables and sets both to 1.0.

7. **Click in the graph and then choose Add Set Variable from the Add Controls item in the Graph menu.**  
The SetVariable Control dialog appears.  
A SetVariable control provides a way to display and change the value of a variable.
8. **Choose ymult from the Value pop-up menu.**
9. **Set the high, low and increment values to 10, 0.1 and 0.1 respectively.**
10. **Enter a width of 80 (pixels) and click OK.**
11. **Click Do It.**

A SetVariable control attached to the variable ymult appears in the upper-left of the control bar.

## Chapter I-2 — Guided Tour of Igor Pro

---

12. Double-click the ymult control.

The SetVariable Control dialog appears.

13. Click the Duplicate button (in the bottom row).

14. Choose xmult as the variable.

15. Click Do It.

16. Choose Add Button from the Add Controls item in the Graph menu.

The Button Control dialog appears.

17. Enter "Update" in the Title box.

18. Click the New Procedure button.

The Control Procedure dialog appears.

19. Edit the procedure text so it looks like this:

```
Function ButtonProc(ctrlName) : ButtonControl
    String ctrlName

    Wave spiralsY, spiralsX
    NVAR ymult, xmult

    spiralsY= x*sin(ymult*x)
    spiralsX= x*cos(xmult*x)
End
```

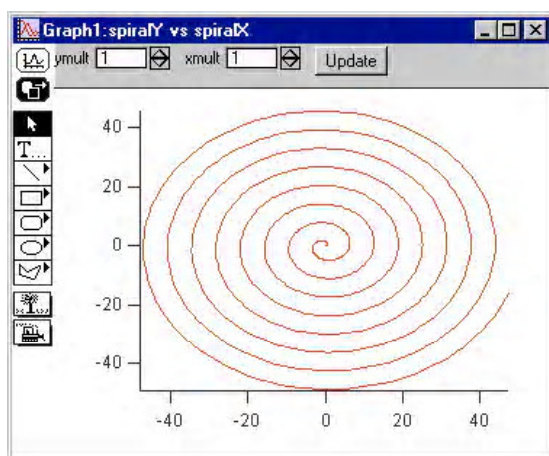
20. Click the Save Procedure Now button.

The Control Procedure dialog disappears and the text you were editing is inserted into the (currently hidden) procedure window.

21. Click Do It.

A Button control is added to the control bar.

The three controls are now functional but are not esthetically arranged.



22. Use the arrow tool to rearrange the three controls into a more pleasing arrangement. Expand the button so it doesn't crowd the text by dragging its handles.
23. Click the top icon in the tool palette to enter "operate mode".
24. Choose Hide Tools from the Graph menu or type command-T (*Macintosh*) or Ctrl+T (*Windows*).
25. Click the up-arrow in the ymult control.  
The value changes to 1.1.
26. Click the Update button.  
The spiralY and spiralX waves are recalculated according to the expressions you entered above and the graphs are updated.
27. Experiment with different ymult and xmult settings as desired.
28. Set both values back to 1 and click the Update button.  
You can use the tab key to select a value and then simply type a 1 on your keyboard followed by the return or enter key.

### Creating a Dependency

1. Click in the command window to bring it to the front.
2. Type the following lines pressing return or enter after each:  

```
spiralY := x*sin(ymult*x)  
spiralX := x*cos(xmult*x)
```

This is exactly what you entered before except here := is used in place of =. These "recalculation expressions" create dependencies. For instance, the wave spiralY is dependent on the variable ymult. If a new value is stored in ymult then the values in spiralY are automatically recalculated from the expression.
3. Click in the graph with the spiral (Graph1) to bring it to the front.
4. Adjust the ymult and xmult controls but do not click the Update button.  
The spiralY and spiralX waves are recalculated automatically and both graphs are updated.
5. On the command line, type the following and press return or enter:  

```
ymult := 3*xmult
```

Note that the ymult SetVariable control as well as the graphs are updated.
6. Adjust the xmult value.  
Again notice that ymult as well as the graphs are updated.

## Chapter I-2 — Guided Tour of Igor Pro

---

**7. Choose Object Status from the Misc menu.**

The Object Status dialog will appear. This dialog allows you to examine Igor objects that might otherwise have no visual representation such as string and numeric variables.

**8. Click the “The Current Object” pop-up menu and choose spiralY from the Dependent Objects item (*Macintosh*) or drop-down list (*Windows*).**

The list on the right indicates that spiralY depends on the variable ymult.

**9. Double-click the ymult entry in the right hand list.**

ymult becomes the current object. The list on the right now indicates that ymult depends on xmult.

**10. Click the Delete Formula button.**

Now ymult no longer depends on xmult.

**11. Click Done.**

**12. Adjust the xmult setting.**

The ymult value is no longer automatically recalculated but the spiralY and spiralX waves still are.

**13. Click the Update button.**

**14. Adjust the xmult and ymult settings.**

The spiralY and spiralX waves are no longer automatically recalculated. This is because the ButtonProc function called by the Update button does a normal assignment using = rather than := and that action removes the automatic recalculation expression.

*Note:* In real work, you should avoid the kind of multilevel dependencies that we created here because they are too confusing.

