IDL Wavelet Toolkit User's Guide

Wavelet Version 1.0
September, 1999 Edition

Acknowledgments

IDL® is a trademark of Research Systems Inc., registered in the United States Patent and Trademark Office, for the computer program described herein.

Numerical Recipes™ is a trademark of Numerical Recipes Software. Numerical Recipes routines are used by permission.

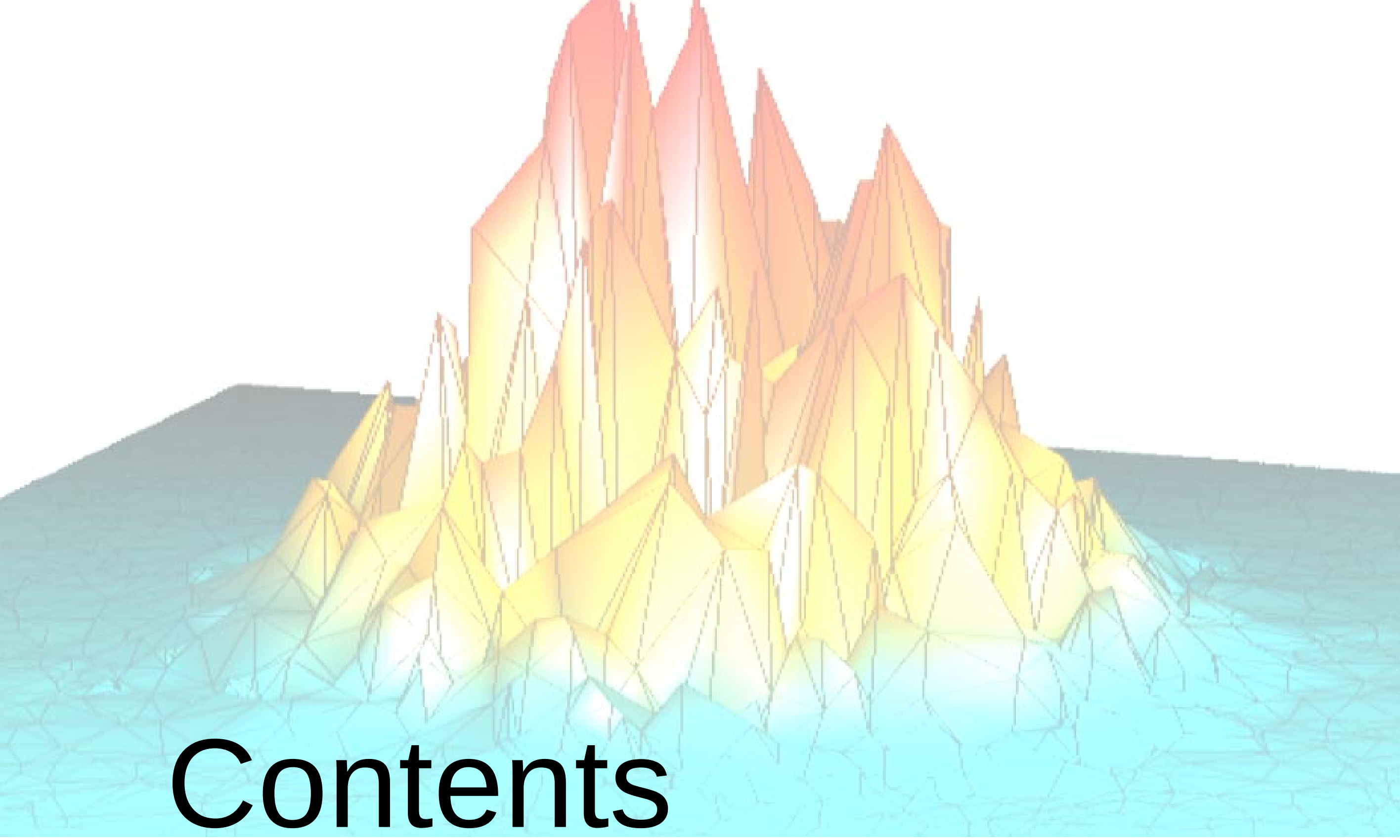
GRG2™ is a trademark of Windward Technologies, Inc. The GRG2 software for nonlinear optimization is used by permission.

NCSA Hierarchical Data Format (HDF) Software Library and Utilities
Copyright 1988-1998 The Board of Trustees of the University of Illinois
All rights reserved.

Portions of this software are copyrighted by INTERSOLV, Inc., 1991-1998.

Other trademarks and registered trademarks are the property of the respective trademark holders.

Research Systems, Inc. documentation is printed on recycled paper. Our paper has a minimum 20% post-consumer waste content and meets all EPA guidelines.



Contents

Chapter 1:
Introduction to the IDL Wavelet Toolkit 7

What is the IDL Wavelet Toolkit? 8

IDL Wavelet Toolkit Architecture 11

File Organization 11

Structure 11

Chapter 2:
Using the IDL Wavelet Toolkit 13

Starting the Toolkit 14

Menu Description 16

Dataset Viewer 20

Importing Data 25

ASCII Files 25

Binary Files 25
Image Files 26
WAV Audio Files 26
IDL Command Line 26
Wavelet Viewer 27
Wavelet Power Spectrum 31
Multiresolution Analysis 35
Denoise Tool 37
Adding User Tools 41

Chapter 3:
Theory and Examples 43

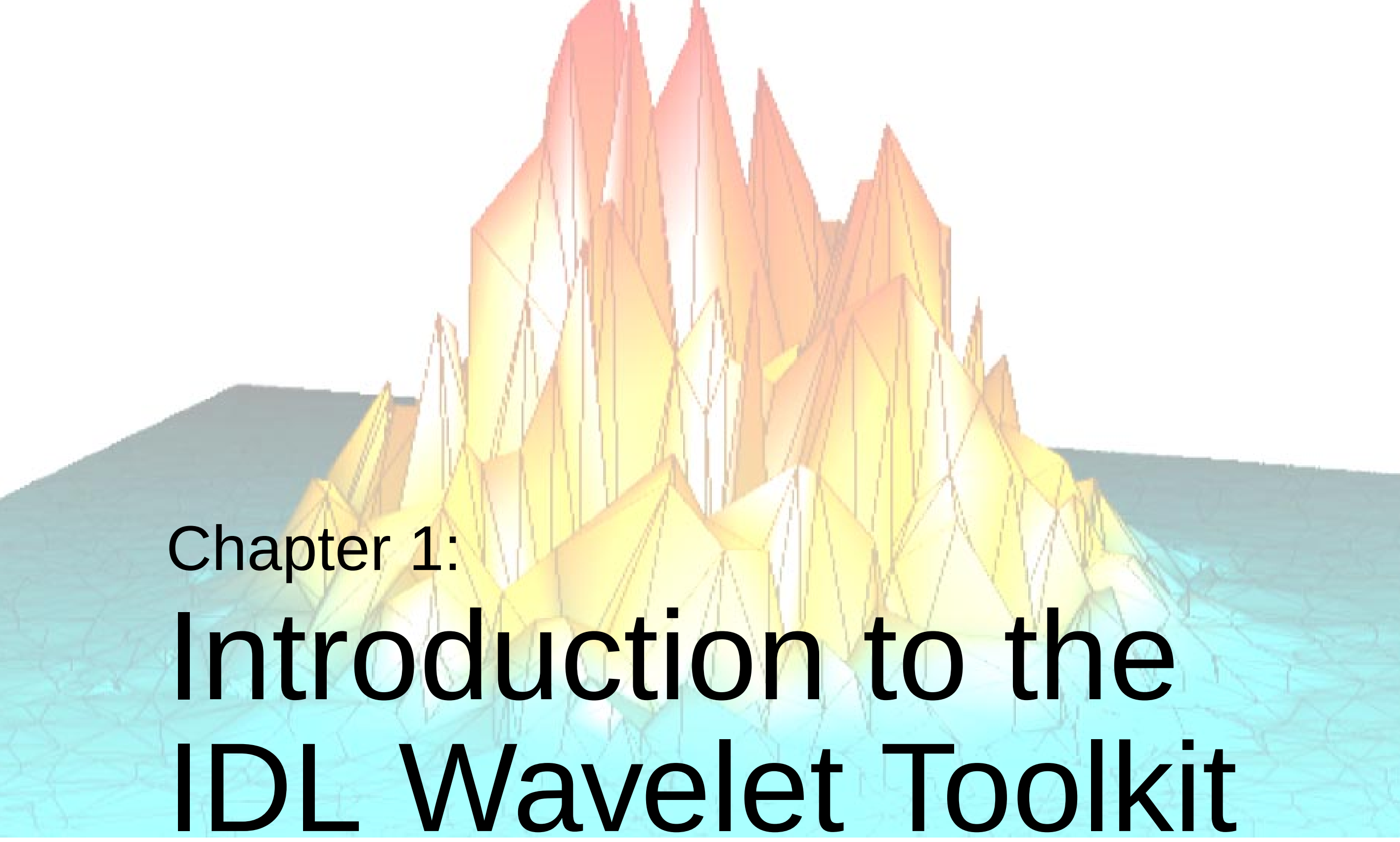
Wavelet Transform 44
Wavelet Power Spectrum 45
Denoise 47
Multiresolution Analysis 51
Bibliography 52

Chapter 4:
IDL Wavelet
Toolkit Reference 53

List of Commands by Functionality 54
WV_APPLET 56
WV_CW_WAVELET 58
WV_DWT 61
WV_FN_COIFLET 63
WV_FN_DAUBECHIES 65
WV_FN_HAAR 67
WV_FN_SYMLET 69
WV_IMPORT_DATA 71
WV_IMPORT_WAVELET 73
WV_PLOT3D_WPS 74
WV_PLOT_MULTIRES 76

WV_PWT 78
WV_TOOL_DENOISE 80

Index 83



Chapter 1:

Introduction to the IDL Wavelet Toolkit

This chapter discusses the following topics:

What is the IDL Wavelet Toolkit? 8
IDL Wavelet Toolkit Architecture 11

What is the IDL Wavelet Toolkit?

The IDL Wavelet Toolkit consists of a set of graphical user interfaces (GUI) and IDL routines for wavelet analysis of multi-dimensional data.

Motivation

Wavelet analysis is becoming a popular technique for data and image analysis. By decomposing a signal using a particular wavelet function, one can construct a picture of the energy within the signal as a function of both spatial dimension (or time) and wavelet scale (or frequency). The wavelet transform is used in numerous fields such as geophysics (seismic events), medicine (EKG and medical imaging), astronomy (image processing), and computer science (object recognition and image compression). The technique is flexible and robust, yet it is fast enough to be used in real-time image processing.

A set of standard wavelet techniques have been developed which make it possible for the average user to apply the wavelet method with confidence. Recent advances in significance testing and cross-wavelet analysis have also enhanced the acceptability of wavelet analysis within the scientific community. Nevertheless, the calculation of the wavelet transform and the display of the output requires considerable experience.

Users

The IDL Wavelet Toolkit is designed for a wide audience, ranging from the casual user who wishes to explore the possibilities of wavelet analysis, to the scientist or engineer who wants to produce robust and complex results.

Potential users and their applications include:

- Students— introduction to wavelets, graphical analysis;
- Engineers— data analysis, signal processing, data compression;
- Scientists— data analysis, filtering and denoising, cross-wavelet;
- Computer scientists— image compression, speed of operations;
- Mathematicians— explore wavelet families, test out new analysis techniques.

Applications

Examples of specific applications are:

- Time-series analysis— time-scale power spectrum, noise filtering, multiresolution analysis;
- Self-similar series— fractals, long-memory processes;
- Turbulence— detection of coherent structures;
- Signal processing— filtering and denoising;
- Image processing— edge detection, compression, enhancement.

Features

The IDL Wavelet Toolkit has the following features:

Wavelet Applet

The Toolkit Applet lets you manage your projects, import data and wavelets, visualize the results, and add your own user tools.

Discrete Wavelet Transform

You can compute the discrete wavelet transform (partial or full) on multi-dimensional data. These routines are written in C and contained in the IDL wavelet dlm.

Wavelet Functions

The Toolkit comes with several wavelet functions that are accessible both inside the Applet and from your own programs. You can easily add your own wavelet functions to the Toolkit.

3D Wavelet Power Spectrum

Callable from within the Applet and from your own programs, the visualizer plots the wavelet power as a three-dimensional surface, with optional contour lines. You can rotate, translate, and find the power at a particular location.

Multiresolution Analysis

Stand alone or callable from the Applet, this routine produces plots for the smooth (low pass), detail (band-pass), and rough (high-pass) components of your data.

Denoise Tool

This widget tool enables you to denoise your vector or image array by thresholding (hard or soft) either by cumulative power or coefficient number;

Dataset Viewer

Manage the datasets within each project by importing new data, viewing data values, and customizing the data fields.

Import Data

You can import data from a variety of file formats: ASCII, binary, image (BMP, GIF, JPEG, PNG, PPM, SRF, TIFF, DICOM), and WAV audio. Image files can be either indexed color (8- or 16-bit) or true color (24-bit). You can also import data directly from the IDL> command prompt.

User Tools

You can extend the functionality of the Wavelet Toolkit by adding your own tools.

