

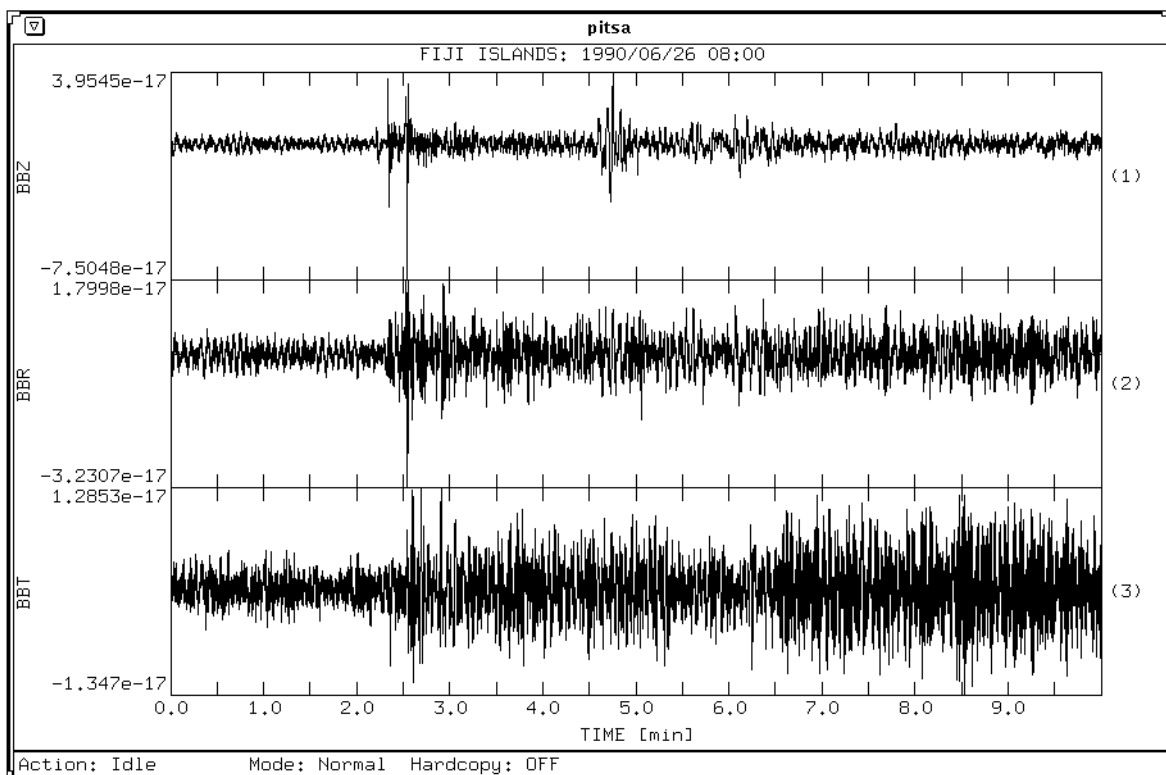
PITSA

by

FRANK SCHERBAUM
JAMES JOHNSON
ANDREAS RIETBROCK

Additional Contributions by

ELKE SCHMIDTKE
JOACHIM WASSERMANN
MATTHIAS HARDT
MATTHIAS OHRNBERGER



Version 5.0

January, 1999

Table of Contents

CHAPTER 1 *Introduction*

1.1	The Growth of an Idea	2
1.2	Acknowledgments	3

CHAPTER 2 *Software Installation*

2.1	User Installation	1
------------	--------------------------------	----------

CHAPTER 3 *Basic Operations*

3.1	Why You Should Read the Rest of this Chapter.....	1
3.2	Typeface Conventions	1
3.3	Starting PITSA	1
3.3.1	Colormap Allocation Problems	3
3.3.2	Command Line Arguments.....	3
3.4	Message Windows	4
3.5	Selecting a Menu Item.....	4
3.6	Dialog Boxes and Text Input	5
3.6.1	Date-Time Formats	6
3.7	Dialog Menus	7
3.8	Channel Lists.....	7
3.9	Applying Tools to Multiple Channels	7
3.10	Accepting New or Processed Traces	8
3.11	Status Line	8
3.12	Moving and Changing the Size of the Main Window	9
3.13	Function keys	9
3.14	Last Command	9
3.15	Quit.....	9

CHAPTER 4 *Setup*

4.1	Color Settings	1
------------	-----------------------------	----------

4.1.1 Example: Changing the Drawing Color 1.....	1
4.2 Program and Track Mode.....	3
4.2.1 Re-prompting for Input with the “?” Key	4
4.2.2 Input from a Text File with the “~” Key	4
4.2.3 Looping in Track Mode.....	4
4.2.3.1 Example: Looping with Track Mode.	5
4.2.4 Using the Default Value of Input with the “\$” Key.....	6
4.3 Saving Images from the Screen.....	6
4.3.1 Hardcopy Mode	6
4.3.2 Hardcopy Destination	7
4.3.2.1 Writing Hardcopies to Screendump Files	7
4.3.3 Use of the F2 and F3 Function Keys.....	8
4.3.4 Doubling the Font Size	8
4.4 Plot All Flag	10

CHAPTER 5 *Files and Traces*

5.1 Data Traces within PITSA.....	1
5.2 Supported File Formats.....	2
5.2.1 The ISAM Database File System	2
5.2.2 ASCII Text File Format.....	2
5.2.3 GSE Format.	2
5.3 Changing Directories.....	3
5.4 Retrieve Files	3
5.4.1 Reading Data From ISAM File Systems	3
5.4.1.1 Example: Reading ISAM files.	4
5.4.2 Reading Plain ASCII Files.....	5
5.4.2.1 Single Trace ASCII File.	7
5.4.2.2 Multiple Trace ASCII File.	7
5.4.3 Reading GSE Format	8
5.5 Save Files.....	8
5.5.1 Writing an ISAM File System	9
5.5.2 Writing ASCII Files	9
5.5.3 Writing GSE Files	9
5.6 Add Traces From File	9
5.7 Copy Traces	10
5.8 Rearrange Traces.....	11
5.9 Delete Traces.....	12

CHAPTER 6 *Routine Tools*

6.1 Zoom	2
6.1.1 Tapered vs. Untapered Windows	2
6.1.1.1 Tapering Functions	2
6.1.2 Channel List for Zooming	2
6.1.3 Zoom Window Selection Methods	2
6.1.3.1 Single Cursor	3
6.1.3.2 Double Cursor.....	3
6.1.3.3 Index	3
6.1.3.4 Accept window as is	3

6.1.4 Accepting the Zoomed Window	4
6.1.5 Selecting the Tapering Fraction	4
6.2 Edit	6
6.2.1 Point Editing	6
6.2.2 Glitch Editing	8
6.2.3 Despiking	9
6.3 Instrument Simulation	9
6.3.1 Please Use These Tools with Great Care!	10
6.3.2 Instrument Response Files and Formats	10
6.3.3 Choose Instrument Menu	11
6.3.4 Multiple Traces	11
6.3.5 Example: Creating an all-pass instrument response file	11
6.4 Phase Picking	12
6.4.1 Auto Pick Phase(s)	13
6.4.2 Adjust Phase(s)	14
6.4.2.1 <i>Working with Many Traces</i>	14
6.4.2.2 <i>Controlling the Zoom Box</i>	14
6.4.2.3 <i>Controlling Which Traces are Displayed for Phase Adjustment</i>	15
6.4.2.4 <i>Picking Phases in Adjust Phase(s)</i>	16
6.4.2.5 <i>The display of overlapping phase labels</i>	19
6.4.2.6 <i>The active phase</i>	20
6.4.2.7 <i>Phase Labels</i>	20
6.4.2.8 <i>The Phase Menu</i>	22
6.4.3 Show False Polarities	29
6.4.4 Show Greatest Residuals	29
6.4.5 Sort Traces by Distance	29
6.4.6 Sort Traces by Azimuth	29
6.4.7 Output Phase(s)	29
6.4.7.1 <i>Processing of HYPO71 Phase Output Files</i>	30
6.4.7.2 <i>Processing of phase output files in the ISOP format</i>	30
6.4.7.3 <i>Processing of phase output files in the Long format</i>	30
6.4.7.4 <i>Phase Output Status Message</i>	30
6.4.8 Output Station(s)	31
6.4.9 Run Location Program	31
6.4.9.1 <i>Running HYPO71</i>	31
6.4.9.2 <i>Solving Problems with HYPO71</i>	32
6.4.10 Clear Phases	32
6.4.10.1 <i>Clearing All Phases</i>	32
6.4.10.2 <i>Clearing Selected Phases</i>	32
6.4.11 Delete Traces	33
6.4.12 Plot All	33
6.4.13 DONE	33
6.5 Magnitudes	34
6.6 Integrate	40
6.7 Differentiate	41
6.8 Baseline Corrections	42
6.8.1 Running Average (Remove or Keep)	43
6.8.2 Offset	43
6.8.3 Linear Trend	43
6.8.4 Linear Regression	44

CHAPTER 7 *Advanced Tools*

7.1 Filter	2
7.1.1 Forward-Backwards (Bilateral) Filtering	2
7.1.2 Butterworth Band Pass	2
7.1.3 Butterworth Low Pass	4
7.1.4 Butterworth High Pass	5
7.1.5 De-ringing Filter	5
7.1.6 Gaussian Band Pass	6
7.1.7 Notch Filter	7
7.1.8 Seidl's simulation	8
7.1.9 Difference Equation	8
7.1.10 GSE Calibration File	10
7.1.11 Evaluate GSE Response	10
7.2 Spectrum	11
7.2.1 Number of Points Used for FFT	12
7.2.2 FFT	12
7.2.3 Inverse FFT	15
7.2.4 Multiplex to FFT Format	16
7.2.5 Demultiplex from FFT Format	17
7.2.6 Maximum Entropy Spectrum	17
7.2.7 POW Spectrum	18
7.2.8 Wavelet Transform	19
7.2.8.1 <i>Wt</i>	20
7.2.8.2 <i>inverse Wt</i>	24
7.3 Cross Correlation	25
7.4 (De)convolution	26
7.4.1 Convolution (FREQ)	26
7.4.2 Convolution (TIME)	27
7.4.3 Deconvolution	27
7.5 Particle Motion	28
7.5.1 2D Snake	29
7.5.1.1 <i>Estimating polarization angles from the 2-D display</i>	31
7.5.1.2 <i>Alternative displays of the particle motion snake</i>	33
7.5.1.3 <i>Phase picking from the particle motion plot</i>	33
7.5.2 3D Snake	33
7.6 Hilbert Transform (Envelope)	37
7.7 Rotate Components	38
7.7.1 Back-azimuth	38
7.7.2 Incidence Angle	39
7.7.3 Rotate N',E' -> N,E	39
7.7.4 Rotate N,E -> R,T	39
7.7.4.1 <i>Determination of apparent incidence angle</i>	39
7.7.5 Rotate Z,N,E -> L,Q,T	41
7.8 Polarization Filter	42
7.8.1 Montalbetti and Kanasevich Filter	42
7.8.2 Vidale Filter	44
7.9 Cross Spectrum	46
7.10 Coherence Spectrum	47
7.11 Response Spectrum	48

7.12 Spectral Ratio	48
----------------------------------	-----------

CHAPTER 8 *Utilities*

8.1 Test Signals.....	1
8.1.1 Add Noise.....	2
8.1.2 Sine/Cosine.....	3
8.1.3 Spike(s)	3
8.1.4 Constant	4
8.1.5 Brune Source Signal.....	5
8.1.6 Mixed Signal.....	5
8.1.7 Ricker Wavelet	6
8.1.8 Seismometer	7
8.1.9 Chirp Signal	7
8.1.10 RPN Equation	8
8.2 Trace Utilities.....	9
8.2.1 Single Channel Utilities	10
8.2.2 Double Channel Utilities	11
8.2.3 Triple Channel Utilities	12
8.2.4 Scale Traces Together.....	12
8.2.5 Re Autoscale Traces.....	13
8.2.6 Change Axis Type	14
8.2.7 Stack Traces	14
8.2.7.1 Receiver Beam	14
8.2.7.2 Source Beam.....	14
8.2.7.3 Plain Sum	15
8.2.8 Unify Traces	15
8.2.9 Normalize Traces.....	15
8.3 Header Access	15
8.3.1 Station	15
8.3.2 Event	16
8.3.3 Record.....	17
8.3.4 Plot.....	17
8.3.5 Output	18
8.4 Phase Utilities.....	18
8.4.1 Mark slowness window at receivers.....	18
8.4.2 Align phases for receiver slowness.....	19
8.4.3 Cut phase windows	19
8.4.4 Relative onset time determination	19
8.5 Resampling.....	22
8.5.1 Simulate discretization in time	23
8.5.2 Reconstruction of discretized signals	24
8.5.3 Integer decimation	25
8.5.4 Resampling of interpolation polynomial (4-deg)	25
8.5.5 Simulate A/D Converter	26
8.5.5.1 How Gain-Ranging Works	27
8.5.5.2 Input Parameters	27

CHAPTER 9 *Special Plots*

9.1	Epi-Distance Plot	1
9.2	Hypo-Distance Plot.....	3
9.3	Receiver Array Vespagram.....	3
9.4	Receiver Array Slowness Power Spectrum	4
9.5	Superimpose Traces.....	7

APPENDIX A *References*

APPENDIX B *The PITSA Configuration File*

B.1	General Parameters:	2
B.1.1	BACKGROUND_COLOR.....	2
B.1.2	CANVAS_HEIGHT.....	2
B.1.3	CANVAS_WIDTH	2
B.1.4	DRAW_1_COLOR	2
B.1.5	DRAW_2_COLOR	2
B.1.6	COLOR_MODE	3
B.1.7	KILL_NOTE.....	3
B.1.8	MAX_NOCH	3
B.1.9	MAX_TRLEN.....	3
B.1.10	MENU_POST	3
B.1.11	MENU_SLCT	3
B.1.12	SET_FOCUS.....	4
B.1.13	SET_FOCUS_LOOP	4
B.1.14	TEXT_MARG.....	4
B.1.15	PICK_ADJ_WINDOW.....	4
B.2	Automatic Phase Picker Configuration Parameters:	4
B.2.1	P_DUR.....	4
B.2.2	PRESETDUR	5
B.2.3	TDOWNMAX.....	5
B.2.4	TUPEVENT	5
B.2.5	THRSHL1, THRSHL2.....	5
B.3	Configuration Parameters for GSE writing.....	5
B.3.1	GSE_IOR2	5
B.3.2	GSE_FORM	5
B.3.3	GSE_NDIFF	6
B.3.4	GSE_RESOL.....	6

APPENDIX C *Printer Configuration File*

C.1	Printer Configuration Parameters	1
C.1.1	#UPPER_LEFT_CORNER.....	1
C.1.2	#UPPER_RIGHT_CORNER	1
C.1.3	#LOWER_LEFT_CORNER.....	1
C.1.4	#PLOT_HEAD	2
C.1.5	#PLOT_TAIL	2
C.1.6	#PLOT_MOVETO.....	2
C.1.7	#PLOT_LINETO.....	2

C.1.8 #PLOT_STROKE.....	2
C.1.9 #PLOT_MAX_LINE.....	3
C.1.10 #PLOT_START_AXIS	3
C.1.11 #PLOT_END_AXIS	3
C.1.12 #PLOT_START_CHAR	3
C.1.13 #PLOT_END_CHAR	3
C.1.14 #PLOT_START_STRING.....	3
C.1.15 #PLOT_END_STRING.....	3
C.1.16 #PLOT_STRING_HEIGHT_X.....	4
C.1.17 #PLOT_STRING_HEIGHT_Y.....	4
C.1.18 #PLOT_COMMENT_START	4
C.1.19 #PLOT_COMMENT_END	4
C.1.20 #PLOT_COLOR1	4
C.1.21 #PLOT_COLOR2	4
C.1.22 #PLOT_OUTPUT_FILE_NAME	4
C.1.23 #PLOT_PRINT_COMMAND	4
C.1.24 #PLOT_RM_COMMAND.....	5
C.1.25 #PLOT_CAT_COMMAND.....	5

APPENDIX D *Command Files*

APPENDIX E *GSE waveform format*

APPENDIX F *GSE Instrument Response File*

APPENDIX G *Automatic Phase Picking*

APPENDIX H *Problem-Solving with HYPO71*

PITSA is a flexible and powerful system for doing signal processing on digital seismic data. It has been used extensively in research and education, and can be used also for routine processing (such as is required at a seismic observatory). However, the extreme flexibility of PITSA makes it less suitable for this last task than a dedicated program (if one is available!).

The user can perform data editing functions such as trend removal and removal of spikes and glitches. PITSA offers a number of utilities for manipulation of seismic traces, such as shifting or scaling of traces, adding and concatenating traces, and stacking. For enhanced visualization of seismic traces, the user can create seismogram section plots with hypocentral distance on the horizontal axis and time on the vertical axis. PITSA automatically adjusts its scaling to different sampling rates.

PITSA contains extensive filtering functions, including Butterworth bandpass, low-pass, and highpass filters, Gaussian bandpass filters, notch filters, de-ringing filters, and polarization filters. Additional filters can be defined by the user by providing either the ARMA coefficients for the filter difference equation or the corresponding pole and zero distribution in the complex plane.

PITSA can be used to estimate earthquake magnitudes, integrate or differentiate seismic traces, analyze particle motions in 2 or 3 dimensions, perform convolution or deconvolution, correct for the transfer function of instruments and simulate arbitrary instruments from broadband signals, rotate components, perform baseline corrections, calculate wave envelopes, cross-correlate traces, and calculate FFTs and inverse FFTs. In addition, the user will find tools for cross-spectral analysis, coherence estimation, and the calculation of maximum entropy spectra.

Several tools are provided for onset time determination, including standard interactive phase picking, phase picking from particle motion diagrams and automatic P wave determination using the P phase picker of Baer and Kradolfer (1987). Phase picks which have been made automatically can be edited interactively by the user. The onset times can be directly used for locating earthquakes using HYPO71PC (Lee and Valdés, 1989).

In addition to its use for processing actual data, PITSA can create various test signals, resample traces and simulate the performance of gain-ranging A/D converters. This makes PITSA a versatile tool for demonstrating the principles of digital signal processing in a classroom environment. These capabilities are used extensively in

the book, **Short Course on First Principles of Digital Signal Processing for Seismologists** by Frank Scherbaum (1993).

Internally, PITSA uses a data format designed for the specific needs of earthquake seismology. In addition to the trace data, PITSA stores logistical information on each trace (plotting information as well as event and station information). The information is kept in ISAM (Indexed Sequential Access Method) file systems for fast access. In addition, the input of plain ASCII files and files using the GSE (Group of Scientific Experts) format is possible.

PITSA is “trainable”, which means the user can teach it to repeat certain tasks without having to input redundant information. It is a common requirement of seismological research to process a series of traces in an identical manner. For example, it could be necessary to load a trace, perform a baseline correction, integrate it, calculate an FFT, and finally create a hardcopy. In its *Program mode*, PITSA records user keystrokes and menu traverses in a file. In its *Track mode*, PITSA uses this file to repeat the same sequence of operations except for the cases where user interaction is required. The user can even instruct PITSA to obtain input from a file during this process. For example, instead of having PITSA ask repeatedly for the file name of a data file to process, the file names could be read from a designated input file. Hence, for repetitive tasks the amount of user interaction—and the likelihood of errors—can be kept to a minimum.

PITSA has a menu-based user interface which guides the user through the many options available. Screen colors and drawing colors can be changed to suit the user. Dialog boxes and pop-up menus can be moved around freely to make optimal use of the screen. Hardcopies of the screen can be made at any time and sent to a printer or to a file.

1.1 *The Growth of an Idea*

The idea for PITSA started to develop during the spring of 1987 while one of us (Frank Scherbaum) was working at the Cooperative Institute for Research in Environmental Sciences (CIRES) in Boulder, Colorado, USA. Facing the need to analyze spectral parameters for about 3000 microearthquake records, the concept—an interactive set of signal processing tools which could be combined in an arbitrary fashion and programmed to do seismological routine analysis on a PC (running DOS of course) with minimal human interaction—popped up during a hike in the Indian Peaks Wilderness in Colorado. PITSA, for **P**rogrammable **I**nteractive **T**ool-box for **S**eismological **A**nalysis, sounded like a good name for such a satisfying idea. The first version of PITSA was programmed during the summer of 1987 and was used in the context of performing an inversion of spectral parameters of microearthquakes in Hawaii (Scherbaum, 1990; Scherbaum and Wyss, 1990).

In 1988, during a subsequent stay at the National Research Center for Disaster Prevention in Tsukuba, Japan, PITSA got its first major overhaul. Some of the concepts used in the original version (e.g., static memory allocation, EGA-specific graphics toolkit) put too many constraints on the type of computer that could be used and the length of the data which could be analyzed. PITSA was partially rewritten in the context of doing a study of regional events of much longer trace duration than the microearthquake signals analyzed previously (Scherbaum and Sato, 1991). The main changes at that time were the implementation of dynamic memory allocation and the use of Metagraphics' MetaWINDOW™/PLUS graphics toolkit for the graphics functions.

During the summer of 1989, at the 25th General Assembly of IASPEI in Istanbul, Turkey, PITSA was demonstrated for the first time to a larger audience. The decision to produce a version of PITSA which could be distributed to other people was strongly motivated by the response to these demonstrations. Recognizing the variety of computer platforms people were using, and in anticipation of the growing importance of Unix workstations, it also seemed to be a worthwhile goal to try to make PITSA more hardware independent. Since the first two versions of PITSA were born out of the need to perform special tasks within the context of particular studies, most of the programming had been done in a 'quick and dirty' way. If Jim Johnson had not come to Munich in the spring of 1990 to start working on the X-Window version of PITSA, it certainly would have stayed that way.

It quickly became clear that it would not be possible to support a variety of computer architectures with a common code base without making major changes to the existing DOS-specific code. Following Jim's re-design of the plotting interface, virtually the entire program was rewritten to his strict standards of coding and error checking. In addition, we decided to use a virtual memory management system under DOS. Although we were largely successful in creating a common code base which would support both the DOS and Unix operating systems, management of the code size under DOS became a nightmare, threatening further development of PITSA. Therefore when the DOS version of PITSA was published in Volume 5 of the IASPEI Software Library, we decided to discontinue development of PITSA for the DOS operating system.

Current development of PITSA is focused solely on Unix-like operating systems using the X-Window graphic environment. This does not prevent the use of PITSA by people with PC hardware, however. By using the free Linux operating system, PCs can now run the latest version of PITSA, with capabilities far beyond those of the old DOS version.

This edition of the PITSA Manual is the first update in nearly 5 years. In that time, PITSA has changed tremendously, mostly through the hard efforts of Andreas Rietbrock, Joachim Wassermann, Matthias Ohrnberger, and Matthias Hardt, who developed many new tools to support various research efforts. Typical of graduate students, however, their efforts at documentation have not always been as keen as their devotion to their research. In 1997 the PITSA Team moved from the University of Munich to a new home at the University of Potsdam, where maintenance and development of PITSA is being turned over to a professional programmer, and PITSA's code is once more being cleaned up and revised. Some of the tools which were familiar in previous versions of PITSA have disappeared, for various reasons. An updated Manual was desperately needed!

1.2 Acknowledgments

Many of our colleagues, friends, and students have contributed to the development of PITSA through their encouragement, supportive criticism, suggestions and proofreading of the manuscript: Manfred Baer, Paul Bash, Nicolas Deichman, Thomas Dreher, Julien Frechet, Manfred Henger, Thomas Ketter, Michael Korn, Willie Lee, Anthony Lomax, Hansruedi Maurer, Dieter Mayer-Rosa, Christian Rocken (we are still waiting for the interpolator), Peter Rummel, Haruo Sato, Eberhard Schmedes, Dieter Stoll, and Peter Troitsky. We owe special thanks to Matthias Hardt, Andreas Rietbrock, and Joachim Wassermann for their help with debugging and but also with the coding of some of the tools.

Jim Johnson's stay at the Institut für Allgemeine und Angewandte Geophysik der Ludwig Maximilians Universität München was made possible through funding by Lennartz electronic, Tübingen, FRG, and the Institute of Geophysics of the ETH Zürich, Switzerland.

In 1991 IRIS became involved in the project by providing financial support to Jim Johnson and by loaning us a Sun workstation and a large hard disk. The University NAVSTAR Consortium (UNAVCO) also began to provide indirect support of PITSA by providing Jim Johnson with office space and computer support. Many thanks go out to Scott Watanabe and Myron McCallum of UNAVCO for helping Jim deal with numerous UNIX, X-Window, C and System Administration problems.

Personal Note by Frank Scherbaum: While I was fighting the memory- and code size problems on the PC and trying to keep my sanity in the presence of DOS, I was more than once close to giving up the whole project. If there hadn't been the support by Marie-Jose, the talks with Jim on martial arts and computer programming, and the music of Tom Waits (Well there's diamonds on my windshield and these tears from heaven...) I certainly would have given up.

Personal Note by James Johnson: I would like to thank Frank for getting this whole project going. I have learned a lot and will learn a lot more as I work on this project. Sylvia Chamberlain had to put up with my obsessive hours behind the computer screen and without her support and understanding, I would have never survived.



In the typical multi-user computer network PITSA should be installed in a system area by the system administrator so that all users of the system will have access to a single copy of the programs. In a single-user system PITSA could be installed in a users own home area. In this example, we will assume that the system administrator will be installing PITSA into the directory `/usr/home/pitsa`.

The PITSA program files and sample data files are distributed as a compressed tarfile named of `pitsa.dist.050193.tar.Z` where 050193 is the date of the release. The actual date of the release you use and the exact contents of the compressed tar files will vary slightly from the example given here. The file can be un-compressed and un-tarred with the following Unix commands:

```
cd /usr/home/pitsa
zcat pitsa.dist.050193.tar.Z | tar xvf -
chmod -R a+rX.
```

The above commands assume that the directory `/usr/home/pitsa` exists and that the tarfile `pitsa.dist.050193.tar.Z` has been moved to that directory. After the files have been un-tarred, the PITSA distribution file can be removed.

In the event that PITSA comes on some other media, follow the instructions that come with it.

2.1 User Installation

Each user who wants to use PITSA will have to make a few simple changes to her own environment, specifically, to the `.cshrc` file in her home directory. Because its name (like those of several other system files) starts with a dot, `.cshrc` is normally invisible. The Text Editor tool included with OpenWindows can open these files for editing, using the option to show invisible files. A similar text editor should be available under any other operating system.

First, the following lines should be added to the users `.cshrc` file. The lines beginning with `#` are comment lines, so they can be omitted. These environment variables tell the operating system where PITSA and its various supporting files and secondary programs (e.g., HYPO71) reside on the computer's file system. We assume that

PITSA has been installed by the system administrator in the directory `/usr/home/pitsa`.

```
# to minimize typing
setenv PITSAHOME /usr/home/pitsa
# puts PITSA in users path
set path = ($path $PITSAHOME)
# path for pitsa.cfg file
setenv PITSA_CONFIG_PATH_ENV $PITSAHOME/
# path for printer definition files
setenv PITSA_PRINTDEF_PATH_ENV $PITSAHOME/pltdef/
# default printer definition file
setenv PITSA_PRINTDEF_NAME_ENV 8X11_landscape.PS
# path to HYP071
setenv PITSA_HYP071_PATH_ENV $PITSAHOME/hypo71
```

Now the modified `.cshrc` file needs to be re-introduced to the operating system by using the UNIX command `source`. This only needs to be done during a login session in which the `.cshrc` file has been changed, because the operating system checks the `.cshrc` file at each login

```
source ~/.cshrc
```

Now you are ready to run PITSA.



3.1 Why You Should Read the Rest of this Chapter

Even if you are not a dedicated reader of manuals, you should take the time to read at least this chapter. It will give you the essential information about PITSA's user interface and basic operations which are common to many of PITSA's tools. With this knowledge you will be able to explore the many tools and features of PITSA efficiently.

3.2 Typeface Conventions

We will use the following font conventions in this Manual:

Table 3-1. Typeface conventions.

Typeface	Usage
Plain Courier	Used for user input, computer output, names of data files and directories, content of data files.
<i>Italicized Times Roman</i>	Used to indicate PITSA menu choices or options.

3.3 Starting PITSA

Assuming someone has installed PITSA according to the instructions in Chapter 2, you are now ready to start PITSA. You must be in OpenWindows (or some X-Window environment). Then type the following command (in a Command Tool, Shell Tool or xterm window):

```
pitsa&
```

The "&" starts PITSA in the background and returns a prompt in the Command Tool window so you can issue additional commands. Depending on the contents of

the PITSA configuration file (Appendix B), some text similar to that shown below should appear in the window where PITSA was started:

```
PITSA Version 5.0 from Thu Feb 02 15:16:48 MET DST 1999
now loading.....
```

```
by
Frank Scherbaum, James Johnson and Andreas Rietbrock
```

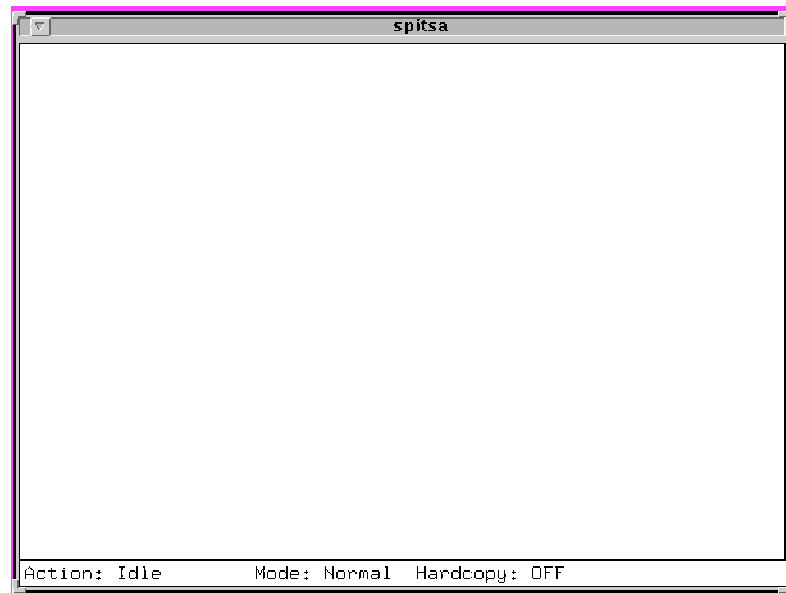
```
Additional contributions by
Elke Schmidtke, Joachim Wassermann, Matthias Hardt and
Matthias Ohrnberger
```

```
HEADER: 11744 2936
Configuration path: /usr/home/pitsa/
Hypo71 path: /usr/home/pitsa//hypo71
LocSAT path: ./
MENU_POST: <Btn3Up>
MENU_SLCT: <Btn1Up>
TEXT_MARG: 4
CANVAS_WIDTH: 600
CANVAS_HEIGHT: 400
DRAW_1_COLOR: black
DRAW_2_COLOR: red
BACKGROUND_COLOR: white
KILL_NOTE: FALSE
SET_FOCUS: TRUE
SET_FOCUS_LOOP: 10
MAX_NOCH: 30
MAX_TRLIN: 8192000
Printer definition file path: /usr/home/pitsa//
pltdef/
Printer definition file name: 8X11_landscape.PS

Screen size is: 1152 X 900
```

Then the PITSA main window should appear (Figure 3-1).

Figure 3-1. The PITSA window at startup.



Admittedly, it is not very exciting, but soon you will learn how to add some seismological content. The identification “spitsa” in the title bar of Figure 3-1 simply reflects the name given to the executable file of the developmental version of PITSA used to make this screen-shot.

3.3.1 Colormap Allocation Problems

When you issue the command to start PITSA in a Command Tool window, you may also see a long list of warning messages like:

```
Warning: Cannot allocate colormap entry for "gray79"
Warning: Cannot allocate colormap entry for "gray78"
Warning: Cannot allocate colormap entry for "gray77"
...
```

The reason for these warnings is that another program (FrameMaker is one known culprit) which is already running on your workstation has allocated these colormap entries. This can cause display problems in PITSA, so if it occurs you should logout and login again before starting PITSA. If you need to run PITSA simultaneously with an offending program, start PITSA first. The other program will probably be able to build its own colormap if it encounters a conflict.

3.3.2 Command Line Arguments

There are two command line arguments that can be specified when starting PITSA. If the `-c` option is used, the argument following the `-c` is taken to be the name of a command file (Appendix D) to execute. After the command file is finished, PITSA will be in its home state. The `-C` option works in the same way except that after the command file is finished, PITSA will terminate. An example might be:

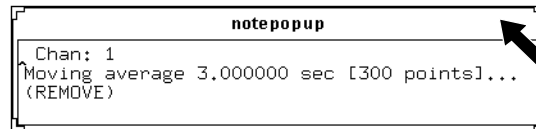
```
my_machine:/usr/a_user% pitsa -c load_data.cmd
```

This example would execute the command file `load_data.cmd` (see Section 4.2) before allowing the user to make menu selections.

3.4 Message Windows

On occasion PITSA will display a message window providing special information (e.g., an error message, results of an analysis, etc.).

Figure 3-2. A message window.