## Saving Your Work

- 1. Choose Save Experiment As from the File menu.**
- 2. Type “Tour #1 e” in the name box and click Save.**

## End of the General Tour

This is the end of the general tour of Igor.

If you want to take a break, you can quit from Igor now.

### Analysis Tour #1

In this section we will concentrate on the data analysis features of Igor Pro. We will generate synthetic data and then manipulate it using sorting and curve fitting.

#### Launching Igor Pro

1. **Double-click the Igor Pro application file on your hard disk.**  
If Igor was already running, choose New Experiment from the File menu.
2. **Use the Misc menu to turn preferences off.**

#### Creating Synthetic Data

We need something to analyze, so we generate some random x values and create some y data from a math function.

1. **Type the following in the command line and then press return or enter:**  

```
SetRandomSeed 0.1
```

This initializes the random number generator so you will get the same results as this tutorial.
2. **Type the following in the command line and then press return or enter:**  

```
Make/N=100 fakeX=enoise(5)+5,fakeY
```

This generates two 100 point waves and fills fakeX with evenly distributed random values ranging from 0 to 10.
3. **Execute this in the same way:**  

```
fakeY= exp(-(fakeX-4)^2)+gnoise(0.1)
```

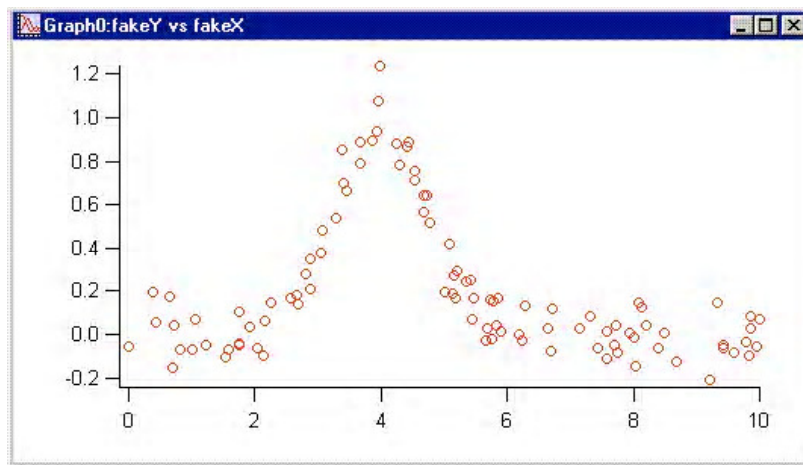
This generates a Gaussian peak centered at 4.
4. **Choose New Graph from the Windows menu.**
5. **In the Y Wave(s) list, select “fakeY”.**
6. **In the X Wave list, select “fakeX”.**
7. **Click Do It.**  
The graph will be a rat’s nest of lines because the X values are not sorted.
8. **Double-click in the center of the graph.**  
The Modify Trace Appearance dialog appears.
9. **From the Mode pop-up choose Markers.**
10. **From the pop-up menu of markers choose the open circle.**

## Chapter I-2 — Guided Tour of Igor Pro

---

**11. Click Do It.**

Now the graph makes sense.



### Curve Fitting to a Gaussian

The synthetic data was generated using a Gaussian function so let's try to extract the original parameters.

**1. From the Analysis menu, choose Curve Fitting.**

The curve fitting dialog appears.

**2. Click the Function and Data tab.**

**3. From the Function pop-up menu, choose gauss.**

**4. From the Y Data pop-up menu, choose fakeY.**

**5. From the X Data pop-up menu, choose fakeX.**

**6. Click the Data Options tab.**

The Weighting pop-up should read “\_none\_”.

**7. Click the Output Options tab.**

The Destination pop-up should read “\_auto\_” and Residual should read “\_none\_”.

**8. Click Do It.**

The curve fit starts, progresses through a few passes and waits for you to click OK.

**9. Click OK.**

The curve fit results are printed in the history.

The amplitude parameter (A) should be 1.0 and the position (x0) should be 4.0. We got  $0.99222 \pm 0.0299$  for the amplitude and  $3.9997 \pm 0.023$  for the position.

### Sorting

In the next section we will be curve fitting to a subrange of the data. For this to work, the data must be sorted by x values.

1. **Double-click one of the open circle markers.**

The Modify Traces Appearance dialog appears with fakeY selected. If fakeY is not selected, click it.

2. **From the Mode pop-up choose Lines between points and click Do It.**

The fakeY trace reverts to a rat's nest of lines.

3. **From the Analysis menu, choose Sort.**

The Sorting dialog appears.

4. **If necessary choose Sort from the Operation pop-up menu.**

5. **Select "fakeX" in the "Key Wave" list and both "fakeX" and "fakeY" in the "Waves to Sort" list.**

6. **Click Do It.**

The rat's nest is untangled. Since we were using the lines between points mode just to show the results of the sort, we now switch back to open circles but in a new way:

7. **Press the control key and click (Macintosh) or right-click (Windows) on the fakeY trace.**

A pop-up menu appears with the name of the trace at the top. If it is not "Browse fakeY" try again.

8. **Choose Markers from the Mode item.**

### Fitting to a Sub-Range

Here we will again fit our data to a Gaussian but using a subset of the data. We will then extrapolate the fit outside of the initial range.