IDL Wavelet Toolkit Architecture

File Organization

The Toolkit consists of the following components:

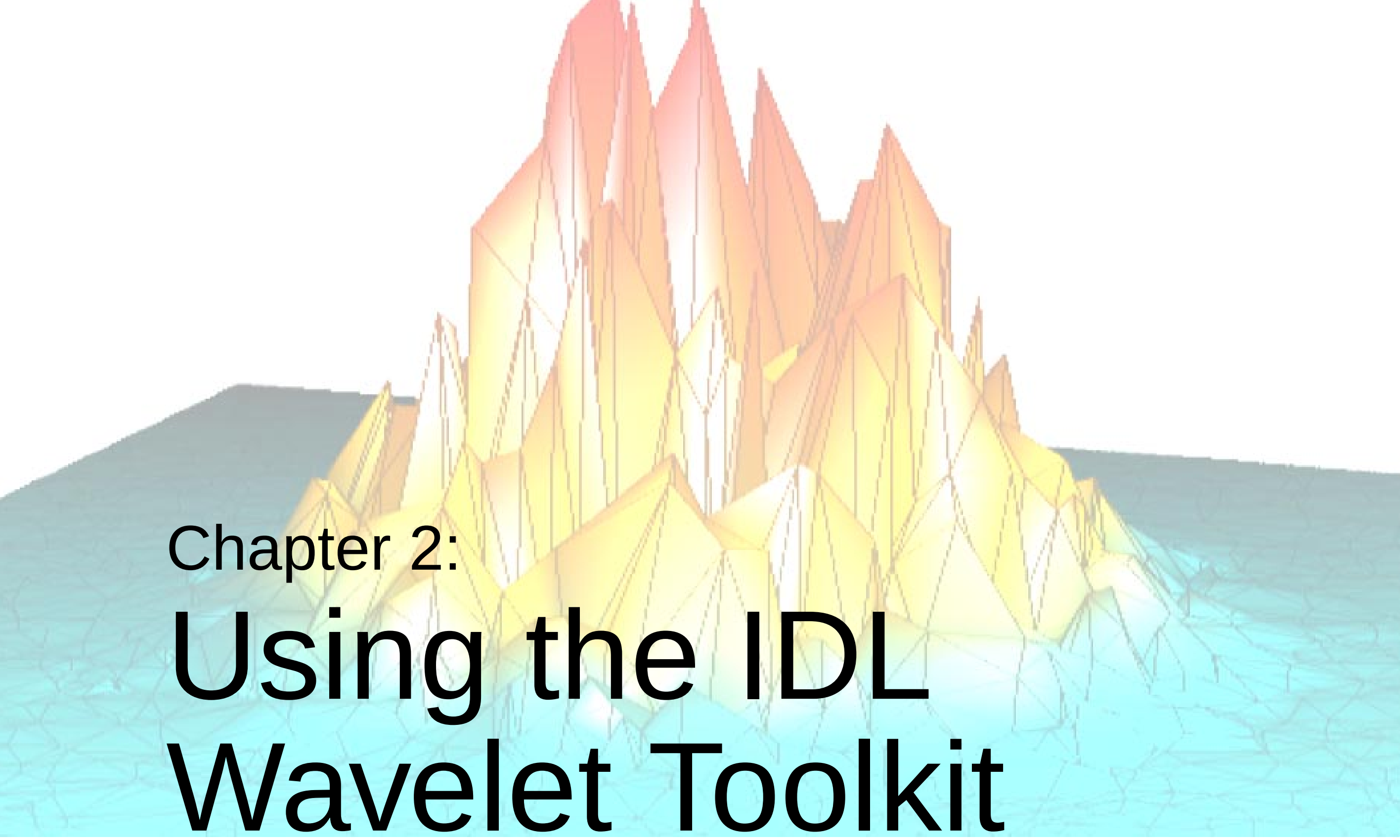
- Source (`.pro`) files in the `wavelet` directory;
- A `bitmaps` subdirectory with button bitmaps;
- The `data` subdirectory with sample data files;
- The Online help manual in the IDL `help` directory;
- The DLM (Dynamically Loadable Module) in the IDL `bin` directory.

Note

You are encouraged to view the source files for details on implementation and technique. You are also welcome to modify the source files, however, it is strongly encouraged that you copy the files to your own directory first. By modifying the IDL !PATH variable you can ensure that your routines are compiled first.

Structure

The IDL Wavelet Toolkit consists of three layers. The topmost layer is the Wavelet Applet, which allows you to import data and wavelet functions, and access various visualization and tool routines. The middle layer is the set of compound widgets and widget tools for visualization and analysis. These tools are accessible both from the Wavelet Applet and from your own routines. The lowest layer are the wavelet API (application programming interface) that consist of the wavelet functions, the wavelet transform, and the import data routine.



Chapter 2: Using the IDL Wavelet Toolkit

This chapter discusses the following topics:

Starting the Toolkit 14
Menu Description 16
Dataset Viewer 20
Importing Data 25
Wavelet Viewer 27
Wavelet Power Spectrum 31
Multiresolution Analysis 35
Denoise Tool 37
Adding User Tools 41

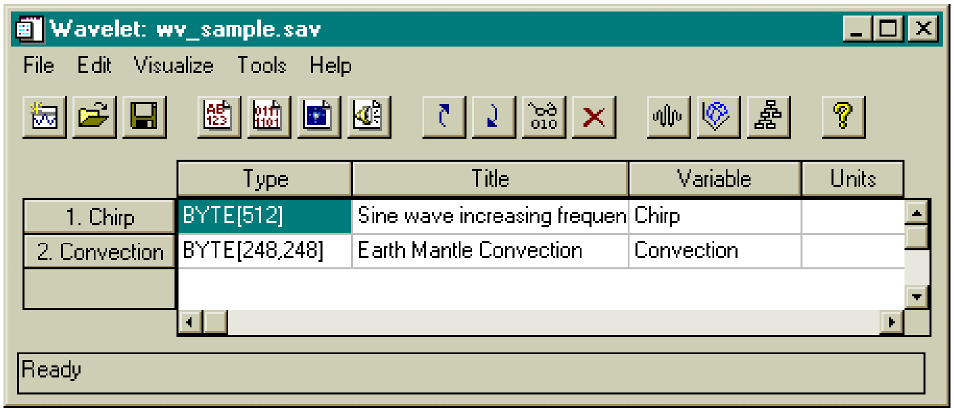
Starting the Toolkit

To start the IDL Wavelet Toolkit type the following at the IDL> command prompt:

```
wv_applet
```

This action compiles the `wv_applet` routines and starts up the main window, shown in the following figure. For other startup options see the WV_APPLET procedure.

The window consists of Menu Items, the Toolbar, the Dataset Viewer, and a Status Bar at the bottom.



	Type	Title	Variable	Units
1. Chirp	BYTE[512]	Sine wave increasing frequen	Chirp	
2. Convection	BYTE[248,248]	Earth Mantle Convection	Convection	

Figure 2-1: The main Wavelet Toolkit window.

Menu Items

The menu items, located at the top of the IDL Wavelet Toolkit window, allow you to perform various actions. These menu items are described in the next section.

Toolbar

The toolbar is divided into four sections: File, Import, Edit, Visualize, and Help. The toolbar buttons allow you to easily access various menu items.

When you position the mouse pointer over a toolbar the Status Bar displays a description of its function.

Dataset Viewer

The variables contained in your dataset are displayed in the dataset table, described in "Dataset Viewer" on page 20.

Status Bar

The Status Bar displays descriptions of the Toolbar buttons and the status of various actions such as Open, Import, and Save. The Status Bar also provides warnings if, for example, you select "Visualize..." without selecting a variable.

Menu Description

The main window has five items: File Menu, Edit Menu, Visualize Menu, Tools Menu, and Help Menu. Each menu and its submenus is described below.

File Menu

The File menu accesses and manipulates files.

New Applet

This menu item or button starts a new Wavelet Toolkit applet with an empty dataset.

Open Dataset...

This menu item or button closes the current dataset and allows you to open up a different dataset. Wavelet datasets have the default filename suffix `.sav` and are written in IDL SAVE format.

If the previous dataset has not been saved, then you will be prompted to save the previous dataset first.

Save

Select this menu item or button to save the current dataset and preferences. If the dataset has not yet been saved, then you are prompted for a filename with the Save As dialog.

Save As...

This menu item allows you to choose a new filename for the current dataset using the Save As dialog, and then saves the dataset to this file.

Import...

Select this menu item to import an external data file into the current dataset. Details on allowable file formats and import options can be found in “Dataset Viewer” on page 20. You can also import data from the current IDL session using the WV_IMPORT_DATA procedure.

Preferences...

This menu item opens up a Preferences dialog in which you can customize your interaction with the Wavelet Toolkit. The Default button restores the built-in default options for all of the preferences. The OK button keeps all of the changes to Preferences. The Cancel button discards all of the changes.

Note

The Preferences are saved within each dataset rather than in a separate preferences file; each dataset can therefore have its own set of preferences. Note, however, that opening a new dataset may change the current preferences. These new preferences will remain in effect until changed either via the Preferences window or by opening a different dataset.

Exit

This menu item will close the current Wavelet Toolkit applet. Other Wavelet applets (either started from the command line or via the “New Applet” menu item) are unaffected.

If you have made changes to the current dataset then you will be prompted to save the dataset before exiting.

Edit Menu

The Edit Menu manipulates the Dataset Viewer.

Move Variable Up

Select this menu item or button to move the currently-selected variable up in the order.

Move Variable Down

Select this menu item or button to move the currently-selected variable down in the order.

View Data Values

This menu item or button displays the values for the currently-selected variable.

Delete Variable

Select this menu item or button to delete the currently-selected variable or variables. You are asked for confirmation before the variables are removed.

Visualize Menu

The Visualize Menu contains methods to graphically display and manipulate the wavelet transform.

Wavelets

This menu item or button starts up the wavelet compound widget, which allows you display the available wavelet functions and their properties. You can also start the wavelet viewer using the WV_CW_WAVELET function from the IDL> command prompt. The wavelet widget is described in "Wavelet Viewer" on page 27.

Wavelet Power Spectrum

This menu item or button starts the three-dimensional viewer for the wavelet power spectrum, using the currently-selected variable. You can also start the viewer using the WV_PLOT3D_WPS function from the IDL> command prompt. The wavelet power spectrum viewer is described in "Wavelet Power Spectrum" on page 31.

Multiresolution Analysis

This menu item or button starts the viewer for multiresolution analysis of the currently-selected variable. You can also start the viewer using the WV_PLOT_MULTIRES function from the IDL> command prompt. The Multiresolution viewer is described in "Multiresolution Analysis" on page 35.

Tools Menu

The Tools Menu contains built-in and user-defined tools.

Denoise

This menu item starts the widget for denoising, filtering, and compression of the currently-selected variable. You can also start the viewer using the WV_TOOL_DENOISE function from the IDL> command prompt. The Denoise tool is described in "Denoise Tool" on page 37.

Other user tools...

If you have added other tools then they will be displayed here. The currently-selected variable will be passed to the tool function. See "Adding User Tools" on page 41.

Help Menu

The Help Menu provides various help functions.

IDL Help

This menu item will start up the IDL Online Help manual.

IDL Wavelet Toolkit Help

This menu item or button will start up the online help manual for the IDL Wavelet Toolkit.

Wavelet Readme

This menu item will display the Readme.txt file that contains additional information about using the Toolkit and contacting RSI for support.

Wavelet Release Notes

This menu item will display the Release_notes.txt file that contains the most recent information about the Toolkit.

About IDL Wavelet Toolkit...

Select this menu item to display information about the current version of IDL and the IDL Wavelet Toolkit.

Dataset Viewer

Your dataset can consist of several different variables, each with a different data format. The Dataset Viewer, located in the middle of the Wavelet Toolkit applet, allows you to organize and manipulate your dataset.

The variables are assigned a number and a name derived from the Variable name. You can sort the variables using the Move Variable Up and Move Variable Down buttons.

Variable Information

Each variable contains a one-dimensional vector or two-dimensional array of data values. The data values can be of any numeric type, such as BYTE, INTEGER, FLOAT, etc.

The variable also has several descriptor fields which you can modify, described below and summarized in the following table. To modify a field, double-click with the left-mouse button on the field. After editing the field press the <Return> key to keep your changes or click outside of the table to discard your changes.

Type

This string shows the numeric type and the array size of the data. It is not modifiable by the user.

Title

This string contains the overall name of the variable. The Title field is used to label the Wavelet Power Spectrum and Multiresolution widgets. The default is the null (`''`) string.

Variable

This string provides a short name for the variable. The Variable is used to label plots, and for the labels in the Dataset Viewer. For a one-dimensional vector (e.g. a time series) the Variable is equivalent to Ytitle. The default is either the name of the import file or `'Data'` if imported from the IDL> command prompt.

Units

This string gives the units of the variable, and is used to label various plots. For a one-dimensional vector (e.g. a time series) the Units is equivalent to the Yunits. The default is the null (`''`) string.

Field	Type	Example (1D vector)	Example (2D array)
Title	STRING	Wave audio recording	IEEE Test Image
Variable	STRING	Channel1	IEEEtest
Units	STRING	Amplitude	intensity
Xname	STRING	Time	X
Xunits	STRING	seconds	pixels
Xstart	STRING	0	0
Dx	STRING	1d0/22050	1
Yname	STRING		Y
Yunits	STRING		pixels
Ystart	STRING		0
Dy	STRING		1
Xoffset	LONG	0	0
Xcount	LONG	16384	256
Xstride	LONG	1	2
Yoffset	LONG		0
Ycount	LONG		256
Ystride	LONG		2
Source	STRING	wavelet/data/hello.wav	wavelet/data/IEEEtest.tif
Notes	STRING	Voice saying ‘hello’	IEEE test image

Table 2-1: Data fields in the Dataset Viewer.