Messages sent to the authors of PITSA should always be non-volatile!

Message windows come in two flavors. If a message vanishes automatically after a few seconds (“volatile message”, no further interaction is required. If a message does not vanish automatically after a few seconds (“non-volatile message”, it expects to be acknowledged by clicking the CONTINUE button.

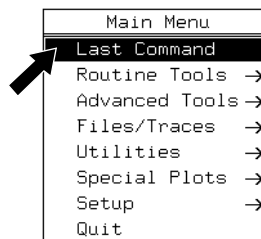
Both volatile and non-volatile messages can be moved by clicking the left mouse button in the border of the message and dragging it to a new location. In Figure 3-2 above, the arrow indicates one place the mouse pointer could be in order to move a message.

3.5 Selecting a Menu Item



PITSA’s tools are all accessed through menu commands, but when the program is started (Figure 3-1), no menus are visible. This is because PITSA uses a pop-up style main menu. To see this menu, just click the right mouse button while the mouse pointer is somewhere within the PITSA main window. Press then release the mouse button; the menu will not appear until the mouse button is released. When this is done, the following pop-up main menu will appear (Figure 3-3):

Figure 3-3. PITSA’s main menu.

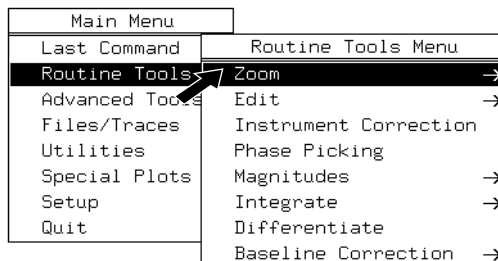


In the above figure, the menu item *Last Command* is highlighted. This is because the mouse pointer is inside the menu items box. The highlighted command may be activated by clicking the left mouse button.

Except for the *Last Command* and *Quit* selections, all the menu items in the main menu have sub-menus. In order to access these sub-menus, PITSA uses a ‘walking menus’ approach. If the mouse pointer is moved (no buttons need be pressed) over

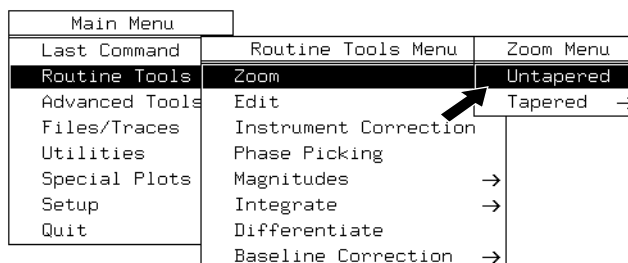
the arrow to the right of the menu item, a new menu box will appear as in Figure 3-4.

Figure 3-4. The sub-menu of *Routine Tools* has been activated by moving the mouse pointer over the arrow to the right of the *Routine Tools* selection.



In the above figure, the *Routine Tools* sub-menu has been activated. Notice that some of the sub-menu selections themselves have sub-menus (which we resist calling sub-sub-menus). If the left mouse button were now clicked while in the *Routine Tools -> Zoom* menu item, no action would take place and the entire menu would disappear. This is because the *Routine Tools -> Zoom* item has additional selections that must be made. In order to make a valid selection, the user must ‘walk’ all the way down to the bottom of the string of sub-menus, which means getting to a menu item with no arrow to the right of it. Figure 3-5 shows the menu after selecting the sub-menu of *Routine Tools -> Zoom*.

Figure 3-5. The Main Menu with two levels of sub-menus activated. In this case, if the left mouse button were to be clicked, the selection *Routine Tools -> Zoom -> Untapered* would be made.



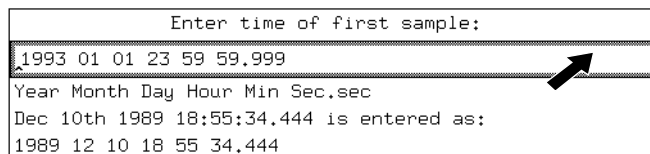
Remember: Menu names and options are *italicized* in this manual.

3.6 Dialog Boxes and Text Input

Many tools in PITSA require the user to enter some additional parameters or information. Text input is obtained through dialog boxes. Unlike message windows, these cannot be moved.

Figure 3-6. An example of a PITSA text input window.

Enter time of first sample:						
1993 01 01 23 59 59.999						
Year	Month	Day	Hour	Min	Sec.	sec
Dec 10th 1989 18:55:34.444 is entered as:						
1989	12	10	18	55	34.444	



The first line is the general question, followed on the second line by the text input area which is followed by an explanation of the text to be input.

Notice that the text input area is highlighted. This means that the text input window has keyboard focus (any key strokes will be directed to this window). In most cases text input windows will have keyboard focus when they appear. If they do not, you will have to move the cursor into the text input area.

On many occasions (as in the example shown), PITSA will display a dialog box with some default value(s) for the input it is expecting. You can accept the default value(s) by simply hitting RETURN.

If you do not wish to use the default value(s), the text in the input window can be edited with standard key strokes, such as the forward and backward arrows. The small carot at the baseline before the first character of the text input is the cursor. The following table lists actions that can be taken on the text input window. Two useful actions to remember are that a `Shift-DEL` will clear the input window and a double click with the mouse pointer in the text input window will act as a RETURN.

Table 3-2. Editing keys for text in dialog boxes.

->:	Move to the next character position on the right
<-	Move to the previous character position on the left
BACKSPACE, DEL	Delete the last character and move one position left.
Shift-DEL	Clear input window.
Shift-Click	Move to beginning of character at mouse position.
ESC	Restore entry field and return to top level of PITSA.
CR, Double-Click	Accept current entry and continue.

3.6.1 Date-Time Formats

Figure 3-6 shows an example of a PITSA dialog box which requests the user to input a date and time. It includes a template for such input, in which each item is separated by a space. PITSA will accept some variations on this format. Here are the rules which govern the formatting of date-time entries to PITSA, not only in dialog boxes, but also in reading ASCII-format data files (Section 5.4.2):

- Year is given with 4 digits.
- All six elements (year, month, day, hour, minute, seconds) must be given, in this order.
- The elements of the format may be separated by spaces, even multiple spaces, or single hyphens or colons. No spaces are allowed around hyphens or colons.

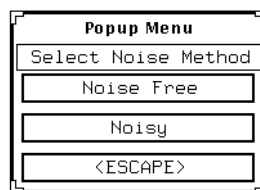
Some examples of legal date-time specifications:

```
1998 10 24 10 30 12.512345
1997  9  1  1 55 20.2
1998-5-24 10:30:1
```

3.7 Dialog Menus

PITSA also gets user input from stand alone pop-up menus. Figure 3-7 shows an example of a dialog menu:

Figure 3-7. An example of a PITSA dialog pop-up menu.



A menu selection is made by simply clicking (left mouse button) on the desired menu option. The *<ESCAPE>* option will terminate the current operation and bring PITSA back to the main menu state.

3.8 Channel Lists

PITSA organizes traces by numbering them, starting with 1. In many tools, PITSA will ask for a channel list to process. To specify the traces that are desired, the user enters the trace numbers separated by commas. There are also some abbreviations that can be used to select a range of traces.

The colon ':' provides an easy way to define a range of channels. If the user wants to read in channels *n1* up to *n2*, entering *n1:n2* would be equivalent to entering the complete channel list from *n1* to *n2*. (e.g., 1:3 would be equivalent to 1, 2, 3).

The semicolon ';' is understood as "up to the last channel". Hence, *n1;* would select all channels from *n1* to the end or the maximum number of channels allowed. If you wanted to specify all ten traces, you would only have to enter '*;*'.

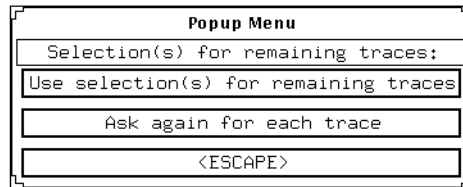
Repeated indices are allowed, and channel indices can be specified in any order.

3.9 Applying Tools to Multiple Channels

When a channel list of more than one trace is selected for a tool, PITSA will pop-up a message window to ask an additional question after all processing steps have been completed (Figure 3-8). In most tools, a number of selections are made pertaining to the way the tool processes a trace. For example, when an FFT is performed, the number of points for the FFT has to be selected. PITSA gives you the

option of applying all the selections that have been made for the first trace to the remaining traces. This is accomplished by selecting the *Use selection(s) for remaining traces*. Alternately, you can have PITSA ask for the input values individually for each trace by selecting *Ask again for each trace*. The *<ESCAPE>* option will abort the entire procedure.

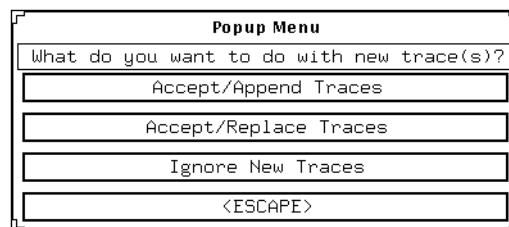
Figure 3-8. Additional menu that appears after processing the first trace of a multi-trace channel list.



3.10 Accepting New or Processed Traces

Most of the tools in PITSA either alter existing traces or create new traces. In this case, PITSA always requests instructions from the user as to what to do with these traces (Figure 3-9).

Figure 3-9. Menu that appears after PITSA has either processed existing traces or created new ones.

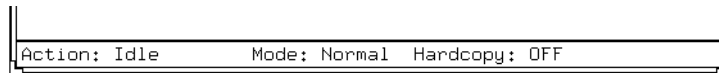


The *Accept/Append Traces* option will cause PITSA to add the new traces to the current list. In the case that traces have been altered by some processing step, such as integration, PITSA can also replace the source traces with the processed traces with the *Accept/Replace Traces* option. If new traces have been created, such as in creating test signals, the new traces can only be appended. The *Accept/Replace Traces* option will still appear, but its effect will be the same as *Accept/Append Traces*. The *Ignore New Traces* option will toss out the new traces while the *<ESCAPE>* option will abort the procedure. The *<ESCAPE>* option is different from the *Ignore New Traces* option in the event that more than one trace is being operated on. In the case of *Ignore New Traces*, PITSA will continue on to the next trace in the channel list (Section 3.8) while *<ESCAPE>* will put PITSA back into its home state.

3.11 Status Line

At the bottom of the main window there is a status line (Figure 3-10) which displays the state of some of the setup options.

Figure 3-10. The status line at the bottom of PITSA's main window.



Action indicates what PITSA is currently doing (i.e., plotting, thinking, etc.), Mode shows if PITSA is in *Program* or *Track Mode* and Hardcopy shows if the *Hardcopy Mode* is ON or OFF.

3.12 *Moving and Changing the Size of the Main Window*



The PITSA main window can be both moved and re-sized like standard OpenWindows applications. The only stipulation is that the PITSA main window should only be re-sized when it is in its home state—all loaded traces are displayed and the main menu is available. If PITSA is resized during the operation of one of its tools, the actual plotting area will not change until PITSA returns to its home state.

3.13 *Function keys*

Earlier versions of PITSA included on-line context-sensitive help, activated by the F1 function key. This feature is, regrettably, absent from the current version. An alternative strategy, based on an HTML version of this manual, is being developed.

Of the remaining function keys, only F2 and F3 are currently used in PITSA, and they are used for printing functions (Section 4.3.3), sending high- and low-resolution screendumps to the printer or to a file. Printing can only be done when the *Hardcopy Mode* is set to ON using the *Setup* menu (Chapter 4).

3.14 *Last Command*

PITSA contains a miniature history function which can be used to repeat the last command which has been selected from the main menu at the top level of the program. Simply select the *Last Command* option from the main menu. Notice, that this feature only repeats the main menu selection, however, none of the submenu selections which may pop up at a later time. Nevertheless, in many situations this feature provides a convenient shortcut.

3.15 *Quit*

Selecting *Quit* will terminate the PITSA session (Figure 3-11).

Figure 3-11. The *Quit* menu option will terminate the PITSA session.



This chapter describes how to use the *Setup* menu options (Table 4-1), which control how PITSA looks and acts. Aspects of PITSA's behavior which may be modified fall into four main categories:

- Choice of screen colors
- Running PITSA in a semi-automated fashion using *Program mode* and *Track mode*
- Saving screen shots as graphic files or sending them to a printer
- Whether or not to replot all traces after an action has been performed.

The current settings for the *Program* or *Track* modes and the hardcopy option are always shown in the status line at the bottom of the main screen.

4.1 Color Settings

On a color monitor, there are three different colors which you may set to change the visual appearance of the PITSA plotting screen (Table 4-1). We'll demonstrate this by showing how to change the setting of *Drawing Color 1*.

4.1.1 Example: Changing the Drawing Color 1.

- Select the *Setup -> Drawing Color 1* option.
- After this selection is made, the entire PITSA window will be painted with the current *Drawing Color 1* (the default is black). To change the color, clear out the entry in the text window and type the name of the color you want. The text window can be quickly cleared by typing a <Shift>. For example, if you want the *Drawing Color 1* to be yellow, simply type that into the text input window as in Figure 4-1.

Table 4-1. Menu selections available under *Setup*.

Menu Selection	Description
Background Color	Changes the background color.
Drawing Color 1	Changes the color PITSA uses for all primary graphics.
Drawing Color 2	Changes the color PITSA uses for all secondary graphics, including things like resulting traces, phase picks, etc.
Program Mode	If <i>Program Mode</i> is not currently on, this selection turns on the <i>Program Mode</i> . If <i>Program Mode</i> is already on, then the current command file is closed and <i>Program Mode</i> is turned off. When PITSA is in <i>Program Mode</i> , all commands and input are stored in a command file that can be later re-played with <i>Track Mode</i> .
Track Mode	<i>Track Mode</i> executes a command file that has been created while PITSA was in <i>Program Mode</i> .
Hardcopy Mode ON	Turns on the <i>Hardcopy Mode</i> —a screendump can be created by pressing the F2 key. When <i>Hardcopy Mode</i> is on, plotting is slowed down considerably and thus it is preferable to keep <i>Hardcopy Mode</i> off until it is needed.
Hardcopy Mode Off	Turns PITSA's <i>Hardcopy Mode</i> off.
Double Font Size	Will double the size of text on the computer screen. This is useful when creating hard copies in order to get larger text.
Normal Font Size	Turns off the <i>Double Font Size</i> option.
Turn Plot All On	By default, PITSA replots all channels currently loaded after every operation is complete. For a large number of traces, this can take a long time. This option can be turned off with <i>Turn Plot All Off</i> (next selection) and on with this selection.
Turn Plot All Off	Turns off the <i>Plot All</i> option as described above.

Figure 4-1. Text input window that appears for changing Drawing Color 1.

Enter the name for the new Drawing 1 color:
yellow
The current Drawing 1 color is being displayed

- After you hit Enter, PITSA will fill the window with the color you selected and prompt you with the following question “Take the new Drawing 1 color? (y/n)” (Figure 4-2).

Figure 4-2. Text input window to verify the new Drawing 1 Color.

Take the new Drawing 1 color? (y/n)
y

- If you like the color choice you made, you can hit RETURN and it will be accepted. If you don't like the color, you can enter 'n' and re-enter the color you want. As with all input windows in PITSA, you can hit the ESC key to cancel the operation.

4.2 Program and Track Mode

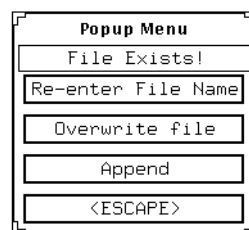
The *Program / Track Mode* combination is a powerful way to process large amounts of data routinely. Especially if the same sequence of operations is needed with different data files, but with only slightly different input parameters for each file.

If *Track Mode* is set to ON, PITSA is running under partial control of a so-called command file which contains the sequence of operations PITSA should perform. Normally the command file that is executed is created by recording steps that are taken in PITSA while the *Program Mode* is on. The two modes are analogous to record and play on a cassette tape deck. Commands are recorded while in the *Program Mode* and then later played back in the *Track Mode*. The *Program Mode* is a little more sophisticated than simply recording steps since conditions can be placed on some input so that when the command file is played back in *Track Mode*, some input windows will not appear.

To set *Program mode* to ON, select *Program Mode* from the *Setup* menu. You will be asked to name of the command file you are creating. After you enter a valid Unix file name (e.g., `test.cmd`) you can proceed to perform the sequence of operations you want to perform. Commands and input are recorded in the command file until the *Program Mode* is toggled to OFF, by selecting *Setup -> Program Mode* again from the main menu.

If the command file name you specified already exists, the following menu dialog will appear (Figure 4-3):

Figure 4-3. Menu dialog that appears if the command file specified for Program mode already exists.



Re-enter File Name will return you to the text input window that asks for the file name, *Overwrite File* will erase the current file and start 'recording' at the beginning of an empty file, and *Append* will append to the end of the existing file. *<ESCAPE>* will cancel the operation.

The directory that PITSA will use to read and write command files will initially be the *directory where PITSA was started*. This directory can be changed by selecting *Files/Traces -> Change Directory* (Section 5.3).

The command file is an ASCII text file which can be viewed with any text editor. For information on the format of this file, see Appendix D.

The command file can be executed at any time by toggling the *Track Mode* to ON by selecting *Setup -> Track Mode* from the main menu. PITSA will ask for the name of the command file to process (e.g., `test.cmd`). Once a valid file name is entered (one that exists and is in the correct format), PITSA will be run under control of the command file.

4.2.1 Re-prompting for Input with the “?” Key

There won't be many procedures that will be repeated with exactly the same input parameters for all the input required. Usually you will want to define a procedure where some of the input parameters will be fixed while others are specified interactively. This is quite easy to do in *Program Mode*. By default, PITSA (in *Track Mode*) replays all text input exactly as it was typed while *Program Mode* was ON. In fact, PITSA does not even pop up the input windows since it already knows what the input should be. To force PITSA to re-prompt for input at a certain step, the corresponding text that you input while in *Program Mode* should be preceded with a question mark ‘?’. PITSA will strip off the question mark before using the input (for the current process) and make a note in the command file that this input should be acquired from the user interactively when the command file is executed. While in *Track Mode*, all the input which was not preceded with a ‘?’ will have disappeared and the corresponding values will be taken from the command file.

4.2.2 Input from a Text File with the “~” Key

PITSA can also be directed to read parameters from a text file. While in *Program Mode*, instead of giving the answer to a question directly, you can enter the file name of a file where PITSA will find the answer; to tell PITSA that you are referring to a file, precede the file name with a tilde ‘~’. PITSA will read the first line of this file and assume it contains the answer to the question being asked. It will then create a new file with the same name with the first line stripped off—PITSA “consumes” one line of the text file each time it needs that particular input parameter.

This input file must exist before using the “~” key option in either *Program* or *Track Mode*, as the input file is stripped of the first line in both modes. Therefore it is not easy to see how one can create the file at all. If you know PITSA well enough, you might be able to construct the file directly. However, you can also build input files while running PITSA in *Program Mode* by editing the input file and a “master input file” in two other Unix windows while issuing commands to PITSA. When PITSA requires text from the input file (because you used the “~” key), you would copy a single line into the input file (before hitting the RETURN key), and also append the line to the end of the master input file. After the *Program Mode* is finished, the master input file can be copied to be used in the *Track Mode*.

4.2.3 Looping in Track Mode

It is sometimes useful to run PITSA in a semi-automated endless loop, and this can be done with the modifier keys introduced above. The basic technique is to enter *Track Mode* from within *Program Mode*. Then enter the name of the command file currently being created as the name of the command file to read from. In this case PITSA will turn the *Program Mode* OFF and start executing the command file from the beginning in *Track Mode*, repeating the operations that have just been performed. Wherever input was preceded with a ‘?’, the user will be asked for new input, while the other parameters are taken from the command file. Wherever input was preceded with a ‘~’, PITSA will read the first line of this parameter file and take it as the answer to the question being asked. As soon as the *Track Mode* gets to the bottom of the command file, it will “rewind” the command file and start again. This is not really an endless loop if input is being taken from a file since after each input item is taken from the file, it becomes one line shorter. Eventually, the input file will become empty and *Track Mode* will terminate with an error (Figure 4-4):

Figure 4-4. Error message displayed when the input file becomes empty while in a *Track Mode* loop.



4.2.3.1 Example: Looping with Track Mode.

Suppose you had 5 ISAM file systems with a single trace in each file on which you wanted to perform a baseline correction followed by an integration. For this example you should use the test signals utilities (Section 8.1) to create 5 ISAM files with names TLOOP1 through TLOOP5, and save them in a directory named demodata below the PITSA directory.

Issue the following commands to start PITSA.

```
cd ~/pitsa/demodata
cp input.sav input.fil
rm testloop.cmd
rm tloop?n.*
pitsa&
```

What we just did was to change into the directory where the demo data files live, and create a file named `input.fil` that contains the names of the five ISAM file systems we want to operate on (you could also use your favorite editor to do this). Then we deleted files that would have been created if this example had been run before, and started PITSA. The file `input.sav` is the master input file that we don't want destroyed, which is why we made a copy of it to 'feed' to PITSA. You should view the contents of `input.sav` so you can follow along with what is being input to PITSA as `input.fil` is being consumed. The input file has the ISAM file names listed twice since we will first read them in (requiring the file names), operate on them, then save them (again requiring a file name). We started PITSA in the background so that the Unix window in which PITSA was started would not be tied up. Once PITSA has started, make the following menu selections:

- Select *Setup -> Program Mode*
- Enter `testloop.cmd` as the name of the command file.
- Select *Files / Traces -> Retrieve Files -> ISAM*
- Enter `~input.fil` for the name of the ISAM file.
- Select *Routine Tools -> Baseline Correction -> Offset Removal*
- Select *Accept/Append Traces*
- Select *Routine Tools -> Integrate -> Tick's rule*
- Enter 2 for the channel to process.
- Select *Accept/Append Traces*
- Select *Files / Traces -> Save Files -> ISAM*
- Enter `~input.file` for the name of the ISAM file system.
- Enter `;` for the channel list to write. This will be the default.
- Select *Setup -> Track Mode*
- Enter `testloop.cmd` as the name of the track file.
- Select *CONTINUE*

When we start tracking the command file that we are building, PITSA will execute the commands in a loop until all input is taken from the input file and an error message will appear. Once the command file finishes, there will be 5 new ISAM file systems with three traces in each file system (the original trace, the trace with offset removed, and the integrated trace).

4.2.4 Using the Default Value of Input with the “\$” Key

There is one more text input modifier that can be used to control how the *Track Mode* deals with text input. If the first character of the line is a dollar sign ‘\$’ then the default value that is passed to the text input window from PITSA will be used. This can be useful when creating command files that will be used on data sets of unknown length. For example, suppose it is desired to take the FFT of any trace. If the number of FFT points is entered as a specific number, then the FFT will always use the same number of points. If, however, the ‘\$’ modifier is used, then the value that PITSA calculates for the number of points for the FFT (calculated as the minimum power of 2 that covers all points) will be used.

Table 4-2. Input modifiers for text input windows in *Program* and *Track Mode*.

Input Modifier	Description
? <text>	When <i>Track Mode</i> gets to this input window, it will stop and re-ask the question instead of using the text input.
\$ <text>	When <i>Track Mode</i> gets to this input window, it will use the default text passed into the text window instead of the text input.
~ <file name>	Input will be obtained from in input file named after the ~. This is true for both <i>Track</i> and <i>Program Mode</i> .

4.3 Saving Images from the Screen

Several options in the Setup menu relate to a very important task—saving screen images, either as printed hardcopy or as graphic files. Saving a graphics file to the hard disk is considered to be another form of “hardcopy”.

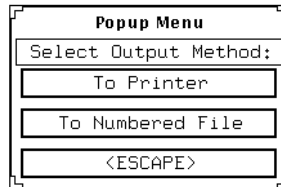
4.3.1 Hardcopy Mode

In order to make hardcopies, PITSA must first be in the hardcopy mode. This is achieved by selecting *Setup -> Hardcopy On*. Once the hardcopy mode is on, you can make a hardcopy at any time by hitting the F2 or F3 function keys (the difference is explained in Section 4.3.3). This will cause a copy of the current screen image to be written to a screendump file or to the printer directly. When the hardcopy mode is on, PITSA will be quite a bit slower displaying graphics. This is because PITSA’s hardcopies are not simply pixel dumps of the screen. PITSA stores all the information needed to generate high resolution vector plots of the screen, which means that PITSA has to record a lot of extra information in temporary files which slows down the graphics. In order to turn the hardcopy mode off, select *Setup -> Hardcopy Mode Off* from the main menu.

4.3.2 Hardcopy Destination

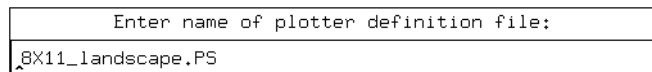
When *Setup -> Hardcopy Mode On* is selected, the user will be asked several questions. The first question will have to do with where the screendumps are sent, either directly to the printer or to a numbered file. The following menu will appear:

Figure 4-5. The menu dialog pop-up that appears when *Setup -> Hardcopy Mode On* is selected.



If you select the *To Printer* option, PITSA will then ask for the name of the plotter definition file. This is the file PITSA uses to build printer commands. PITSA expects this file to be in the directory defined by the system variable `PITSA_PRINTDEF_PATH_ENV`, which you defined in your `.cshrc` file. PITSA will insert the default printer definition file name that is defined by the environment variable `PITSA_PRINTDEF_NAME_ENV`, which should also be in the `.cshrc` file, but others could be defined as well.

Figure 4-6. Text input window that asks for the name of the plotter definition file.



The default printer definition file which comes with the standard distribution of PITSA is called `8X11_landscape.PS`. This printer definition file will generate screendumps compatible with standard PostScript printers using a page size of 8-1/2 x 11 inches in landscape mode.

After the plotter definition file has been specified, you can send a screendump to the printer at any time by pressing the F2 or F3 keys (Section 4.3.3).



Note: In order for the hardcopy feature to work correctly, the user must make sure that the printer configuration file contains the correct information (see Appendix C).

4.3.2.1 Writing Hardcopies to Screendump Files

If *To Numbered File* is selected in Figure 4-5, then an additional text input window will appear. This text input window will ask for a base filename to use for the plot output files, which will be stored in the current directory by default. The file names will be based on the specified base filename, starting with `plotout<base-name>.001` for the first file and `plotout<base-name>.002`, `plotout<base-name>.003`, etc. for subsequent files.

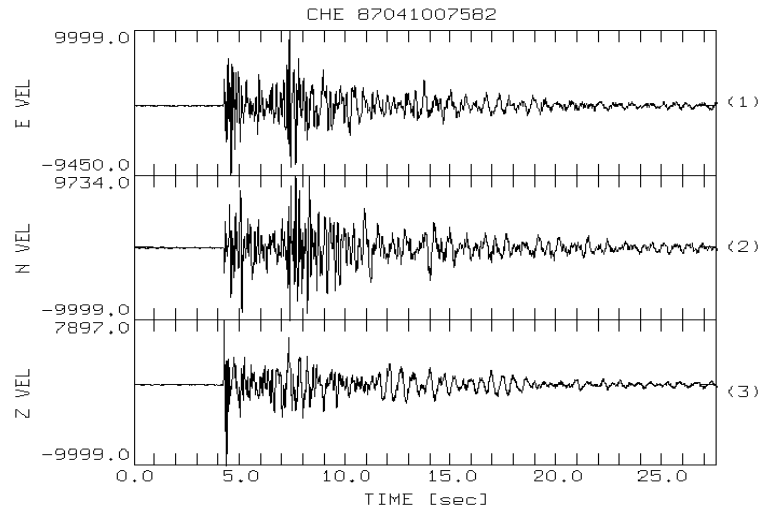
As with the printer option discussed above, the F2 and F3 keys are used to save a screen image to a file (Section 4.3.3).

You can print the screendump files using the Unix command `lpr`. To print the file `test1.001`, type:

```
lpr test1.001
```

If a screendump file is a PostScript file, then it can be viewed with the pageview program (provided with Sun's OS) (Figure 4-7).

Figure 4-7. A PostScript screendump file viewed with the program pageview.



4.3.3 Use of the F2 and F3 Function Keys

To send the current screen image to the printer or save it to a file, press the F2 or F3 function key. The F2 key makes a high-resolution image—all the vector drawing information is retained and used. The F3 key implements a low-resolution format, a bitmapped image from the screen. Obviously, file sizes will be smaller and print times will be shorter with the F3 option.

4.3.4 Doubling the Font Size

When screen images are created—especially when they are to be printed or saved to a file—it is sometimes desirable to increase the font size so that text is more legible. This can be easily accomplished by selecting *Setup -> Double Font Size*. When this item is selected, all characters will be doubled in size. An example of the effect of this option is shown in Figure 4-8 and Figure 4-9.

*Or perhaps your eyes
are going bad from
staring at computer
screens too long!*

Figure 4-8. A sample display with normal font size.

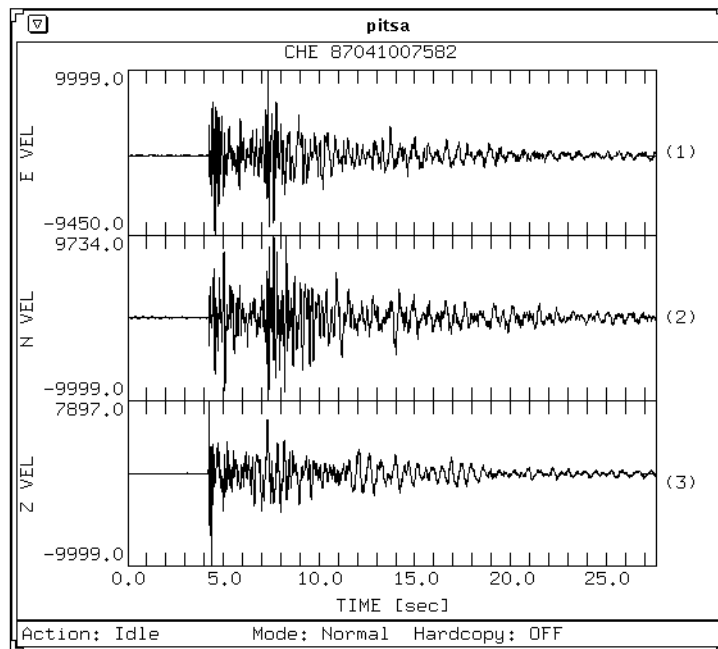
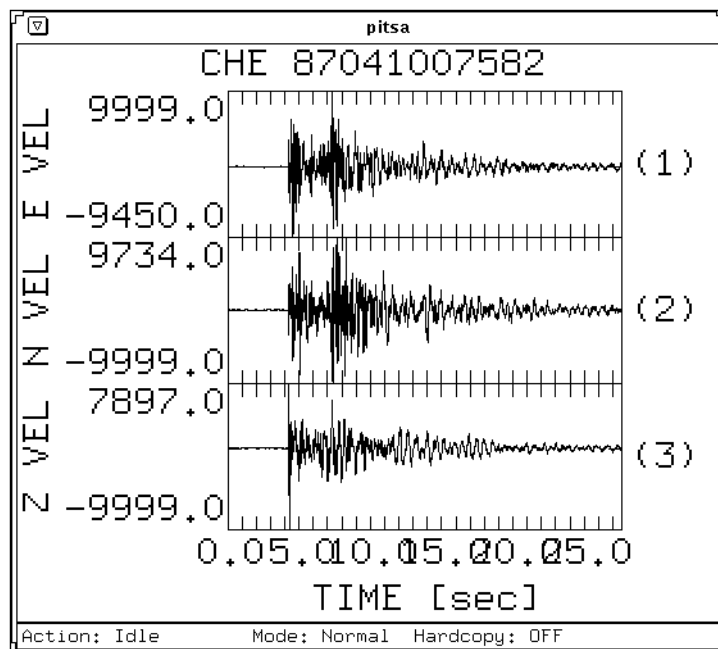


Figure 4-9. Same as Figure 4-8 with the font size set to double.



In Figure 4-9 the time labels on the x-axis overlap because of the large font size. The window could be made larger so that the numbers do not overlap. On the screen, the characters look rough. However, when a screendump is created, the characters will be plotted smoothly.

4.4 *Plot All Flag*

PITSA normally replots the screen after any operation which causes the screen to change, no matter how insignificant the change. Although we have tried to reduce the plotting overhead as much as possible, in some cases (e.g., if you are working with a large number of long traces) you may want to override PITSA's default behavior. Setting the *Plot All* flag to OFF prevents PITSA from replotting the main screen after each processing function. Turning the *Plot All* flag OFF does not affect any other plotting operations. You can turn off the *Plot All* flag by selecting *Setup -> Turn Plot All Off* from the main menu and turn it on by selecting *Setup -> Turn Plot All On*. When the *Plot All* flag is OFF, PITSA will display a message window showing how many traces are currently loaded.

This chapter covers basic trace management and file I/O options in PITSA. Here you will find information on how to load and save data from and to disk, and how to copy, rearrange, or delete data traces.

Specifically, this chapter covers the commands and options available under the *Files/Traces* menu:

Table 5-1. A list of the options available under *Files/Traces*.