1. **Choose Show Info from the Graph menu.**

A cursor info panel is appended to the bottom of the graph.

2. **Place cursor A (the round one) on the fakeY trace and move to it point #14.**

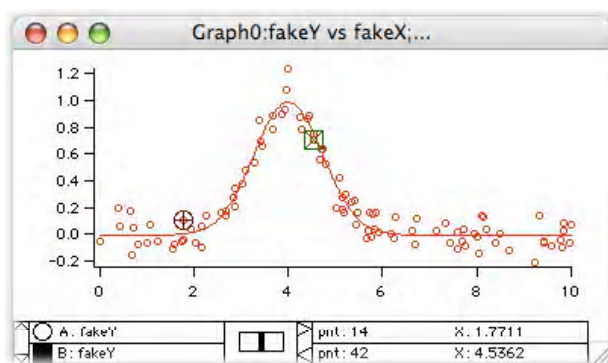
You can use the pop-up menu in the name area or just drag the cursor from the dock to the trace. Tip: if the slider moves the cursor too fast, click the area on either side or use your keyboard's arrow keys to move more slowly.

3. **Click the dock for cursor A to deselect it**

This is so you can adjust cursor B without affecting the position of cursor A.

## Chapter I-2 — Guided Tour of Igor Pro

4. Place cursor B (the square one) on the fakeY trace and move it to point #42. Your graph should look like this:



5. From the Analysis menu, choose Curve Fitting and then click the Function and Data tab.

The curve fitting dialog appears and the settings should be as you left them. Check that the function type is gauss, the y data is fakeY, the x data is fakeX.

6. Click the Data Options tab.
7. Click the Cursors button in the Range area.

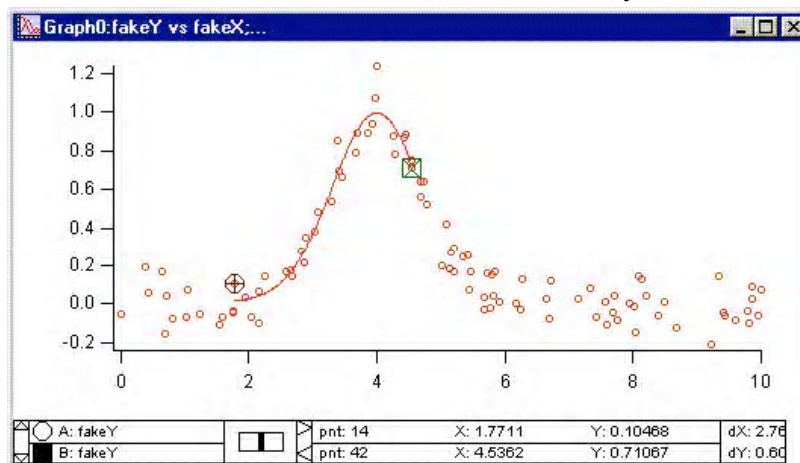
This puts the text "pcsr(A)" and "pcsr(B)" in the range entry boxes.

8. Click Do It.

The curve fit starts, does a few passes and waits for you to click OK.

9. Click OK.

Note that the auto trace curve is evaluated only over the subrange.



### Extrapolating a Fit

When we chose “\_auto\_” from the Destination pop-up menu in the curve fit dialog, Igor created a wave named fit\_fakeY to show the fit results. This is just an ordinary wave whose X scaling is set to the extent of the X values used in the fit.

To extrapolate, we simply change the X scaling and reexecute the fit equation.

1. **Choose Change Wave Scaling from the Data menu.**
2. **If you see a button labeled More Options, click it.**
3. **From the SetScale Mode pop-up menu, choose Start and End.**
4. **Double-click “fit\_fakeY” in the list.**

This reads in the current settings. The starting X value will be about 1.77 and the ending X will be about 4.53.

5. **Type tab until the Start box is selected and enter 1.0.**
6. **Tab to the End box and type 8.0.**
7. **Click Do It**

The fit\_fakeY trace is stretched out and now runs between 1 and 8.

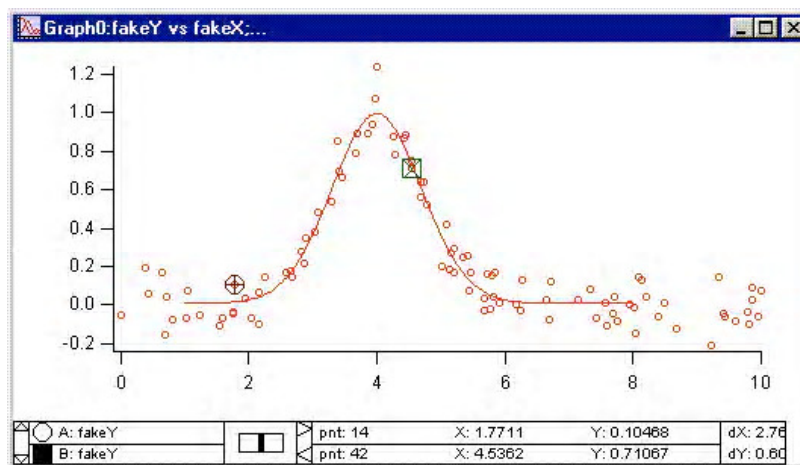
Now we need to recalculate the fit expression.