Xname

This string is the name of the independent variable for the first data dimension (“X”), and is used to label the x-axis. The default is the null (`''`) string.

Xunits

This string gives the units of X. The default is the null (`''`) string.

Xstart

This string gives the value of the first X coordinate. The default is `'0'`. Xstart can contain complicated mathematical expressions, although the result must be a scalar number.

Dx

This string gives the sampling interval between the X coordinates. The default is `'1'`. Dx can contain complicated mathematical expressions, although the result must be a scalar number.

Yname

This string is the name of the independent variable for the second data dimension (“Y”), and is used to label the y-axis (for a one-dimensional variable this is actually equivalent to the name of the dependent Variable). The default is the null (`''`) string.

Yunits

This string gives the units of Y. The default is the null (`''`) string.

Ystart

This string gives the value of the first Y coordinate. The default is `'0'`. Ystart can contain complicated mathematical expressions, although the result must be a scalar number.

Dy

This string gives the sampling interval between the Y coordinates. The default is `'1'`. Dy can contain complicated mathematical expressions, although the result must be a scalar number.

Xoffset

This long integer gives the offset along the first data dimension at which to start. The default is 0L.

Xcount

This long integer gives the number of data points to use along the first data dimension. The default is the size of the first dimension.

Xstride

This long integer gives the sampling interval along the first data dimension. The default is 1L.

Yoffset

This long integer gives the offset along the second data dimension at which to start. The default is 0L.

Ycount

This long integer gives the number of data points to use along the second data dimension. The default is the size of the second dimension.

Ystride

This long integer gives the sampling interval along the second data dimension. The default is 1L.

Source

This string describes the original source or location of the data. The default is either the full filename (if the data was from a file) or `'Imported'` (if the data was from the IDL> command prompt).

Notes

You can enter miscellaneous information into the Notes string. The default is the null (`''`) string.

Mathematical Expressions

For Xstart, Dx, Ystart, and Dy it is highly recommended that whenever possible you enter mathematical expressions rather than converting to numbers. For example, in the above table, the sampling rate for hello.wav is 22050 Hz. One could have entered Dx as 0.00004535 rather than ‘1d0/22050’. Nevertheless, the latter is not only more accurate (limited only by your computer’s precision) but is also much more informative. (Note that the ‘1d0’ forces the computation to be done in double precision.)

You may also enter IDL functions in these strings. For example, if your X coordinate was in Julian days, starting from say 29 February 2000, you could set Xstart = ‘`JULDAY(2,29,2000)`’.

Selecting Variables

To select a particular variable for visualization or some other action, click the mouse on any field for that variable, or click the mouse on the Table row label to highlight the entire row.

To select multiple variables for deletion, click the mouse on any field and drag down to select the list of variables, or click once on the row label, scroll down and hold the <Shift> key while clicking on the last row label.

Importing Data

You can import data in multiple file formats into the IDL Wavelet Toolkit.

ASCII Files

Select this menu item or button to import data from an ASCII text file. After choosing the file using the Select Import File dialog, you can specify the particular format for the file using the ASCII_TEMPLATE dialog.

The ASCII_TEMPLATE routine handles ASCII files consisting of an optional header of a fixed number of lines, followed by columnar data. The procedure consists of three steps:

1. “Define Data Type/Range”— Specify whether the data is in fixed width columns or separated by commas or spaces. The first 50 lines are displayed. Choose the first line of data and click on the Next > button;
2. “Define Fields”— Choose the number of fields per line and then click Next >;
3. “Field Specification”— You can change the names and data types for the various fields. The Field names can also be changed once the data is imported into the Toolkit. Click on the Finish button to import the data into the Wavelet Toolkit.

Once the data is successfully imported you can change the default names for the variable Title, Variable, etc.

Independent Variable

By default only the first column is imported into the Wavelet Toolkit. For files with multiple columns, if the first column is determined to be monotonically increasing in value, it is assumed to be the “independent variable” (for example, “time”); the second column is then imported as the “dependent variable.”

Two-Dimensional Arrays

For two-dimensional data, under “Field Specification” (Step #3) the “Group All” button should be used to connect all of the columns into one field.

Binary Files

Select this menu item or button to import data from a binary data file. After choosing the file using the Select Import File dialog, you can specify the particular format for the file using the BINARY_TEMPLATE dialog.

The BINARY_TEMPLATE routine handles raw binary files consisting of headers and multiple data fields. The dialog consists of a Binary Template window where you can define various fields within the file. Each field will be imported into the Wavelet Toolkit as a separate variable.

Image Files

Select this menu item or button to import an image file. The function DIALOG_READ_IMAGE is used to select the image file. For files with multiple images you can choose the particular image you wish to import.

For true color (24-bit) images you will then be asked how you wish to convert the three channels into a single two-dimensional image. You have the option to scale the data into an intensity from 0–255, quantize the 24-bit colors down to 256 colors, or split the three channels into separate red, green, and blue images.

WAV Audio Files

Select this menu item or button to import a .WAV (RIFF) audio file as a one-dimensional vector. The file must be in uncompressed PCM format. Multiple channels are imported as separate variables, one for each channel.

IDL Command Line

You can also import data directly from the IDL> command prompt using the WV_IMPORT_DATA command:

```
WV_IMPORT_DATA, variable
```

where `variable` is either a data vector or array, or a structure of data tags (see the WV_IMPORT_DATA procedure for tag information).

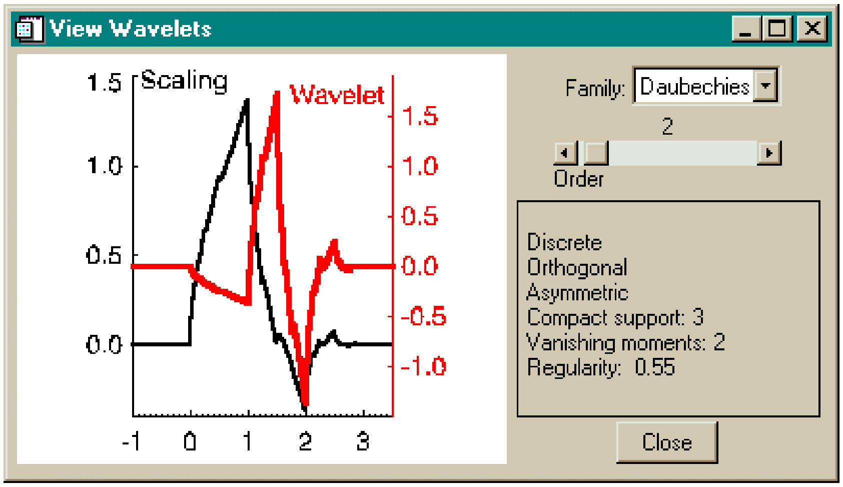
If there is more than one Wavelet Toolkit applet currently running, then variables are entered into the one that was most-recently active.

Wavelet Viewer

The Wavelet Viewer is accessible from the Visualize menu or button, and can also be started from the IDL> command prompt using the WV_CW_WAVELET function:

```
wId = WV_CW_WAVELET( )
```

The Wavelet Viewer consists of a graph of the currently-selected wavelet function, a selection area for the wavelet function, and an information area, shown in the following figure:



Family: Daubechies
Order: 2

Discrete
Orthogonal
Asymmetric
Compact support: 3
Vanishing moments: 2
Regularity: 0.55

Figure 2-2: The Wavelet Viewer.

Wavelet and Scaling Functions

The wavelet consists of two components, the scaling function which describes the low-pass filter for the wavelet transform, and the wavelet function which describes the band-pass filter for the transform.

Changing Wavelets

The droplist contains the names of all currently-available wavelets. The “Family” refers to the overall properties of the wavelet, while the “Order” determines the particular wavelet within each family.

Wavelet Information

After you select a wavelet family and order, the following information will be displayed:

Discrete/Continuous

Discrete wavelet functions are used with the discrete wavelet transform, which provides the most compact representation of the data. The discrete transform is very fast and is best suited for image processing, filtering, and large arrays.

Continuous wavelet functions are used to approximate the continuous wavelet transform, which provides a highly-redundant transformation of the data. The continuous wavelet transform is much smoother than the discrete transform and is better suited for time-series analysis on small arrays (less than 20000 data points).

Orthogonal/Nonorthogonal

Orthogonal wavelet functions will have no overlap with each other (zero correlation) when computing the wavelet transform, while nonorthogonal wavelets will have some overlap (nonzero correlation). Using an orthogonal wavelet, you can transform to wavelet space and back with no loss of information.

Nonorthogonal wavelet functions tend to artificially add in energy (due to the overlap) and require renormalization to conserve the information.

In general, discrete wavelets are orthogonal while continuous wavelets are nonorthogonal.

Symmetry

This flag describes the symmetry of the wavelet function about the midpoint. Symmetric wavelets show no preferred direction in “time,” while asymmetric wavelets give unequal weighting to different directions.

Compact Support

This value measures the effective width of the wavelet function. A narrow wavelet function such as the Daubechies order 2 (compact support=3) is fast to compute, but the narrowness in “time” implies a very large width in “frequency.” Conversely, wavelets with large compact support such as the Daubechies order 24 (compact support=47) are smoother, have finer frequency resolution and are usually more efficient at denoising.

Vanishing Moments

An important property of a wavelet function is the number of vanishing moments, which describes the effect of the wavelet on various signals. A wavelet such as the Daubechies 2 with vanishing moment=2 has zero mean and zero linear trend. When the Daubechies 2 wavelet is used to transform a data series, both the mean and any linear trend are filtered out of the series. A higher vanishing moment implies that more moments (quadratic, cubic, etc.) will be removed from the signal.

Regularity

The regularity gives an approximate measure of the number of continuous derivatives that the wavelet function possesses. The regularity therefore gives a measure of the smoothness of the wavelet function with higher regularity implying a smoother wavelet.

User-Defined Wavelets

You can easily extend the IDL Wavelet Toolkit by adding more wavelet functions. These wavelet functions should follow the same calling mechanism as the built-in wavelet functions such as WV_FN_DAUBECHIES. In addition, your wavelet function should begin with the prefix `’wv_fn_’`.

1. Let’s say you would like to add a wavelet function called “Spline” giving the Daubechies “Spline” wavelets. To do this, first create a wavelet function to return the wavelet coefficients and the information structure:

```
FUNCTION wv_fn_spline, Order, Scaling, Wavelet, Ioff, Joff
; compute coefficients here...
   ...
; find support, moments, and regularity
   ...
  info = {family:'Spline', $
    order_name:'Order', $
    order_range:[1,5,1], $
    order:order, $
    discrete:1, $
    orthogonal:1, $
    symmetric:0, $
    support:support, $
    moments:moments, $
    regularity:regularity}
  RETURN, info
END
```

2. Save this function in a file `’wv_fn_spline.pro’` that is accessible from your current IDL path.

3. Now start the Wavelet Toolkit with your new wavelet function:

   ```
   WV_APPLET, WAVELETS='Spline'
   ```

   Or, if you are already running the Wavelet Toolkit:

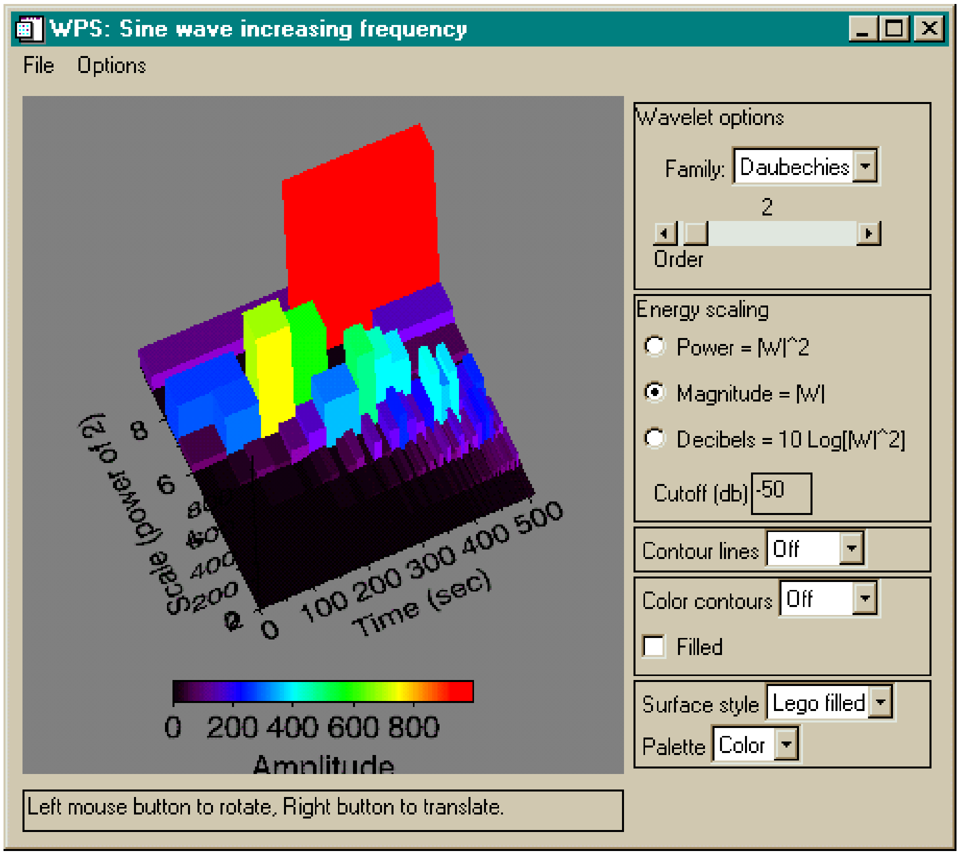
   ```
   WV_IMPORT_WAVELET, 'Spline'
   ```

Your new wavelet function should appear in the list of current wavelet functions, and should be accessible from any of the wavelet tools.