Menu Selection	Description
Change Directory	Changes the directory PITSA uses to read and write data files.
Retrieve Files	Retrieve data files from disk, replacing all current traces.
Save Files	Saves a user-selectable list of currently loaded traces to disk.
Add Traces From File	Retrieves data files from disk, but unlike <i>Retrieve Files</i> , new traces are appended to the current list of traces.
Copy Traces	Creates copies of selected existing traces.
Rearrange Traces	Rearranges the display order of existing traces.
Delete Traces	Removes selected traces from memory (not disk).

5.1 Data Traces within PITSA

A data trace within PITSA consists of a collection of information, kept within a data structure (C language). The pointer to the actual trace is just one of the elements of this 'data header'. Other elements contain plotting information (e.g., the current zooming information, labels, etc.), station information (coordinates, name, etc.), event information (coordinates, magnitudes, etc.), and record information (time of first sample, sampling rate, phase picks, etc.). Some of this information is essential to PITSA (e.g., the sampling rate), others are only used occasionally (e.g., picked phases).

5.2 Supported File Formats

PITSA currently supports several standard file formats, but not as many as some earlier versions of the program. In particular, support for SEED and SUDS is no longer provided. The main reason for this reduction in scope is the difficulty in maintaining support for multiple formats which are themselves undergoing continual evolution. The issue of supporting various popular formats is never finally decided, however, and it is possible that some of these formats could make a return, or that new ones will be added. The authors welcome your comments regarding file formats which should be supported, and especially your source codes for implementing those formats. Currently PITSA supports the following types of data files:

Table 5-2. Types of data files supported by PITSA.

Format Name	Description
ISAM	Indexed Sequential Access Method. This is PITSA's native data format and all header information, such as scaling and zooming parameters, are stored in this type of data file.
ASCII	This format contains data points, one per line, in an editable ASCII text file. In addition to the data points, a minimal amount of header information is required.
GSE	The GSE waveform format is an ASCII file format and therefore platform independent. It contains a one or two line short standardized header (as was specified by the GSE group), the data sections and a checksum entry at the end for verification purposes.

5.2.1 The ISAM Database File System

On disk, data are kept in an ISAM (Indexed Sequential Access Method) database file system for fast and easy access. ISAM file systems consist of at least three files: two files containing the headers and the indexing information for all the traces and one trace file per channel. The trace file is simply a binary image of the floating point data which—in principle—can be accessed independently. All files in an ISAM file system have the same filename base, and different extensions to indicate their function. The extensions are `.nx0` and `.dt` for the database files, and `.001`, `.002`, etc. for the trace files.

5.2.2 ASCII Text File Format

In addition to ISAM file systems, PITSA can read and write data files in a plain ASCII text file. A number of header fields are defined for this format, but none of them are required—only a special line must appear at the beginning of each trace. The trace data are given, one point per line. ASCII files can contain multiple data traces.

5.2.3 GSE Format.

In the Conference on Disarmament an Ad Hoc Group of Scientific Experts (GSE) was established in 1976 working on the task of giving guidelines for international cooperative measures to detect and identify seismic events for the use of monitoring a Comprehensive Test Ban Treaty. Since then the GSE has developed several data exchange systems which have been widely tested in two technical tests

(GSETT1 and GSETT2). The complete GSE message concept allows to exchange quite more information than only seismic waveform data. However at this point we refer in our context only to the GSE waveform format and in Chapter 6.3 to the format used for specifying calibration (instrument response) information. GSE is widely used and at present implemented in most AutoDRM systems for quick and reliable data exchange worldwide. There exist several subformats inside GSE, the most common are supported in PITSA. For a further description see Appendix E.

5.3 Changing Directories

The directory that PITSA will use to load and save data files will initially be the *directory where PITSA was started*. This will also be the directory where PITSA will read and write command files for the *Program* and *Track Mode*. This directory can be changed by selecting *Files/Traces -> Change Directory*. When this selection is made, the following text input window will appear (Figure 5-1):

Figure 5-1. The Change Directory dialog window.

Enter the new path name for data files
~/
Current path is ./

To enter a new directory, type Shift-Delete to clear the default entry and then type the desired path. Paths can be either relative or absolute.

Note that PITSA does not check that the path you specified is valid! If you are subsequently unable to load the files you want, double-check the directory path which PITSA is using.

5.4 Retrieve Files

In order to read in data from disk, start PITSA from the directory containing the data (or use the *Change Directories* command) and select *Retrieve Files* from the *Files/Traces* menu. The user will be prompted for the file type. The *Retrieve Files* command will first clear PITSA of any currently-loaded traces. Traces may be added to the currently-loaded set using the *Add Traces from File* menu command (Section 5.6).

5.4.1 Reading Data From ISAM File Systems

To read data from an ISAM file system, select *ISAM* as the file type with the *Retrieve Files* menu command. You will be asked to enter the name of the ISAM file system. Enter only the basename without any extension.

If you have selected a valid ISAM file name, PITSA will check for the existence of the corresponding files on disk, make a consistency check and determine the total number of traces within the file system. Then you will be prompted for the channel list to load. See Section 3.8 for details of channel list specification.

5.4.1.1 Example: Reading ISAM files.

For this example, we will load in an ISAM file system named `isam1` that is stored in the `PITSA/demodata` directory. You should have received a sample dataset with the PITSA distribution, but it will not necessarily be the same one. Therefore, substitute the appropriate path, file names, channel numbers, etc., for your own dataset in the commands given below:

- Select *Files/Traces -> Retrieve Files -> ISAM* from the main menu.
- Enter `isam1` as the name of the file system.

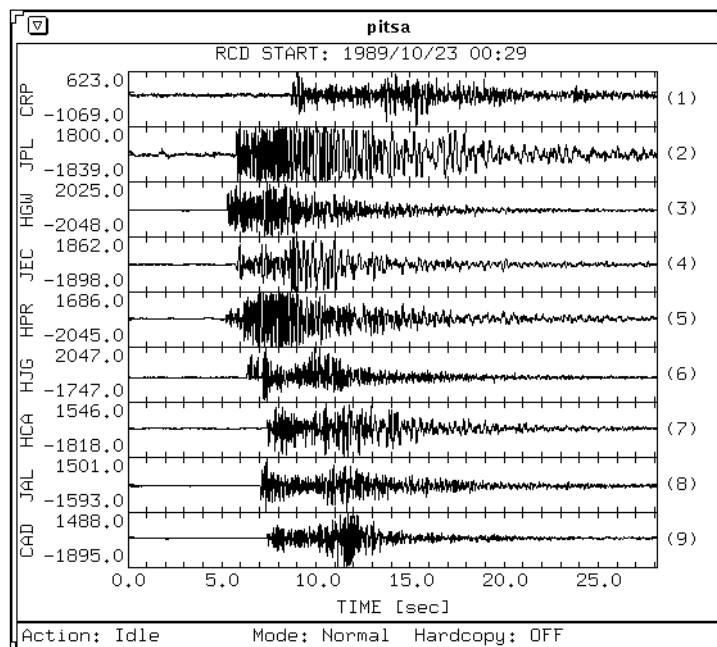
PITSA will now be asking for the channel list to load. In order to read in channels 1, 3, 4, 5, 2, 9, 10, 11, and 12 from the example file system, enter: `1,3:5,2,9;` followed by a CR (Figure 5-2).

Figure 5-2. Abbreviated channel list selection for trace input.

Enter channel list to load: (; loads all)
1,3:5,2,9;
file: isam1
no. of channels: 12

Next, PITSA will check the file system for internal consistency and extract the data headers corresponding to the selected channels. Finally it will retrieve the data traces and display them on the screen (Figure 5-3). Of course your display will be different if you used a different dataset. Notice that PITSA re-numbers the traces starting with 1.

Figure 5-3. Reading a selected number of channels from an ISAM file system.



5.4.2 Reading Plain ASCII Files

To load a plain ASCII datafile into PITSA, select option *ASCII* for the file type with the *Retrieve Files* menu command. In addition to the data values which are stored one per line, optional header information may be included at the top of the file. A line containing header information starts with a hash (#) symbol, followed immediately by the name of the header parameter to be given, a space, and the value of that header parameter. It is required that each trace begin with a line containing a hash (#) symbol in the first position. This may be followed by header information, or the remainder of the first line may be blank. Header information may be given in any order.

Although it is possible to read a file of trace data with no header information, this will seldom be a useful option for anything except demonstration purposes. PITSA will use default values for any parameters it requires for plotting or other functions. At the very least you will probably want to specify the sampling rate (PITSA uses 100 sps as default) and the start time of the trace.

Table 5-3. Header Fields Defined for the ASCII File Format

Header Field	Definition
EVT_X	Event longitude or x-coordinate (degrees or km)
EVT_Y	Event latitude or y-coordinate (degrees or km)
EVT_Z	Event focal depth (km)
EVT_COORD_TYPE	Event coordinate units: 0 = degrees/degrees/km 1 = km/km/km
EVT_TIME	Origin time (see Section 3.6.1 for formats)
EVT_COMMENT	Comment (39 characters max)
EVT_AZ	Azimuth (epicenter to station)
EVT_BACK_AZ	Back-azimuth (station to epicenter)
EVT_HYPO_DIST	Hypocentral distance (km)
EVT_EPI_DIST	Epicentral distance (degrees)
EVT_TYPE	E.g., “Local”, “Tele” (5 characters max)
EVT_LOCAL_MAG	Local magnitude (M_L)
EVT_BODY_WAVE_MAG	Body wave magnitude (m_b)
EVT_SURFACE_WAVE_MAG	Surface wave magnitude (M_S)
EVT_DURATION_MAG	Duration magnitude (M_D)
EVT_MOMENT_MAG	Moment magnitude (M_W)
START_TIME	Trace start time (see Section 3.6.1 for formats)
SAMP_FREQ	Sample frequency (samples/sec)
NDAT	Number of data points
STA_CODE	Station code (5 characters max)
STA_CHAN	Channel code, e.g., bhz, spn (5 characters max)
STA_TYPE	E.g., “WWSSN” (7 characters max)
STA_X	Station longitude or x-coordinate (degrees or km)
STA_Y	Station latitude or y-coordinate (degrees or km)
STA_Z	Station elevation, km
STA_COORD_TYPE	Station coordinate units: 0 = degrees/degrees/km 1 = km/km/km
STA_COMMENT	Comment (39 characters max)
STA_LOCATION_NUMBER	Station location ID code (integer)
INSTRUMENT_NUMBER	Instrument ID number (integer)

Reading ASCII files gives you the flexibility to quickly visualize a wide variety of data types. However, in order to use the full power of PITSA you may want to provide more information (e.g., label settings, station and/or event information, etc.). You can set most of the header parameters from within PITSA using the *Header Access* option from the *Utility* menu.

5.4.2.1 Single Trace ASCII File

The following lines show the beginning of an ASCII format data file containing a single trace (although it could just as well be the beginning of a multi-trace file), including a number of header items. A few points of the trace data are shown after the last header line.

```
#SAMP_FREQ 100.163937
#START_TIME 1989-10-24 10:42:23.391325
#STA_X -121.096001
#STA_Y 36.661301
#STA_Z 0.488000
#STA_COORD_TYPE 0
#STA_CODE BEMV
#STA_CHAN spV
#EVT_X -121.714729
#EVT_Y 37.227287
#EVT_Z 0.000000
#EVT_COORD_TYPE 0
#EVT_TIME 1989 10 24 10 42 23.391325
-57
-53
-55
-59
...
```

Note that different formats have been used to input date-time parameters for the header fields `START_TIME` and `EVT_TIME`. See Section 3.6.1 for a discussion of legal formats for these fields.

An example of the beginning of an ASCII data file with no header information is given next:

```
#
11715
7662
6470
7424
6708
7900
...
```

After PITSA processes the file, you will be shown the channel selection dialog box as usual—in this case it will contain only a single channel.

5.4.2.2 Multiple Trace ASCII File

Multiple-trace ASCII files can be constructed by concatenating single-trace ASCII files. No additional header fields are required.

A simple multiple-trace ASCII file is illustrated in the following lines:

```
#START_TIME 1997-01-31 02:44:35.000000
#NDAT 18000
#SAMP_FREQ 100.000000
#STA_CODE MIE_
11715
7662
6470
(17994 points not shown)
-2112
-1397
-1874
#START_TIME 1997-01-31 02:44:35.000000
#NDAT 18000
#SAMP_FREQ 100.000000
#STA_CODE NUN_
-3748
-2317
-1364
(17997 points not shown)
```

After PITSA processes the file, you will be shown the channel selection dialog box as usual.

5.4.3 Reading GSE Format

PITSA is able to read GSE files in single-trace or multi-trace files. However, not all possible flavors of GSE are supported. The currently supported format types are given in Table 5-4. For conversion between different flavors of GSE, please use the conversion routine 'codeco' from Urs Kradolfer (for more information on conversion utilities, please see <http://orfeus.knmi.nl/other.services/conversion.html>).

To load a GSE datafile into PITSA, select option *GSE* for the file type with the *Retrieve Files* menu command. If the GSE-file contains more than one trace the user is asked for the channel list to be loaded.

Table 5-4. Supported formats of GSE in PITSA

GSE format	GSE subformat	Differences
GSE1.0	CMP6 (6 bit compressed ASCII)	0-2
GSE1.0	INTV (variable integer format, ASCII)	0-2
GSE2.0	CM6 (6 bit compressed ASCII)	2
GSE2.0	INT (integer format, ASCII)	0

5.5 Save Files

The user can write trace data from PITSA to disk in either ISAM , GSE, or ASCII format.

5.5.1 Writing an ISAM File System

First the user selects *ISAM* as file format from the *Save file* menu. Next, they enter a valid file name for the new ISAM file system without extension. Before writing the data to disk, PITSA will check for the existence of a file system with that name. If it finds one, the user will be asked if they want to append to it (enter letter *a*), overwrite it (enter letter *o*), or cancel the whole output (enter letter *i* for ignore or hit the <Escape> key). If the user has selected the overwrite option it may take a couple of seconds to delete all the corresponding files, depending on the actual size of the file system.

Next, the user will be asked for the list of channels to write (Section 3.8). During the trace output, you will see note windows popping up, informing you about the trace file names currently being written.

5.5.2 Writing ASCII Files

To save the currently loaded traces in ASCII format, select *ASCII* as the output file format from the *Save Files* menu. In a submenu you can specify if you want to write all channels to a single file - single file option - or each selected channel to an extra file - multiple file option. In the multiple file selection each file will get an automatically created extension '*.axx*', where '*xx*' is running from '*01*' to '*99*'. You will be asked to enter a filename and the channel list of selected channels to write. If the specified file name already exists, PITSA will ask you for overwrite permission.

PITSA will write all available header information for each trace to the file, including any header fields defined during processing with the *Header Access* command.

5.5.3 Writing GSE Files

To save the currently loaded traces in GSE format, select *GSE* as the output file format from the *Save Files* menu. As for the ASCII output either a single file or a multiple file can be written according to your selection in the following sub-menu. You will be asked to enter a filename and the channel list of selected channels to write. When writing multiple files, each file will get an automatically created extension as '*.gxx*', where '*xx*' is a running from '*01*' to '*99*'.

The type of GSE chosen for writing is given by four entries in the *pitsa.cfg* file, named *GSE_1* or *GSE_2*, *GSE_FORM*, *GSE_NDIFF*, *GSE_RESOL*. Here you can select between the subformats of GSE as given in Table 5-4. A more detailed description of these entries is to be found in Appendix B. If no entries could be found in the *pitsa.cfg* file then *GSE1.0*, variable Integer format *INTV*, and 2nd differences with a resolution of 16 are used as default values.

PITSA will write all available header information for each trace to the file, as long as they are used in the GSE format.

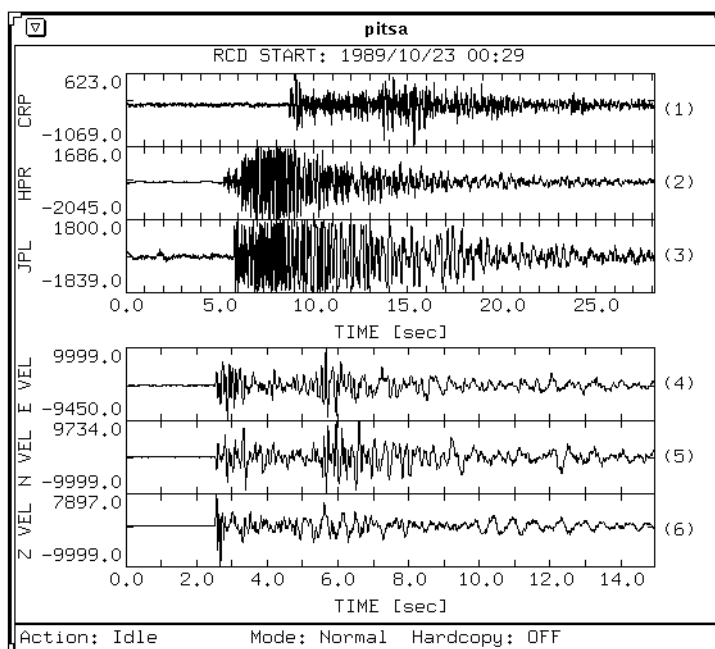
5.6 Add Traces From File

Selecting this option will allow the user to read additional channels into PITSA from an ISAM, ASCII or GSE file and append them to the currently loaded traces. If you load traces with the *Retrieve Files* option, PITSA first clears out all the exist-

ing channels before loading in the new channels. Using *Add Traces from File*, you can combine traces from different file formats, e.g., adding traces from an ASCII file to traces currently loaded from an ISAM file system.

When you select *Add Traces from File*, you will be asked for the name of the file from which to add traces, and then for the channel list of the channels to add. New traces are always appended at the bottom of the screen, but they can be re-arranged afterwards (Section 5.8). Figure 5-4 shows the result of loading three channels, and then using the *Add Traces from File* command to add three additional channels with different sample rates. PITSA adjusts its labeling depending on the sampling rate and the trace length.

Figure 5-4. Adding traces from an ISAM file system. The top three channels were loaded first.

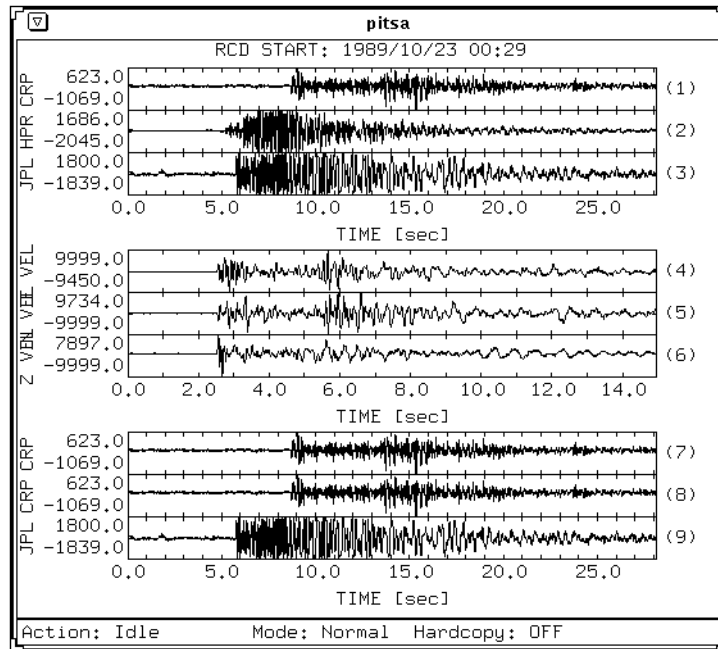


5.7 Copy Traces

It is often useful in the research or educational context to duplicate a trace for further (or alternative) processing. This is done with the *Copy Traces* menu Command. When you select this command, you will be asked for the source channel list (to copy from) and the destination channel list (to copy to). The number of channels in these lists must match. Undefined traces are filled with copies of the first trace. Existing traces will be over-written if they appear in the destination channel list.

Figure 5-5 shows the result of using *Copy Traces* to duplicate channels 1 and 3 of Figure 5-4 into channels 8 and 9. Channel 7 was not defined in the process and it was filled automatically by another copy of channel 1.

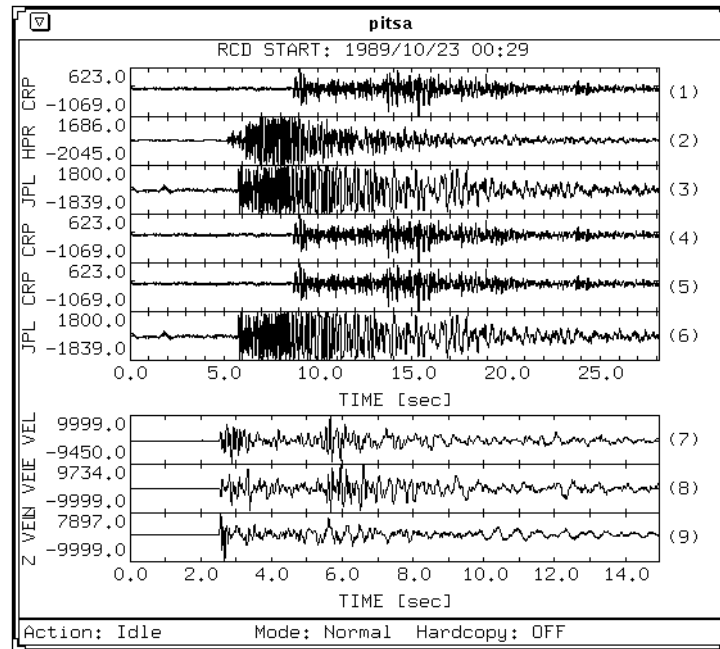
Figure 5-5. Illustration of Copy Traces.



5.8 Rearrange Traces

The *Rearrange Traces* command allows the user to rearrange the order of traces on the screen (e.g., to change the sequence of a three-component record from vertical, N-S, E-W to E-W, N-S, vertical). The user must only enter two sets of channel lists, corresponding to the current and the desired sequence. The number of channels in the two lists must, obviously, be equal. No traces are created or destroyed with this command. Figure 5-6 shows the result of interchanging the traces 7-9 with traces 4-6 in Figure 5-5.

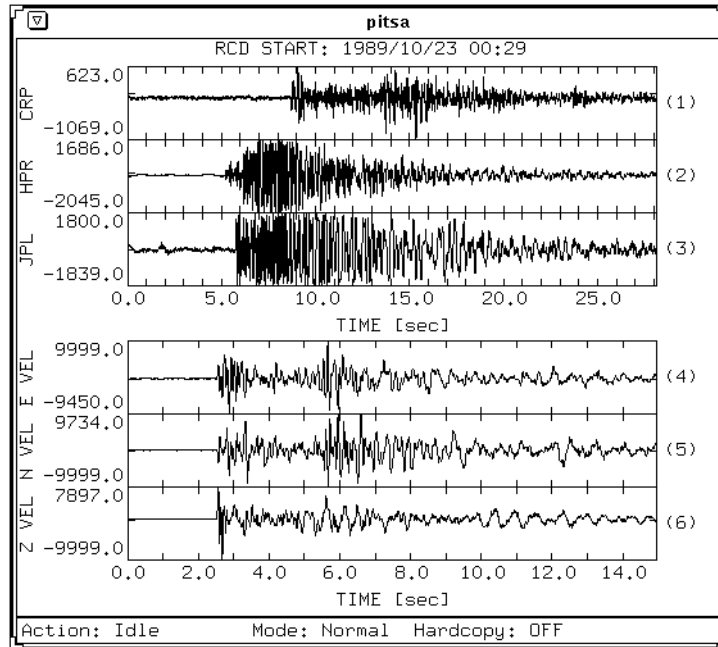
Figure 5-6. Using the Rearrange Traces command to exchange the positions of traces 7, 8, and 9 in Figure 5-5 with traces 4, 5, and 6, respectively.



5.9 Delete Traces

This option is used to remove selected traces from PITSA's memory. After selecting *Files/Traces -> Delete Traces*, you will be asked for the channel list of traces to delete. After deletion, traces will be lost from memory and can only be recovered by re-loading from the hard disk (if they have been saved) or some backup medium. Disk files are *not* affected by this option. You must use a Unix command, outside of PITSA, to delete a file from disk. Figure 5-7 shows the result of using *Delete Traces* to remove channels 4-6 from Figure 5-6.

Figure 5-7. Deleting traces 4, 5, and 6 from Figure 5-6.



This chapter describes the tools in PITSA which you are likely to use most frequently. They fall into two categories:

- Tools for manipulating trace data, such as “zooming in” for a closer view of a portion of a trace or cutting a small portion out for further analysis, performing simple point-by-point data editing functions, making baseline corrections, and integrating or differentiating traces.
- Basic “seismological” tools, such as phase picking, instrument correction, earthquake location, and magnitude estimation.

These tools are accessed through the *Routine Tools* command in the main menu. The options available under *Routine Tools* are shown in Table 6-1:

Table 6-1. The options available under *Routine Tools*.

Menu Selection	Description
Zoom	Select a portion of a trace using the cursor or by giving the indices of the points desired.
Edit	Simple data editing tasks such as single point editing, glitch editing, and automatic spike removal.
Instrument Simulation	Simulating some standard seismometer using the PREPROC software library written by Zmeskal and Plesinger.
Phase Picking	An extensive set of tools for picking and editing phase arrivals, an automatic P picker, and a location routine.
Magnitudes	Calculating several types of magnitudes.
Integrate	Digital integration on selected traces.
Differentiate	Digital differentiation on selected traces.
Baseline Correction	Correct baseline offset of traces by several methods

6.1 Zoom

In order to zoom in on a portion of the traces, the *Zoom* tool from the *Routine Tools* menu is selected.

6.1.1 Tapered vs. Untapered Windows

First you will be prompted for the type of zooming window. There are two choices:

- *Untapered* zooming
- *Tapered* zooming.

6.1.1.1 Tapering Functions

If you select *Tapered* zooming, the data within the selected window will be multiplied by a taper function which rises from a value of 0 at the margins to a value of 1 in the central portion of the window. The extent of the regions (at the left and right margins of the window) over which the taper function will be applied is determined later by the choice of the taper fraction parameter (Section 6.1.5).

The types of taper functions which are available are:

- Bartlett
- Cosine
- Hamming
- Hanning
- Boxcar
- Parzen
- Welch

For a discussion of the characteristics of different taper functions, see Press et al. (1988). PITSA includes so many taper functions primarily for educational purposes. The choice of taper function is mainly important if you intend to carry out additional operations in the frequency domain.

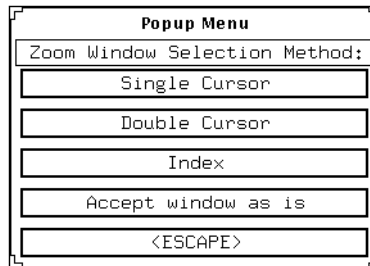
6.1.2 Channel List for Zooming

As with most tools in PITSA, you will be asked to specify the list of channels on which to operate. It is possible to zoom on more than one trace at a time, however, the traces must all have the same sampling rate, trace lengths and onset times (PITSA will confirm this).

6.1.3 Zoom Window Selection Methods

Next you will be asked to select a method for specifying the zoom window (Figure 6-1).

Figure 6-1. Options for selecting a zoom window.



Several combinations of cursor selection and index selection are available.

6.1.3.1 Single Cursor

To look at a window of predetermined length but select the onset of the window using a cross hair cursor, choose the option *Single Cursor*. You will be prompted for the window length (in points) and then be given a cross hair cursor to interactively select the first point of the zoom window. Click the left mouse button to select a point.

6.1.3.2 Double Cursor

To select both margins of the zoom window interactively, choose *Double cursor*. You will be presented with the cross hair cursor twice. Click the left mouse button to select each point.

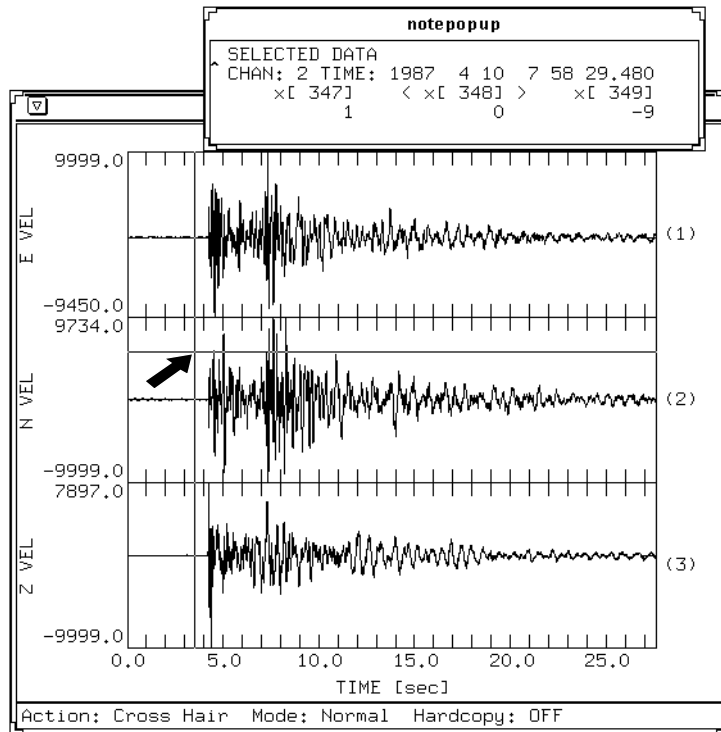
6.1.3.3 Index

To zoom in on a trace portion of n points in length starting at point m , select option *Index* and enter the appropriate values for the index of the starting point and the length of the desired zoom window.

6.1.3.4 Accept window as is

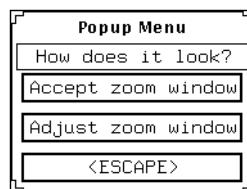
To take the complete trace length, select *Accept window as is*. The zooming procedure is often automatically activated as the first option for other tools, but it is not always desirable to use it. The *Accept window as is* option aborts the zoom tool.

Figure 6-2 shows an example of the use of the cursor to select the left margin of a zoom window. The note window shows the actual cursor position in absolute time, and the indices and data values of the current sample point and its nearest neighbors.

Figure 6-2. Selection of a window margin using the cross hair cursor.

6.1.4 Accepting the Zoomed Window

After selecting the window margins, you will be prompted for acceptance of the selection (Figure 6-3).

Figure 6-3. Accepting the zoomed window.

If *Adjust zoom window* is selected the current zoom window will be re-displayed and you will be able to adjust the margins. If you select the left (or right) margin during adjusting, PITSA assumes that you want the left (or right) margin extended back to the first point. Using this technique you can increase the size of the zoom window without having to exit the zoom tool. When you are satisfied with the selection of the window margin, choose *Accept zoom window*.

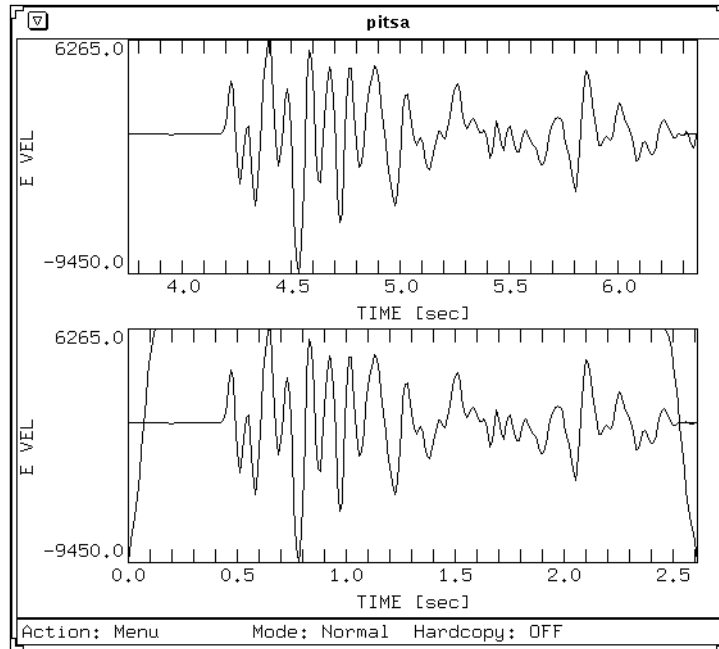
6.1.5 Selecting the Tapering Fraction

If you have chosen to use a tapering function in the zoom tool, PITSA will now (i.e., after the zoom window has been selected) prompt you for the tapering fraction which defines the proportion of the zoom window to which the tapering function will be applied.

The taper fraction parameter was a value between 0 and 1. A value of 0 would cause no tapering at all, and a value of 1 would cause the entire zoom window (100%) to be affected by the taper function (except a single point in the center of the zoom window). With a value of 0.5, 50% of the window would be affected (25% on each side).

After the taper fraction is selected, PITSA will display the zoomed up trace(s) followed by the trace(s) with the taper function applied. Overlaid on the tapered traces will be the actual taper function used (Figure 6-4).