8. **In the history, find the line that starts “fit\_fakeY=” and click it.**

The entire line is selected. (The line in question is a few lines after the curve fit command.)

9. **Press return or enter once to copy the selection from the history to the command line and a second time to execute it.**

The fit\_fakeY wave now contains valid data between 1 and 8.



## Chapter I-2 — Guided Tour of Igor Pro

### Appending a Fit

When we chose “\_auto\_” from the Destination pop-up menu in the curve fit dialog, Igor created a wave named fit\_fakeY to show the fit results. If we want to show the results of several fits to the same data, we will have to somehow protect the fit wave from being overwritten. This is done by simply renaming it.

1. Choose **Rename** from the **Data** menu.
2. Select the wave named **fit\_fakeY** in the list.
3. Change the name to “**gaussFit\_fakeY**” and click **Do It**.
4. Position the **A** and **B** cursors to point numbers 35 and 65, respectively.

*Tip:* click in the dock for a given cursor to enable/disable its being moved by the slide control. Click to either side of the central slide or use the arrow keys on your keyboard to move the cursor(s) one point number at a time.

5. From the **Analysis** menu, choose **Curve Fitting** and click the **Function and Data** tab.

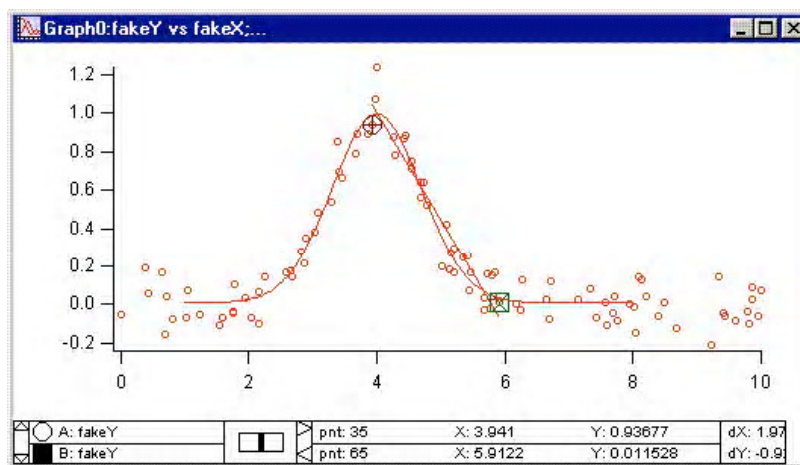
The curve fitting dialog appears and the settings should be as you left them.

6. Change the function type from **gauss** to **line**.

Verify that Y data is fakeY and X data is fakeX.

7. Click **Do It**.

The curve fit is performed without a dialog since the line fit is not iterative.



This concludes Analysis Tour #1.

### Analysis Tour #2

In this tour we will explore the Histogram operation and will perform a curve fit using weighting. The optional last portion creates a residuals plot and shows you how to create a useful procedure from commands in the history.

#### Launching Igor Pro

1. **Double-click the Igor Pro application file on your hard disk.**  
If Igor was already running, choose New Experiment from the File menu.
2. **Use the Misc menu to turn preferences off.**

#### Creating Synthetic Data

As in Tour #1, we need something to analyze, so we generate some random values.

1. **Type the following in the command line and then press return or enter:**  

```
SetRandomSeed 0.1
```

This initializes the random number generator so you will get the same results as this tutorial.
2. **Type the following in the command line and then press return or enter:**  

```
Make/N=10000 fakeY= enoise(1)
```

This generates a 10,000 point wave filled with evenly distributed random values ranging from -1 to 1.

#### Histogram of White Noise

Here we will generate a histogram of the evenly distributed “white” noise.

1. **Type the following in the command line and then press return or enter:**  

```
Make/N=100 histData
```

This is where the histogram output will be stored.
2. **Choose Histogram from the Analysis menu.**  
The Histogram dialog appears.
3. **Select fakeY in the Source Wave list.**
4. **Choose Select Existing Wave from the Output Type pop-up menu. Then select histData in the list on the right.**
5. **Click the Auto-set bins radio button.**
6. **Click the Do It button.**  
The mathematical histogram operation is performed. Now we need to display the results.

## Chapter I-2 — Guided Tour of Igor Pro

---

7. Choose New Graph from the Windows menu.
8. Select histData in the Y Wave(s) list and “\_calculated\_” in the X list.
9. Click the Do It button.

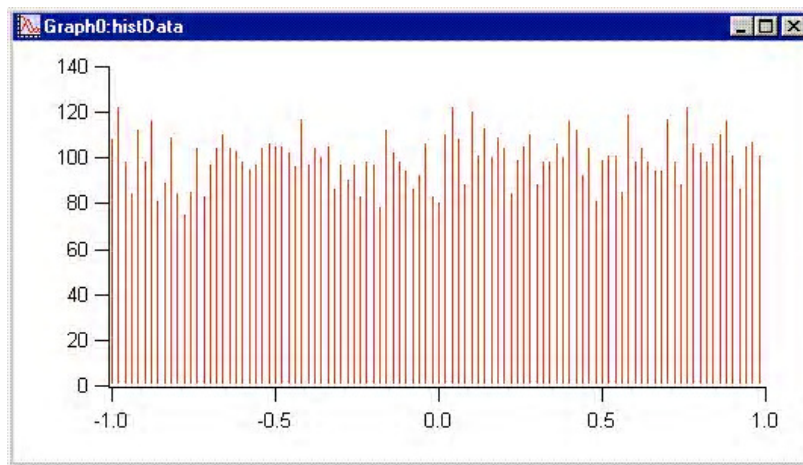
A graph is created showing the histogram results. We need to touch it up a bit.
10. Double-click the trace in the graph.