Wavelet Power Spectrum

The wavelet transform converts the data array into a series of wavelet coefficients, each of which represents the amplitude of the wavelet function at a particular location within the array and for a particular wavelet scale.

The Wavelet Power Spectrum viewer, shown in the following figure, allows you to visualize the wavelet power as a three-dimensional surface plot, where the height of the surface represents the magnitude of the wavelet coefficients.



Family: Daubechies

Order: 2

Power = |W|^2 [✓ Magnitude = |W|] Decibels = 10 Log[|W|^2]

Cutoff (db) -50

Contour lines Off

Color contours Off

Filled ☐

Surface style Lego filled

Palette Color

Left mouse button to rotate, Right button to translate.

Figure 2-3: The Wavelet Power Spectrum 3D viewer.

File Menu

Copy To Clipboard

This menu item makes a copy of the current graphics image and places it on the system clipboard.

Save As VRML

This menu item allows you to save the graphics image as a three-dimensional VRML file.

Close

This menu item closes the Wavelet Power Spectrum viewer.

Options Menu

Color Table

Selecting this item brings up the XLOADCT color table editor. You can then choose different color tables for the graphics image.

Drag Quality

This submenu has three different settings:

- Low— only the axes are exposed for graphics manipulation such as rotation and translation;
- Medium— low resolution graphics are used for graphics manipulation;
- High— full resolution is used for all graphics manipulations

Reset View

If you select this menu item, the view will be reset to zero rotation and translation.

Wavelet Options

You can change the current wavelet family or the order. The plot will be automatically updated.

Energy Scaling

These buttons control the scaling of the wavelet magnitude in the Z-direction.

Power

The power is the absolute-value-squared of the wavelet coefficients. The height of each point measures the contribution to the total energy.

This scaling emphasizes large peaks and sharp discontinuities, and de-emphasizes low-amplitude background noise.

Magnitude

The magnitude is the absolute value of the wavelet coefficients, and provides a measure of the relative amplitude of each point.

This scaling reduces the weighting given to large peaks and can bring out finer-detail features.

Decibels

The power can also be displayed in decibels, normalized relative to the mean of the wavelet power spectrum.

Since decibels are a logarithmic scale, the smallest wavelet coefficients are given just as much weight as the largest coefficients. This scaling is most useful for data that contain a broad range of energy, or that contain a single sharp spike embedded in small-amplitude noise.

Cutoff (db)

You can specify the lower cutoff for the Decibel plot. The default is –50 db.

Contour Lines

You can choose to include contour lines at the top of the plot, the bottom, or three dimensional.

Color Contours

You can also put color contours at the top, bottom, or 3D. The color contours can either be open or filled. The color palette is the same as that used for the surface plot.

Surface Style

There are seven different surface plots that you can choose from:

- Points— places colored dots at each location/height;
- Mesh— creates an unfilled surface plot;

- Surface— creates a shaded filled surface;
- XZ Lines— draws lines parallel to the X-axis, one for each Y location;
- YZ Lines— draws lines parallel to the Y-axis, one for each X location;
- Lego— draws a lego-block plot with mesh sides;
- Lego filled— draws a lego-block plot with solid sides.

You can also change the color palette between Black, Gray, and Color.

Power Display

The graphics window contains the three-dimensional image and a color palette.

If you move the mouse cursor over points in the image, then the current location and power will be displayed in the Status Bar.

Rotation

Click on the image using the left mouse button, and, holding the button down, drag the mouse pointer to rotate the image about the midpoint.

Translation

Click on the image using the right mouse button (on the Macintosh hold down the command key while clicking the button). Holding the button down, drag the mouse pointer to translate the image in the view window.

Multiresolution Analysis

Multiresolution Analysis uses the wavelet transform to decompose a data series in a cascade from the smallest scales to the largest. At each scale there are three components: the Smooth (or low-pass filtered) data series, the Details (or band-pass) data series, and the Rough (or high-pass).

For one-dimensional vectors this can be views as a hierarchy of x-y plots, as shown in the following figure:

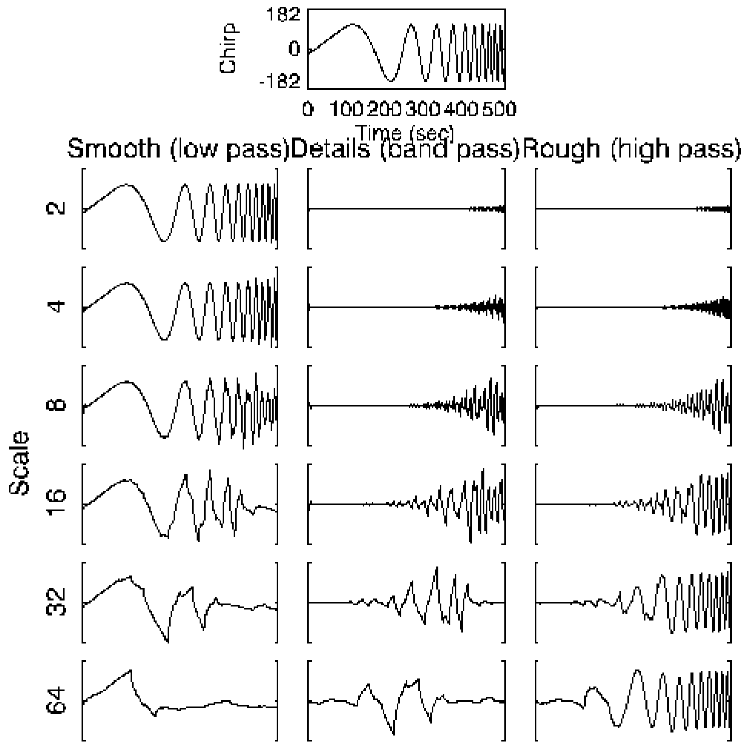


Figure 2-4: Multiresolution analysis of the "chirp" variable.

For two-dimensional arrays the multiresolution analysis gives a series of images.

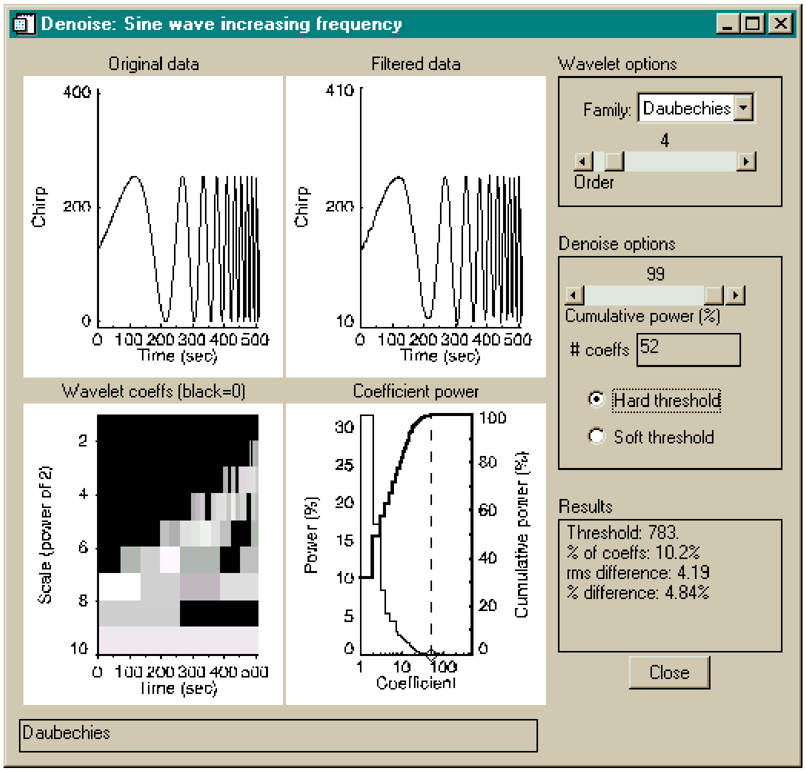
Wavelet Options

You can change the current wavelet family or the order. The plot will be automatically updated.

Denoise Tool

You can use the Denoise Tool to explore different techniques for removing noise and data compression using the wavelet transform.

The Denoise Tool is shown in the following figure. The plots and options are described below.



Threshold: 783.
% of coeffs: 10.2%
rms difference: 4.19
% difference: 4.84%

Figure 2-5: The Denoise Tool

Original Data

This window displays a graph of the original one-dimensional vector or two-dimensional image. For images, all values are converted to an intensity (0–255) and a grayscale color palette is used.

Filtered Data

This window displays the data after filtering using the wavelet function and options given on the right. For images, all values are converted to an intensity (0–255) and a grayscale color palette is used.

Wavelet Coefficients

The filtered coefficients are displayed as a two-dimensional image using a logarithmic energy scaling. The method is as follows:

1. Find the maximum value “P_m” of the original, unfiltered, wavelet power (absolute-value squared of the wavelet coefficients);
2. Square the filtered wavelet coefficients to get wavelet power, then take the base-10 logarithm of each;
3. Scale this logarithmic power from the range [$-10 \, \text{Log}_{10}(P_m)$, $\text{Log}_{10}(P_m)$] into the range [32, 255]. Values greater than zero but less than $-10 \, \text{Log}_{10}(P_m)$ are set equal to 32.
4. Set all values removed by the filter to zero (0).
5. Display the image using a grayscale color palette.

Using the above method, all retained coefficients will appear in the image, shaded from dark gray (32) to white (255). Coefficients that have been removed will be black.

Coefficient Power

This graph shows the wavelet power for each coefficient, sorted into decreasing order, and scaled so that the total power is 100%. The wavelet power is also shown as a cumulative plot where each point represents the sum of all of the previous points. Both curves are plotted on a logarithmic x-axis so that the largest coefficients are easily visible.

The dashed line shows the current cutoff value that you have selected.

Wavelet Options

You can change the current wavelet family or the order. Since all of the denoise options remain constant, you can compare the effects of different wavelet orders and families.

Denoise Options

Cumulative Power

The slider bar allows you to set the cutoff threshold for cumulative power. Coefficients to the right of the dotted line in the Coefficient Power graph will be excluded. The # Coeffs box is adjusted accordingly.

Note

At low cumulative power you may notice that the slider adjusts itself in uneven increments. This is designed so that at least one additional coefficient is either discarded (as the slider moves left) or retained (as the slider moves right). These jumps in power correspond to the discrete steps in the coefficient power graph.

Number of Coefficients

You can also specify the exact number of coefficients that you wish to retain. The cumulative power slider bar will be adjusted accordingly.

Hard Threshold

The hard threshold removes all discarded wavelet coefficients by setting them to zero and computing the inverse wavelet transform. For details see "Denoise" on page 47.

Soft Threshold

The soft threshold also sets all discarded wavelet coefficients to zero. However, it also linearly reduces the magnitude of the each retained wavelet coefficient by an amount equal to the largest discarded coefficient. For details see "Denoise" on page 47.

Results Window

This text window contains the following output results:

Threshold

The threshold is the actual wavelet power (in the variable's units squared) that is used for the cutoff value.

Percent of Coefficients

This is the percent number of coefficients used in the reconstruction. The smaller the percent coefficients the more efficient the filter.

RMS Difference

This is the root-mean-square difference between the original data (upper-left plot) and the filtered data (upper-right plot) in the variable units. A smaller number implies a more accurate reconstruction.

Percent Difference

This is the percent difference between the original and filtered data, and is equal to 100% x (RMS difference/StdDev) where StdDev is the standard deviation of the original data. The smaller the percent difference the more accurate the reconstruction.

Adding User Tools

You can extend the capabilities of the IDL Wavelet Toolkit by adding in your own user-defined tool functions. These wavelet functions should follow the same calling mechanism as the built-in tool functions such as WV_TOOL_DENOISE. In addition, your tool function should begin with the prefix `'wv_tool_'`.