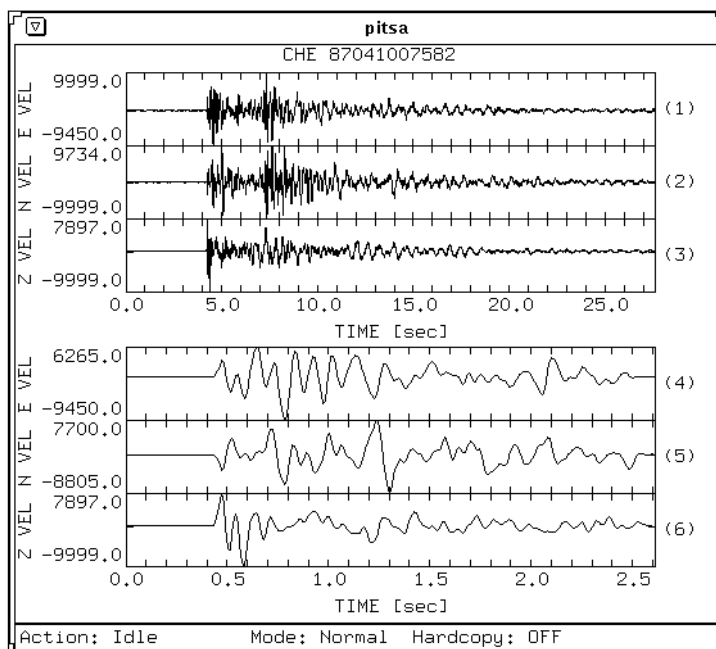
Figure 6-4. The zoom window with tapering.



There is a slight difference in entering the amount of tapering to be applied to the zoom window if the user has chosen the *Index* selection mode. In this case, tapering is defined by 4 indices: *n1*, *n2*, *n3*, and *n4*. *n1* and *n4* are simply the indices of the left and right zoom window margins, respectively. Indices *n2* and *n3* define where the tapering of the left and the right margin, respectively, stop. In this case, the user will be asked to enter values for indices *n2* and *n3* instead of entering a taper fraction. It is permissible to enter values which yield different tapering window lengths at the left and right margins.

Next you are presented with the standard PITSA menu window to select what to do with the new traces. Figure 6-5 shows the PITSA window after zoomed traces have been appended.

Figure 6-5. The PITSA window after zooming in on the P arrivals of the top three traces and selecting the *Accept/Append Traces* option.



If multiple traces were selected for tapered zooming, the menu for trace acceptance will be shown for each trace (Figure 6-3). This can be averted by selecting *Use selection(s) for remaining traces* when PITSA asks how to deal with remaining traces.

6.2 Edit

The tools under the *Routine Tools -> Edit* menu are used to correct minor problems in a data trace, involving at most a few specific data values. The options are:

- Point Editing
- Glitch Editing
- Despiking

The point and glitch editing tools are cases for which you are first given the chance to zoom in on a portion of the trace(s) in order to obtain a better visual resolution. This works in an identical manner to *untapered zooming* (Section 6.1.1).

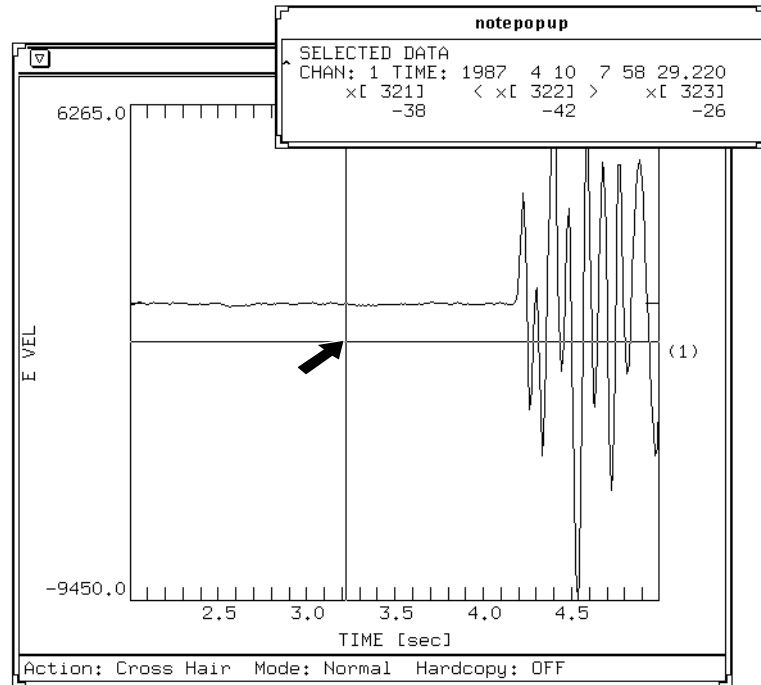
6.2.1 Point Editing

Point Editing accesses individual data point values. You can manually change a selected data value or interpolate the selected value by using the average value of the two neighboring points.

Point Editing will first show a cross hair cursor to select the point to edit (Figure 6-6). The note window on the top of the PITSA window shows the currently selected channel, absolute time of the actual sample, and the indices and data values of the

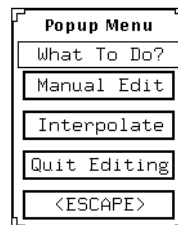
current sample and its neighbors. The current sample is selected for point editing by clicking the left mouse button.

Figure 6-6. Point editing. The user moves the cross hair cursor to select the point to be edited.



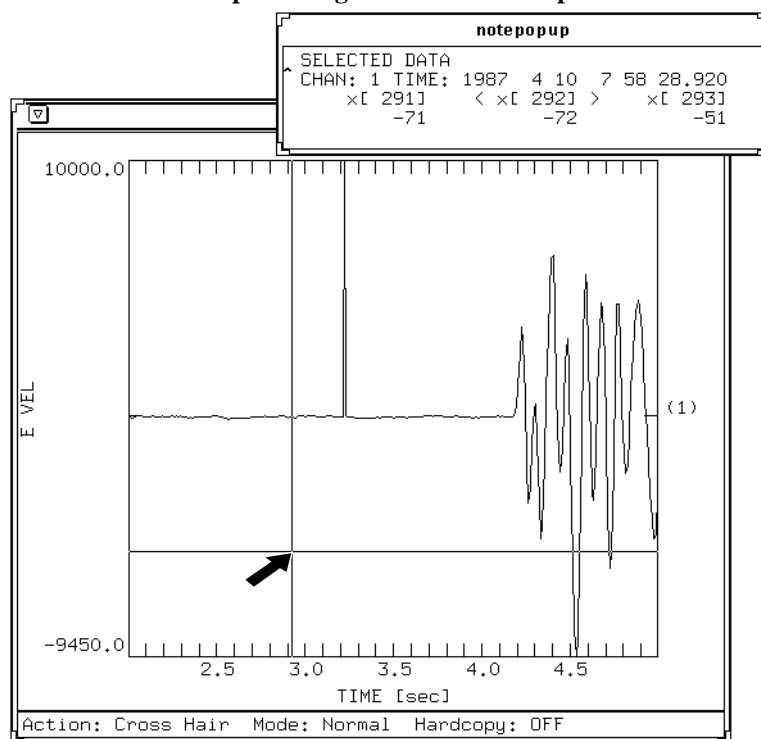
PITSA will now pop up a menu asking the user how to edit the point (Figure 6-7).

Figure 6-7. Pop-up menu for point editing methods.



Selecting *Manual Edit* will allow the user to enter a value for the selected point, while the *Interpolate* option will change the value of the selected point to the average of the previous and successive data points. In either case, the display returns to the zoomed trace with whatever changes have been made.

Figure 6-8 shows the result of selecting the point shown in Figure 6-6 and changing its value to 10000.0 with the *Manual Edit* option—creating a spike. To remove this spike, you could select this data point with the cross hair cursor and select the *Interpolate* option in Figure 6-7.

Figure 6-8. Creation of a spike using the *Manual Edit* option.

To exit the point editing routine, select any point and then select either *Quit Editing* or *<ESCAPE>*. The currently selected point will not be changed. Any other points that were changed by point editing, however, will retain their new values even if the *<ESCAPE>* option is selected.

6.2.2 Glitch Editing

Glitch Editing is used to remove corrupted sequences of data. It is a fairly crude tool, in the sense that no point-by-point editing is possible—all data points in the selected trace section will be given values which are interpolated between the two end points. It is generally used to remove a pulse of high-amplitude noise.

You will first be given a chance to zoom in on a small portion of the trace which is of interest. Selection of the trace section to edit is made using the cursor (and left mouse button) twice. After selecting the two end-points of the section to be edited, you will be presented with a limited menu of choices, shown in Figure 6-9.

Figure 6-9. The Glitch Editing menu.



The only choice for actually editing the selection is the *Interpolate* option, in which case the values inside the glitch (all points between, but not including, the margin points) will be linearly interpolated between the selected end points. The display returns to the zoomed trace.

To exit the tool, select an arbitrary set of glitch margins and then select *Quit Editing* or *<ESCAPE>*. As in *Point Editing*, all changes which have been made to the data are permanent, even if the *<ESCAPE>* option is used.

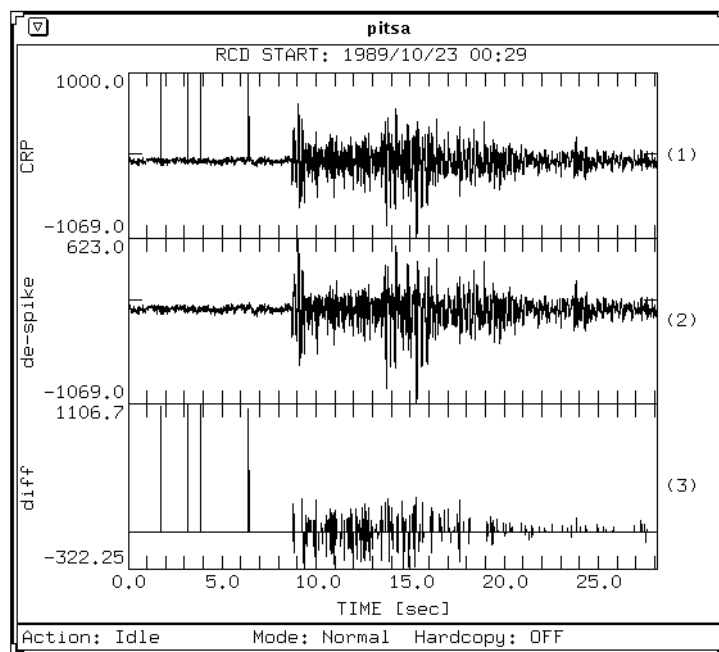
6.2.3 Despiking

The *Despiking* tool automatically remove spikes from data traces. It should be used with care, especially with traces for which the sampling rate is not much higher than the frequency of energetic signal components. In such cases this tool can distort the signal significantly (see Figure 6-10). It is mainly useful when picking P-arrivals automatically, to remove spurious signals which would otherwise be chosen as P arrivals.

After a channel list has been selected, PITSA will prompt for a smoothing level. Picking this value involves more art than science. The smoothing level parameter is greater than zero, and smaller values result in more smoothing (i.e., more aggressive removal of things that look like spikes). The default value is 0.5.

After a trace has been processed, PITSA will display the “before” and “after” traces and provide the usual menu of choices for what to do with the new trace. If the channel list contained more than one trace, PITSA will provide the option to use the two input parameters (the smoothing value and “what should be done”) for the remaining traces.

Figure 6-10. The Despiking tool was used to remove spikes from trace 1. The result is in trace 2. Trace 3 shows the difference between the original trace and the despiked trace. Notice that the despiked trace has been distorted in the signal region.



6.3 Instrument Simulation

Seismic signals as recorded at a seismic station differ considerably from the true ground motion. The recording system acts as a filter which changes the signal con-

tents of the seismic waveforms. Before we can interpret recorded signals in terms of properties of the source and/or the Earth we have to correct for the recording process. By applying signal processing techniques to the recorded data we can attempt first to obtain an approximation to the true ground motion at the surface (“restitution”). We simulate a standardized recording process (e.g., for magnitude determination) by applying an additional filter—representing the response characteristics of the desired instrument—to the true ground motion signal.

These important and difficult processes are performed in PITSA with the options under the *Routine Tools -> Instrument Simulation* menu.

6.3.1 Please Use These Tools with Great Care!

The implementation of instrument simulation in PITSA is far from thorough. We have borrowed source code to implement a subset of the functions in the software package PREPROC (written by Miroslav Zmeskal and Axel Plesinger). For many cases the results in PITSA will be “correct”, but there are also many situations in which the codes we are using may fail—meaning not so much that the program will crash, but that the results will not represent what we think they should. PREPROC contains extensive and elaborate testing and decision-making routines, with alternative algorithms to deal with most circumstances which can arise. The routines in PITSA will be most useful in an educational context and for exploration of various datasets. We recommend against using PITSA for research or analysis in which the reliability of the instrument simulation routines must be guaranteed, unless you have specifically verified the results against a known benchmark. For such purposes you should really use PREPROC or a specially-written and well-tested code which can deal with your particular requirements. Eventually, we hope to incorporate the full power of PREPROC for this kind of work in PITSA.

Probably the most important *caveat* we can offer to the use of the instrument simulation routines in PITSA is to ensure that the original recording system has a bandwidth at least as wide as that of the instrument to be simulated. PITSA will allow you to “simulate” a broadband recording from a short-period one, but the results will have more to do with round-off errors and algorithmic lapses than with seismology.

6.3.2 Instrument Response Files and Formats

The response characteristics of a seismic system are defined in a so-called “instrument response file” or “calibration file”, an ASCII text file which encodes the ground velocity frequency response of the recording system in one of three formats which are listed in Table 6-2.

Table 6-2. Formats for instrument response files.

FAP•	frequency - amplitude - phase triplets
PAZ•	poles and zeroes of the transfer function
FIR•	coefficients of the corresponding FIR filter

These formats conform to the recommendations of the GSE (Group of Scientific Experts) and are a subset of the GSE calibration section format. They are described explicitly in Appendix F. Instrument response files can be created during conversion to PITSA's ISAM format or by using a text editor. PITSA “knows” the calibration data for some standard seismic systems (Table 6-3).

When the *Instrument Simulation* menu item is selected, PITSA will ask first for a list of channels to process, and then for the name of a calibration file. PITSA will look for this file in the current PITSA directory (Section 5.3).

6.3.3 Choose Instrument Menu

PITSA will next display the *Choose Instrument* menu which asks the user which instrument should be simulated. The instrument choices are listed in Table 6-3:

Table 6-3. Instrument simulation options

Menu Option	Description
WWSSN (SP)	World-Wide Standard Seismograph Network, short period
WWSSN (LP)	World-Wide Standard Seismograph Network, long period
Kirnos	Medium period, wide-band system
Wood-Anderson	Used for determination of M_L
Ground displacement	Restitution of true ground motion
User-defined	Arbitrary

After you select a standard instrument or *Ground displacement* from the options listed above, PITSA will display the original trace and the simulated trace and display the normal menu of choices for “what to do” with the new trace. If *User-defined* is selected, you will first be asked for the name of the corresponding calibration file.

6.3.4 Multiple Traces

If multiple traces are selected in the channel list, PITSA assumes they all have the same calibration file. If this is not true, you must process the traces individually or in groups which do share a calibration file.

After the first trace of a multiple trace selection has been processed, PITSA will ask if the input entered for the first trace should be used for the remaining traces, or if new input values should be taken for each trace. PITSA’s behavior in this situation is predictable.

6.3.5 Example: Creating an all-pass instrument response file

It should be obvious that the simulation process requires accurate calibration files for both the actual recording system and the system to be simulated. However, it often happens that we do not know the exact properties of the seismic recording system used to record our data, but we still want to approximate how the recording would have looked on a different instrument. As long as the instrument we want to simulate is more narrow-band than the original recording system, we can approximate the system to be deconvolved by one of constant amplification and zero phase shift (either in displacement or in velocity). The issue of what constitutes sufficiently “more narrow band” in this context is beyond the scope of this Manual.

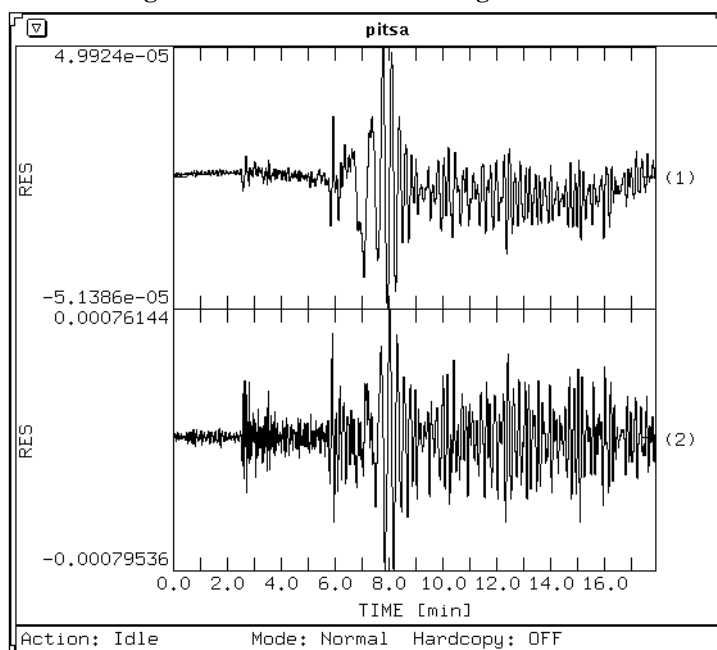
We will create an instrument response file in PAZ format for a hypothetical recording system with flat response.

For the PAZ format, the instrument response file contains the positions of the poles and zeros in the complex S plane in the format defined in Appendix F. Since the corresponding transfer function for a flat response should have no poles and zeros at all, lines 2 and 3 should contain a 0. The scaling factor should be set to 1.0E09 to make up for the fact that the GSE format expects the scaling factor to cause a normalization into nanometers, or nanometers/sec, respectively. PITSA, however, works consistently in SI units (m and m/sec, respectively). The instrument response file (which we might call `allpass.cal`) will be quite simple:

```
CAL1                                PAZ
0
0
1.0E09
```

Figure 6-11 shows the results of reading an unknown “broadband” signal (trace 1), deconvolving it with `allpass.cal` as the calibration file, and simulating a WWSSN short period signal (trace 2).

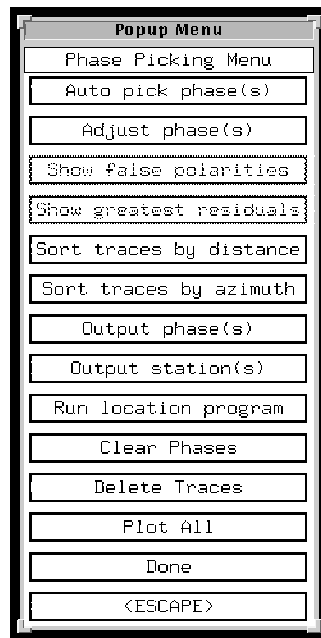
Figure 6-11. Use of a generic “allpass” calibration file to simulate a short-period recording from a broadband recording.



6.4 Phase Picking

The *Routine Tools -> Phase Picking* option places PITSA into its phase picking mode of operation. All the tools needed to perform routine phase picking and event location have been grouped together and are available when PITSA is in this mode. PITSA will pop up a menu of the options available (Figure 6-12) and after an operation is complete, PITSA will return to this menu. PITSA will not re-plot all the traces loaded in memory when it returns to the phase picking menu as it does when it returns to the main menu state. Instead, it will just leave up the last plot that was created before returning to this menu. It is analogous to having the Plot All Flag off while PITSA is in its standard mode of operation.

Figure 6-12. The Phase Picking Pop-up Menu.



Since phase picking is a very time-consuming task, PITSA has implemented a P-phase picker (Baer and Kradolfer, 1987) in order to determine the phase onset and the phase description automatically (option *Auto pick phase(s)*). Under this option you can edit the picked phase(s), change their phase description interactively, pick additional phases and/or delete phases (option *Adjust phase(s)*). The traces loaded into PITSA can be sorted by their distance from the event and plotted (options *Sort traces by distance* and *Plot All*).

Phase and station files can be written to files in various formats (HYPO71 or Long) for use by other programs. Once phase and station files have been created, you can run HYPO71 directly from PITSA and load in the phase residuals (options *Run location program* and *Read location results*).

PITSA will remain in the phase picking mode until either the *DONE* or *<ESCAPE>* option is selected, at which point PITSA will return to its standard mode of operation.

Note: Phase pick information is stored in the data headers for the current PITSA session, i.e., in memory. You must explicitly save the data files (as ISAM files) in order to retain the phase pick information in the PITSA headers permanently, i.e., on disk. All the information in memory is lost upon exiting PITSA.

Phases can also be picked from within the particle motion option.

6.4.1 Auto Pick Phase(s)

To automatically pick phases, select *Auto pick phase(s)* from the *Phase Picking* menu. After entering the list of channels to process, determination of onset times, first motion polarities, and evaluation of the reliability of the picks is done automatically using the algorithm of Baer and Kradolfer (1987)—see Appendix G.

There is no interactive control of the picking process, but the parameters can be modified in PITSA's configuration file (Appendix B).

After the automatic phase picking is completed, the phase picking menu will reappear. Normally, the next step would be to interactive edit and adjust the automatic picks, using the *Adjust phase(s)* option.

6.4.2 Adjust Phase(s)

Selecting the *Adjust phase(s)* option of the *Phase Picking* menu provides tools to adjust and edit the description of existing phases that were either selected by the user or created with the automatic phase picker.

6.4.2.1 Working with Many Traces

After asking for a channel list, PITSA will display a text input box, asking for the number of traces to look at in each step. For a small number of traces—say, up to five or six—it will be fine to just take the default value (all traces) for this parameters. For larger numbers of traces, however, the interface will be more effective if you work with subsets of traces.

If you enter a number which is smaller than the total number of traces loaded, PITSA will display a second text input box asking for the number of traces to *increment* at each step. It is straight-forward to specify some small number—say, three—for both parameters and have PITSA cycle through all the traces three at a time. By specifying the number to increment as one less than the number to view at each step, however, you can retain the last trace with adjusted picks as a guide to pick adjustments in the next group of traces.

For example, if you have selected 12 traces on which to adjust phase picks, and you then entered 5 for the number to look at in each step and 4 for the number of channels to increment in each pass, then PITSA would first display traces 1 through 5 and allow the user to pick, adjust and edit phases. Once the user is finished with these first 5 traces, PITSA would then present the user with traces 5 through 9, repeating the display of trace 5. On the next step, you would view traces 9 through 12.

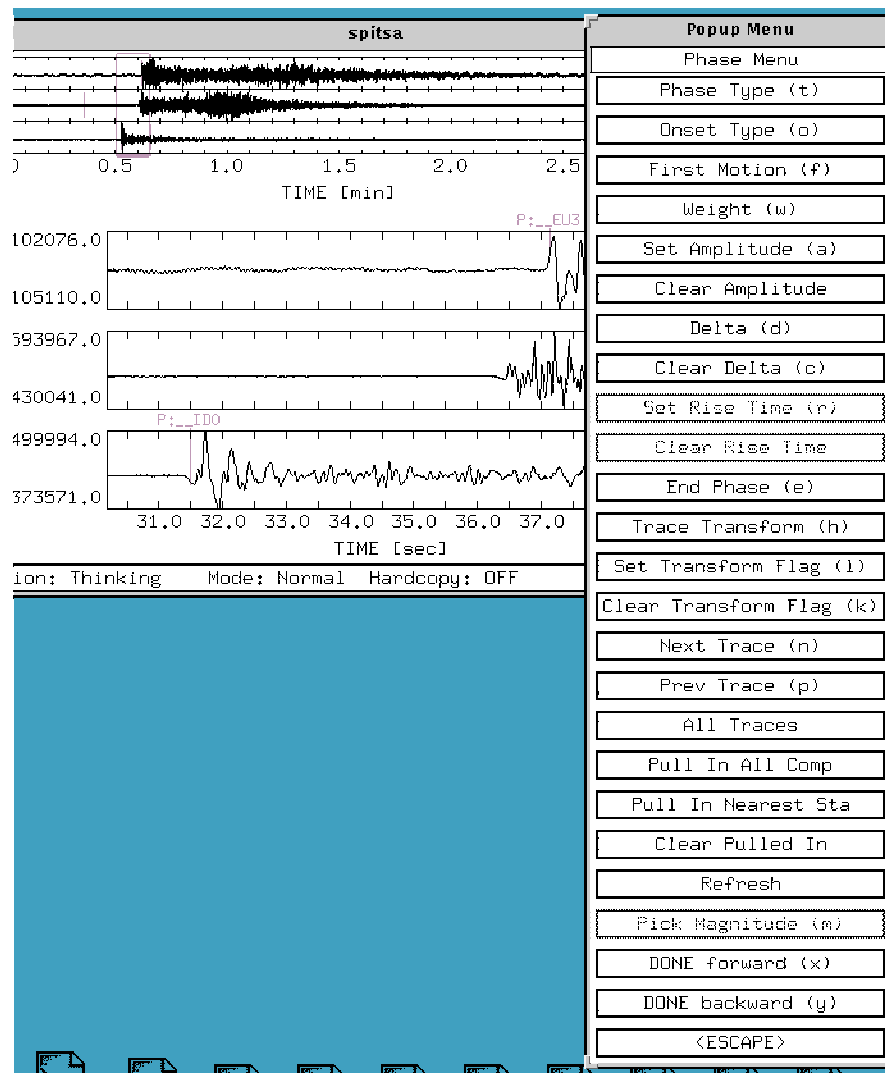
Use the *DONE forward* and *DONE backward* options of the *Adjust phase(s)* menu to move through the incremental display of traces.

6.4.2.2 Controlling the Zoom Box

Figure 6-13 shows how PITSA looks when it is in the phase pick/adjust mode. In this case, PITSA has presented the user with three traces at a time. The upper part of the screen is the zoom control area and shows the traces in their entirety with a zoom box overlaid.

The zoom box controls what part of the traces are visible in the bottom portion of the screen, which is the phase picking area. The portions of the traces that are zoomed in the bottom part of the screen are controlled by moving the zoom box to the area of interest in the upper part of the display. To move the zoom box, place the cursor inside the zoom box, then drag the box while depressing the left mouse button.

Figure 6-13. The Phase Adjust pop-up menu and screen display.



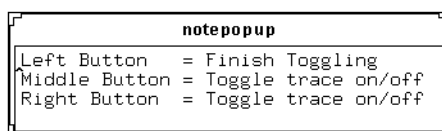
To change the size of the zoom box place the cursor just outside of a margin of the zoom box and drag the margin to a new location, again holding down the left mouse button.

6.4.2.3 Controlling Which Traces are Displayed for Phase Adjustment

In addition to controlling the position and size of the zoom box, the upper window also controls which traces are visible in the bottom (zoomed) part of the screen. For adjusting phase picks, the user may want maximum resolution and want to see only the current trace in the zoomed portion of the screen. The selection of which traces (of those displayed in the upper part of the screen) are displayed in the zoomed portion of the display is controlled by using all three mouse buttons.

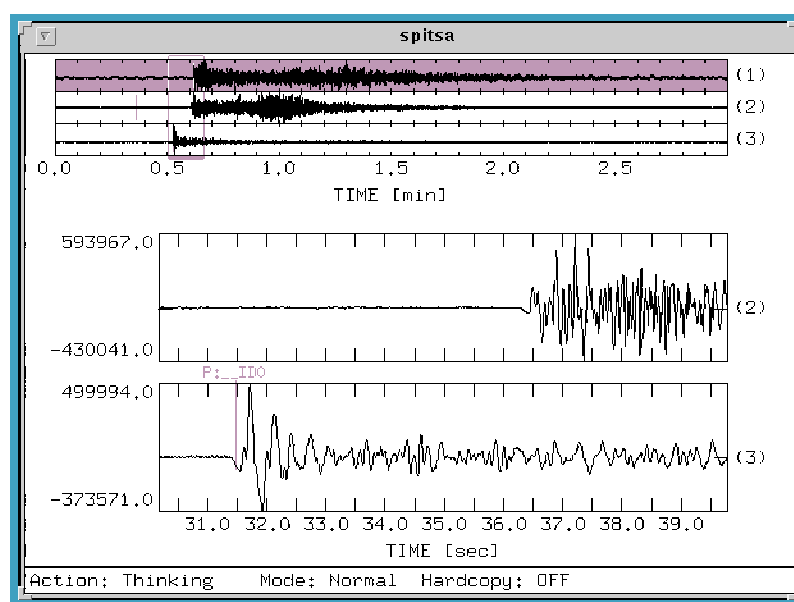
Clicking either the middle or right mouse button while the pointer is in one of the upper traces causes the trace will be inverted (meaning it will not be displayed in the zoomed portion of the display and will not be available for phase adjustment) and a message (Figure 6-14) to appear on the screen. You are now in trace selection mode.

Figure 6-14. Pop-up note for trace selection in phase picking.



For example, if the user were to click the right mouse pointer while in the top trace in the upper portion of the screen, then the PITSA window would look like Figure 6-15. The upper trace has been inverted to indicate that it will not be displayed in the bottom portion of the screen. Since replotting the lower traces is time consuming and the user may want to toggle on/off several traces, PITSA does not replot the bottom part of the screen until the user has clicked the left button to leave trace selection mode.

Figure 6-15. Trace selection example: trace 1 has been turned off.



While PITSA is in the trace selection mode, the user can not perform any other actions such as picking a phase or selecting one of the *Phase Menu* options; the buttons in the *Phase Menu* will become disabled. Trace selection can also be controlled with the *Next Trace*, *Prev Trace* and *All Traces* options available in the *Phase Menu*.

6.4.2.4 Picking Phases in Adjust Phase(s)

Manual phase picking (and editing of picks, either automatic or manual) is carried out using the cursor and all three mouse buttons on the zoomed traces in the lower part of the main display. The cursor must be inside one of the zoomed trace boxes in order for the mouse buttons to function in this context. Each mouse button has a different function:

Table 6-4. Mouse button usage for editing phase picks.

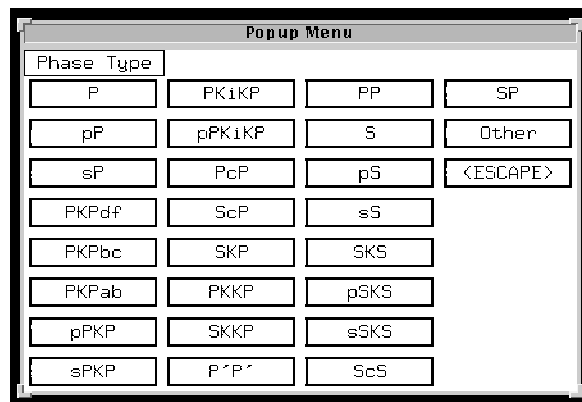
Right Mouse Button	Delete the phase closest to the current mouse point.
Middle Mouse Button	Insert a new phase at the current mouse pointer location.
Left Mouse Button	Adjust the phase closest to the mouse pointer.

PITSA contains some rules about phase picking which are important to understand before attempting to perform manual phase editing with the cursor and mouse buttons. If you're having trouble picking or adjusting phases, it is very likely that you are trying to violate one of these restrictions:

- You should select the phase type (using the *Phase Type* pop-up menu) before you try to pick a phase. Even if you have not yet picked any phases on the current traces yet, PITSA will be using an assumed phase type, for example, the last phase you picked on a previous trace.
- PITSA keeps the same phase type in memory until you change it.
- You cannot pick a phase earlier than an existing P phase on the same trace (because it is the first-arriving phase). You must either adjust the position of the existing pick, or delete it and pick a new P phase.
- You cannot pick the same phase twice on a single trace. You must delete the existing pick and re-pick it.
- Because of the previous restriction, you cannot pick phases for more than one event on a given set of traces.

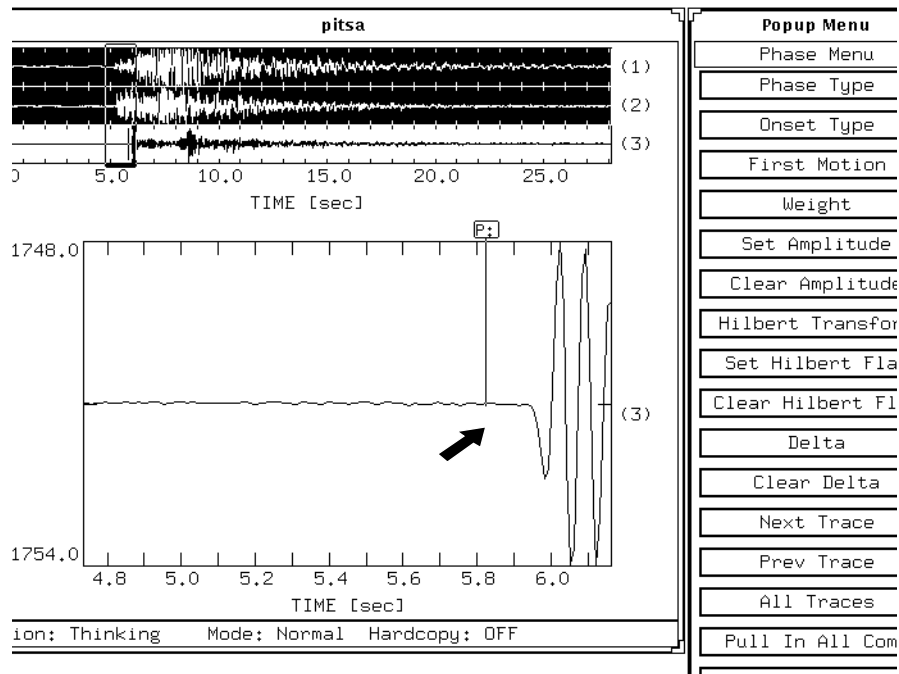
Insert a new phase pick

To insert a phase, you must first select the phase type. Bring up the *Phase Type* pop-up menu by clicking on the appropriate option of the *Phase Menu*, and select one of the phase types.