The Modify Trace Appearance dialog appears.
11. Choose “Sticks to zero” from the Mode pop-up menu and click Do It.

A graph is redrawn using the new display mode.
12. Double-click one of the tick mark labels (e.g., “100”) of the left axis.

The Modify Axis dialog appears, showing the Axis Range tab.
13. From the two pop-up menus in the When Autoscaling area, choose “Round to nice values” and “Autoscale from zero”.
14. Choose Bottom from the Axis pop-up menu.
15. From the two pop-up menus in the When Autoscaling area, choose “Round to nice values” and “Symmetric about zero”.
16. Click the Do It button.

Your graph should now look like this:



### Histogram of Gaussian Noise

1. Type the following in the command line and then press return or enter:

```
fakeY= gnoise(1)
```
2. Choose Histogram from the Analysis menu.

The dialog should still be correctly set up from the last time.

**3. Click the Do It button.**

The histogram operation is performed and the graph of histData is updated. Note the roughly Gaussian shape.

### Curve Fit of Histogram Data

A histogram operation counts the number of source values that fall within a given destination bin. The expected standard deviation for count data is roughly the square root of the number of counts — at least when the number of counts is substantial. We will use this property to provide weighting for a curve fit to a Gaussian function.

**1. Choose Duplicate Waves from the Data menu.**

**2. Choose histData from the Template pop-up menu.**

**3. In the first two Names boxes, enter “histWeight” and “histResids”.**

**4. Click Do It.**

The histWeight wave will be used to provide weighting for the curve fit and the histResids wave will be used later when we calculate the residuals of the fit.

In general, you should use the Duplicate operation when you need new waves with the same number of points and the same data type as a particular wave.

**5. Type the following in the command line and then press return or enter:**

```
histWeight= sqrt(histData)
```

The weighting wave should contain the standard deviation of the data.

**6. Choose Curve Fitting from the Analysis menu.**

**7. Click the Function and Data tab and choose gauss from the Function menu.**

**8. Choose histData from the Y Data menu.**

**9. Leave the X Data pop-up menu set to “\_calculated\_”.**

**10. Click the Data Options tab and click the Clear button in the Range section.**

**11. Choose histWeight from the Weighting pop-up menu.**

**12. Just under the Weighting pop-up menu there are two radio buttons. Click the top one which is labelled “Standard Deviation”.**

**13. Click the Output Options tab and choose “\_auto\_” from the Destination pop-up.**

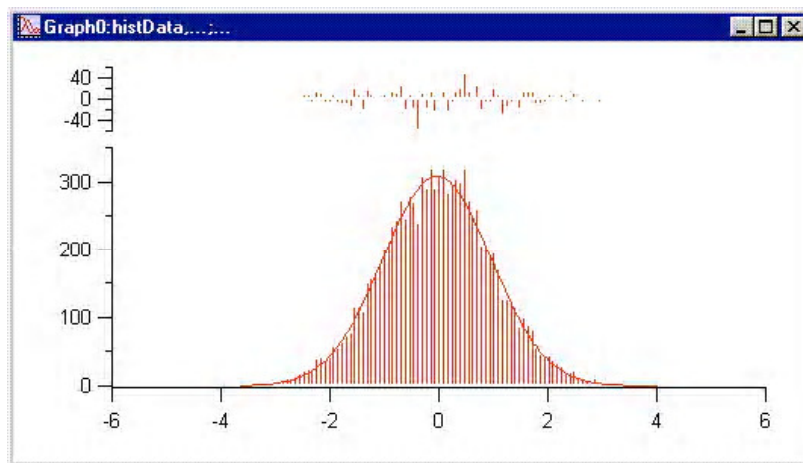
**14. Set the Residual pop-up menu to “\_auto trace\_”.**

Igor will automatically calculate the residuals and add them with the curve fit to our graph.

## Chapter I-2 — Guided Tour of Igor Pro

### 15. Click Do It.

The curve fit starts, does a few passes, and waits for you to click OK.



(This is a quick and dirty curve fit to a histogram. In real life, you would do some extra steps to increase accuracy. See **Curve Fitting to a Histogram** on page III-175 for details.)

## Curve Fit Residuals (Optional)

This section and the next one are primarily of interest to people who want to use Igor programming to automate tasks.

In the next section, as an illustration of how the history area can be used as a source of commands to generate procedures, we will create a procedure that appends residuals to a graph. The preceding section illustrated that Igor is able to automatically display residuals from a curve fit, so the procedure that we write in the next section is not needed. Still, it demonstrates the process of creating a procedure. In preparation for writing the procedure, in this section we append the residuals manually.