1. Let's say you want to add a wavelet tool called "Edge Detect" that uses the wavelet transform to detect edges in images. To do this, first create a tool function that accepts a data array and possibly other variable parameters:

```
FUNCTION wv_tool_edgedetect, $
  Array ; 1D vector or 2D array
  [,X]  ; X coordinates of array
  [,Y]  ; Y coordinates of array
  [, GROUP_LEADER=group_leader]
  [, TITLE=title] [, UNITS=units]
  [, XTITLE=xtitle] [, XUNITS=xunits]
  [, YTITLE=ytitle] [, YUNITS=yunits]
  [, XOFFSET=xoffset] [, YOFFSET=yoffset]
; start the edge detection applet...
   ...
; return the Widget ID for the applet
  RETURN, wID
END
```

2. Save this function in a file `'wv_tool_edgedetect.pro'` that is accessible from your current IDL path.

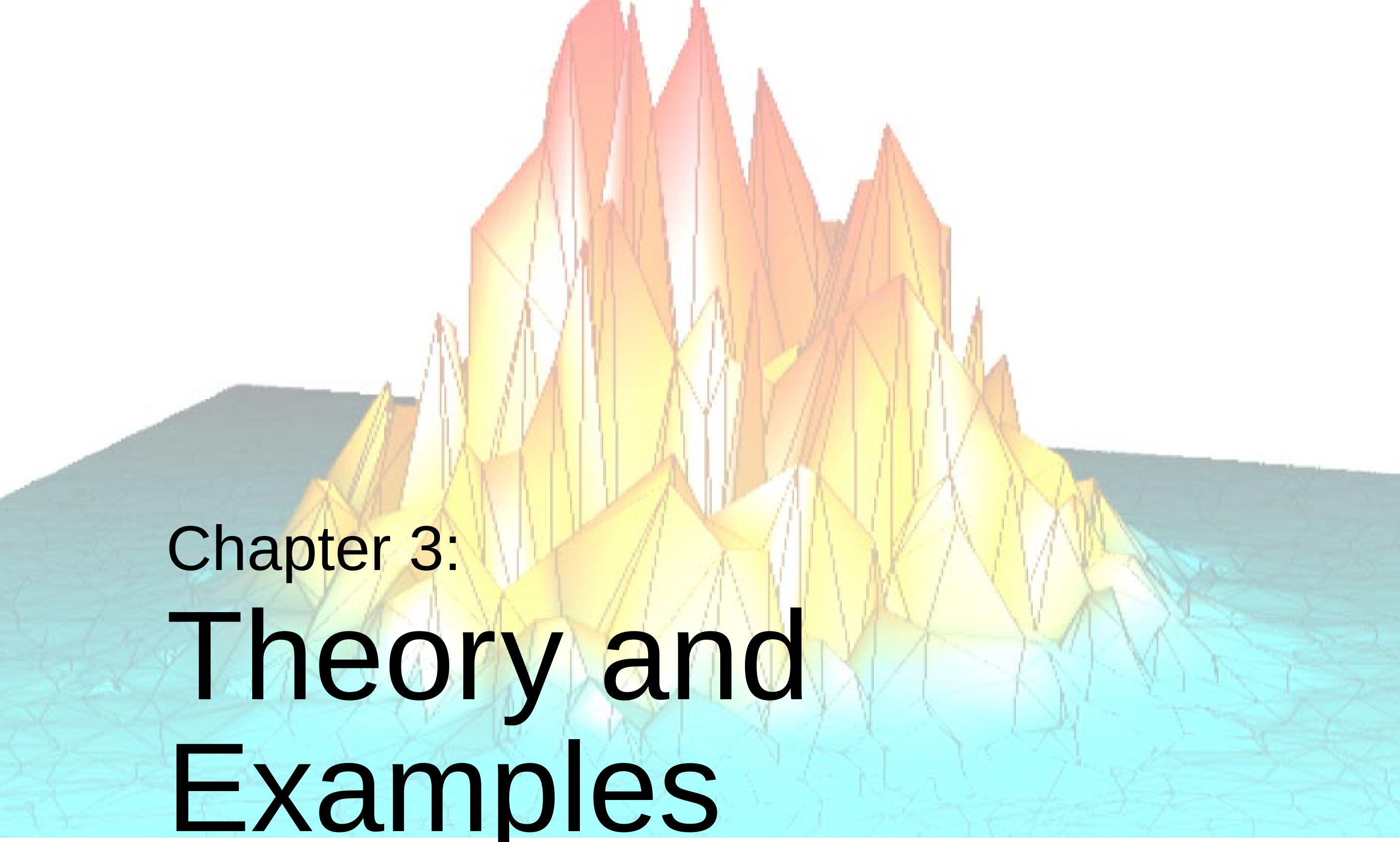
3. Now start the Wavelet Toolkit with your new wavelet function:

```
WV_APPLET, TOOLS=['Edge Detect']
```

Your new tool should appear in the Tools Menu. The actual function name is constructed by removing all white space from the name and attaching a prefix of WV_TOOL_.

Note

At a minimum your tool function must accept a data *Array*. All other parameters (such as *X* and *Y*) and keywords (GROUP_LEADER, TITLE, etc.) are optional. The IDL Wavelet Toolkit will automatically pass in only those parameters and keywords that are usable by your tool function.



Chapter 3: Theory and Examples

This chapter discusses the following topics:

Wavelet Transform 44
Wavelet Power Spectrum 45
Denoise 47
Multiresolution Analysis 51
Bibliography 52

Wavelet Transform

Background

Wavelet analysis is a technique to transform an array of N numbers from their actual numerical values to an array of N wavelet coefficients.

Each wavelet coefficient represents the closeness of the fit (or correlation) between the wavelet function at a particular size and a particular location within the data array. By varying the size of the wavelet function (usually in powers-of-two) and shifting the wavelet so it covers the entire array, you can build up a picture of the overall match between the wavelet function and your data array.

Since the wavelet functions are compact (hence the term wave-*let*), the wavelet coefficients only measure the variations around a small region of the data array. This property makes wavelet analysis very useful for signal or image processing; the “localized” nature of the wavelet transform allows you to easily pick out features in your data such as spikes (for example, noise or discontinuities), discrete objects (in, for example, astronomical images or satellite photos), edges of objects, etc.

The localization also implies that a wavelet coefficient at one location is not affected by the coefficients at another location in the data. This makes it possible to remove “noise” of all different scales from a signal, simply by discarding the lowest wavelet coefficients.

For a general introduction to the wavelet transform and its applications see Hubbard (1998).

Method

The IDL Wavelet Toolkit uses the discrete wavelet transform (DWT), calculated by the pyramidal algorithm. Details on this technique can be found in Daubechies (1992) and Mallat (1989). A good introduction to the DWT and multiresolution analysis is given in Lindsay et al. (1996).

The DWT routines are based on the routines described in section 13.10 of *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* (Cambridge University Press), and are used by permission.

An introduction to the continuous wavelet transform for time series analysis can be found in Torrence and Compo (1998), along with a discussion of statistical significance testing.

Wavelet Power Spectrum

Background

The wavelet coefficients yield information as to the correlation between the wavelet (at a certain scale) and the data array (at a particular location). A larger positive amplitude implies a higher positive correlation, while a large negative amplitude implies a high negative correlation.

A useful way to determine the distribution of energy within the data array is to plot the wavelet power, equivalent to the amplitude-squared. By looking for regions within the Wavelet Power Spectrum (WPS) of large power, you can determine which features of your signal are important and which can be ignored.

Method

Given the wavelet transform W_i of a multi-dimensional data array, A_i, where i=0...N–1 is the index and N is the number of points, then the Wavelet Power Spectrum is defined as the absolute-value squared of the wavelet coefficients, $|W_i|^2$.

One-dimensional Vector

For a vector (such as a time series) the coefficients of wavelet power can be rearranged to yield a two-dimensional picture, where the first dimension is the independent variable (e.g. time) and the second dimension is the wavelet scale (e.g. 1/frequency).

Two-dimensional Array

The wavelet transform of a 2D array is also two-dimensional, and is arranged so that the smallest scales are in the upper-right quadrant (assuming that index [0, 0] is in the lower-left).

Example

Use the “Chirp” dataset that is included in the Wavelet sample file. This dataset contains a time series with a sine wave that has an exponentially-increasing frequency. You can use the Multiresolution Analysis viewer to examine the time series.

Try the following steps:

1. Select the Chirp dataset and start up the Wavelet Power Spectrum viewer using either the Visualize Menu or the Toolbar button. The WPS can be seen under "Wavelet Power Spectrum" on page 31.
2. You should be able to see the exponential increase in frequency as a band of high power extending from left to right, and ranging from about Scale=8 near the beginning to Scale=4 near the end of the time series.
3. To bring out the features more clearly, change the Energy Scaling from Power to Magnitude
4. Notice the large peak near Scale=9. This is primarily due to the discontinuity that occurs when the dataset is wrapped around from the end back to the beginning. Switch to a smoother wavelet (such as the Daubechies 4) to remove most of this artifact.
5. Using the mouse button, rotate the plot to a angle that is better for viewing the decreasing scale (increasing frequency).
6. Add Filled Color Contours to the Top of the plot, and then select Options–> Reset View.

Question

What effect does changing the wavelet function and order have on the location and amplitude of the wavelet power?

Denoise

Background

One of the most useful applications of wavelet analysis is to remove unwanted noise from a dataset. This noise could be due to measurement errors or instrument noise. In image processing the “noise” might be small-scale features or artifacts.

You could try to remove noise from the signal by using a low-pass or band-pass Fourier filter. There are two problems with this approach:

1. You need to carefully choose the width and shape of your filter, both to avoid removing too much of your signal and to decrease “ringing” from peaks and discontinuities, and,
2. In many cases the noise is “white,” in other words, it is distributed across all frequencies or spatial scales.

Wavelet analysis, on the other hand, offers a scale-independent and robust method to filter out noise. The basic technique involves computing the wavelet transform of your data and then decreasing or discarding the smallest wavelet coefficients. The inverse transform of these coefficients will then be a filtered version of your data.

Method

We assume that you have computed the wavelet transform W_i of a multi-dimensional data array, A_i, where i=0...N–1 is the index and N is the number of points.

You then compute a threshold level W_0. This threshold level can be based on the percent of wavelet power that you wish to retain, the number of coefficients, or some other method. Suggestions for choosing the threshold are given in Donoho and Johnstone (1994). Wavelet coefficients smaller than this threshold are discarded while those above are retained. There are two methods for thresholding:

Hard threshold

The hard threshold removes all discarded wavelet coefficients by setting them to zero and computing the inverse wavelet transform. This can be defined as:

$$W_i = \begin{cases} W_i & |W_i| > W_0 \\ 0 & |W_i| \le W_0 \end{cases}$$

where W_i is the wavelet coefficient and W_0 is the chosen threshold level.

Soft Threshold

The soft threshold also sets all discarded wavelet coefficients to zero. However, it also linearly reduces the magnitude of the each retained wavelet coefficient by an amount equal to the largest discarded coefficient, i.e.:

$$W_i = \begin{cases} \operatorname{sgn}(W_i)(|W_i| - W_0) & |W_i| > W_0 \\ 0 & |W_i| \le W_0 \end{cases}$$

where sgn(W_i) is the sign of W_i.

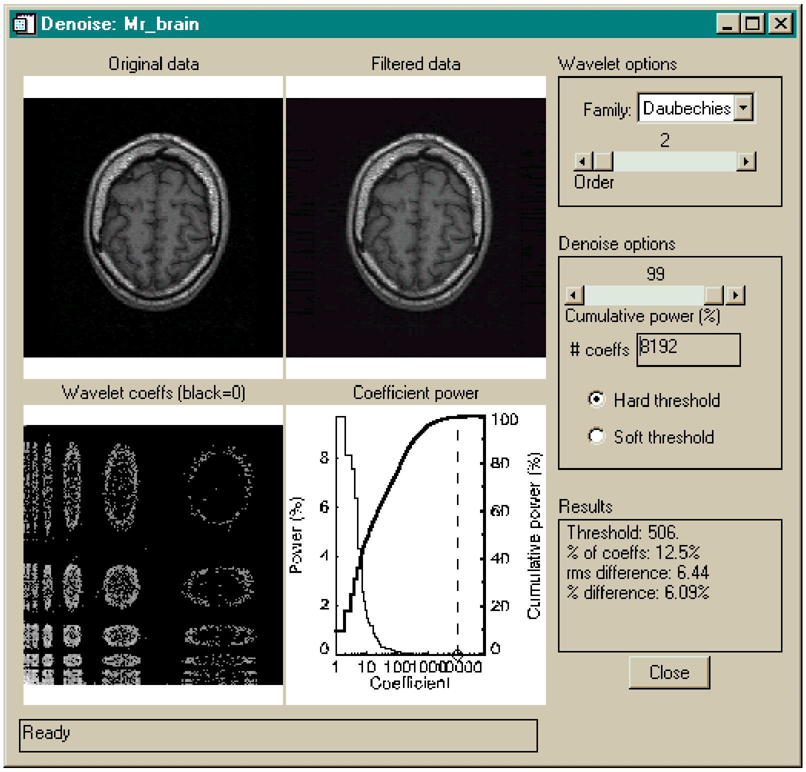
Example

We will look at a magnetic-resonance image (MRI) of the brain, and use the Denoising widget tool to filter out unwanted speckles and compress the size of the image.

Try the following steps:

1. In WV_APPLET, choose File–>Import–>Image File, and navigate to the `examples/data` directory in the IDL distribution.
2. Import the file `mr_brain.dcm`. The file should contain a 256x256 unsigned integer (UINT) image.
3. In the Dataset Viewer change the Title field to ‘MRI Brain Image’ and the Variable field to ‘Brain’.
4. Select the Brain dataset and start up the Denoise tool from the Tools Menu. You should see the Denoise widget, with the threshold set to 100% and all coefficients retained.

5. Set the ‘# coeffs’ threshold to 8192 points. You should then see a view similar to that of the following figure.



Wavelet options
Family: Daubechies
Order: 2

Denoise options
Cumulative power (%): 99
# coeffs: 8192
[✓ Hard threshold]
Soft threshold

Results
Threshold: 506.
% of coeffs: 12.5%
rms difference: 6.44
% difference: 6.09%

Figure 3-1: The Denoise widget for the MRI brain scan

Notice that you have retained 12.5% of the coefficients and have discarded 87.5%. The black regions of the “Wavelet Coeffs” plot shows the discarded coefficients. The percent difference between the original and filtered image is about 6%. Examining the filtered image, you will notice that much of the speckling around the outside is now gone. In addition, some of the small-scale features and low-contrast regions within the image have been diminished. Finally, the dotted line on the Cumulative Power graph indicates that although you are only retaining 12.5% of the information you are preserving almost 100% of the variance, or power.

Question

Increase the Wavelet order and notice what happens to the Results information, especially the percent difference. Finally, try a different wavelet function. At this threshold, which wavelet and order gives the smallest error?