Figure 6-16. The Phase Type menu

To actually set the new pick, move the cursor to the desired position and click the middle mouse button (Figure 6-17). When a phase is picked, a box is drawn above the trace that shows the phase id and a line is drawn down to the trace. Also, a vertical line is drawn at the phase pick location in the zoom control trace in the upper part of the display.

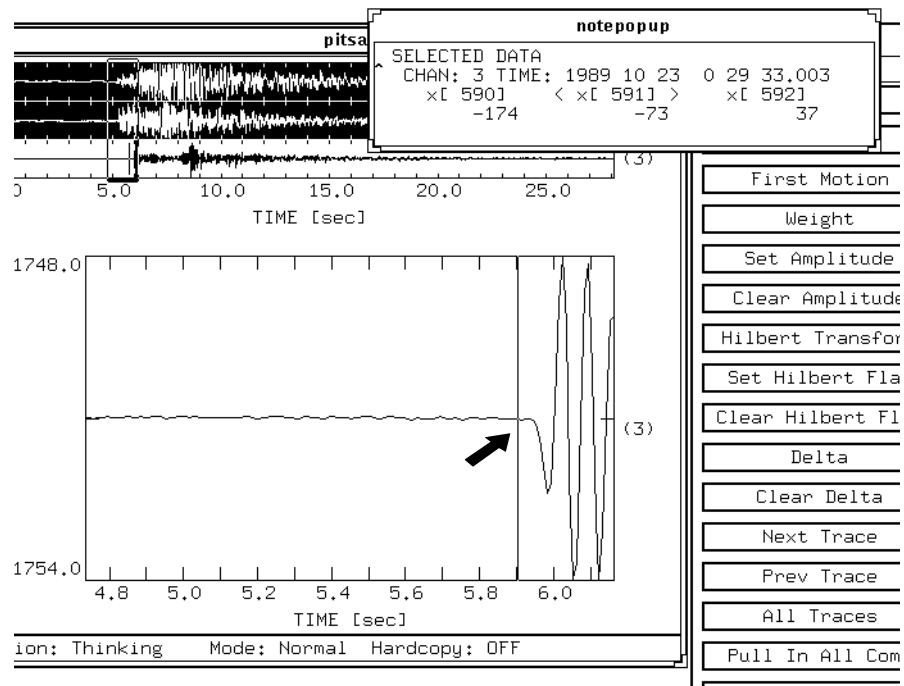
Figure 6-17. Inserting a phase pick manually.



Moving a phase pick

Once a pick has been made as in Figure 6-17, its position can be adjusted by using the left mouse button. When the left mouse button is depressed PITSA will look for the closest pick (meaning the line dropped from the top of the zoom box to the trace) within about 100 screen pixels from the mouse pointer position. If PITSA finds a phase pick close enough to the cursor, it will replace the phase pick with a vertical line running from the top of the trace to the bottom as in Figure 6-18. Also, a note will pop up showing the absolute time and data value at the current location of the pick, and for the two immediate neighbors.

Figure 6-18. Adjusting the location of a phase pick.



As long as the left mouse button is held down, the vertical line will follow the cursor, allowing the user to adjust the position of the phase pick. If the trace on which the pick is being made is highly magnified, you may observe the vertical line jumping from one trace point to the next; this is because PITSA only allows picks to be made on actual data points. When the left mouse button is released, the vertical line is replaced by a standard phase label.

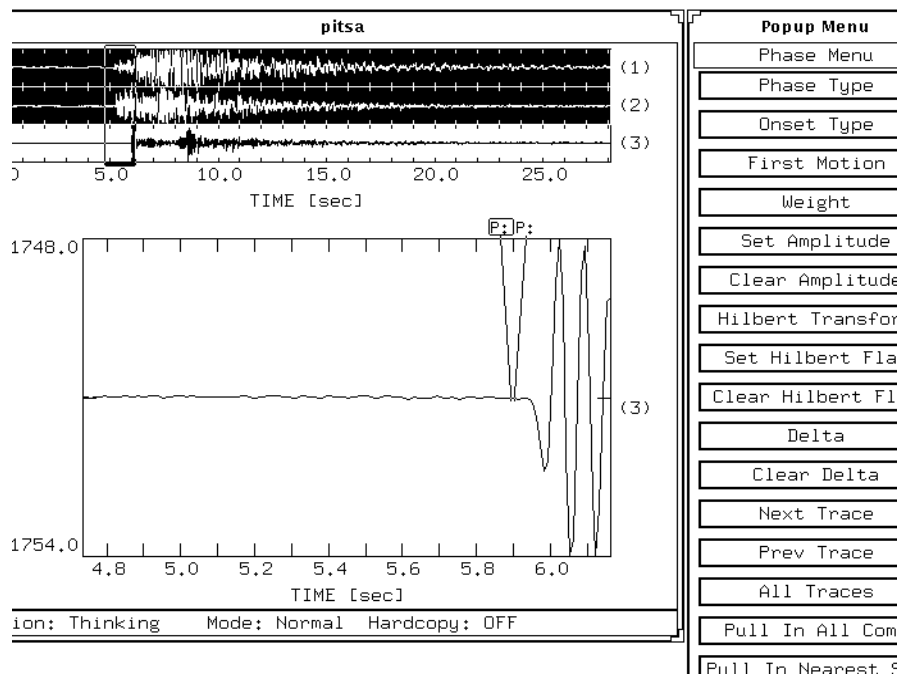
Delete a phase pick

To delete a phase pick, position the cursor near the phase location and click the right mouse button. PITSA will find a phase pick within about 100 screen pixels of the tip of the cursor.

6.4.2.5 The display of overlapping phase labels

PITSA tries to place the phase label directly above the location of the pick on the trace. If there are several phase picks close together, then this is not possible without overlapping the phase labels. Instead, PITSA will move the phase labels to the side so that all phase labels can be plotted without overlapping. Then a line is drawn from the bottom middle of the phase label to the phase location on the trace (Figure 6-19).

Figure 6-19. Display of overlapping phase labels.



6.4.2.6 The active phase

Notice that in Figure 6-19 one phase label has a box around it while the other does not. This is because the box indicates that a phase is “active” and that phase-specific *Phase Menu* selections will apply to that phase. Only one phase at a time can be active and to change the active phase, the user selects the desired phase by clicking near it with the left mouse button. This in effect adjusts the phase, but if the mouse is not moved in between the button down and up events, then the phase location will be unaffected.

6.4.2.7 Phase Labels

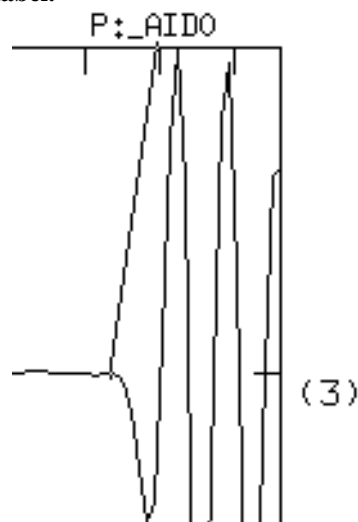
A phase label in PITSA has several parts. First there is a phase id followed by a colon which is followed by five single-character phase descriptor fields (<phase id>:12345). The phase id can be any string up to 19 characters long. Each phase descriptor encodes an attribute of the phase pick. An underscore in a phase descriptor field indicates that the attribute has not been set. Trailing underscores are not displayed, in order to reduce confusion in the display. Table 6-5 lists the options available for the different phase descriptor fields:

Table 6-5. The meaning of the phase descriptor characters.

Character Position	Meaning
1	If an H is present in this position then the phase was either picked or adjusted while the corresponding trace was Hilbert transformed.
2	If an A is present in this location then amplitude information has been input for this pick.
3	Onset of the phase. Possibilities are I for impulsive and E for emergent.
4	HYPO71 codes for first motion of the phase. Possibilities are: U -> Up D -> Down + -> Poor up - -> Poor down N -> Noisy ? -> Unreadable
5	HYPO71 codes for weight of the phase: 0 -> Full weight 1 -> 3/4 weight 2 -> 1/2 weight 3 -> 1/4 weight 4 -> 0 weight

Figure 6-20 shows a phase label with a phase id of P and four set attributes. PITSA has the ability to read in synthetic or residual phases and these are displayed as only the phase id string without a colon. These phases cannot be adjusted or edited as they are meant to be references.

Figure 6-20. A phase label.



6.4.2.8 The Phase Menu

Figure 6-13 shows the PITSA *Phase Menu* on the right side of the PITSA main window. When *Routine Tools -> Adjust phase(s)* is entered, this menu will pop up on the screen and remain until the user selects one of the *DONE* options or *<ESCAPE>*. Some of the options in this menu apply only to the active phase (Section 6.4.2.6); PITSA will ignore these commands unless a phase has been selected first.

The following sections describe in detail how the different options work.

Phase Type

This option is used to specify the phase type before a new phase is picked. In earlier versions of PITSA it could be used to change the phase ID of the active phase, but this is no longer implemented. To change a phase ID, you must delete a phase and re-pick it.

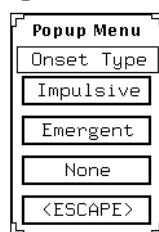
PITSA will pop up the menu shown in Figure 6-16 on page 6-17. This menu has a number of pre-defined phase IDs or you can select the *Other* option and specify an arbitrary phase ID of up to 19 characters.

Selecting *<ESCAPE>* will pop down the menu. The selected phase type will become the default phase type for subsequent phase picks until it is changed again.

Onset Type

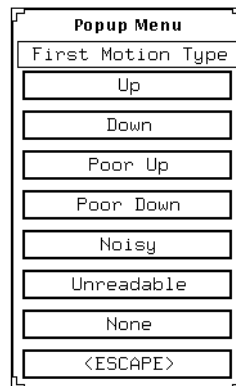
This option is used to change the onset type for the selected phase which is stored in the third phase descriptor field of the phase label (Table 6-5 on page 6-21). It operates on the active phase. The pop-up menu is shown in Figure 6-21. Selecting *Impulsive* or *Emergent* resets the third character of the phase descriptor field. Selecting *None* clears the onset character field in the phase label and replaces it with an underscore. *<ESCAPE>* will end the operation without changing the onset character.

Figure 6-21. The *Onset Type* pop-up menu.



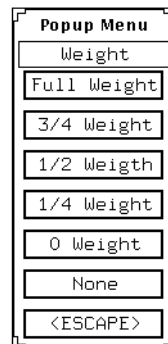
First Motion

This option is used to set the polarity of the first motion of the active phase, which is stored in the fourth phase descriptor field of the phase label. The pop-up menu is shown in Figure 6-22. See Table 6-5 on page 6-21 for description of the label codes. Selecting *None* clears the first motion field of the phase label and sets this character to an underscore. *<ESCAPE>* exits the pop-up menu, leaving the descriptor field unchanged.

Figure 6-22. The *First Motion* pop-up menu.

Weight

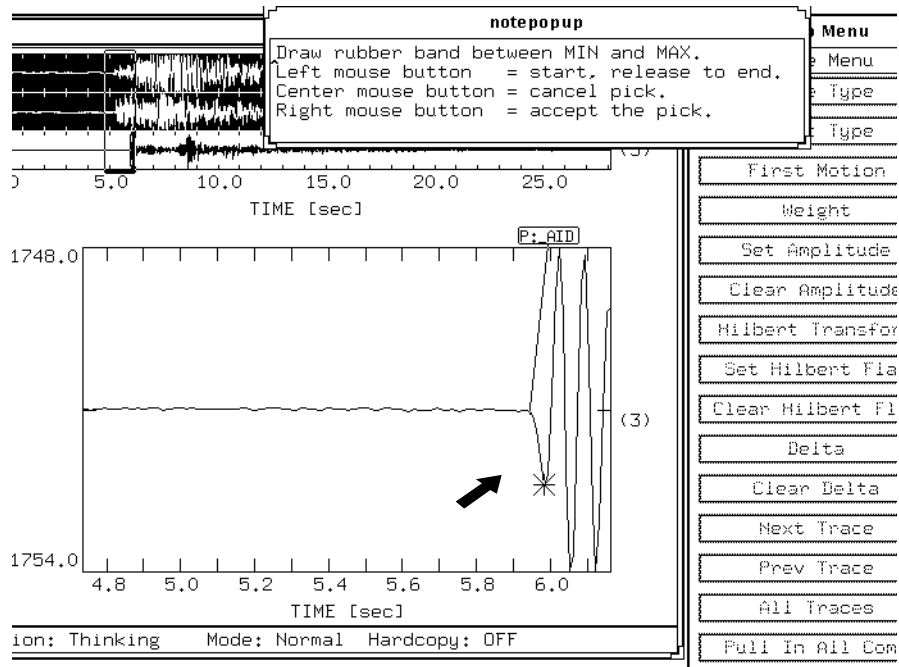
This option is used to set, clear or change the weight of the active phase (the fifth position of the phase descriptor field), which is used in earthquake location programs. The pop-up menu is shown in Figure 6-23. See Table 6-5 on page 6-21 for the encoding of the weight character in the phase descriptor. Selecting *None* clears the entry in the phase label and sets this character to an underscore. *<ESCAPE>* exits, leaving the descriptor field unchanged.

Figure 6-23. The *Weight* pop-up menu.

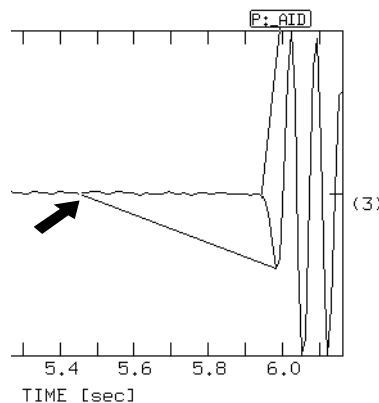
Set Amplitude

The *Set Amplitude* option is used to measure the amplitude and period of the active phase, which are stored in the data header of the trace. You will select two extrema of opposite polarity (e.g., from a maximum to the following minimum); PITSA will measure the peak-to-peak amplitude and period (twice the time length between the extremal points) of the phase from these points.

When this option has been selected, PITSA displays a star that tracks the cursor and snaps to local minima or maxima along the trace (see Figure 6-24). A note window pops up to provide help.

Figure 6-24. The *Set Amplitude* display and note window.

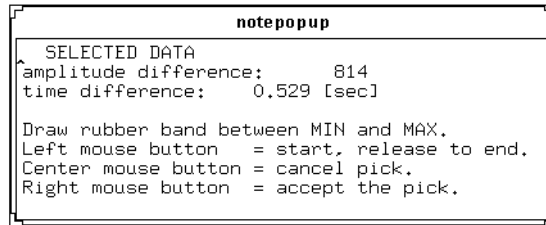
When the star is at the extremal point you wish to use as the beginning of the measurement window, hold the left mouse button down and drag the cursor to the following extremal point. As you drag the cursor, PITSA draws a line from the first extremal point of the pick to the nearest local extremal point (Figure 6-25).

Figure 6-25. The *Set Amplitude* display, showing the tracking of the cursor as it is dragged away from the first extremal point.

When the left mouse button is released, PITSA will pop up a message similar to the one in Figure 6-26 which shows the amplitude and period the user selected. The star will have returned and follows the cursor as before. At this point, the user has the option of re-selecting the amplitude (in the manner just described), canceling the operation (clicking the center mouse button) or accepting the values presented (clicking the right mouse button).

If the measured values are accepted, the second character in the phase descriptor field (Table 6-5 on page 6-21) will be set to A. Currently amplitude information is only used when phase output files are created See "Output Phase(s)" on page 29..

Figure 6-26. Display of measured amplitude and period information.



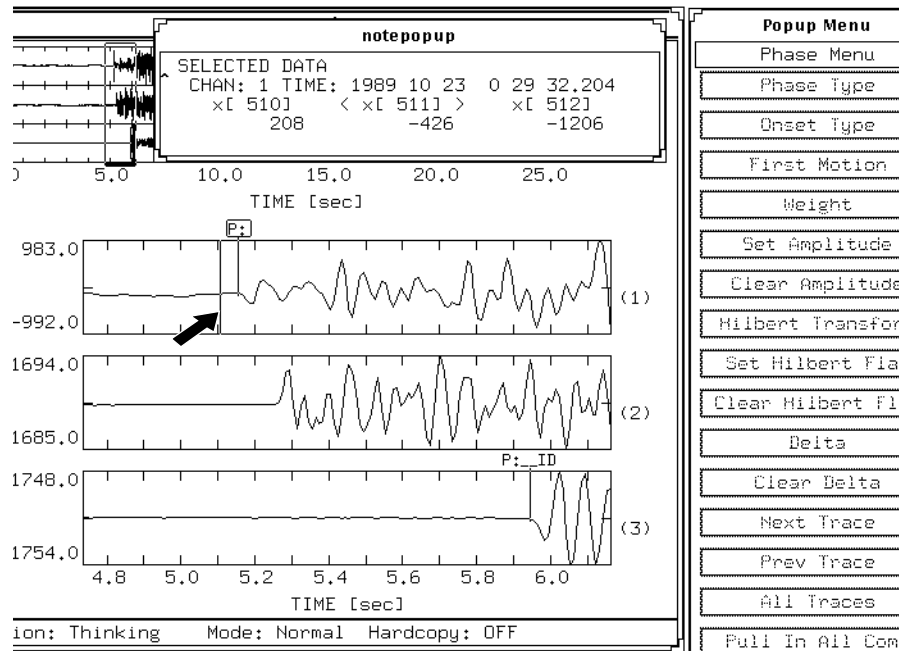
Clear Amplitude

This option is used to quickly clear the amplitude and period measurements associated with the active phase. The A in the second position of the phase descriptor field of the active phase will be reset to an underscore.

Delta

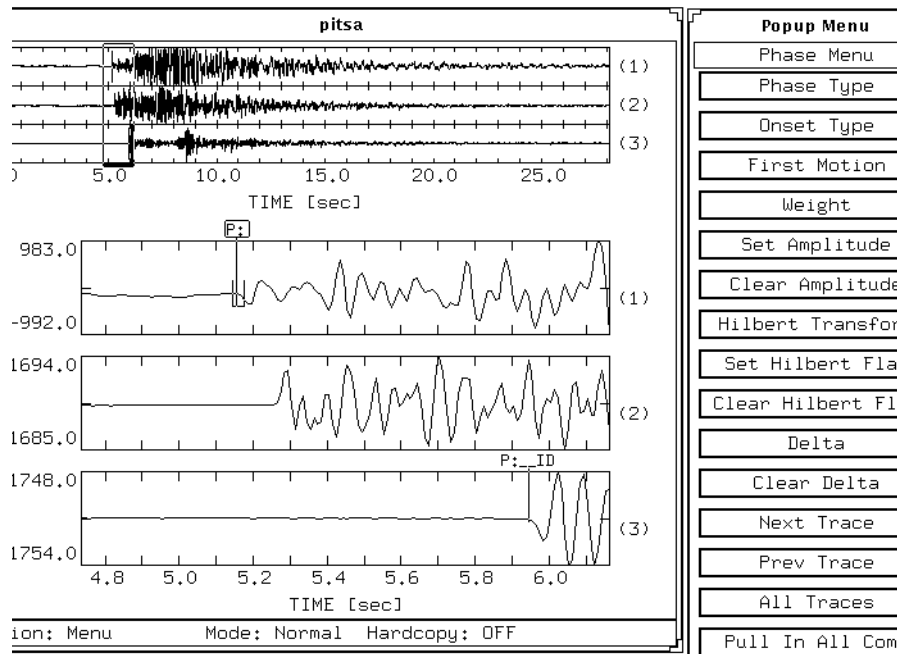
This option is used to set the region of confidence of the onset time of the active phase. The window representing the confidence of the pick need not be symmetric about the onset time. When selected, PITSA will produce a vertical line which tracks the movement of the cursor on the trace that contains the selected phase (Figure 6-27).

Figure 6-27. Measurement of the region of confidence of the onset time of a phase.



The region of confidence is specified by clicking the mouse button at its boundaries. In order to be valid, the two mouse clicks must be on opposite sides of the selected phase.

Once a region of confidence has been selected, PITSA will display an error bar along with the phase pick as in Figure 6-28. If a phase pick is adjusted after this step, then PITSA will clear the existing region of confidence and you will have to re-select it.

Figure 6-28. The error bar indicating the region of confidence for a phase pick.**Clear Delta**

If the currently selected phase has an error bar, then this option can be used to clear it. The error bar will be removed and the confidence region will be reset for the phase pick.

Set Rise Time

In the stand-alone version of PITSA, this option will be dimmed in the *Phase Menu*, meaning it is unavailable. It is available when PITSA is used within the GIANT software package.

Clear Rise Time

In the stand-alone version of PITSA, this option will be dimmed in the *Phase Menu*, meaning it is unavailable. It is available when PITSA is used within the GIANT software package.

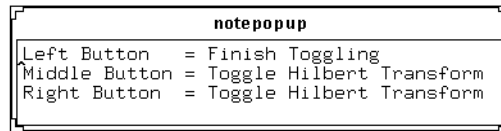
End Phase

In some applications it is useful to mark the end of a phase. This option is used to create a companion phase (the “end phase”) to the active phase for this purpose. The new phase will initially have the same onset time as the original phase, and will have the same phase ID with the text “.end” appended. After this option is selected, clicking once with the left mouse button on the original phase will cause the phase labels to be replotted with some separation. Then the onset time of either phase can be adjusted in the normal manner.

Trace Transform

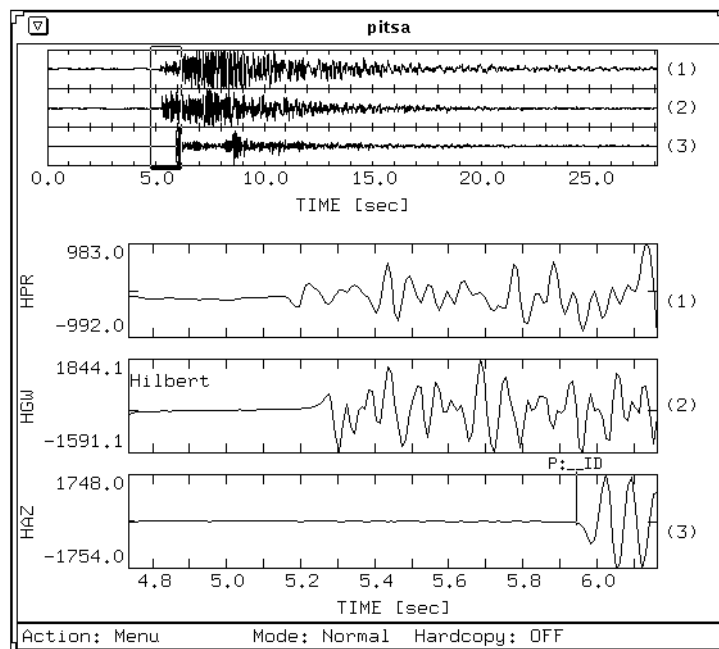
This option is used to apply the Hilbert Transform to the traces in the zoomed boxes of the display. When this selection is made, the note window shown in Figure 6-29 will appear.

Figure 6-29. The Trace Transform note window.



Move the cursor inside the zoomed box of the trace to be transformed and click either the middle or right mouse button. The transformed trace will be redrawn in red and the letter “H” will appear at the beginning of the trace. You can toggle the Hilbert transform on and off in any of the zoomed boxes until the left mouse button is clicked. Figure 6-30 shows the PITSA window after trace (2) has been Hilbert transformed.

Figure 6-30. The PITSA window after trace (2) has been Hilbert transformed.



If the boundaries of the zoom box are changed, any traces that have been Hilbert transformed will be un-transformed before the new zoomed traces are redrawn.

Note that the labelling of phases to show that they were picked from a transformed trace is now entirely manual. The next two options of the *Phase Menu* control these functions.

Set Transform Flag

This option will set the active phase's descriptor to indicate that it was picked while the Hilbert transform was on, by setting the first position of the phase label's phase descriptor field to an “H”.

Clear Transform Flag

This option will set the active phase's descriptor to an underscore character to indicate that it was picked on an un-transformed trace.

Next Trace

This option will change which traces are displayed in the bottom part of the screen (the zoomed up traces). When this selection is made, PITSA looks for the first trace in the upper part of the screen that is on, toggles it off, turns the next trace on and turns all other traces off. This can be used to conveniently step through the traces at maximum resolution.

Prev Trace

This option is the reverse of *Next Trace*. PITSA looks at the traces in the upper part of the screen from bottom to top for the first trace that is on, toggles it off, turns the trace that is above it on, and toggles the remaining traces off.

All Traces

This option turns on all the traces in the upper part of the screen so that all traces are displayed in the lower part of the screen (the zoomed traces).

Pull In All Comp

This option is used to ensure that all components of a given station are loaded into the phase picking screen. After selecting this option, move the cursor onto the trace of interest (either in the upper or lower set of traces); As the pointer moves into different traces, a note window displays the station name and component of the trace. When any mouse button is clicked, PITSA looks for any traces with the same station name that are not loaded into the phase picking screen. If it finds some, they are added to the channel list that is being picked.

Pull In Nearest Sta

This option works nearly the same as the *Pull In All Comp* option except that PITSA looks for the closest station (calculated by looking at the station coordinates) that is not already loaded into the phase picking screen and adds it to the channel list that is being picked.

Clear Pulled In

If traces have been pulled in with either the *Pull In All Comp* or *Pull In Nearest Sta* options, then they can be taken out of the channel list loaded into the phase picking screen with this option.

Refresh

This option causes the phase picking screen to be redrawn. This can be useful when placement of phase labels has become messy because of adjusting, adding or deleting phases.

Pick Magnitude

In the stand-alone version of PITSA, this option will be dimmed in the *Phase Menu*, meaning it is unavailable. It is available when PITSA is used within the GIANT software package.

DONE forward

This option is used when the user is done picking phases (i.e., using the options available under the *Phase Menu*) on the channels loaded into the phase picking screen. If there are more traces to look at in the main channel list, PITSA loads the next batch of traces. When all traces have been processed, control returns to the *Phase Picking Menu*.

DONE backward

This option is identical to *DONE forward*, but loads new sets of traces in the reverse order.

<ESCAPE>

This option returns control immediately to the *Phase Picking* menu.

6.4.3 Show False Polarities

In the stand-alone version of PITSA, this option will be dimmed in the *Phase Menu*, meaning it is unavailable. It is available when PITSA is used within the GIANT software package.

6.4.4 Show Greatest Residuals

In the stand-alone version of PITSA, this option will be dimmed in the *Phase Menu*, meaning it is unavailable. It is available when PITSA is used within the GIANT software package.

6.4.5 Sort Traces by Distance

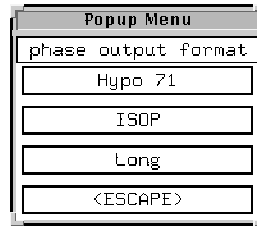
When this option is selected, PITSA will sort the traces loaded into PITSA by either hypocentral or epicentral distance, if the necessary information is available in the data headers. If the information is not available, this field is grayed out and inactive.

6.4.6 Sort Traces by Azimuth

When this option is selected, PITSA will sort the traces loaded into PITSA by azimuth, if the necessary information is available in the data headers. If the information is not available, this field is grayed out and inactive.

6.4.7 Output Phase(s)

This option is used to write phase pick data to a file in several standard formats. When this option is selected, a menu pops up that gives the user three format options (Figure 6-31). These options are *HYPO71*, *ISOP* and *Long*. The *HYPO71* format is directly readable by the *HYPO71* program (Lee and Lahr, 1975; Lee and Valdés, 1989). The *Long* format is a general format that includes all the phase pick information.

Figure 6-31. The phase output menu.

After a format has been selected, PITSA will ask the user for a channel list and a name for the output file. The file will be written in the current PITSA data directory. If the file name entered does not already exist, then PITSA will create the file and add all the phase picks in the channel list to the file. If, however, the file does exist, PITSA will overwrite the file without warning.

6.4.7.1 Processing of HYPO71 Phase Output Files

The HYPO71 program expects to see only P and S arrivals and only one P or S arrival per station. PITSA will group the traces together by station and if there are more than one P or S arrivals, then PITSA selects the phase pick with the highest quality by looking at the weight and whether or not the phase is impulsive or emergent. Also, HYPO71 requires three additional pieces of information besides the onset time and phase name. It must know if it is impulsive or emergent, the first motion and the weight. PITSA checks to make sure all these values are set and if they are not, it fills these values in with default values. The default values are impulsive, unknown first motion, and full weight (Table 6-5 on page 6-21).

6.4.7.2 Processing of phase output files in the ISOP format

The ISOP format can be used to dump picked phases into a file following the format description of the ISOP project. There are no default values used in this format.

6.4.7.3 Processing of phase output files in the Long format

This output format contains most information about the picks made inside PITSA. It just dumps all information that have been found for the picked phases in a file. As for the ISOP format there are no default values defined, if information is missing.

6.4.7.4 Phase Output Status Message

After PITSA has finished writing to the phase output file, it will pop up a status message in the upper right hand corner of the screen (Figure 6-32).

Figure 6-32. The phase output status message.

Total number of phases in new file	= 2
Number of new phases used	= 2
Number of new phases ignored	= 0
Number of old phases used	= 0
Number of old phases ignored	= 0
Number of warnings	= 0
CONTINUE	

6.4.8 Output Station(s)

This option is used to generate station files in the format used by HYPO71. PITSA asks for a channel list and the name of the file to create. The file will be created in PITSA's current directory.

As in *Output phase(s)*, if the file already exists, PITSA will overwrite this file without warning.

6.4.9 Run Location Program

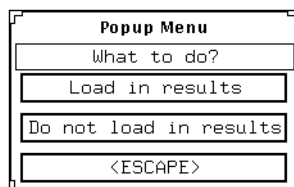
PITSA can run HYPO71 directly from the program and load in the phase residuals of the run. In order to do this, the user must first generate a phase and station file to be used as input to the location program.

6.4.9.1 Running HYPO71

PITSA will ask the user for a number of files that will be used as input and output. The first file that is asked for is the velocity model file, which is `velmod.hdr` by default. PITSA will look in the path set by `PITSA_HYPO71_PATH_ENV` for the velocity model Section 2.1. Next PITSA will ask for the name of the control header file to use which is `control.hdr` by default; PITSA will also look in the `PITSA_HYPO71_PATH_ENV` directory for this file. Next PITSA will ask for the station and phase file to use and will look for these files in PITSA's current directory Section 5.3. Finally PITSA will ask for the name (`hypo_out` by default) of the output file which is where HYPO71 will write the results of the location run.

After all the input is entered, PITSA will issue a system command to run HYPO71 and wait for it to finish. Any output generated by HYPO71 will appear in the window where PITSA was started. Then PITSA will ask the user if they want to load in the results of the location (Figure 6-33). If the user selects *Load in results*, then PITSA will read in the new location from the output file HYPO71 created and create synthetic phases for each residual. If there were errors in the HYPO71 run, the user may want to skip trying to load in results by selecting the *Do not load in results* option.

Figure 6-33. The menu PITSA pops up after the HYPO71 location program has finished. The user can either load in the new location and phase residuals with the *Load in results* option or skip reading in this information by selecting *Do not load in results*.



If the *Load in results* option is selected, then PITSA will first clear out any synthetic phases that exist (remember that synthetic phases do not have a colon after the phase id). PITSA will then display a message in the upper left hand corner of the screen that shows the new location and some statistics (Figure 6-34).

Figure 6-34. After PITSA loads in a new location and phase residuals, it pops up this message in the upper left hand corner of the screen. The statistics

for the P and S residuals are the average and maximum residuals in seconds.

New Location:	36.95 [lat deg]	-121.74 [lon deg]	16.66 [depth km]
P residuals:	0.188 [ave sec]	0.650 [max sec]	11 [# residuals]
S residuals:	0.000 [ave sec]	0.000 [max sec]	0 [# residuals]
11 synthetic phases created			
CONTINUE			

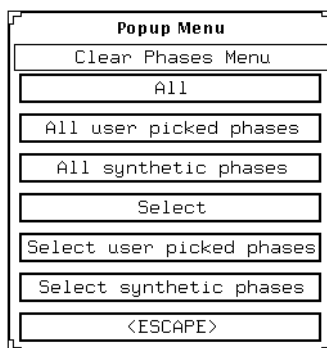
6.4.9.2 Solving Problems with HYPO71

HYPO71 is a complicated program in its own right, and it may be that you encounter problems running it from PITSA. A guide to solving problems in HYPO71 is included here as Appendix H. More complete information will be found in the HYPO71 Manual.