If the curve fit to a Gaussian function went well and if the gnoise function truly produces noise with a Gaussian distribution, then a plot of the difference between the histogram data and the fitted function should not reveal any curvature.

0. To remove the automatically generated residual from the Gaussian fit in the previous section, control-click (*Macintosh*) or right-click (*Windows*) directly on the residual trace at the top of the graph and select **Remove Res\_histData** from the pop-up menu.
1. In the history area of the command window, find the line that reads:  

```
fit_histData= W_coef[0]+ W_coef[1]*exp(-(x-W_coef[2])/W_coef[3])^2)
```
2. Click once on the line to select it and then press return or enter once.  
The line is transferred to the command line.

3. **Edit the line to match the following (all on one line):**

```
histResids= histData - ( W_coef[0]+  
                        W_coef[1]*exp(-(x-W_coef[2])/W_coef[3])^2) )
```

In other words, change “fit\_histData” to “histResids”, click after the equals and type “histData - (” and then add a “)” to the end of the line.

**Note:** if the fit had used an X wave rather than calculated X values then it would have been necessary to substitute the name of the X wave for the “x” in the expression.

4. **Press return or enter.**

The wave assignment command calculates the difference between the measured data and the fitted values.

Now we will append the residuals to the graph stacked above the current contents.

5. **Choose Append Traces to Graph from the Graph menu.**
6. **Select histResids from the Y wave(s) list and “\_calculated\_” from the X wave list.**
7. **Choose New from the Axis pop-up menu under the Y Wave(s) list.**
8. **Enter “Lresid” in the Name box and click OK.**
9. **Click Do It.**

The new trace and axis is added.

10. **Double-click the far-left axis.**

The Modify Axis dialog will appear. If any other dialog appears, cancel and try again making sure the cursor is over the axis.

11. **Click the Axis tab.**

The Left axis should already be selected in the pop-up menu in the top/left corner of the dialog.

12. **Set the Left axis to draw between 0 and 70% of normal.**
13. **Choose Lresid from the Axis pop-up menu.**
14. **Set the Lresid axis to draw between 80 and 100% of normal.**
15. **Enter 0 in the “Free axis position:” “Distance:” box.**

This moves the Lresid axis so it is in line with the Left axis.

16. **Choose Bottom from the Axis pop-up menu.**
17. **Click the Axis Standoff checkbox to turn standoff off.**

Just a couple more touch-ups and we will be done. The ticking of the Lresid axis can be improved. The residual data should be in dots mode.

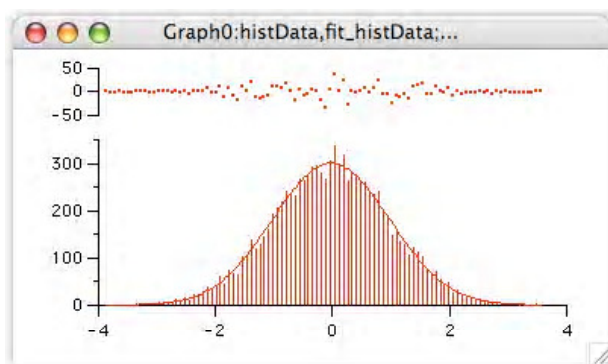
18. **Choose Lresid from the Axis pop-up menu again.**
19. **Click the Auto/Man Ticks tab.**

## Chapter I-2 — Guided Tour of Igor Pro

---

20. Change the Approximately value to 2.
21. Click the Axis Range tab.
22. In the When Autoscaling area, choose “Symmetric about zero” from the menu currently reading “Zero isn’t special”.
23. Click the Do It button.
24. Double-click the histResids trace.  
The Modify Trace Appearance dialog appears with histResids already selected in the list.
25. Choose Dots from the Mode pop-up menu
26. Set the line size to 2.00.
27. Click Do It.

Your graph should now look like this:



### Writing a Procedure (Optional)

In this section we will collect commands that were created as we appended the residuals to the graph. We will now use them to create a procedure that will append a plot of residuals to a graph.

1. Click the zoom icon (*Macintosh*) or the maximize button (*Windows*) of the command window to enlarge it to fill the screen.
2. Find the line that reads:  

```
·AppendToGraph/L=Lresid histResids
```

This is the 5th line from the bottom.
3. Select this and the remaining lines and press command-C (*Macintosh*) or Ctrl+C (*Windows*) to copy them to the clipboard.
4. Click the zoom icon (*Macintosh*) or the restore button (*Windows*) of the command window to return it to its normal size.
5. Choose Procedure from the New item in the Windows menu.

6. **Type Append Residuals in the Document Name box and click New.**

A new procedure window will appear. We could have used the always-present built-in procedure window, but we will save this procedure window as a stand-alone file.

7. **Add a blank line to the window and then type Function AppendResiduals() and press return or enter.**
8. **Type command-V (Macintosh) or Ctrl+V (Windows) to paste the commands from the history into the new window.**
9. **Type “End” and press return or enter.**
10. **Select the five lines that you pasted into the procedure window and then choose Adjust Indentation from the Edit menu.**

This removes the bullet characters from the history and prepends tabs to apply the normal indentation for procedures.

If you are running on an Asian-language system, you will have asterisks at the start of each line and you must remove them manually.