Multiresolution Analysis

Background

The wavelet transform can be thought of as a band-pass filter, where the location and width in Fourier space depends on the wavelet scale. Larger scales imply a lower frequency and small bandwidth.

In computing the wavelet transform you change from small scales to larger scales. At each stage you can stop and compute the inverse wavelet transform using the remaining coefficients, while setting the small-scale coefficients to zero. You can then build up a series of smooth (or low-passed), detailed (or band-passed), or rough (high-passed) versions of your original data.

Method

Details on computing the multiresolution analysis can be found in Lindsay et al. (1996).

Example

Use the “Mantle convection” dataset that is included in the Wavelet sample file. This dataset contains an image of convection within the Earth’s mantle.

Try the following steps:

1. Select the Convection dataset and start up the Multiresolution Analysis viewer using either the Visualize Menu or the Toolbar button.
2. As you progress from top to bottom the wavelet scale increases in powers of two. At the smallest scale most of the image is still in the Smooth image. Notice that the Rough image contains only the edges or discontinuities which the small scales can pick out.
3. Change to the Haar wavelet and observe the different structure of the images.

Question

Which scale is best for picking out the largest regions of the plot?

Think of some applications for the small-scale Rough images.

Bibliography

Daubechies, I., 1992: *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 357 pp.

Donoho, D. L. and I. M. Johnstone, 1994: Ideal spatial adaptation by wavelet shrinkage. *Biometrika,* **81,** 425–455.

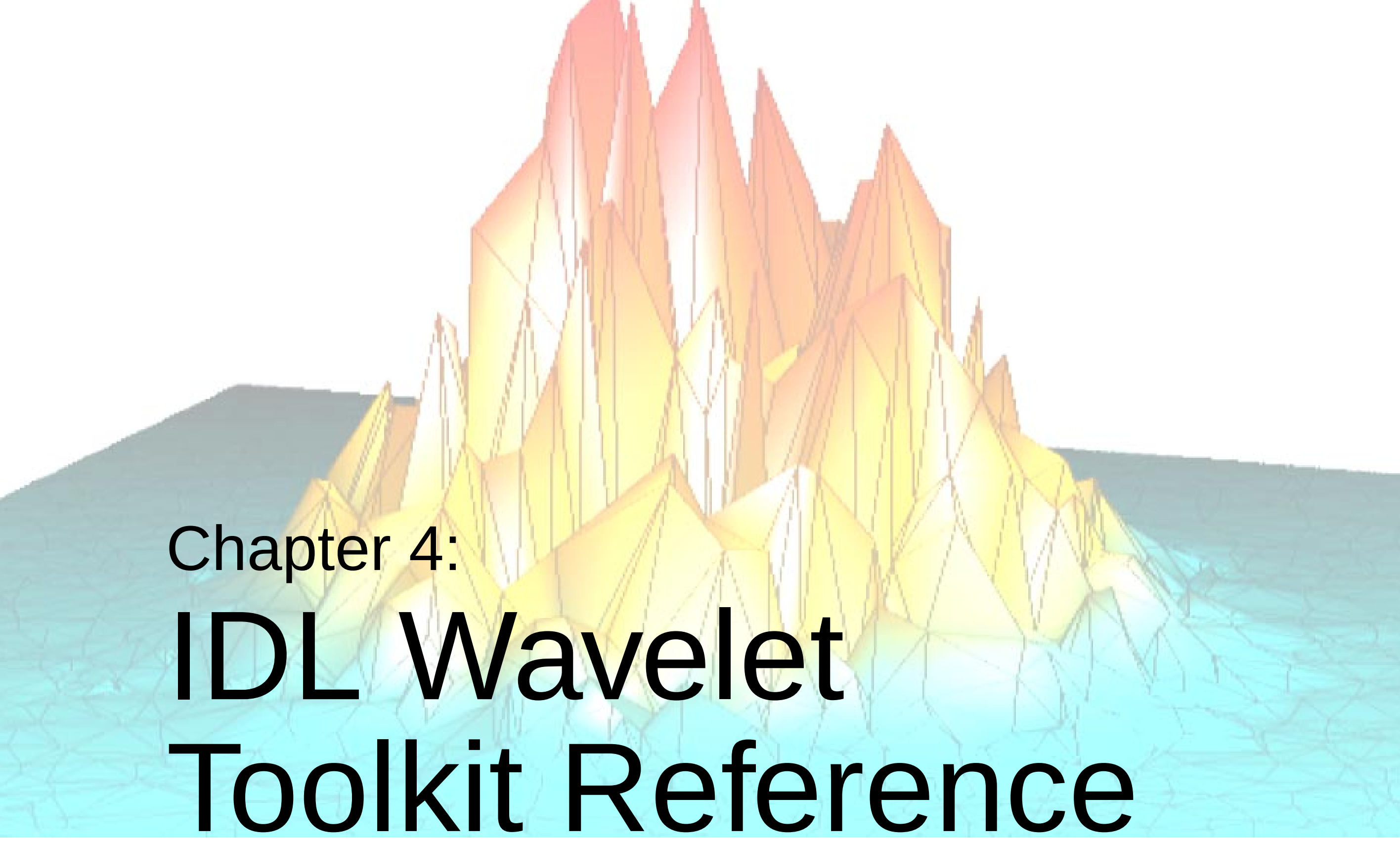
Hubbard, B. B., 1998: *The World According to Wavelets, 2nd ed.* A. K. Peters, Wellesley, Mass., 331 pp.

Lindsay, R. W., D. B. Percival, and D. A. Rothrock, 1996: The discrete wavelet transform and the scale analysis of the surface properties of sea ice. *IEEE Trans. Geosci. Remote Sens.,* **34,** 771–787.

Mallat, S., 1989: Multiresolution approximation and wavelets. *Trans. Amer. Math. Soc.,* **315,** 69–88.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1992: *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* Cambridge University Press, 994 pp.

Torrence, C., and G. P. Compo, 1998: A practical guide to wavelet analysis. *Bull. Amer. Meteor. Soc.*, **79**, 61–78.



Chapter 4: IDL Wavelet Toolkit Reference

This reference lists the following topics:

List of Commands by Functionality 54
WV_APPLET 56
WV_CW_WAVELET 58
WV_DWT 61
WV_FN_COIFLET 63
WV_FN_DAUBECHIES 65
WV_FN_HAAR 67
WV_FN_SYMLET 69
WV_IMPORT_DATA 71
WV_IMPORT_WAVELET 73
WV_PLOT3D_WPS 74
WV_PLOT_MULTIRES 76
WV_PWT 78
WV_TOOL_DENOISE 80

List of Commands by Functionality

Widget Commands and Visualization Tools

The following table describes the widget and visualization tools:

Command	Description
WV_APPLET	Run IDL Wavelet Toolkit GUI
WV_CW_WAVELET	Compound widget to display and select wavelets
WV_IMPORT_DATA	Import data from the IDL> command prompt
WV_IMPORT_WAVELET	Import wavelet functions into the current applet
WV_PLOT3D_WPS	Run the wavelet power spectrum GUI
WV_PLOT_MULTIRES	Run the multiresolution analysis GUI
WV_TOOL_DENOISE	Run the wavelet de-noising GUI

Table 4-1: Widget commands and tools. (GUI=Graphical user interface.)

Wavelet Transform

The following table describes the wavelet transform commands:

Command	Description
WV_DWT	Compute the discrete wavelet transform of an array
WV_PWT	Compute the partial wavelet transform of a vector

Table 4-2: Wavelet transform commands

Wavelet Functions

The following table describes the built-in wavelet functions:

Command	Description
WV_FN_COIFLET	Construct coiflet wavelet coefficients
WV_FN_DAUBECHIES	Construct Daubechies wavelet coefficients
WV_FN_HAAR	Construct Haar wavelet coefficients
WV_FN_SYMLET	Construct symlet wavelet coefficients

Table 4-3: Wavelet basis functions.

WV_APPLET

The WV_APPLET procedure runs the IDL Wavelet Toolkit graphical user interface.

Syntax

WV_APPLET [, *Filename*] [, ARRAY=*array*] [, GROUP_LEADER=*widget_id*] [, /NO_SPLASH] [, TOOLS=*string array*] [, WAVELETS=*string or string array*]

Arguments

Filename

A scalar string giving the name of a Wavelet Toolkit save file to open upon startup. If no filename is specified (and keyword ARRAY is not set) then the example file is opened. If the filename is a null string (`''`) then the example file is not opened.

Keywords

ARRAY

A one- or two-dimensional array to be imported into WV_APPLET. If argument *Filename* is also given then ARRAY is added to the list of variables.

GROUP_LEADER

The widget ID of an existing widget that serves as "group leader" for the newly-created widget. When a group leader is killed, for any reason, all widgets in the group are also destroyed.

A given widget can be in more than one group. The WIDGET_CONTROL procedure can be used to add additional group associations to a widget. It is not possible to remove a widget from an existing group.

NO_SPLASH

If this keyword is set then the splash screen will not be displayed on startup.

TOOLS

A scalar string or vector of strings giving the names of user-defined functions to be included in the WV_APPLET Tools menu. The actual function names are constructed by removing all white space from each name and attaching a prefix of WV_TOOL_.

WAVELETS

A scalar string or vector of strings giving the names of user-defined wavelet functions to be included in WV_APPLET. The actual function names are constructed by removing all white space from each name and attaching a prefix of WV_FN_.

Examples

```
WV_APPLET, TOOLS=['Renormalize','My Tool']
```

The above statement will start up the Wavelet Toolkit, and add the user tools 'Renormalize' and 'My Tool' to the Tools menu. When these are selected the actual functions that will be called are WV_TOOL_RENORMALIZE and WV_TOOL_MYTOOL.

See Also

For an example of TOOLS see WV_TOOL_DENOISE

WV_CW_WAVELET

The WV_CW_WAVELET function is a compound widget that lets the user select and display wavelet functions. WV_CW_WAVELET is accessible from the Visualize Menu of WV_APPLET.

Syntax

Result = WV_CW_WAVELET([*Parent*] [, TITLE=*string*] [, UNAME=*string*] [, UVALUE=*value*] [, WAVELETS=*string array*])

Return Value

The returned value of this function is the widget ID of the newly-created widget.

Arguments

Parent

The widget ID of the parent widget. Omit this argument to created a top-level widget.

Keywords

TITLE

Set this keyword equal to a scalar string containing the title of the top level base. TITLE is not used if the wavelet widget has a parent widget. If it is not specified, the default title is “Wavelets.”

UNAME

Set this keyword to a string that can be used to identify the widget in your code. You can associate a name with each widget in a specific hierarchy, and then use that name to query the widget hierarchy and get the correct widget ID.

To query the widget hierarchy, use the WIDGET_INFO function with the FIND_BY_UNAME keyword. The UNAME should be unique to the widget hierarchy because the FIND_BY_UNAME keyword returns the ID of the first widget with the specified name.

UVALUE

Set this keyword equal to the user value associated with the widget.

WAVELETS

A scalar string or vector of strings giving the names of user-defined wavelet functions to be included in WV_CW_WAVELET. The actual function names are constructed by removing all white space from each name and attaching a prefix of WV_FN_.

Widget Keywords Accepted

The WV_CW_WAVELET also accepts the following WIDGET_BASE keywords: ALIGN_BOTTOM, ALIGN_CENTER, ALIGN_LEFT, ALIGN_RIGHT, ALIGN_TOP, DISPLAY_NAME, FRAME, GROUP_LEADER, KBRD_FOCUS_EVENTS, MAP, NOTIFY_REALIZE, RESOURCE_NAME, SCR_XSIZE, SCR_YSIZE, SPACE, TLB_FRAME_ATTR, TRACKING_EVENTS, UNITS, XOFFSET, XSIZE, YOFFSET, YSIZE.

Keywords to WIDGET_CONTROL and WIDGET_INFO

The widget ID returned by most compound widgets is actually the ID of the compound widget's base widget. This means that many keywords to the WIDGET_CONTROL and WIDGET_INFO routines that affect or return information on base widgets can be used with compound widgets.

In addition, you can use the GET_VALUE and SET_VALUE keywords to WIDGET_CONTROL to obtain or set the value of the wavelet. Use the command:

```
WIDGET_CONTROL, id, GET_VALUE=value
```

to read the current wavelet. To change the current wavelet, use the command:

```
WIDGET_CONTROL, id, SET_VALUE=value
```

In both cases value is an anonymous structure, `{FAMILY: '', ORDER: 0}`, where FAMILY is a string containing the name (for example 'Daubechies'), and ORDER is a variable giving the order number. Depending on the family, ORDER can be of type Integer or Double.

See Compound Widgets in *Building IDL Applications* for a more complete discussion of controlling compound widgets using WIDGET_CONTROL and WIDGET_INFO.

Widget Events Returned by the WV_CW_WAVELET Widget

This widget generates event structures each time the family or order is changed. The event structure has the following definition:

```
Event = { ID:0L, TOP:0L, HANDLER:0L, FAMILY:'', ORDER:0}
```

The ID field is the widget ID of the WV_CW_WAVELET widget. The TOP field is the widget ID of the top-level widget. HANDLER is the widget ID of the widget handler. The FAMILY field contains the family name. The ORDER field contains the order number, and can be an Integer or a Double depending on the family.

See Also

WV_FN_COIFLET, WV_FN_DAUBECHIES, WV_FN_HAAR, WV_FN_SYMLET

WV_DWT

The WV_DWT function returns the multi-dimensional discrete wavelet transform of the input *Array*. The transform is done by WV_PWT using a user-inputted wavelet filter.