6.4.10 Clear Phases

To delete phase picks from the trace data headers, select the *Clear Phases* option. The menu in Figure 6-35 will be displayed:

Figure 6-35. The *Clear Phases* pop-up menu.



6.4.10.1 Clearing All Phases

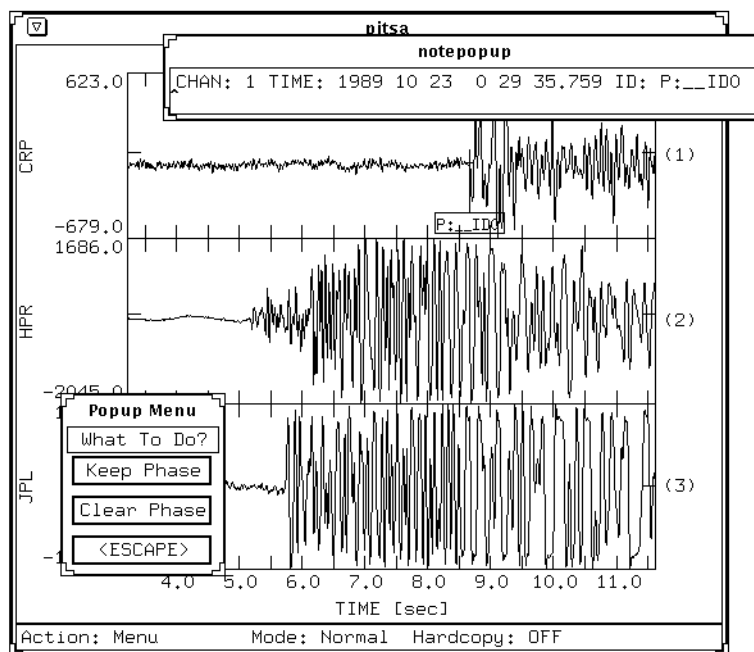
The user has three options when clearing all phases; they can clear user-picked phases and synthetic phases (*All*), just user-picked phases (*All user picked phases*) or just synthetic phases (*All synthetic phases*). When one of these options is selected, PITSA will first ask for a channel list to clear phases from. Then the user will be presented with a pre-zooming window so that they can select the portion of the traces in the channel list to clear. When the zoom window has been selected, PITSA will display all the traces in the channel list with the zoom selected and overlay all phase picks that would be deleted. Then PITSA asks the user if they really want to delete these phase. The user can either enter a *n* (the default) to terminate the operation or enter *y* to go ahead and delete the phases. If the user enters *y*, then PITSA will erase the phase picks overlaid on the traces, otherwise the phase picks will remain. Then PITSA returns to the *Phase Picking Menu*.

6.4.10.2 Clearing Selected Phases

The user has the option of stepping through all selected phases (*Select*), all selected user-picked phases (*Select user picked phases*) or selected synthetic phases (*Select synthetic phases*) one at a time and deciding if the phase should be deleted. First PITSA will ask for a channel list and then let the user pre-zoom the traces in the channel list. PITSA will then display the traces selected in the channel list using

the pre-zoom selected. Then PITSA will start stepping through all phases contained in the traces with in the zoom area (Figure 6-36).

Figure 6-36. PITSA's Clear Selected Phase screen.



For each phase, PITSA will display the phase label surrounded by a box, display the phase information in a note, and ask the user if the phase should be deleted or not. If the user selects *Keep Phase*, then the phase is not removed from the traces header and the phase is left on the screen without the box around it and PITSA moves on to the next phase. If *Clear Phase* is selected, then the phase label is erased from the screen and the phase information is deleted from the traces header before moving on to the next trace. After all phases have been examined, PITSA returns to the *Phase Picking Menu*.

6.4.11 Delete Traces

Warning: Use this option with great care. It will remove files from the memory in PITSA and from your disk! This is the only possibility to physically delete waveforms inside PITSA. So please be careful with this. When using PITSA in connection with the GIANT database, the according waveform files and any other connected entries (i.e. picked phases) are also deleted from the database.

6.4.12 Plot All

After PITSA returns from a *Phase Picking Menu* option, it re-displays the menu but does not re-plot the screen with all the channels loaded into PITSA. If the user would like to see the traces loaded into memory, they can select this option.

6.4.13 DONE

This *Phase Picking Menu* option will end the phase picking mode and return PITSA back to its home state. PITSA's main menu will then become active again.

6.5 Magnitudes

PITSA can be used to estimate earthquake magnitude, but it is important to understand that reliable estimation of magnitude depends critically on the proper calibration of instruments. In practice, many seismological instruments are only roughly calibrated, for example, by using the specifications supplied by the manufacturer.

Five magnitude types are currently supported:

Table 6-6. Magnitude Types and Menu Options.

Menu Option		Description
Bakun/Joyner local magnitude	M_L	Local magnitude (Bakun and Joyner, 1984)
Generic local magnitude	M_{WA}	Wood Anderson magnitude (Richter, 1935)
Body Wave Magnitude	m_b	Willmore, 1979
Surface Wave Magnitude	M_S	Willmore, 1979
Moment Magnitude	M_W	Moment magnitude (Kanamori, 1977)

Before attempting to determine earthquake magnitudes from seismic traces, the user must make sure that the proper corrections for the instrument type have been applied. For m_b and M_S , the trace units are expected to be in ground velocity (m/sec) while for M_L and M_{WA} the units are assumed to be in ground displacement (m) amplified by a factor of 2800 to simulate the output of a Wood Anderson instrument.

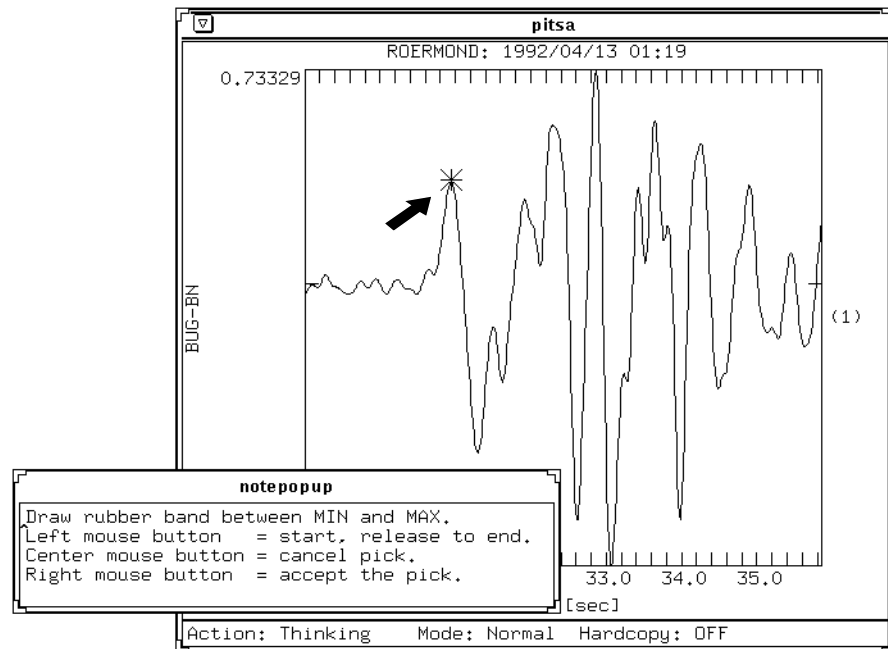
It is advisable to perform an instrument correction and simulation of standardized instruments before estimating magnitudes. See Section 6.3 for the simulation of standardized instruments or Section 8.2 for amplitude scaling of individual traces.

After selecting one of the *Routine Tools* -> *Magnitude* options (Table 6-6), you will be asked for a channel list. Next, you will be given the opportunity to zoom in on the selected traces to focus on the area of maximum amplitudes.

Magnitude calculation is done by selecting the two extremal values in a trace which represent the peak-to-peak amplitude. The same process will determine the corresponding dominant signal period, which is required for M_S or m_b . The process is identical to the operation of the *Phase Menu* -> *Set Amplitude* option discussed in Section 6.4.2

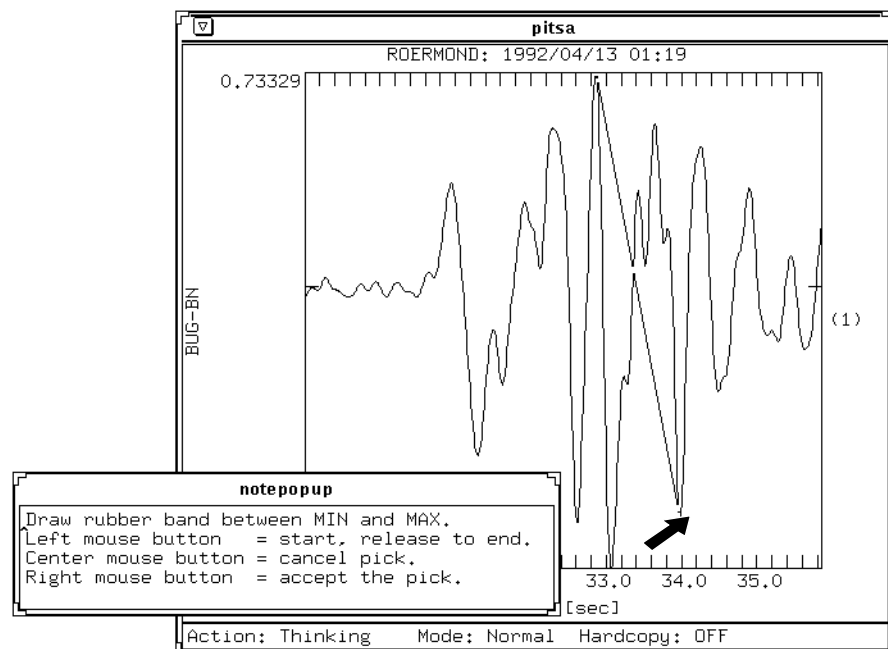
Figure 6-37 shows the display at the start of the magnitude estimation process. A star tracks the motion of the cursor by 'jumping' to nearest local extreme value of the trace.

Figure 6-37. The magnitude determination screen, initial display.



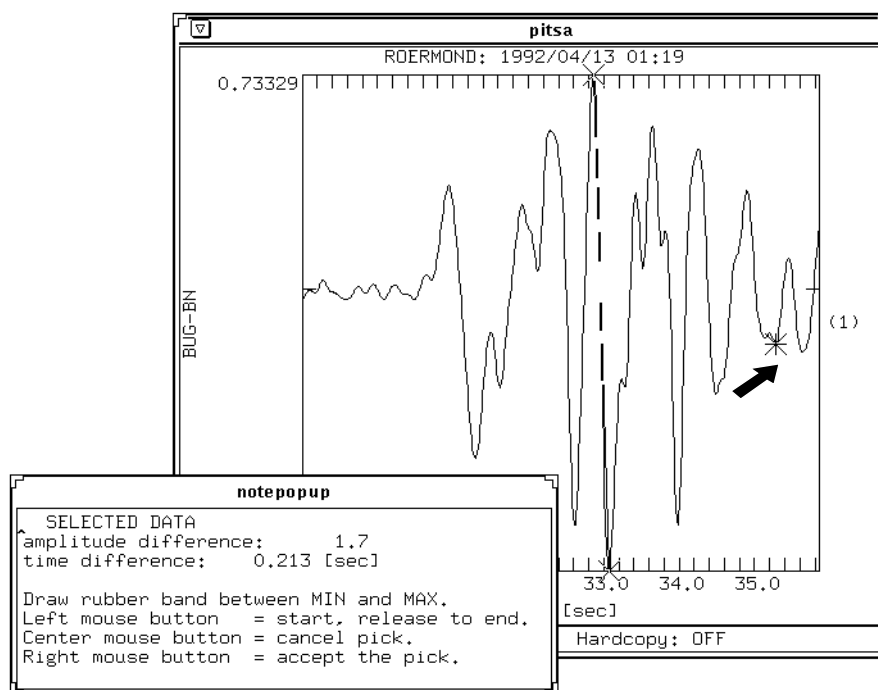
When the star is located on the extremum which you wish to use for the start of the magnitude measurement window, click and hold down the left mouse button—this “locks in” the first extremal point. As you drag the mouse away, a line will be drawn from the first extremal point to the extremum nearest the cursor. Figure 6-38 shows an intermediate step in this process in which the cursor has been moved too far and the line is being drawn to the wrong extremal value.

Figure 6-38. The magnitude determination screen after the first point has been selected and the user is attempting to select the second point.



In Figure 6-39 the cursor has been moved back and the correct extremal value has been selected.

Figure 6-39. The magnitude determination window after two values have been selected.



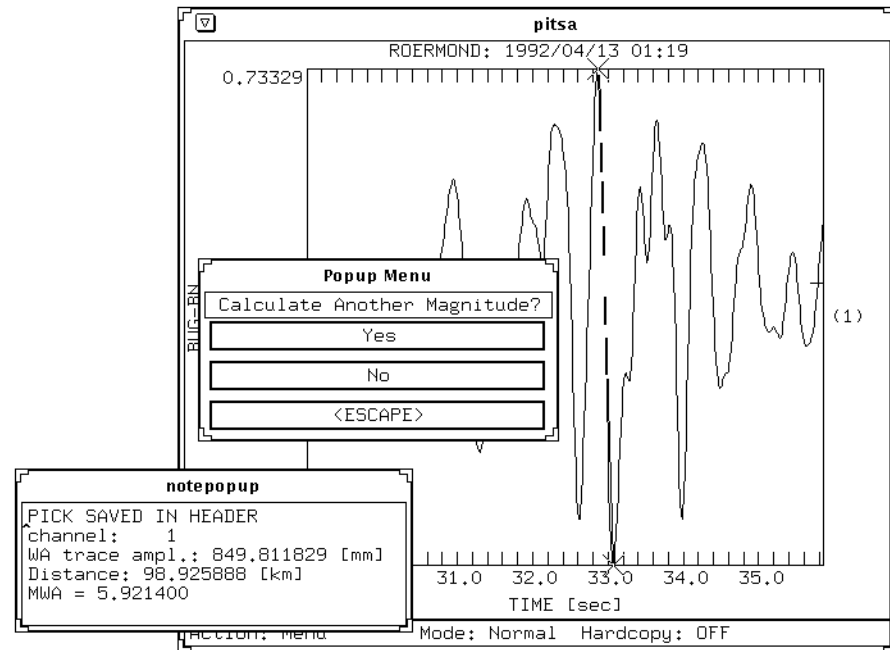
Note in Figure 6-39 that a star continues to track the cursor even after two extremal points have been selected. Once a pick has been made, you may:

- Reselect the start and end points for the magnitude, using the left mouse button to start a new selection line.
- Accept the current selections and calculate a magnitude, by clicking the right mouse button.
- Cancel the operation, by clicking the center mouse button.

Depending on what type of magnitude determination was selected, you may be asked to input additional information for distance-dependent amplitude correction if the needed data values are not set in the trace's header. For the m_b determination, this term (often called Q) depends on the selected wave type and source depth, while for M_{WA} it is only dependent on epicentral distance. For tabulated values of the calibration term see Willmore (1979). For the calculation of M_L and M_S , no further input is required.

When a pick has been made and accepted (Figure 6-40), PITSA will make a magnitude calculation, store it in the trace's header (Section 8.3.2) and display the result of the calculation.

Figure 6-40. The magnitude determination window after the selected pick has been accepted.



After the calculations are made, you will have the option of picking additional magnitudes or exiting.

There are several differences to the just described procedure when using the *Moment Magnitude* option. With this routine the spectra of given phase picks are computed and an inversion is started assuming a Brune source (Brune, 1970). This inversion is based on a hybrid of the simulated annealing and simplex algorithms (Press et al, 1992). The final output is the inverted spectrum and the estimated M_w magnitude. To run this module properly an input file named *annealing.cfg* is needed. In this file the starting values of the simplex and simulated annealing algorithms as well as starting values of the model are given (Figure 6-41).

Figure 6-41. Starting values of the spectral inversion given in annealing.cfg file.

```

MODEL MOMENT 1.0e-6
MOMENT UNSERT 0.0001
MODEL CORNER FREQ 20.0
MODEL CORNER FREQ UNSERT 0.1
MODEL HIGHPASS 0.5
MODEL LOWPASS 30.0
MODEL POLES 2
MODEL INCIDENCE 5.0
MODEL Q 200
MODEL Q UNSERT 0.005
MODEL SEIS Fc 1.0
MODEL SEIS DAMPING 0.7
NUMBER OF RUNS 1000
NUMBER OF ITERATIONS 10
ANNEALING TEMPERATURE 1.0
ANNEALING TEMPERATURE STEP 0.7
RADIATION PATTERN WEIGTH 1.0
VELOCITY (m/s) 5000
DENSITY (m^3/kg) 2700
LOW FREQ INVERS -1.0
HIGH FREQ INVERS 20.0

```

PITSA prompts first for the number of points of the FFT, the tapering factor, pre-pick window length in fraction of the overall FFT window and the number of points for smoothing the spectra (Figure 6-42).

Figure 6-42. Input panel for moment magnitude estimation.

Specify nfft trace, tapering frac, window frac, smooth
256,0.200000,0.050000,4
.separate by commas

A noise sample with the same length as the phase wavelet is taken just in front of the selected pick.

Next you are asked to input the phase ID which you want to use for estimating the moment magnitude (Figure 6-43).

Figure 6-43. Entering the desired phase ID.

Specify phase id
P

In order to estimate an instrument corrected moment magnitude, PITSA prompts for the calibration file of the instrument (Figure 6-44).

Figure 6-44. Name of calibration file for used instrument.

Enter calibration file for instrument
<input type="text"/>
Return if specify instrument manually

If you don't want to use a specific calibration file (or you just don't know it exactly), you are asked to enter *return*. Now, PITSA will ask you to input the calibration information manually (Figure 6-45).

Figure 6-45. Calibration information of used instrument.

Enter calibration fc, damping, gain, sensor type(dis = 1, vel = 2)
1.000000, 0.707000, 1.000000, 2

PITSA will now try to perform an inversion using the given spectra and annealing.cfg file.

The result is plotted as a colored model curve together with the spectra of the selected phase window, the smoothed version and the noise spectrum. Please be aware that PITSA only computes a model spectra if 10 spectral values of the signal are above the noise spectrum.

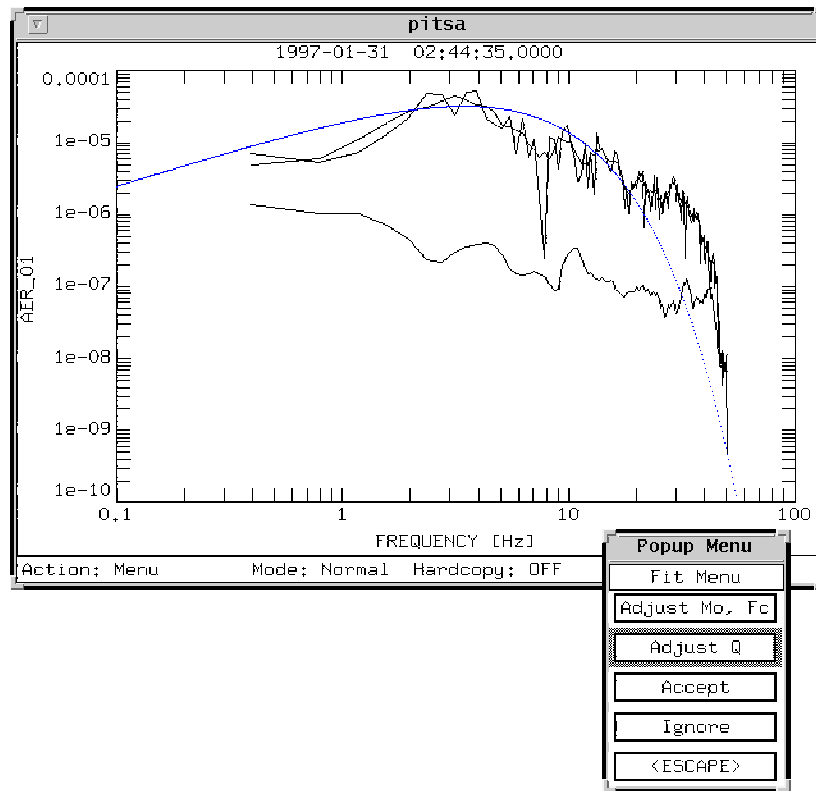
Next a popup menu will appear in the PITSA main screen. Here the user can decide whether to control the output model fit interactively (Figure 6-46).

Figure 6-46. Interactive quality control of the fit.

Popup Menu
Quality Control
Next Trace
Previous Trace
<ESCAPE>

Selecting *Next Trace* or *Previous Trace* results in entering into the interactive mode. <Escape> simply accepts all models and all associated moment magnitudes are stored in the header.

Figure 6-47. Result of inversion in the interactive mode. The blue line is the model curve, the black lines represent the original spectra, a smoothed version and the noise spectra, respectively. Also the popup menu for interactive change of the fit is visible



If you are in the interactive mode, you are able to control the actions via a popup menu (Figure 6-47). *Adjust Mc, fc* simply adjust the Moment and the corner frequency of the assumed Brune source model. A cross hair cursor will appear and you can set the fit to a new value. *Adjust Q* will adjust the Q-value of the model and leaves the rest unchanged. *Accept* will accept either the original model or the manually adjusted one. Selecting *Ignore* deletes this spectra out of the channel list and no magnitude is computed.

The moment magnitude M_W is computed using the equations:

$$M_W = \frac{2}{3} \cdot \text{Log}_{10} M_0 - 6.063 \quad \text{and} \quad M_0 = \frac{4\pi \cdot P \cdot \rho \cdot \Delta \cdot v^3}{R}.$$

M_0 represents the Moment, P the plateau value of the spectrum, ρ the density of the propagation medium, Δ the hypocentral distance, v the P- or S- wave velocity and R the radiation pattern factor.

6.6 Integrate

Numerical integration of a seismic trace is used to convert an acceleration-proportional record to a velocity-proportional record, or a velocity-proportional record to a displacement-proportional record. This may be necessary, for example, in order to make a magnitude measurement (Section 6.5). This is done by selecting one of the methods under *Routine Tools -> Integrate*. The choice of method has consequences for the spectral properties of the resulting signal:

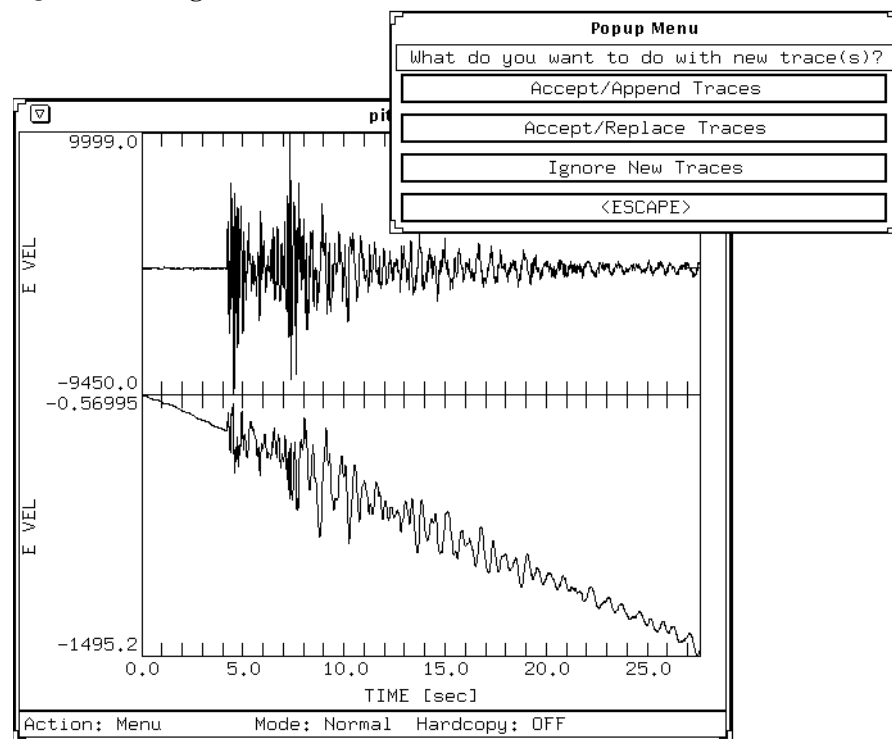
Table 6-7. Spectral properties of integration methods

Method	Effect on spectral properties
<i>Tick's rule</i>	Overemphasizes high frequencies with respect to the true frequency response of an integrating filter.
<i>Trapezoidal rule</i>	Underemphasizes high frequencies with respect to the true frequency response of an integrating filter.

For a discussion of the frequency response of numerical integration using recursive filters see Hamming (1983).

After a channel list has been specified, PITSA will display the first trace along with the integrated trace (Figure 6-48). You have the usual choices of what to do with the new trace, and, in the case of multiple traces, the option to use this choice for all remaining traces as well.

Figure 6-48. Integration result.



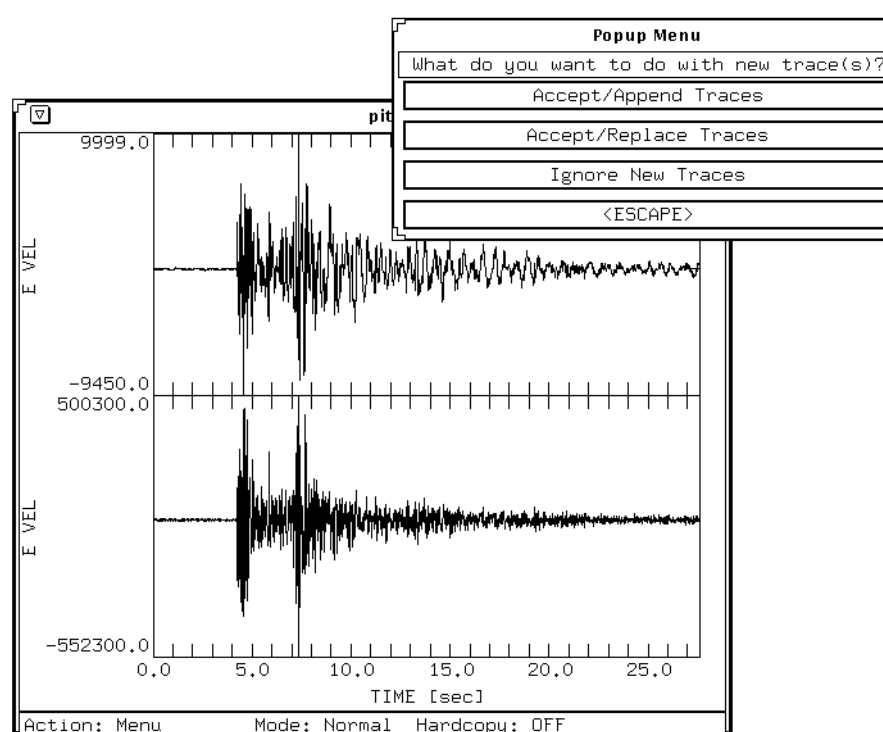
An integrated trace will often have a significant linear trend, as in Figure 6-48, which is caused by even a small amount of baseline offset in the original trace.

6.7 Differentiate

Numerical differentiation of a seismic trace is used to convert a velocity-proportional record to an acceleration-proportional record, or a displacement-proportional record to a velocity-proportional record. This may be necessary, for example, in order to make a magnitude measurement (Section 6.5). This is done by selecting *Routine Tools -> Differentiate*. There are no options for this tool.

After asking for the channel list, PITSA will display the first trace along with the differentiated trace (Figure 6-49). You have the usual choices of what to do with the new trace, and, in the case of multiple traces, the option to use this choice for all remaining traces as well.

Figure 6-49. Differentiation result.



6.8 Baseline Corrections

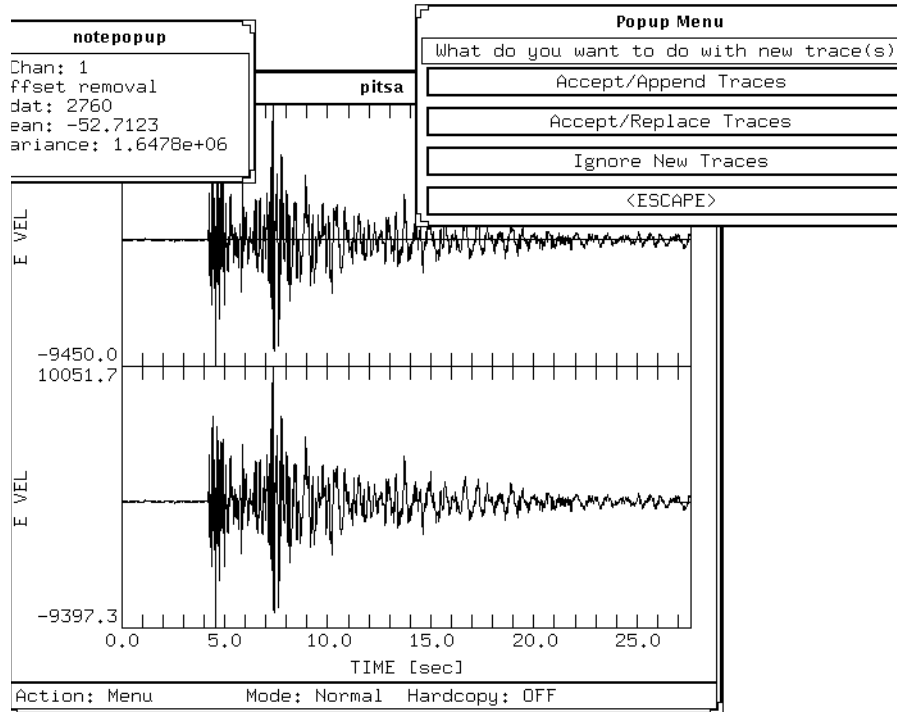
The tools under the *Routine Tools -> Baseline Correction* are used to determine and subtract the slowly varying components of a seismic trace. The options are:

- Running Average (Remove)
- Running Average (Keep)
- Offset Removal
- Linear Trend
- Linear Regression

With any of these options, PITSA will ask for a channel list to process, and then display the first pair of original and baseline-corrected traces (Figure 6-50). You have the usual choices of what to do with the new trace, and, in the case of multiple

traces, the option to use this choice for all remaining traces as well. The main difference in the display for these options is that the pop-up note window will provide different information, corresponding to the specific method used.

Figure 6-50. Baseline correction result.



Some of the problems calling for a baseline correction can also be dealt with by low-pass filtering (Section 7.1.3).

6.8.1 Running Average (Remove or Keep)

For either of these two options, the baseline is calculated from the original trace by sliding an averaging window of a given length over the data series. For each window position, the baseline is calculated as the average value in the data window.

The running average of a data trace is sometimes of interest by itself. It can be saved for further processing by selecting the *Running Average (Keep)* option.

When either option is selected, you will be asked to enter the width of the averaging window in seconds. The window length is directly proportional to the “smoothness” of the correcting baseline.

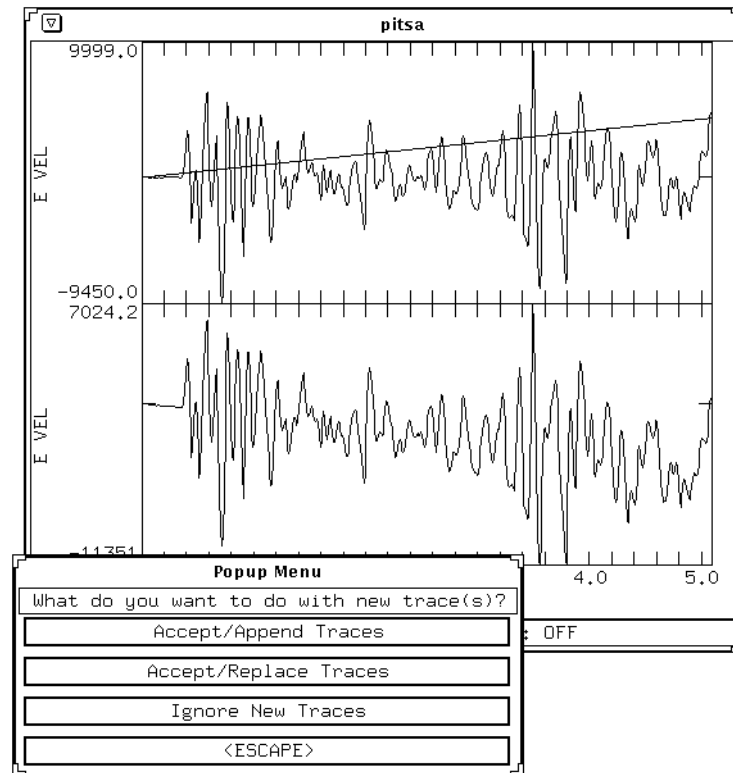
6.8.2 Offset

Here, the baseline is simply the DC component (average value of all points) of the trace.

6.8.3 Linear Trend

This approach calculates the baseline as the connecting line between the first and the last data points of the trace (Figure 6-51).