Your procedure should now look like this, except that the two ModifyGraph commands will be all on one line:

```
Function AppendResiduals()  
    AppendToGraph/L=Lresid histResids  
    ModifyGraph nticks(Lresid)=2,standoff(bottom)=0,  
        axisEnab(left)={0,0.7};DelayUpdate  
    ModifyGraph axisEnab(Lresid)={0.8,1},  
        freePos(Lresid)=0;DelayUpdate  
    SetAxis/A/E=2 Lresid  
    ModifyGraph mode(histResids)=2,lsize(histResids)=2  
End
```

11. **Delete the “;DelayUpdate” at the end of the two ModifyGraph commands.**

DelayUpdate has no effect in a function.

We now have a nearly functional procedure but with a major limitation — it only works if the residuals wave is named “histResids”. In the following steps, we will change the function so that it can be used with any wave and also with an XY pair, rather than just with equally-spaced waveform data.

12. **Convert the first two lines of the function to match the following:**

```
Function AppendResiduals(ywave,xwave)  
    String ywave,xwave  
  
    if (CmpStr("_calculated_",xwave) == 0)  
        AppendToGraph/L=Lresid $ywave  
    else  
        AppendToGraph/L=Lresid $ywave vs $xwave  
    endif
```

## Chapter I-2 — Guided Tour of Igor Pro

---

13. In the last `ModifyGraph` command in the function, change both “`histResids`” to “`$ywave`”.

The “\$” character is a signal to Igor to take the string expression that follows and convert it into the name of an Igor object.

Here is the completed procedure. (Again, the first `ModifyGraph` command will be all on one line.)

```
Function AppendResiduals(ywave,xwave)
    String ywave,xwave

    if (CmpStr("_calculated_",xwave) == 0)
        AppendToGraph/L=Lresid $ywave
    else
        AppendToGraph/L=Lresid $ywave vs $xwave
    endif
    ModifyGraph nticks(Lresid)=2,standoff(bottom)=0,
        axisEnab(left)={0,0.7}
    ModifyGraph axisEnab(Lresid)={0.8,1},freePos(Lresid)=0
    SetAxis/A/E=2 Lresid
    ModifyGraph mode($ywave)=2,lsize($ywave)=2
End
```

Let's try it out.

14. Click the **Compile** button at the bottom of the procedure window to make Igor compile the function.
- If you get an error, edit the function text to match the listing above.
15. From the **Windows** menu, choose **New Graph**.
16. Choose **histData** from the **Y Wave(s)** list and click **Do It**.
17. In the command line, execute the following command:

```
AppendResiduals("histResids", "_calculated_")
```

The `AppendResiduals` function should display the residuals in the graph, above the original histogram data.

Next, we will add a function that displays a dialog so we don't have to type wave names into the command line.

18. Use the **Windows** menu, **Other Windows** submenu, to open the **Append Residuals** procedure window.

19. Enter the following function, below the end of the AppendResiduals function.

```
Function AppendResidualsDialog()  
    String ywave,xwave  
  
    Prompt ywave,"Residuals Data",popup WaveList("",";",",")  
    Prompt xwave,"X Data",popup "_calculated_";"+WaveList("",";",",")  
    DoPrompt "Append Residuals", ywave, xwave  
    if (V_flag != 0)  
        return -1;        // User canceled.  
    endif  
  
    AppendResiduals(ywave,xwave)  
End
```

This function will display a dialog to get parameters from the user and will then call the AppendResiduals function.

Let's try it out.

20. **Click the Compile button at the bottom of the procedure window to make Igor compile our function.**

If you get an error, edit the function text to match the listing above.

21. **Type command-E (*Macintosh*) or Ctrl+E (*Windows*) to send the procedure window to the back, and then activate the graph.**

22. **Control-click (*Macintosh*) or right-click (*Windows*) on the residual trace at the top of the graph and select Remove histResids from the pop-up menu.**

The axis displaying histData will stay short because the residuals were not appended to the graph automatically. Don't worry about it.

23. **In the command line, execute the following command:**

```
AppendResidualsDialog()
```

The AppendResidualsDialog function displays a dialog to let you choose parameters.

24. **Choose histResids from the Residuals Data pop-up menu.**

25. **Leave the X Wave pop-up set to "\_calculated\_".**

26. **Click Continue.**

The graph should once again contain the residuals plotted on a new axis above the main data.

Next we will add a menu item to the Macros menu.

27. **Use the Windows menu, Other Windows submenu, to open the Append Residuals procedure window.**

## Chapter I-2 — Guided Tour of Igor Pro

---

28. Enter the following above the AppendResiduals function:

```
Menu "Macros"  
    "Append Residuals...", AppendResidualsDialog()  
End
```

29. Click the compile button.

Igor compiles the function and adds the menu item to the Macros menu.

30. Type command-E (*Macintosh*) or Ctrl+E (*Windows*) to send the procedure window to the back, and then activate the graph.
31. Control-click (*Macintosh*) or right-click (*Windows*) on the residual trace at the top of the graph and select "Remove histResids" from the pop-up menu.
32. Click the Macros menu and choose the "Append Residuals" item
33. Choose histResids from the Residuals Data pop-up menu.
34. Leave the X Wave pop-up menu set to "\_calculated\_".
35. Click the Continue.