The length of each dimension of *Array* must be either a power of two (2), or must be less than four (4). The transform is not computed over dimensions of lengths less than four (4), but is computed over all other dimensions (for example, the wavelet transform of an array of size [3, 256] is computed over each [1, 256] column vector).

WV_DWT is based on the routine `wtn` described in section 13.10 of *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* (Cambridge University Press), and is used by permission.

Syntax

Result = WV_DWT(*Array*, *Scaling*, *Wavelet*, *Ioff*, *Joff*)

Return Value

The result is an output array of the same dimensions as *Array*, containing the discrete wavelet transform over each dimension, computed using the pyramidal algorithm (Mallat 1989).

Arguments

Array

The input vector or array. The length of each dimension must be either less than four (4) or a power of two (2).

Scaling

A vector of scaling (father) coefficients.

Wavelet

A vector of wavelet (mother) coefficients.

Ioff

An integer that specifies the support offset for *Scaling*.

Joff

An integer that specifies the support offset for *Wavelet*.

Keywords

DOUBLE

Set this keyword to force the computation to be done in double-precision arithmetic.

INVERSE

If set, the inverse transform is computed. By default, the forward transform is computed.

See Also

WV_PWT, WTN

WV_FN_COIFLET

The WV_FN_COIFLET function constructs wavelet coefficients for the coiflet wavelet function.

Syntax

Result = WV_FN_COIFLET([*Order*, *Scaling*, *Wavelet*, *Ioff*, *Joff*])

Return Value

The returned value of this function is an anonymous structure of information about the particular wavelet.

Tag	Type	Definition
FAMILY	STRING	‘Coiflet’
ORDER_NAME	STRING	‘Order’
ORDER_RANGE	INTARR(3)	[1, 5, 1] Valid order range [first, last, default]
ORDER	INT	The chosen *Order*
DISCRETE	INT	1 [0=continuous, 1=discrete]
ORTHOGONAL	INT	1 [0=nonorthogonal, 1=orthogonal]
SYMMETRIC	INT	2 [0=asymmetric, 1=symm., 2=near symm.]
SUPPORT	INT	6**Order* – 1 [Compact support width]
MOMEMTS	INT	2**Order* [Number of vanishing moments]
REGULARITY	DOUBLE	The number of continuous derivatives

Table 4-4: The structure tags for Result.

Arguments

Order

A scalar that specifies the order number for the wavelet. The default is 1.

Scaling

On output, contains a vector of scaling (father) coefficients.

Wavelet

On output, contains a vector of wavelet (mother) coefficients.

Ioff

On output, contains an integer that specifies the support offset for *Scaling*.

Joff

On output, contains an integer that specifies the support offset for *Wavelet*.

Note

If none of the above arguments are present then the function will simply return the *Result* structure using the default *Order*.

Keywords

None.

WV_FN_DAUBECHIES

The WV_FN_DAUBECHIES function constructs wavelet coefficients for the Daubechies wavelet function.

Syntax

Result = WV_FN_DAUBECHIES([*Order*, *Scaling*, *Wavelet*, *Ioff*, *Joff*])

Return Value

The returned value of this function is an anonymous structure of information about the particular wavelet.

Tag	Type	Definition
FAMILY	STRING	‘Daubechies’
ORDER_NAME	STRING	‘Order’
ORDER_RANGE	INTARR(3)	[1, 24, 2] Valid order range [first, last, default]
ORDER	INT	The chosen *Order*
DISCRETE	INT	1 [0=continuous, 1=discrete]
ORTHOGONAL	INT	1 [0=nonorthogonal, 1=orthogonal]
SYMMETRIC	INT	0 [0=asymmetric, 1=symm., 2=near symm.]
SUPPORT	INT	2**Order* – 1 [Compact support width]
MOMEMTS	INT	*Order* [Number of vanishing moments]
REGULARITY	DOUBLE	The number of continuous derivatives

Table 4-5: The structure tags for Result.

Arguments

Order

A scalar that specifies the order number for the wavelet. The default is 2.

Scaling

On output, contains a vector of scaling (father) coefficients.

Wavelet

On output, contains a vector of wavelet (mother) coefficients.

Ioff

On output, contains an integer that specifies the support offset for *Scaling*.

Joff

On output, contains an integer that specifies the support offset for *Wavelet*.

Note

If none of the above arguments are present then the function will simply return the *Result* structure using the default *Order*.

Keywords

None.

WV_FN_HAAR

The WV_FN_HAAR function constructs wavelet coefficients for the Haar wavelet function.

Syntax

Result = WV_FN_HAAR([*Order*, *Scaling*, *Wavelet*, *Ioff*, *Joff*])

Return Value

The returned value of this function is an anonymous structure of information about the particular wavelet.

Tag	Type	Definition
FAMILY	STRING	‘Haar’
ORDER_NAME	STRING	‘Order’
ORDER_RANGE	INTARR(3)	[1, 1, 1] Valid order range [first, last, default]
ORDER	INT	1
DISCRETE	INT	1 [0=continuous, 1=discrete]
ORTHOGONAL	INT	1 [0=nonorthogonal, 1=orthogonal]
SYMMETRIC	INT	0 [0=asymmetric, 1=symm., 2=near symm.]
SUPPORT	INT	1 [Compact support width]
MOMEMTS	INT	1 [Number of vanishing moments]
REGULARITY	DOUBLE	0d [Number of continuous derivatives]

Table 4-6: The structure tags for Result.

Arguments

Order

A scalar that specifies the order number for the wavelet. The default is 1.

Scaling

On output, contains a vector of scaling (father) coefficients.

Wavelet

On output, contains a vector of wavelet (mother) coefficients.

Ioff

On output, contains an integer that specifies the support offset for *Scaling*.

Joff

On output, contains an integer that specifies the support offset for *Wavelet*.

Note

If none of the above arguments are present then the function will simply return the *Result* structure using the default *Order*.

Keywords

None.

WV_FN_SYMLET

The WV_FN_SYMLET function constructs wavelet coefficients for the symlet wavelet function.

Syntax

Result = WV_FN_SYMLET([*Order*, *Scaling*, *Wavelet*, *Ioff*, *Joff*])

Return Value

The returned value of this function is an anonymous structure of information about the particular wavelet.

Tag	Type	Definition
FAMILY	STRING	‘Symlet’
ORDER_NAME	STRING	‘Order’
ORDER_RANGE	INTARR(3)	[1, 15, 4] Valid order range [first, last, default]
ORDER	INT	The chosen *Order*
DISCRETE	INT	1 [0=continuous, 1=discrete]
ORTHOGONAL	INT	1 [0=nonorthogonal, 1=orthogonal]
SYMMETRIC	INT	2 [0=asymmetric, 1=symm., 2=near symm.]
SUPPORT	INT	2*$Order$ – 1 [Compact support width]
MOMEMTS	INT	*Order* [Number of vanishing moments]
REGULARITY	DOUBLE	The number of continuous derivatives

Table 4-7: The structure tags for Result.

Arguments

Order

A scalar that specifies the order number for the wavelet. The default is 4.

Scaling

On output, contains a vector of scaling (father) coefficients.

Wavelet

On output, contains a vector of wavelet (mother) coefficients.

Ioff

On output, contains an integer that specifies the support offset for *Scaling*.

Joff

On output, contains an integer that specifies the support offset for *Wavelet*.

Note

If none of the above arguments are present then the function will simply return the *Result* structure using the default *Order*.

Keywords

None.

WV_IMPORT_DATA

The WV_IMPORT_DATA procedure allows the user to add a variable to the currently active WV_APPLET widget from the IDL> command prompt.

Syntax

WV_IMPORT_DATA, *Data* [, MESSAGE_OUT=*string*] [, PARENT=*variable*]

Arguments

Data

A one- or two-dimensional array of data, or a structure containing the data.

Keywords

MESSAGE_OUT

A scalar string giving a message to be output to the WV_APPLET message bar.

PARENT

A long integer specifying the ID of the WV_APPLET widget in which to import the data. The default is the most-recently active WV_APPLET widget.

Example

To import a 1D or 2D array directly into the active WV_APPLET widget:

```
WV_IMPORT_DATA, Array
```

To import a data structure:

```
WV_IMPORT_DATA, {DATA: PTR_NEW(Array), $
    SOURCE: 'Chandra X-Ray Observatory', $
    TITLE: 'Cygnus X-1 X-Ray Image', $
    VARIABLE: 'Cygnus X-1', $
    UNITS: 'Intensity'}
```

If *Data* is a structure then it must include at the very least a DATA tag containing a pointer to a one- or two-dimensional array. Recognized tags are shown in Table 4-8. Tags other than these will be quietly ignored.

Tag	Type	Definition
DATA	PTR->Array	Pointer to data array
TITLE	STRING	Long name of data variable
VARIABLE	STRING	Short name of data variable
UNITS	STRING	Units for variable
XNAME	STRING	Name of X coordinate
XUNITS	STRING	Units for X coordinate
XSTART	STRING	Starting value for X coord
DX	STRING	Sampling rate for X coord
YNAME	STRING	Name of Y coordinate
YUNITS	STRING	Units for Y coordinate
YSTART	STRING	Starting value for Y coord
DY	STRING	Sampling rate for Y coord
XOFFSET	LONG	Starting index of X coord to use
XCOUNT	LONG	Number of X coords to use
XSTRIDE	LONG	X sampling interval to use
YOFFSET	LONG	Starting index of Y coord to use
YCOUNT	LONG	Number of Y coords to use
YSTRIDE	LONG	Y sampling interval to use
SOURCE	STRING	Filename or contact info
NOTES	STRING	Miscellaneous notes
COLORS	PTR->Bytarr(3,256)	Pointer to color table for Data

Table 4-8: Tags that are recognized by WV_IMPORT_DATA.

WV_IMPORT_WAVELET

The WV_IMPORT_WAVELET procedure allows the user to add wavelet functions to the IDL Wavelet Toolkit.

Syntax

WV_IMPORT_WAVELET [, *Wavelet*] [, /RESET]

Arguments

Wavelet

A string scalar or vector giving the names of the wavelet functions. The actual function names are constructed by removing all white space from each name and attaching a prefix of WV_FN_.

Keywords

RESET

If set, then erase all user-defined wavelets and initialize with only the built-in wavelet functions. If *Wavelet* is also specified then the new wavelets will be appended onto the built-in wavelets.

See Also

WV_APPLET, WV_CW_WAVELET

WV_PLOT3D_WPS

The WV_PLOT3D_WPS function runs the graphical user interface for three-dimensional visualization of the wavelet power spectrum. WV_PLOT3D_WPS is accessible from the Visualize Menu of WV_APPLET.

Syntax

Result = WV_PLOT3D_WPS(*Array* [, *X*] [, *Y*] [, GROUP_LEADER=*widget_id*] [, TITLE=*string*] [, UNITS=*string*] [, XTITLE=*string*] [, XUNITS=*string*] [, YTITLE=*string*] [, YUNITS=*string*])

Return Value

The returned variable is the Widget ID of the newly-created widget.

Arguments

Array

A one- or two-dimensional array of data to be analyzed.

X

An optional vector of uniformly-spaced values giving the location of points along the first dimension of *Array*. The default is 0, 1, 2,..., N_X–1, where N_X is the size of the first dimension.

Y

An optional vector of uniformly-spaced values giving the location of points along the second dimension of *Array*. The default is 0, 1, 2,..., N_Y–1, where N_Y is the size of the second dimension.

Keywords

GROUP_LEADER

The widget ID of an existing widget that serves as "group leader" for the newly-created widget. When a group leader is killed, for any reason, all widgets in the group are also destroyed.

A given widget can be in more than one group. The WIDGET_CONTROL procedure can be used to add additional group associations to a widget. It is not possible to remove a widget from an existing group.

TITLE

A scalar string giving the label to be used for the widget. The default is ‘WPS:’.

UNITS

A scalar string giving the units of *Array*.

XTITLE

A scalar string giving the label to be used for the first dimension.

XUNITS

A scalar string giving the units of *X*.

YTITLE

A scalar string giving the label to be used for the y-axis (for a 1D vector) or for the second dimension (for a 2D array).

YUNITS

A scalar string giving the units of *Array* (for a 1D vector) or the units of *Y* (for a 2D array).

See Also

WV_APPLET

WV_PLOT_MULTIRES

The WV_PLOT_MULTIRES function runs the graphical user interface for multiresolution analysis. WV_PLOT_MULTIRES is accessible from the Visualize Menu of WV_APPLET.

Syntax

Result = WV_PLOT_MULTIRES(*Array* [, *X*] [, *Y*]
[, GROUP_LEADER=*widget_id*] [, TITLE=*string*] [, UNITS=*string*]
[, XTITLE=*string*] [, XUNITS=*string*] [, YTITLE=*string*] [, YUNITS=*string*])

Return Value

The returned variable is the Widget ID of the newly-created widget.

Arguments

Array

A one- or two-dimensional array of data to be analyzed.

X

An optional vector of uniformly-spaced values giving the location of points along the first dimension of *Array*. The default is 0, 1, 2,..., N_X–1, where N_X is the size of the first dimension.

Y

An optional vector of uniformly-spaced values giving the location of points along the second dimension of *Array*. The default is 0, 1, 2,..., N_Y–1, where N_Y is the size of the second dimension.

Keywords

GROUP_LEADER

The widget ID of an existing widget that serves as “group leader” for the newly-created widget. When a group leader is killed, for any reason, all widgets in the group are also destroyed.