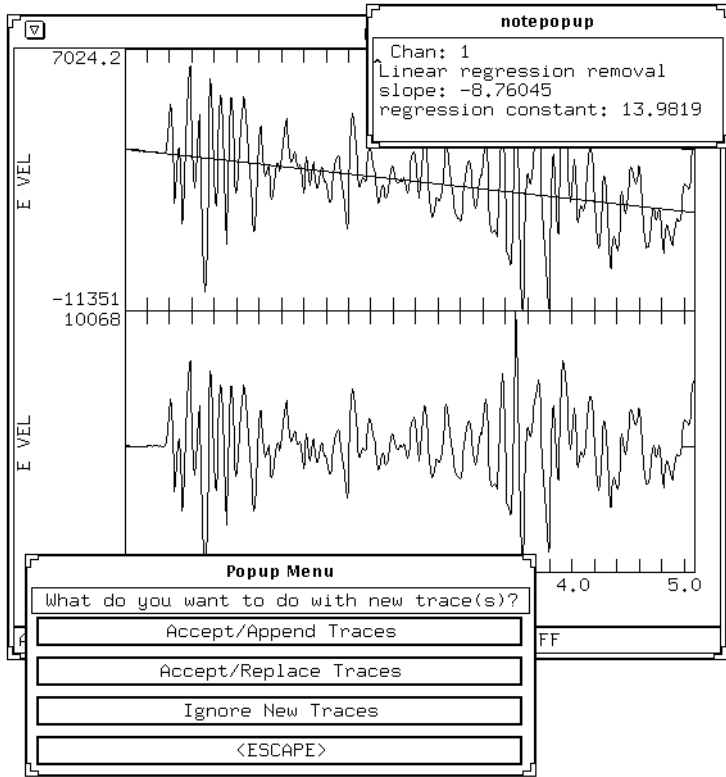
Figure 6-51. Linear trend removal.



6.8.4 Linear Regression

Here, the baseline is determined as the best fitting line calculated from a linear regression analysis of the trace (Figure 6-52).

Figure 6-52. Linear regression baseline correction.



This chapter describes the more sophisticated processing tools to be found in PITSA, under the *Advanced Tools* option of the main menu. Table 7-1 outlines the many tools available.

Table 7-1. The options available under *Advanced Tools*.

Menu Selection	Description
Filter	Various filters including band-, high- and lowpass filters; a de-ringing filter; a notch filter; a user-specified pole-zero filter; and a user-defined difference equation filter.
Spectrum	Calculate FFTs, inverse FFTs and maximum entropy spectra. Demultiplex complex spectra into amplitude, phase, real and imaginary parts; multiplex two real traces into a complex spectrum; Power Spectral Density; discrete wavelet-transform.
Xcorrelation	Scaled and unscaled cross-correlation of two traces.
(De)convolution	Convolution and deconvolution of two traces.
Particle Motion	View ground motion in 2 or 3 dimensions. Also pick phases and angles.
Envelope / Hilbert Trans	Envelope or Hilbert transform of a trace.
Rotate Components	Rotate components in 2 or 3 dimensions.
Polarization Filter	Polarization analysis in 2 or 3 dimensions.
Cross Spectrum	Cross-spectrum of two traces.
Coherence Spectrum	Coherence spectrum of two traces.
Response Spectrum	Response Spectrum
Spectral Ratio	Spectral ratio of two traces

7.1 Filter

To filter selected traces, select one of the *Advanced Tools* -> *Filter* options. From the *Filter* menu, there are 10 options from which to choose (Table 7-2):

Table 7-2. Options under the Advanced Tools -> Filter Menu.

Butterworth Filters (Band-, Low- and Highpass)	Recursive time-domain filters using the bilinear z-transform design of Stearns (1984). Applied in sections of 40 dB/decade or 12 dB/octave for the slope of the transition band. They may be given zero phase characteristic by bilateral filtering.
Gaussian Band Pass Filter	Zero phase frequency-domain filter. Best time resolution for a given bandwidth.
De-Ringing Filter	Removes the effect of reverberations in a water layer or a layer of shallow low velocity sediments (Backus, 1959).
Notch Filter	Recursive time-domain filter using the design routine from Press et al (1988).
Seidl's Simulation	Instrument simulation using the method of Seidl (1980)
Difference Equation	A general linear time-invariant (LTI) system can be completely described by a linear difference equation. Given the coefficients in an ASCII file, PITSA will directly apply the difference equation to the input trace.
GSE Calibration File	Frequency-domain filter for a GSE calibration section provided in an ASCII file.
Evaluate GSE Response	Evaluate and display various aspects of the frequency response function (e.g. group delay, phase delay, etc.) for a GSE calibration section provided in an ASCII file.

7.1.1 Forward-Backwards (Bilateral) Filtering

For some of the filter options, PITSA will ask if the filter should be applied “forward backwards”, meaning bilaterally (Figure 7-1). If so, the trace will be filtered twice (in opposite directions) to produce a zero phase filter. Bilateral filtering will make the filter transition band(s) twice as steep.

Figure 7-1. Bilateral (“forward-backwards”) filter option.

Filter forward backwards (y/n) ?
y
(y -> zero phase filter) NOTE: zero phase option doubles the steepness of the transition band!

7.1.2 Butterworth Band Pass

The user must provide input for three parameters:

- Low and high cutoff frequencies.
- Whether or not to filter “forwards backwards” (Figure 7-1).
- Number of filter sections. Each filter section corresponds to an increase of the slope of the transition band by 20 dB per decade.

Figure 7-2 shows the impulse response of the filter. Trace 1 is an impulse filtered forwards only (trace 2) and then forwards backwards (trace 3). Notice that the zero phase filter is acausal (the output signal starts before the input signal arrives).

Figure 7-2. Butterworth bandpass filtering of an impulsive signal. Trace 2 used cutoff frequencies of 10.0 and 30.0 Hz, forward filtering only, with a three section filter. Trace 3 uses forward-backwards filtering.

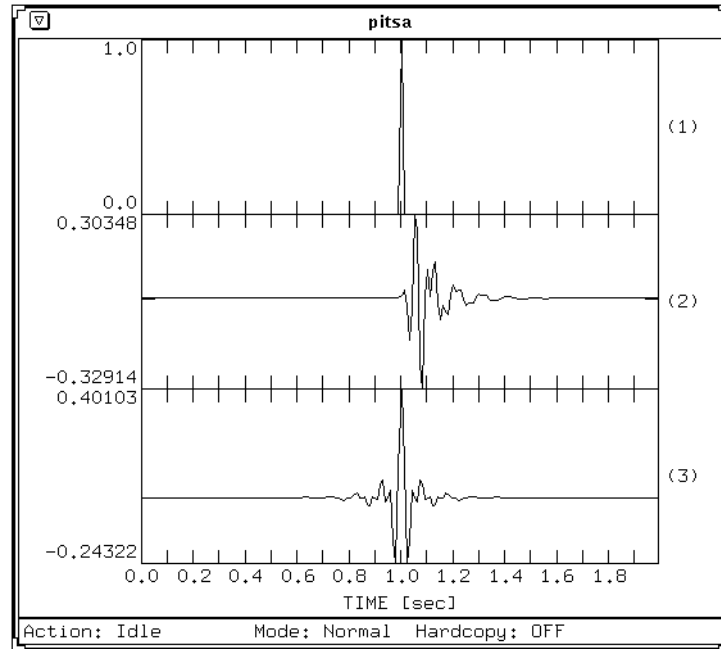
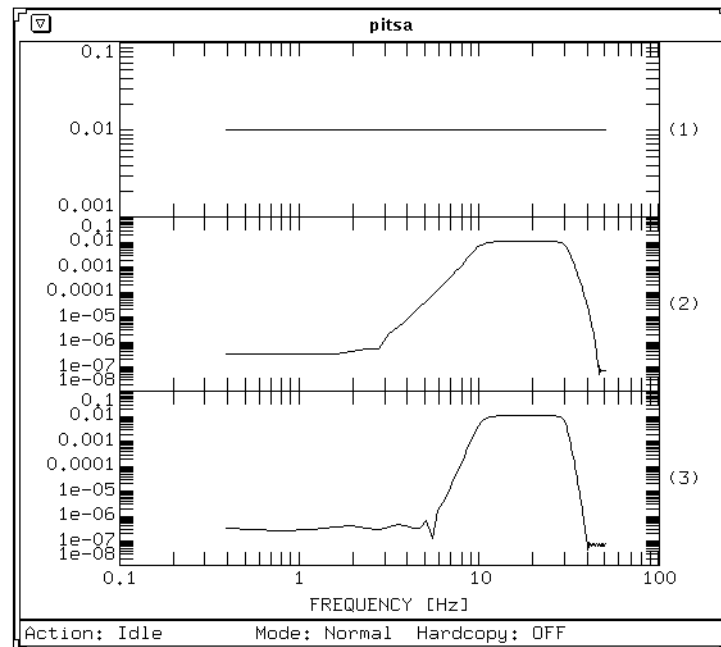


Figure 7-3. FFT of the traces in Figure 7-2. Note the steeper slope of the transition band for the bilateral filter used in trace 3.



7.1.3 Butterworth Low Pass

The user must provide the same input as for the bandpass filter (Section 7.1.2), except only a single cut-off frequency is required.

Figure 7-4. Impulse response of a Butterworth low pass filter (2 section Butterworth Low Pass filter with a corner frequency of 4 Hz) both causal (trace 2) and acausal (trace 3).

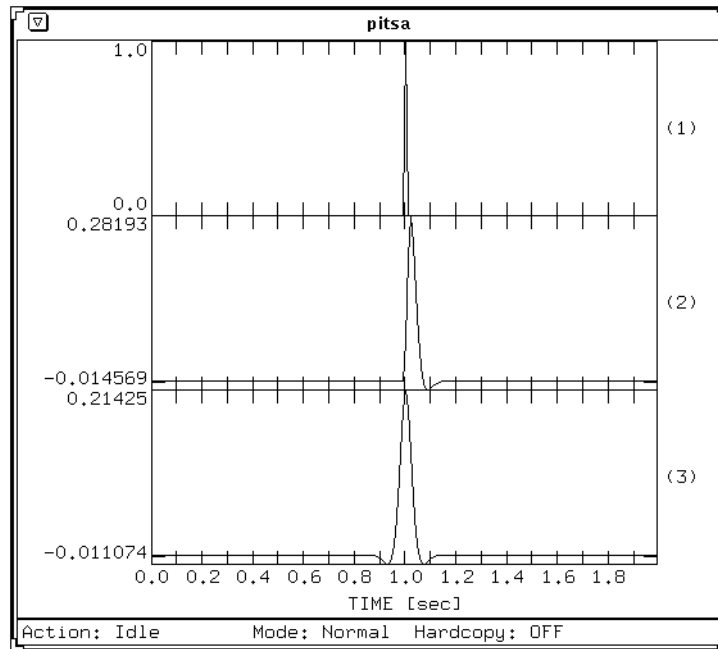
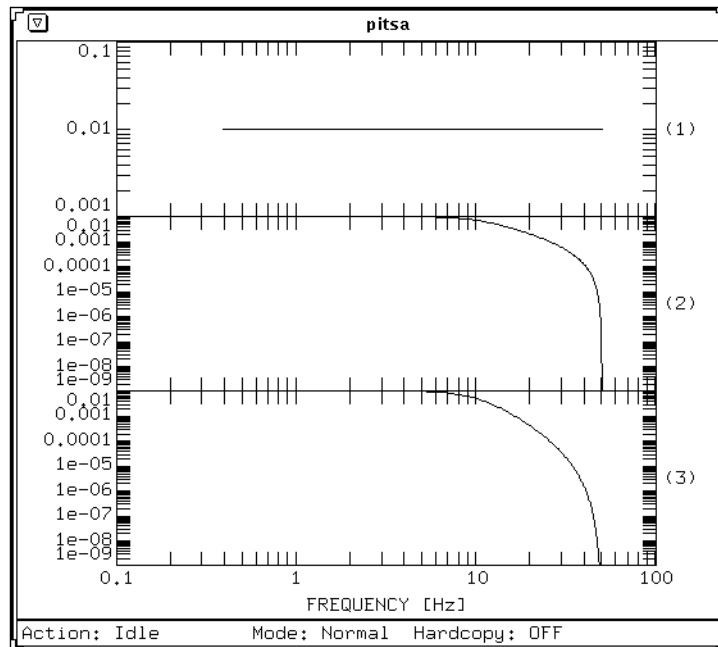


Figure 7-5. The FFTs of the traces in Figure 7-4.



7.1.4 Butterworth High Pass

The user must provide the same input as for the bandpass filter (Section 7.1.2), except only a single cut-off frequency is required.

Figure 7-6. Causal (trace 2) and acausal impulse response of the Butterworth high pass filter.

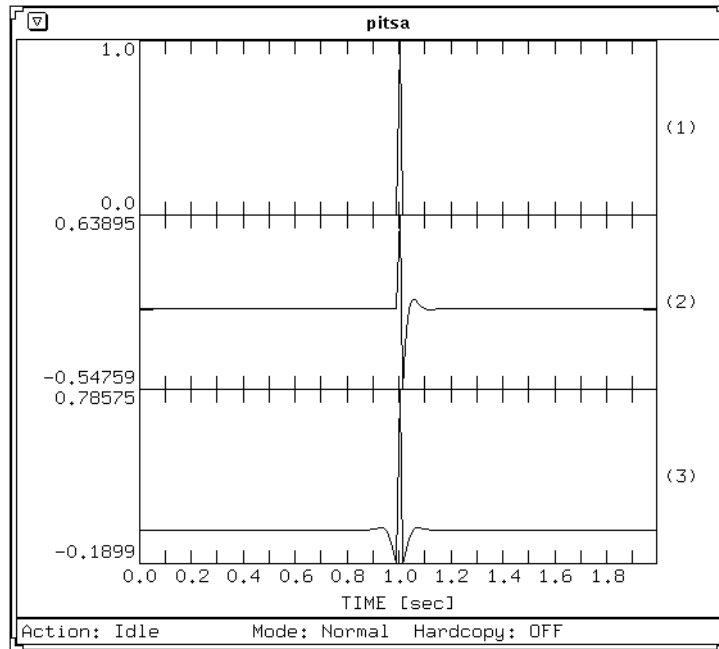
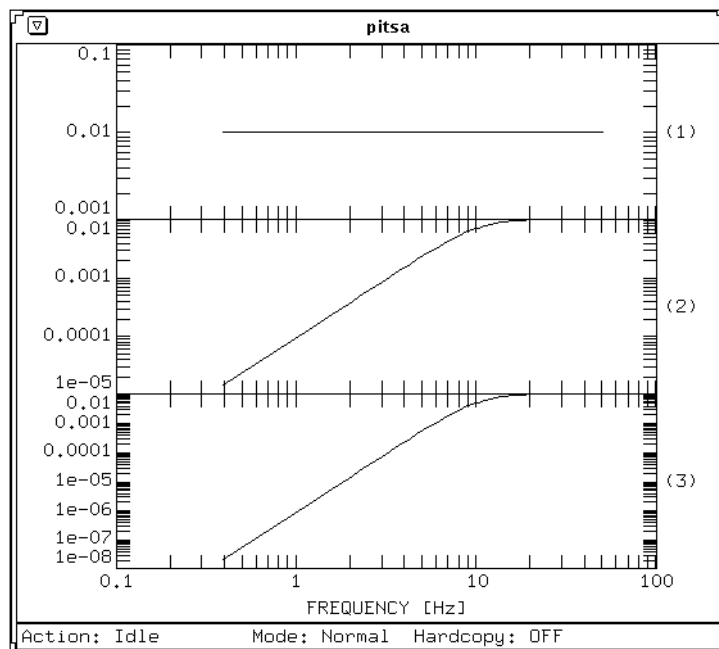


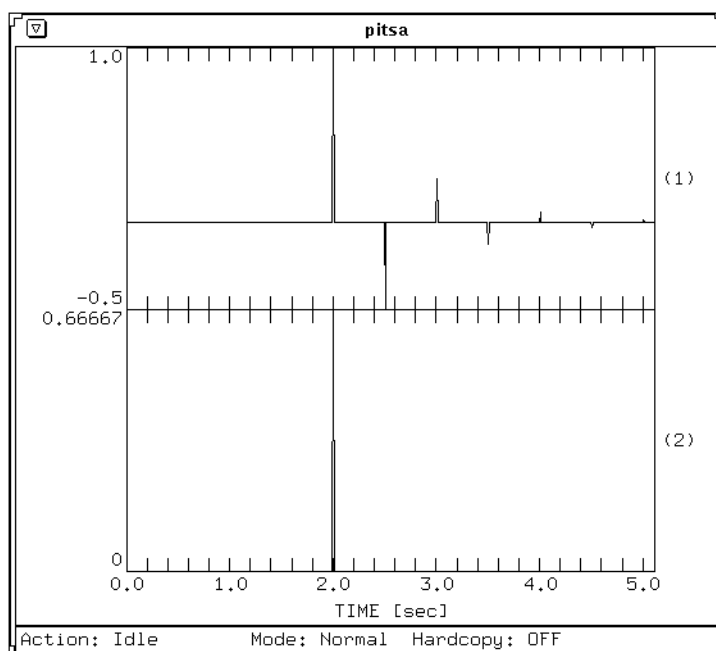
Figure 7-7. The FFTs of the traces in Figure 7-6.



7.1.5 De-ringing Filter

De-ringing filters are used to remove reverberations in shallow low-velocity sediments or in water layers. The user enters the assumed reflection coefficient of the shallow layer boundary and the two-way travel time within this layer. If the input values match the conditions which produce ringing in the seismic record, the de-ringing filter will eliminate all reverberations within this layer (see Figure 7-8).

Figure 7-8. Example of the de-ringing filter. The upper trace shows a synthetic input signal containing multiple reflection signals at a distance of 0.5 seconds. The amplitudes decrease with a factor of 0.5. The bottom trace has been filtered with a de-ringing filter with a reflection coefficient of -.5 and a 2 way travel time of 0.5 seconds.



7.1.6 Gaussian Band Pass

The Gaussian bandpass filter is a non-causal filter which is applied in the frequency domain. For a given center frequency f_{cent} and a given bandwidth α , the Gaussian:

$$\exp\left(\frac{-4\pi^2(f - f_{cent})^2}{4\alpha^2}\right)$$

is calculated and multiplied with the FFT spectrum of the input trace. Subsequently, the inverse FFT is calculated. The user enters values for the center frequency f_{cent} and the bandwidth α .

Figure 7-9. The impulse response of the Gaussian Band Pass filter with a center frequency f_{cent} of 5 Hz and a bandwidth α of 3.5.

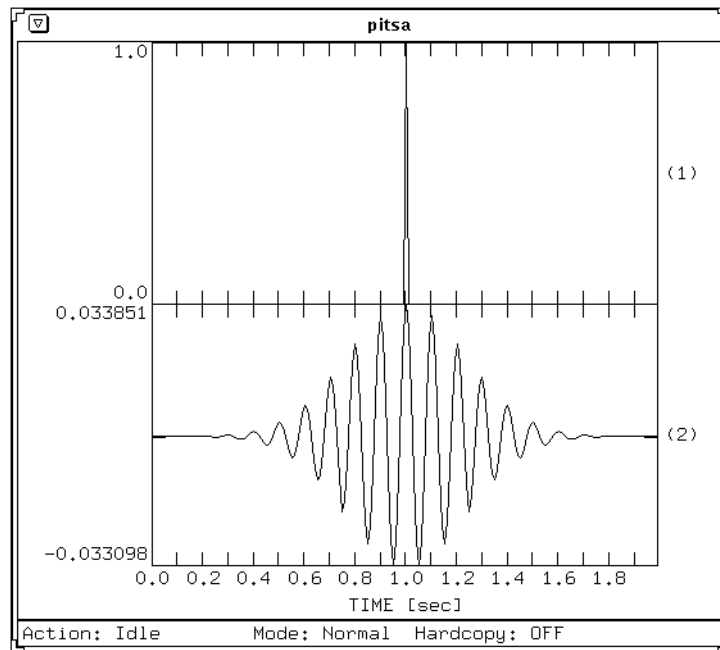
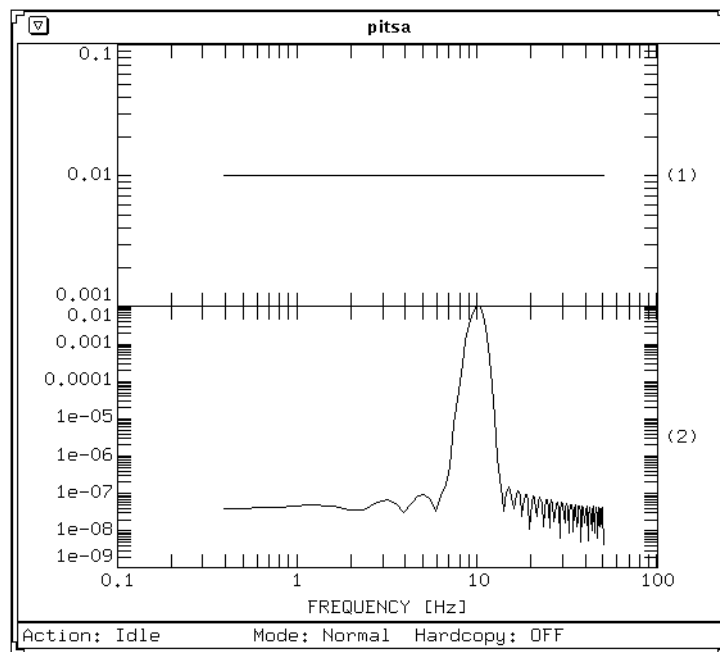


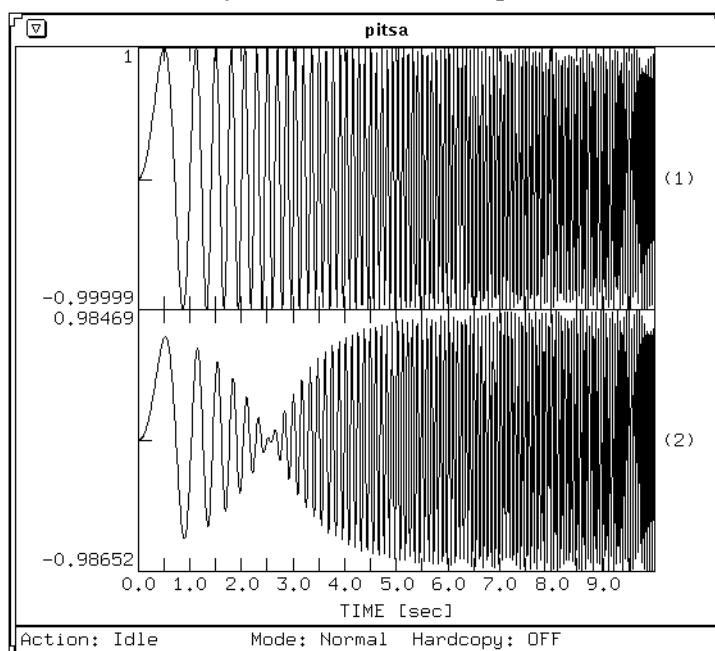
Figure 7-10. The FFT of the traces in Figure 7-9.



7.1.7 Notch Filter

Notch filters are designed to remove only a narrow frequency band around a certain center frequency. User input consists of the notch center frequency and the notch width. Notice that the notch width has to be entered in fractions of the notch center frequency. Due to the underlying design procedure (bilinear z-transform), this filter should only be used for notch center frequencies small in comparison to the Nyquist frequency, due to inherent non-linear distortions of the frequency axis. It is wise to always check the actual notch frequency with synthetic data.

Figure 7-11. Example of a notch filter. The upper trace shows a synthetic input signal containing a chirp signal (Section 8.1.9) for frequencies between 0 and 10 Hz. The bottom trace has been filtered with a notch filter with a center frequency of 7.5 Hz and a notch width of 0.5. Notice that the signal is ‘pinched’ at around 3.0 seconds instead of at 7.5 seconds as one might expect since the equation for the chirp signal is $y(t)=\sin(\theta(t)*t)$ and $\theta(t)=7.5$ Hz when $t=7.5$ seconds. This is because the instantaneous frequency is defined as $d/dt \theta(t)$ for $y(t)=\sin(\theta(t)*t)$ (Gibson p. 140).



7.1.8 Seidl's simulation

The numerical simulation of records of particular instruments is one of the fundamental problems in the processing of digital broadband seismograms. Here, as one of the possible approaches, the method of Seidl (1980) has been implemented. It allows the simulation of arbitrary seismograph-galvanometer systems using a time-domain recursive filter. The user must provide input for three parameters:

- Channel list to process.
- Eigenfrequencies and damping values of the actual seismometer, the simulated seismometer and potentially also a galvanometer. If -1,-1 are given for the galvanometer eigenfrequency and damping, the galvanometer response is ignored. In addition, the gain factor (last parameter in input line) can be used to scale the amplitudes of the traces while simulation filtering is performed.

For numerous examples of the performance of this technique see the original paper of Seidl (1980).

7.1.9 Difference Equation

An important way of describing linear time invariant (LTI) systems is by means of linear difference equations with constant coefficients. A general m -th order difference equation can be described by:

$$y_i = \sum_{k=1}^m a_k y_{i-k} + \sum_{k=0}^n b_k x_{i-k}$$

The coefficients a_k describe the feedback part of the system and are called the autoregressive (AR) coefficients. Coefficients b_k control the “feed-forward” part and are called moving average (MA) coefficients. In order to use the difference equation directly for filtering in PITSA, you must first create an ASCII file containing the sequence of AR parameters on the first line and the MA parameters on the second line. The coefficients on each line are separated by commas. Since the first

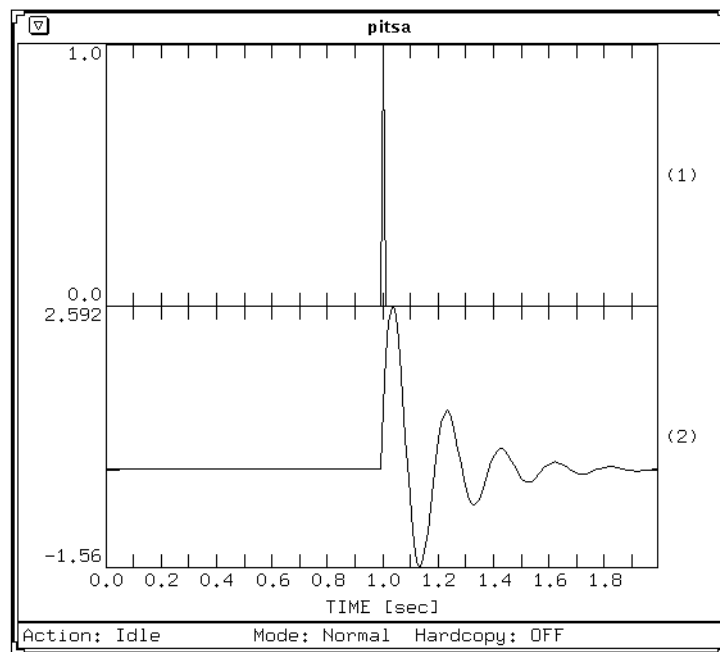
summation $\sum_{k=1}^m a_k y_{i-k}$ starts at index 1, the first parameter on line 1 of the input

file is interpreted as a_1 . In order to simulate a purely autoregressive system, you would enter a single 1.0 for the MA parameter b_0 (line 2). The parameter file for a purely autoregressive filter of second order with AR parameters $a_1 = 1.8$, $a_2 = -.9$ would look like:

```
1.8, -.9
1.0
```

In Figure 7-12, a sequence of 512 points of Gaussian white noise with an offset of 1 has been filtered using the filter parameters above, thus simulating an AR process of second order. After the channel list for processing has been selected, PITSA will prompt for the name of the parameter file. Before filtering, it will display a list of AR and MA coefficients in the upper left hand corner of the screen.

Figure 7-12. Difference equation filter. The upper trace shows the filter input signal. The bottom trace has been filtered using the difference equation with AR parameters $a_1 = 1.8$, $a_2 = -.9$, and $b_0 = 1.0$.



7.1.10 GSE Calibration File

The GSE calibration file filter is a filter which is applied in the frequency domain. The FFT spectrum of the selected trace is calculated and multiplied with the FFT spectrum calculated from the information given in the input file.

The user is prompted for the name of this file which has to be in GSE calibration section format. Next, the user is asked for the number of points for the FFT. PITSA will provide a default value which is large enough to avoid wrap-around effects. You may always increase the number, as long as it is a power of 2, but using a smaller number will lead to spurious results.

Finally, the inverse FFT is calculated and input and output trace are displayed together.

7.1.11 Evaluate GSE Response

For transfer functions of arbitrary instruments given in GSE format (see Appendix F), PITSA allows the evaluation and display of various properties of the frequency response function such as amplitude- and phase response, real and imaginary parts as well as group delay and phase delay.

The user is prompted for the name of an ASCII file containing the instrument description in GSE format. Subsequently, the user has to provide the sampling frequency in Hz and a fictitious time for the frequency response function (here called time of first sample). This time is checked against the validity time provided in the GSE header. If the fictitious time does not fall within the validity time range, a warning is issued (without further consequences).

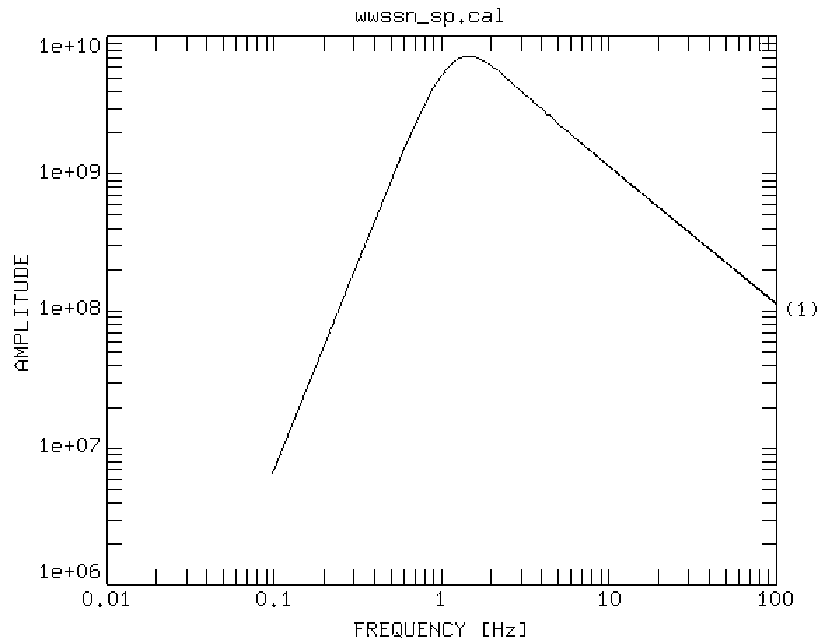
Next, the user is asked for the number of points for the FFT. PITSA will provide a default value which is large enough to avoid wrap-around effects. You may always increase the number, as long as it is a power of 2, but using a smaller number will lead to spurious results. Together with the sampling frequency used, the number of points defines the frequency range used for the evaluation of the frequency response function. The smallest frequency for which the response is evaluated is given by the sampling frequency divided by the number of points used for the FFT. This value defines also the frequency increment between neighboring frequency points. The maximum frequency for which the response function is evaluated and displayed corresponds to half of the sampling frequency.

Next, the user has to select the quantity to display. The possible options are:

- *Amplitude*. In this case the amplitude of the frequency response function is displayed.
- *Phase shift [degrees]*. Displays phase shift in degrees. Note that a simple phase unwrapping algorithm is used.
- *Real part*. Real part of the frequency response function.
- *Imaginary part*. Imaginary part of the frequency response function.
- *Group delay* in seconds
- *Phase delay* in seconds.

After the axis type has been selected from either *LIN-LIN*, *LIN-LOG*, *LOG-LIN*, and *LOG-LOG*, the selected quantity is displayed. As an example, the amplitude part of the frequency response for WWSSN short period instrument is displayed in .

Figure 7-13. Amplitude part of the frequency response function of a WWSSN short period instrument.



7.2 Spectrum

PITSA contains several tools with which to carry out investigations of the spectral content of seismic traces and move between the frequency and time domains. The fundamental tools, of course, are the Fast Fourier Transforms (FFT) and inverse FFT. The spectral values shown have been multiplied by the sampling rate in order to approximate the values of the Fourier transform within the given frequency band (Press et al., 1988). While the type of spectral information displayed may be chosen freely between amplitude, phase, real or imaginary component, internally it is always kept in its full complex form.

To process the amplitude, phase, real, or imaginary component of a spectrum separately (as if they were real valued time series), you can demultiplex the complex spectrum into a real trace containing only the component of interest. The inverse operation, namely to combine (multiplex) two real valued traces containing amplitude and phase, or real and imaginary components of a complex spectrum into its multiplexed form, can be performed as well.

For signals which have been produced by autoregressive processes, the FFT is not the best way to estimate the spectral content of the signal. In this case the maximum entropy spectrum offers a much better spectral resolution.

Table 7-3. Spectral Analysis Tools.