The graph should once again contain the residuals plotted on a new axis above the main data.

### Saving a Procedure File (Optional)

Now that we have a working procedure, let's save it so it can be used in the future.

1. In the Finder (*Macintosh*) or Explorer (*Windows*) create a folder named "Igor Shared Procedures".

Create this folder outside of the Igor Pro Folder. Keeping your personal files outside the Igor Pro Folder will make it easier to upgrade Igor in the future.

2. Activate the Append Residuals procedure window again.
3. Choose Save Procedure As from the File menu.
4. Enter the file name "Append Residuals.ipf".
5. Navigate to your Igor Shared Procedures folder and click Save.

The Append Residuals procedure file is now saved in a stand-alone file.

There are several ways to open the procedure file to use it in the future. One is to double-click it. Another is to choose Procedure from the Open File item in the File menu. A third is to put a #include statement in the built-in procedure window.

6. Click the close box on the procedure window.

Igor will ask if you want to kill or hide the file. Click Kill. This removes the file from the current experiment, but it still exists on disk.

### Including a Procedure File (Optional)

The preferred way to open a procedure window is to use a `#include` statement. This section demonstrates how to do that.

1. **On the desktop, create an alias (*Macintosh*) or shortcut (*Windows*) for your Igor Shared Procedures folder, and drag it into the User Procedures folder inside the Igor Pro folder.**

This allows you to open your procedure file using an include statement, as we will see shortly.

2. **Back in Igor, use the Windows menu to open the built-in procedure window.**
3. **At the top of the built-in procedure window, notice the line that says:**

```
#pragma rtGlobals = 1
```

This is technical stuff that you can ignore.

4. **Under the `rtGlobals` line, leave a blank line and then enter:**

```
#include "Append Residuals"
```

5. **Click the compile button at the bottom of the built-in procedure window.**

Igor compiles the procedure window. When it sees the `#include` statement, it opens the `Append Residuals.ipf` procedure file, but it is hidden.

6. **Use the Windows→Other Windows submenu, verify that the Append Residuals procedure file is in fact open.**

To remove the procedure file from the experiment, you would remove the `#include` statement.

Using the `#include` statement allows you to change the organization of your procedure files without affecting Igor experiments that access them.

This concludes Analysis Tour #2.

### For Further Exploration

We developed the guided tours in this chapter to provide an overview of the basics of using Igor Pro and to give you some experience using features that you will likely need for your day-to-day work. Beyond these fundamentals, Igor includes a wide variety of features to facilitate much more advanced graphing and analysis of your data.

As you become more familiar with using Igor, you will want to further explore some of the additional learning and informational aids that we have included with Igor Pro.

- The Igor Help Browser provides online help, including reference material for all built-in operations and functions, an extensive list of shortcuts, and the ability to search Igor help files, procedure files, examples and technical notes for key phrases. See **Igor Help Browser** on page II-7 for more information.
- The More Help Files folder contains several supplementary help files. You should read Chapter III-14, **Platform-Related Issues** for information about potential problems and work-arounds that may affect your work. Use the File→Open File→Help Files menu item to open help files in the More Help Files folder.
- The Igor Pro manual is, by default, installed on your hard disk in electronic form. You can access it through the Igor Help Browser. The entire manual is also available in hard-copy form from WaveMetrics.
- The Examples folder contains a wide variety of self-documented sample experiments illustrating many of Igor's advanced graphing and programming facilities. Many of these experiments provide useful services in addition to acting as examples. You can access these most easily through the File→Example Experiments submenus.
- The WaveMetrics Procedures folder contains a number of utility procedures that you may find useful for writing your own procedures and for your more advanced graphing requirements. For an overview of the WaveMetrics procedures, choose Windows→Help Windows→WM Procedures Index.
- The Learning Aids folder contains additional guided tours and tutorials. You can access these most easily through the File→Example Experiments→Tutorials submenus.
- The More Extensions folder contains a number of External Operations (XOPs), which add functionality not built into the Igor Pro application. Read the included help files to find out more about the individual XOPs and how to install them, or consult the External Operations Index in the XOP Index help file, which has brief description of each XOP.
- The Technical Notes folder contains miscellaneous additional information and services. Tech Note #000 contains an index to all of the other notes.
- The Igor Pro mailing list is an Internet mailing list where Igor users share ideas and help each other. See **Igor Mailing List** on page II-19 for details on subscribing.

# Volume II

# User's Guide: Part 1

## Table of Contents

II-1	Getting Help .....	II-1
II-2	The Command Window .....	II-21
II-3	Experiments, Files and Folders .....	II-29
II-4	Windows .....	II-57
II-5	Waves .....	II-85
II-6	Multidimensional Waves .....	II-123
II-7	Numeric and String Variables .....	II-133
II-8	Data Folders .....	II-139
II-9	Importing and Exporting Data .....	II-163
II-10	Dialog Features .....	II-215
II-11	Tables .....	II-225
II-12	Graphs .....	II-283
II-13	Category Plots .....	II-381
II-14	Contour Plots .....	II-395
II-15	Image Plots .....	II-425
II-16	Page Layouts .....	II-453