A given widget can be in more than one group. The WIDGET_CONTROL procedure can be used to add additional group associations to a widget. It is not possible to remove a widget from an existing group.

TITLE

A scalar string giving the label to be used for the widget. The default is ‘MRes:’.

UNITS

A scalar string giving the units of *Array*.

XTITLE

A scalar string giving the label to be used for the first dimension.

XUNITS

A scalar string giving the units of *X*.

YTITLE

A scalar string giving the label to be used for the y-axis (for a 1D vector) or for the second dimension (for a 2D array).

YUNITS

A scalar string giving the units of *Array* (for a 1D vector) or the units of *Y* (for a 2D array).

See Also

WV_APPLET

WV_PWT

The WV_PWT function returns the partial wavelet transform of the input vector *A*. The transform is done using a user-inputted wavelet filter. WV_PWT is called by WV_DWT.

WV_PWT is based on the routine `pwt` described in section 13.10 of *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* (Cambridge University Press), and is used by permission.

Syntax

Result = WV_PWT(*A*, *Scaling*, *Wavelet*, *Ioff*, *Joff*)

Return Value

The result is an output vector of the same length as A, containing one stage of the pyramidal algorithm (Mallat 1989).

Arguments

A

The input vector. The length must be either less than four (4) or a power of two (2).

Scaling

A vector of scaling (father) coefficients.

Wavelet

A vector of wavelet (mother) coefficients.

Ioff

An integer that specifies the support offset for *Scaling*.

Joff

An integer that specifies the support offset for *Wavelet*.

Keywords

DOUBLE

Set this keyword to force the computation to be done in double-precision arithmetic.

INVERSE

If set, the inverse transform is computed. By default, the forward transform is computed.

See Also

WV_DWT

WV_TOOL_DENOISE

The WV_TOOL_DENOISE function runs the graphical user interface for wavelet filtering and denoising. WV_TOOL_DENOISE is accessible from the Tools Menu of WV_APPLET.

Syntax

Result = WV_TOOL_DENOISE(*Array* [, *X*] [, *Y*] [, GROUP_LEADER=*widget_id*] [, TITLE=*string*] [, UNITS=*string*] [, XTITLE=*string*] [, XUNITS=*string*] [, YTITLE=*string*] [, YUNITS=*string*])

Return Value

The returned variable is the Widget ID of the newly-created widget.

Arguments

Array

A one- or two-dimensional array of data to be analyzed.

X

An optional vector of uniformly-spaced values giving the location of points along the first dimension of *Array*. The default is 0, 1, 2,..., N_X–1, where N_X is the size of the first dimension.

Y

An optional vector of uniformly-spaced values giving the location of points along the second dimension of *Array*. The default is 0, 1, 2,..., N_Y–1, where N_Y is the size of the second dimension.

Keywords

GROUP_LEADER

The widget ID of an existing widget that serves as "group leader" for the newly-created widget. When a group leader is killed, for any reason, all widgets in the group are also destroyed.

A given widget can be in more than one group. The WIDGET_CONTROL procedure can be used to add additional group associations to a widget. It is not possible to remove a widget from an existing group.

TITLE

A scalar string giving the label to be used for the widget. The default is ‘Denoise:’.

UNITS

A scalar string giving the units of *Array*.

XTITLE

A scalar string giving the label to be used for the first dimension.

XUNITS

A scalar string giving the units of *X*.

YTITLE

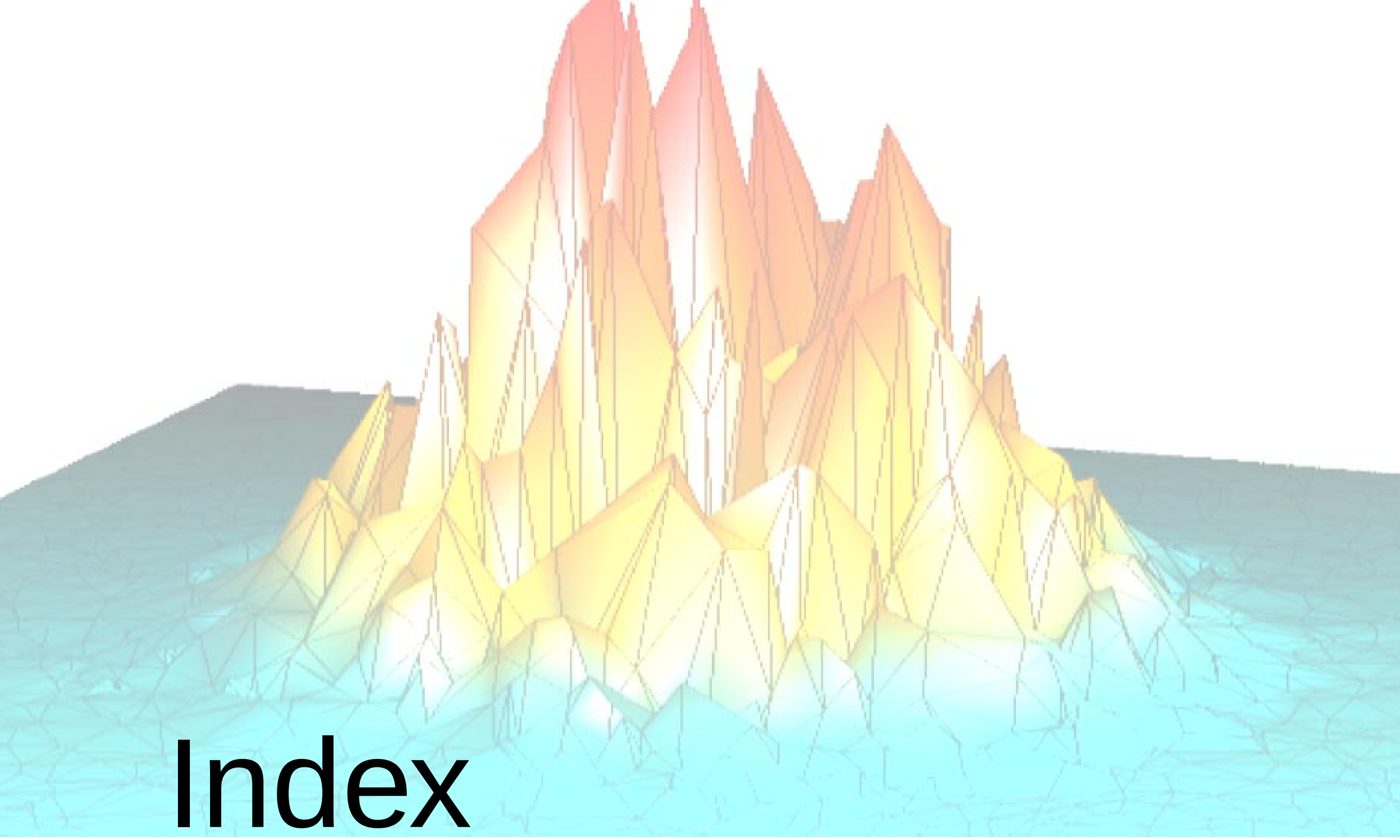
A scalar string giving the label to be used for the y-axis (for a 1D vector) or for the second dimension (for a 2D array).

YUNITS

A scalar string giving the units of *Array* (for a 1D vector) or the units of *Y* (for a 2D array).

See Also

WV_APPLET



Index

Symbols

!PATH system variable, 11

A

ARRAY keyword, 56
ASCII files, importing arrays, 25
ASCII files, independent variable, 25
ASCII_TEMPLATE, 25

B

binary files, importing, 25
BINARY_TEMPLATE, 25

C

cascade plot, *see* multiresolution analysis
coefficient power plot, denoising, 38
coiflet, *see* wavelet functions
color tables in wavelet power spectrum, 32
commands
   list of, 54
   WV_APPLET procedure, 56
   WV_CW_WAVELET function, 58
   WV_DWT function, 61
   WV_FN_COIFLET function, 63
   WV_FN_DAUBECHIES function, 65
   WV_FN_HAAR function, 67
   WV_FN_SYMLET function, 69
   WV_IMPORT_DATA procedure, 71
   WV_IMPORT_WAVELET procedure, 73
   WV_PLOT_MULTIRES function, 76

WV_PLOT3D_WPS function, 74
WV_PWT function, 78
WV_TOOL_DENOISE function, 80
compact support, wavelet functions, 28
continuous wavelet functions, 28
continuous wavelet transform, 28, 44
contours in wavelet power spectrum, 33
copy to clipboard, 32
cumulative power plot, denoising, 38
cutoff in decibels, wavelet power spectrum, 33

D

datasets
mathematical expressions, 23
selecting variables, 23
variable information fields, 20
Daubechies, *see* wavelet functions
decibel scaling, wavelet power spectrum, 33
denoising
coefficient threshold, 39
color scaling for images, 37
cumulative power threshold, 39
Denoise Tool, 37
theory, 47
wavelet coefficient display method, 38
see also WV_TOOL_DENOISE function
detail (band-pass) multiresolution plots, 35
DIALOG_READ_IMAGE, 26
discrete wavelet functions, 28
discrete wavelet transform, 28, 44, 61
DLM (Dynamically Loadable Module), 11
DOUBLE keyword, 62, 78
drag quality, Wavelet Power Spectrum, 32

E

energy scaling, wavelet power spectrum, 32

F

filtering, *see* denoising

G

GROUP_LEADER keyword, 56, 74, 76, 80

H

Haar, *see* wavelet functions

I

IDL Wavelet Toolkit
features, 9
file organization, 11
main window, 14
menus
Edit, 17
File, 16
Help, 19
Tools, 18
Visualize, 18
starting, 14, 56
status bar, 15
structure, 11
toolbar, 14
image compression, *see* denoising
image files, importing, 26
importing data from
ASCII files, 25
binary files, 25
command line, 26, 71
image files, 26
WAV files, 26
importing wavelet functions, 29, 73
INVERSE keyword, 62, 79

L

lego-style surface, 34
localization of wavelet functions, 44

M

Macintosh mouse differences, 34
magnitude scaling, wavelet power spectrum, 33
Mallat, *see* pyramidal algorithm
mathematical expressions, *see* datasets
menus
  IDL Wavelet Toolkit, 16
  WV_PLOT3D_WPS function, 32
MESSAGE_OUT keyword, 71
MRI (magnetic-resonance image) example, 48
multiple channels in WAV files, 26
multiresolution analysis
  cascade plot, 35
  example, 51
  see also WV_PLOT_MULTIRES function

N

NO_SPLASH keyword, 56
noise removal, *see* denoising
nonorthogonal wavelet functions, 28

O

orthogonal wavelet functions, 28

P

parameter passing in user tools, 41
PARENT keyword, 71
partial wavelet transform,
  see wavelet transform
PCM format for WAV audio files, 26
percent difference, denoised, 40
plotting
  multiresolution analysis, 35, 76
  wavelet power spectrum, 31, 74
power scaling, wavelet power spectrum, 33
pyramidal algorithm, 44, 61, 78

R

regularity of wavelet functions, 29
RESET keyword, 73
reset view, wavelet power spectrum, 32
rms difference, denoised, 40
rotation of wavelet power spectrum, 34
rough (high-pass) multiresolution plots, 35

S

scaling functions, 27
significance testing, 44
smooth (low-pass) multiresolution plots, 35
structure tags for WV_IMPORT_DATA, 71
surface style, wavelet power spectrum, 33
symlet, *see* wavelet functions
symmetry of wavelet functions, 28

T

threshold for denoise, hard versus soft, 39, 47
TITLE keyword, 58, 75, 77, 81
tools
  denoise, 37, 80
  user-defined, 18, 41
TOOLS keyword, 41, 56
translation of wavelet power spectrum, 34
true color (24-bit) images, 26

U

UNAME keyword, 58

UNITS keyword, 75, 77, 81
UVALUE keyword, 58

V

vanishing moments, wavelet functions, 29
variable information, *see* datasets
variable selection, *see* datasets
viewing
  data values, 17
  wavelet functions, 27
VRML (Virtual Reality Markup Language), 32

W

WAV audio files, importing, 26
wavelet functions
  built-in
    coiflet, 63
    Daubechies, 65
    Haar, 67
    symlet, 69
  compact support, 28
  discrete versus continuous, 28
  family, 59
  order, 59
  orthogonal versus nonorthogonal, 28
  regularity, 29
  symmetry, 28
  user-defined, 29, 73
  vanishing moments, 29
  viewing, 27
wavelet power spectrum
  energy scaling, 32
  example, 45
  plotting method, 45
  rotation and translation, 34
  saving as VRML, 32
  theory, 45
  see also WV_PLOT3D_WPS function
wavelet transform
  continuous, 28, 44
  discrete, 28, 44, 61
  partial, 78
WAVELETS keyword, 57, 59
WV_APPLET procedure, 56
WV_CW_WAVELET function, 27, 58
  GET_VALUE, 59
  SET_VALUE, 59
  widget events generated, 59
WV_DWT function, 61
WV_FN_COIFLET function, 63
WV_FN_DAUBECHIES function, 65
WV_FN_HAAR function, 67
WV_FN_SYMLET function, 69
WV_IMPORT_DATA procedure, 26, 71
WV_IMPORT_WAVELET procedure, 30, 73
WV_PLOT_MULTIRES function, 76
WV_PLOT3D_WPS function, 74
  File Menu, 32
  Options Menu, 32
WV_PWT function, 78
WV_TOOL_DENOISE function, 80

X

XTITLE keyword, 75, 77, 81
XUNITS keyword, 75, 77, 81

Y

YTITLE keyword, 75, 77, 81
YUNITS keyword, 75, 77, 81