Menu Item	Description
FFT	Fast Fourier Transform.
Inverse FFT	Inverse Fast Fourier Transform.
Multiplex to FFT Format	Combine constituent component traces into full complex spectrum.
Demultiplex from FFT Format	Convert full complex spectrum into constituent component traces.
ME Spectrum	Maximum entropy spectrum.
POW Spectrum	Power Spectral Density
Wavelet Transform	discrete fast Wavelet Transform

7.2.1 Number of Points Used for FFT

The spectral analysis tools in this section make frequent use of the FFT and inverse FFT tools. In each case, PITSA will prompt for the number of points to use, and provide a default value which is large enough to avoid wrap-around effects. You may always increase the number, as long as it is a power of 2, but using a smaller number will lead to spurious results.

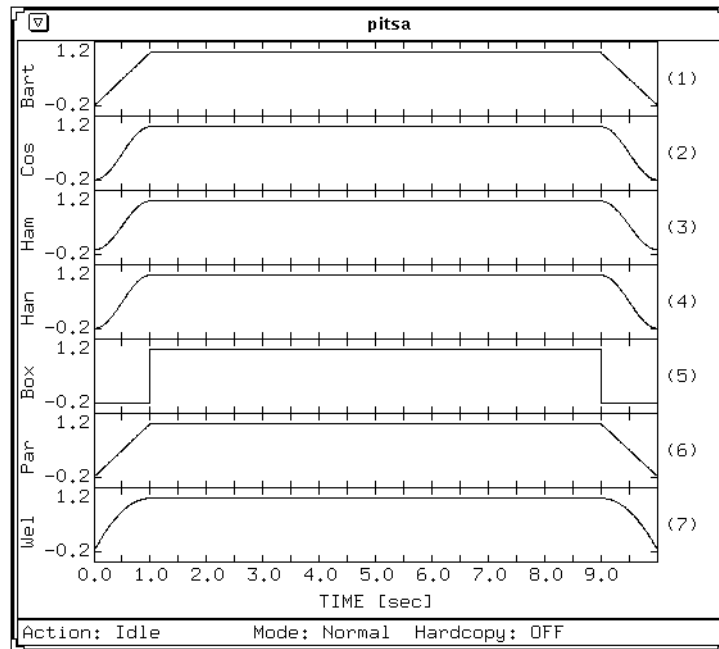
7.2.2 FFT

Before calculating a spectrum, you will have a chance to zoom in on a selected portion of the trace. As described in Section 6.1, this can be done using tapered or untapered zooming (for untapered zoom, the user enters a taper fraction of 0). If spectral leaking is a concern, choose an appropriate taper function and enter a non-zero taper fraction. The following taper functions are implemented:

- Bartlett
- Cosine
- Hanning
- Hamming
- Boxcar
- Parzen
- Welch.

Figure 7-14 shows the different taper functions. For a discussion of the issues involved in selecting a taper function see Press et al. (1988).

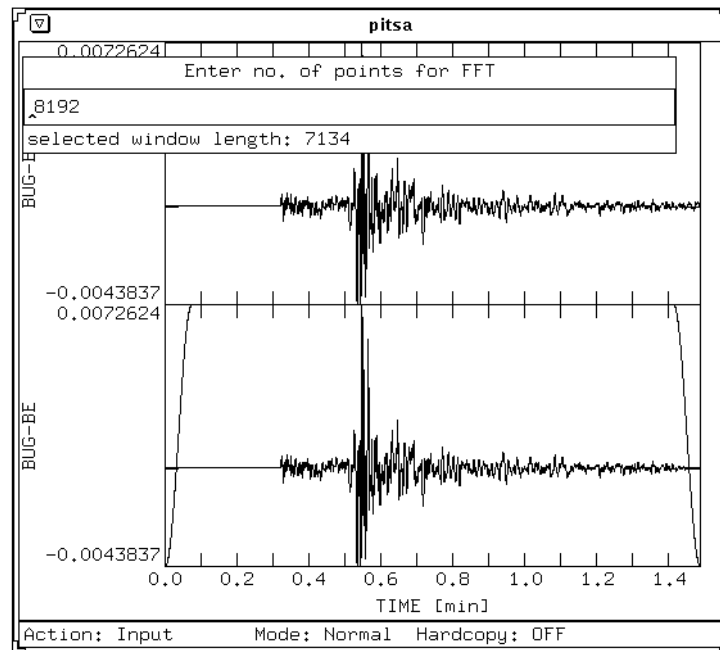
Figure 7-14. The shapes of the different taper functions.



PITSA will then prompt for the channel list, the zooming method, and finally the tapering fraction. The tapering fraction is the fraction of the trace that will be altered by the tapering function. For example, if a tapering fraction of 0.2 is entered, 10% at the beginning and 10% at the end of the input trace will be influenced by the taper function.

Next, PITSA will display the first trace both untapered and tapered (Figure 7-15) and ask the user for the number of points to use for the FFT, which must be a power of 2. PITSA will provide the user with a default value that will be a power of 2 and be greater than or equal to the number of points in the input trace.

Figure 7-15. PITSA prompting for the number of points to use for the FFT.
The actual taper function used is overlaid on the tapered trace.



PITSA will now calculate the FFT for the given data window. The spectrum is kept as a series of complex numbers in a multiplexed format. The following options are available for the display:

Table 7-4. Spectrum display types

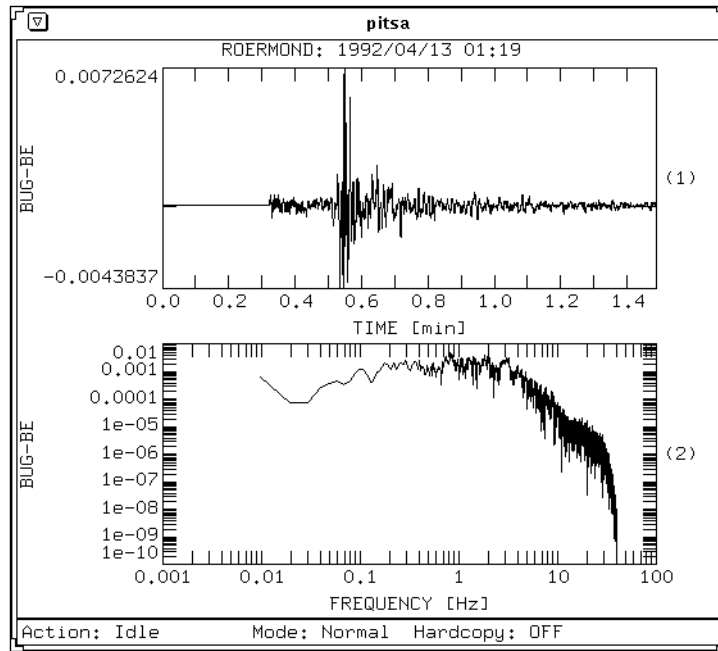
Display Option	Description
Amplitude	Absolute values of the complex spectral values multiplied by the sampling rate
Phase	Phase spectrum to be displayed in radians, using a simple phase unwrapping scheme (Stearns and David, 1988).
Real	Real value of the FFT
Imaginary	Imaginary part of the spectrum

It is also necessary to choose the axis types for the display, where the first specification refers to the x-axis and the second to the y-axis:

- LIN - LIN
- LIN - LOG
- LOG - LIN
- LOG - LOG

Note that logarithmic axes are only permitted for positive values. A popular display type for spectra of seismic traces is the *LOG - LOG* display of the amplitude spectrum. Figure 7-16 shows the display after an FFT has been calculated for a trace and added to the list of traces in memory.

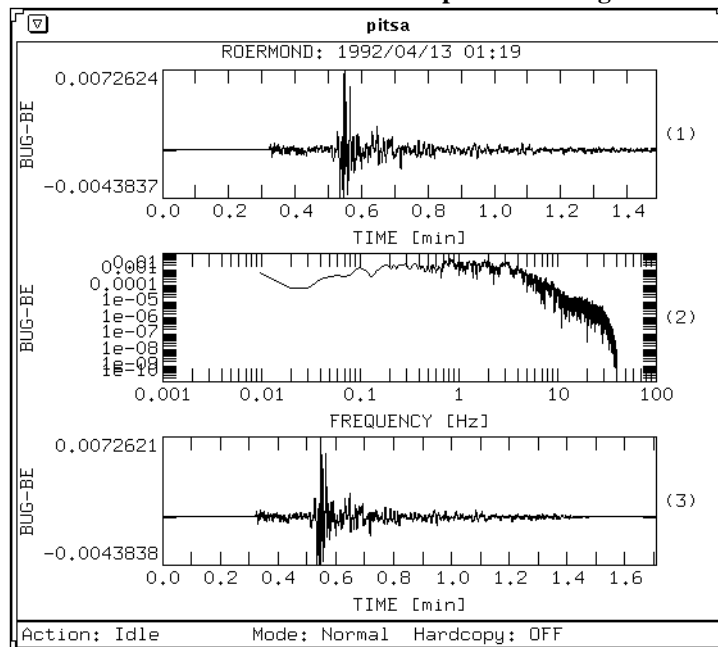
Figure 7-16. FFT Result. The upper trace is the trace that the FFT was calculated for and the lower trace is the FFT. Here the amplitude is displayed with a *LOG - LOG* scale.



7.2.3 Inverse FFT

The *Inverse FFT* tool is used to calculate the time series corresponding to a spectrum. PITSA will check to make sure that the selected traces are in the right format (i.e., multiplexed). Figure 7-17 shows the final result of taking the inverse FFT of the spectra in Figure 7-16. Notice that the length of the time series resulting from an inverse FFT is always a power of 2.

Figure 7-17. Inverse FFT calculated from the spectrum in Figure 7-16.



7.2.4 Multiplex to FFT Format

The *Multiplex to FFT format* tool is used to combine a pair of traces representing the components of a complex spectrum into the corresponding multiplexed spectrum. This tool is used in specialized circumstances where you may want to perform certain filter operations on spectral components separately, recombine them, and finally transform the resulting spectrum back into the time domain.

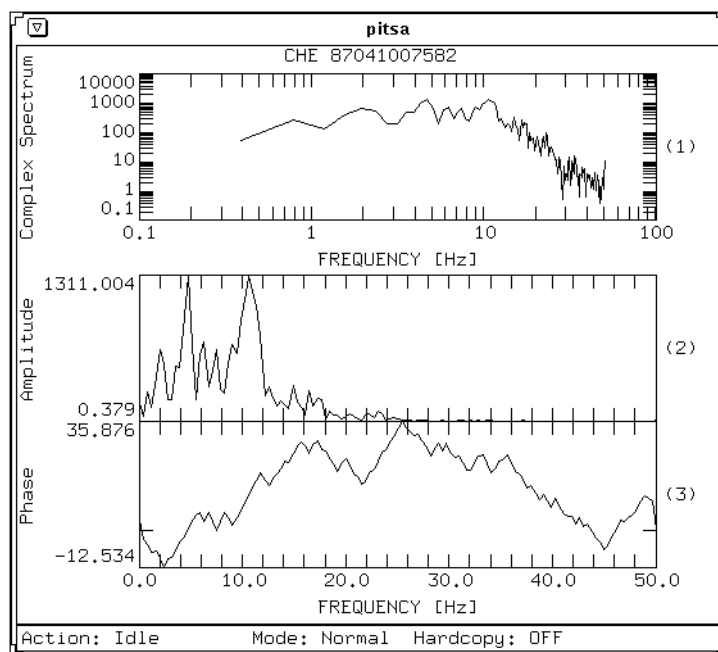
There are two options, depending on the components represented by the trace pair:

- Amplitude, Phase
- Real, Imaginary

The channel list should specify the component traces in the order given for each option. Both traces must be a length of a power of 2. The resulting multiplexed trace will have the same format as a complex spectrum and can be treated as such, e.g., for calculating the inverse transform. PITSA will prompt for the axis type to use in the display (Section 8.2.6).

Figure 7-18 displays a complex spectrum in trace 1; even though only the amplitude part can be shown, the trace represents the full multiplexed spectrum. Most of PITSA's tools cannot be used with this trace.

Figure 7-18. Display of a complex spectrum (trace 1) and its corresponding amplitude and phase components (traces 2 and 3).



In the same figure, traces 2 and 3 display the amplitude and phase components of the complex spectrum represented in trace 1. They can be operated on by most of PITSA's tools, and they can be recombined into the full multiplexed spectral representation. If they were displayed with the same axis types, traces 1 and 2 would be appear identical, even though they represent different things.

7.2.5 Demultiplex from FFT Format

Certain tools within PITSA can only be applied to real-valued traces, hence they do not apply to spectral traces in their complex format. If you wish to process the amplitude or phase (or, alternatively, the real or imaginary) components of a spectrum separately (e.g., to integrate it or smooth it, etc.), you can use the *Demultiplex from FFT Format* tool to separate the complex spectrum into traces containing only selected components, which are reflected in the options to this tool:

- Amplitude
- Phase
- Real part
- Imaginary part

The selected component will be extracted as a real-valued trace. Of course this tool only works on a trace representing a complex spectrum (e.g., trace 1 in Figure 7-18).

7.2.6 Maximum Entropy Spectrum

The *ME Spectrum* tool is used to calculate a spectrum from a time series using the maximum entropy method. Maximum entropy spectral analysis is a data-adaptive tool for the analysis of signals produced by autoregressive processes. Its strongest point is its very good spectral resolution. Its weakest point is that the order of the process must be known. It should be emphasized that the maximum entropy spectral analysis method is only applicable to feedback systems (Kanasewich, 1981).

There are two options for this tool:

- Normalized (scaled to a maximum value of 1)
- True Value (no scaling)

PITSA will prompt for the name of the AR output file in which to store the coefficients of the prediction error filter which are calculated in the course of the maximum entropy spectral estimation. File extensions will be added automatically, beginning with a01, a02, a03, and so forth.

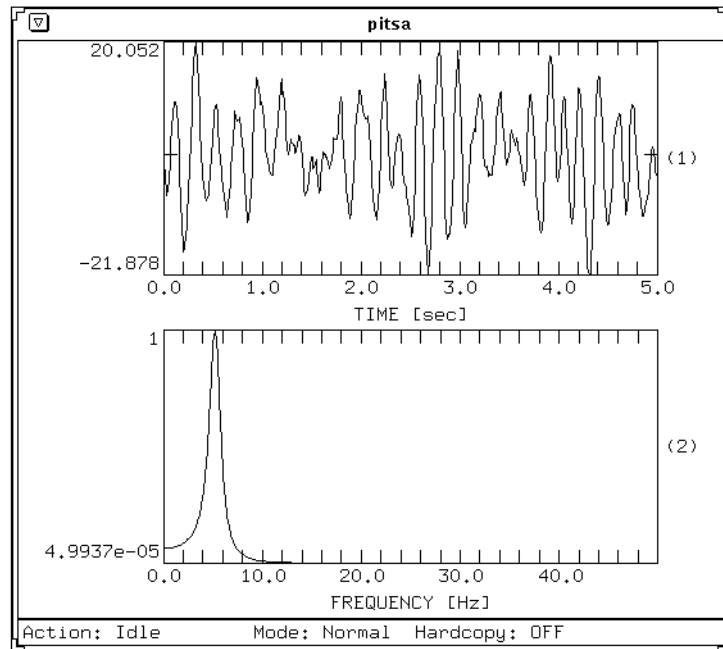
PITSA will prompt for three further input parameters:

- The order of the process.
- The maximum frequency for which a spectral value should be calculated.
- The step width Δf [Hz].

Spectral values are calculated from 0 frequency up to the maximum frequency in steps of Δf Hz. The AR parameters a_1, a_2, \dots, a_m (with m being the specified order of the process) will be written to the output file. This file can be used directly as input for filtering using the *Difference Equation* filter option (Section 7.1.9).

For details on the estimation of the AR order, the calculation of the ME spectrum and further discussion of its properties see Kanasewich (1981), and Press et al. (1988). Figure 7-19 shows an example of a ME spectrum calculated for an autoregressive process of order 2. The time series was produced using the *Difference Equation* filter option (Section 7.1.9) with AR coefficients $a_1 = 1.8$, and $a_2 = -0.9$ on Gaussian noise.

Figure 7-19. Maximum entropy spectrum.



The corresponding AR parameter file contains the following lines:

```
#1.813720      1  -AR-Parameter
#-0.916940     2  -AR-Parameter
#Pm = 0.011553 N = 512   FPE = 0.011689
1.813720 , -0.916940
#No MA parameters
1.0
```

Lines starting with # are comment lines. Pm, N, and FPE are the power of the prediction error signal, trace length in points, and final prediction error, respectively.

7.2.7 POW Spectrum

The *POW Spectrum* tool is used to calculate the Power Spectral Density (PSD) of the input time series. There are various ways to compute the PSD. We focus here solely on the algorithm called time averaged Periodogram introduced by Welch (1967). There are two options for this tool:

- Normalized
- True PSD Value.

The first simply normalize the spectrum to 1.0. The latter computes the PSD value according to:

$$P_{Total} = \sum_f P(f) = \sigma^2$$

Here $P(f)$, P_{Total} represent the PSD Spectrum and the total Power of a stationary random process with normal distribution. σ^2 is the variance of the process. After asking for the input channel list, PITSA prompts for zooming the input trace, the length of the segment in points (power of two). Please keep in mind that the frequency resolution is given by:

$$\Delta f = \frac{\Delta_s}{N}.$$

Here N represents the number of points per segment and Δ_s is the sampling rate. Next you are asked to use 50% overlapping segments or no overlapping in order to compute the PSD. Each segment is also tapered using a Welch window. Next PITSA prompts for the number of segments to use for PSD. Default is the maximum number of segments (according to the number of points per segment given before and the overall trace length). Finally you are asked after how many segments you want to plot a first approximation of the PSD. This is especially useful testing the assumption of stationary time series. Before plotting PITSA will prompt for the display type of the traces.

Figure 7-20. Power Spectral Density after 10 steps.

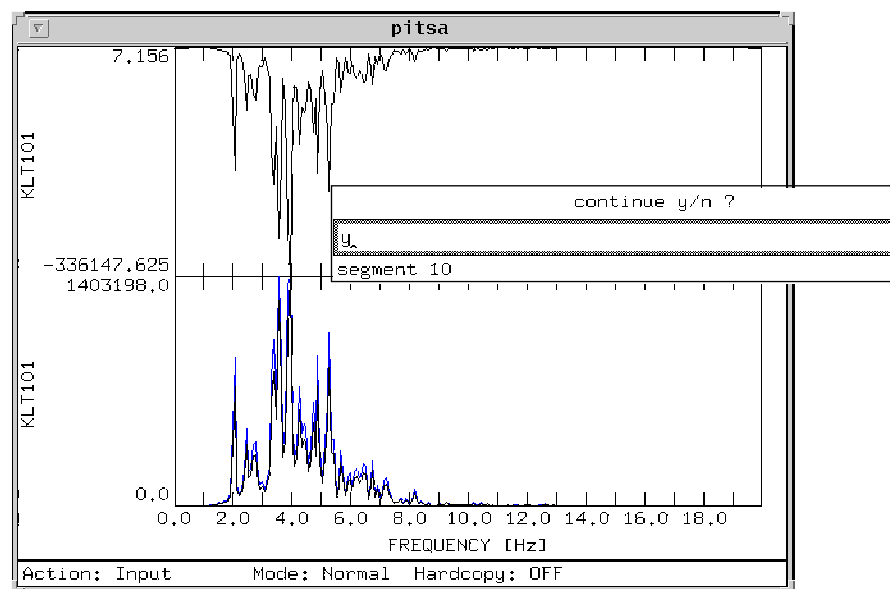


Figure 7-20 shows the PITSA screen after 10 segments FFTs are stacked. At the top panel the difference between the actual PSD and the result of eight FFT stacks is shown. On the lower panel both spectra are shown. The color line represent the old the black line the new PSD. Now, you are asked for continue or stop adding new segments.

7.2.8 Wavelet Transform

Warning: this tool is for advanced users only!

The wavelet transform tool is used to compute spectrograms or to be more strict so called “scalograms”. The final plot represents the time - scale (frequency) distribution of a single input time trace. For a more detailed description we refer to the excellent text book of Chui (1992) or for a more brief discussion to Press et al. (1992).

The input signal is decomposed using a family of different orthogonal signals (called wavelets) which are subsequently shifted over the trace. Different scales represent different contraction or dilatation of the wavelets (and therefore different center frequencies and bandwidths). Thus the final output is a projection of the signal onto the orthogonal family of wavelets. In addition, it is possible to transfer the wavelet coefficients (i.e. the scaling factors for the different wavelets) to the PITSA main screen.

There are two options for this tool:

- Wt
- Inverse Wt

The first option performs the wavelet transform, the latter enables the user to compute a recovered version of a signal using given wavelet coefficients.

7.2.8.1 Wt

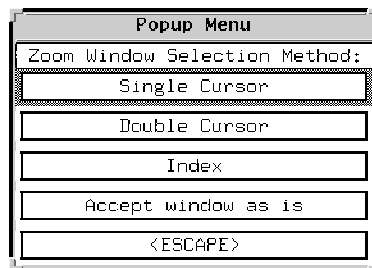
First of all PITSA prompts for either scrolling over the data or computing the Wt of the whole data trace (Figure 7-21). In the first case only the scalograms are shown and no additional output possibility is given.

Figure 7-21. What to do popup menu.



After specifying the input channel list PITSA prompts for zooming a time window out of the trace. The user can either accept the time window or zoom some specific time using the given options.

Figure 7-22. Zooming popup menu.

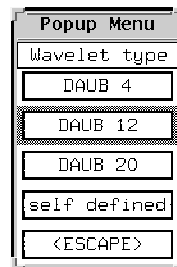


Warning: the Wt tool only works correctly if the user specifies a number of points which is a power of two! Otherwise the trace is truncated to power of two next to the input length. In order to display the resulting scalogram only a number of points less or equal to the double horizontal number of screen pixels is allowed. In this case the PITSA main frame has to be enlarged to the full screen size. A good

choice for a 20" screen is 2048 points. This is not cross checked by PITSA! It is only possible to compute Wt for a longer time series using the scrolling option in the Wt tool.

After selecting the time window, PITSA prompts for the wavelet family used for calculation (Figure 7-23). There are three standard wavelets implemented in PITSA (Daubechies 4, 12 and 20). For the exact coefficients and the overall behavior we refer to Press et al. (1992). It is also possible to import your own wavelet by selecting *self defined*. In this case PITSA prompts for the ASCII input file name. The wavelet coefficients has to be written in a separate line each.

Figure 7-23. Wavelet choice popup menu.



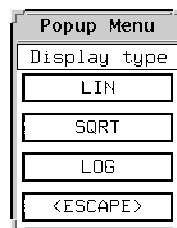
Next PITSA asks the user to specify the length of the data trace which must also be a power of 2 (Figure 7-24).

Figure 7-24. Data length input window.

Enter number of data points for WT
1024
Must be a power of two, 1024 points already selected

In order to display the absolute values of the wavelet coefficients, the user is able to chose between three different types (Figure 7-25).

Figure 7-25. Display type popup menu.



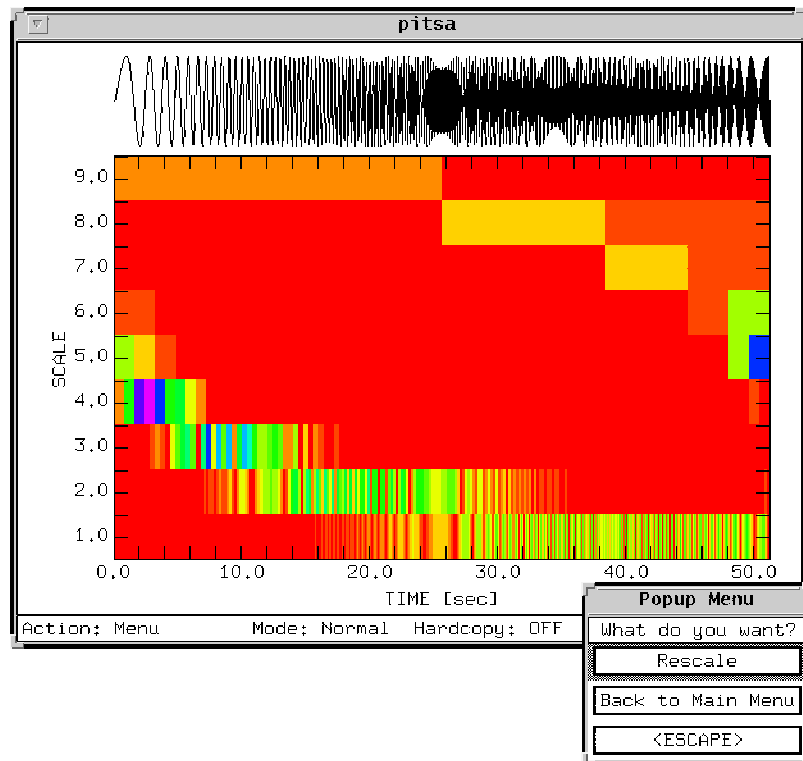
Before plotting the result of the Wt, PITSA prompts for changing the scaling for the minimum and maximum values of the wavelet coefficients (Figure 7-26). The user is also able to change this input settings while displaying the resulting map.

Figure 7-26. Changing the scaling of the wavelet map.

Enter Minimum and Maximum of the Scale:	
7.07805e-07	3.83351
Trace_min: 7.07805e-07 Trace_max: 3.83351	

If the COLOR_MODE in the pitsa.cfg file is set to COLOR_SCALE the result will be displayed in color (Figure 7-27). Otherwise it will be presented in grayscale.

Figure 7-27. Colored wavelet scalogram. Also the rescaling popup menu is visible



PITSA only supports dyadic shifts of the wavelet and dilatation is limited to the factor of two. Therefore the first scale represents roughly the frequency band of $f_{Nyquist} - f_{Nyquist}/2$, the second scale is $f_{Nyquist}/2 - f_{Nyquist}/4$ etc. On the other hand time resolution is changing from $t_s/2$ in the first scale to $t_s/(scale \cdot 2)$ in the following scales.

Selecting *Back to Main Menu* PITSA results in the popup menu for the display type of the wavelet coefficient trace. It is possible to append this trace to the existing traces in the PITSA main window. Please be aware that all coefficients are written in one trace all one type of scale. The consequence is that the last 50% of points of the written trace represent the coefficients of scale 1. The coefficients representing the scale 2 are stored in the last half of the remaining part of the trace and so on. Finally the first two points represent the remainder of the signal.

The absolute size of the coefficients of the Wt represent directly the contribution of the scaled wavelet to the signal at that specific time and scale. Therefore a noise reduction of the input trace can easily implemented by just ignoring small wavelet coefficients. This can be done by choosing a threshold in percent of the peak value (Figure 7-28).

Figure 7-28. Input panel for lower threshold for output wavelet coefficients.

Enter lower threshold fraction for WT-coefficients
-1.000000
-1 ignores

Choosing a value of 0.1 will discharge all Wt coefficients smaller than 10% of the maximum value. Finally PITSA prompts for *Accept/Append* the wavelet coefficient traces (Figure 7-29).

Selecting the *just scrolling* in the first panel of the Wt tool is acting only on the input length of each Wt slice and the appearance of a *Forward, Backward* option (Figure 7-30).

Figure 7-29. Wavelet coefficients trace written to the main frame of PITSA.

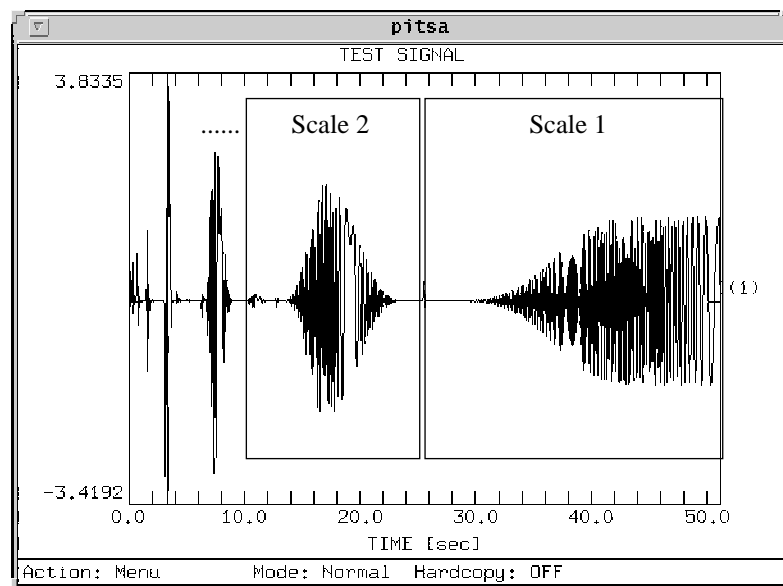
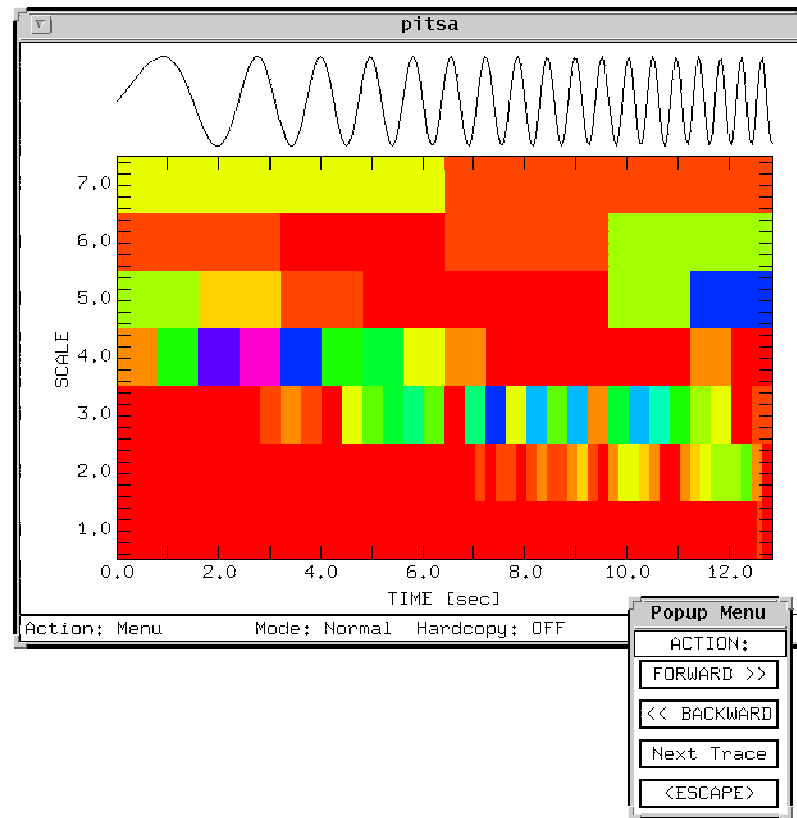


Figure 7-30. Wavelet scalogram in the scrolling mode.

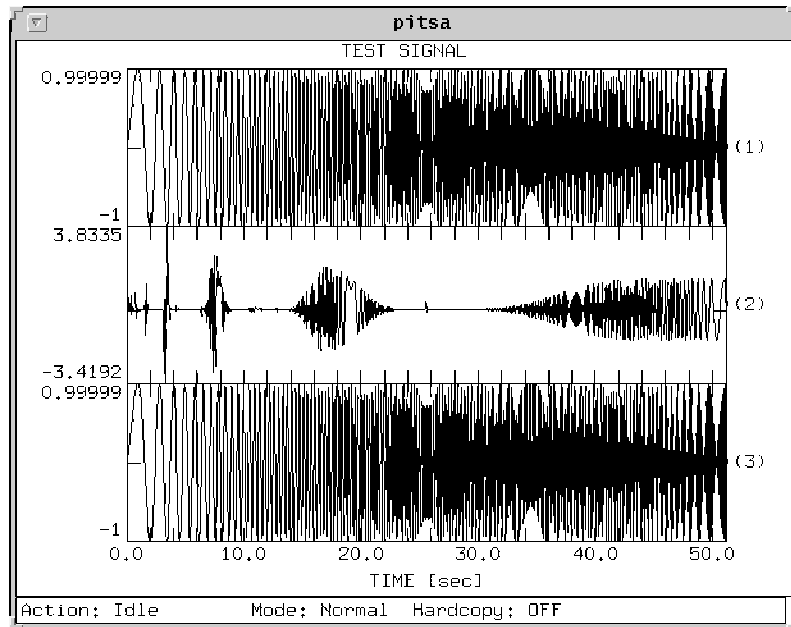


Choosing the *Forward/Backward* option enables the user to select the next or previous part of the input trace. In addition no wavelet coefficient trace will be written using this mode *Next Trace* will load the scalogram of the next trace in the input channel list.

7.2.8.2 *inverse Wt*

The *inverse Wt* option compute a time domain signal using the Wt coefficients. Only those traces of wavelet coefficients are allowed that follow the output convention of the Wt module implemented in PITSA. In order to get meaningful results the same wavelet family for forward and backward transforming should be used. The final output is the recovered time trace (Figure 7-31).

Figure 7-31. Original trace, wavelet coefficients and recovered trace using the Wt and inverse Wt tool of PITSA.



7.3 Cross Correlation

Cross-correlation in PITSA is calculated in the spectral domain by complex multiplication and subsequent inverse transform. To calculate the autocorrelation function of a single trace, the user enters the same index twice. There are two options for the *Cross Correlation* tool:

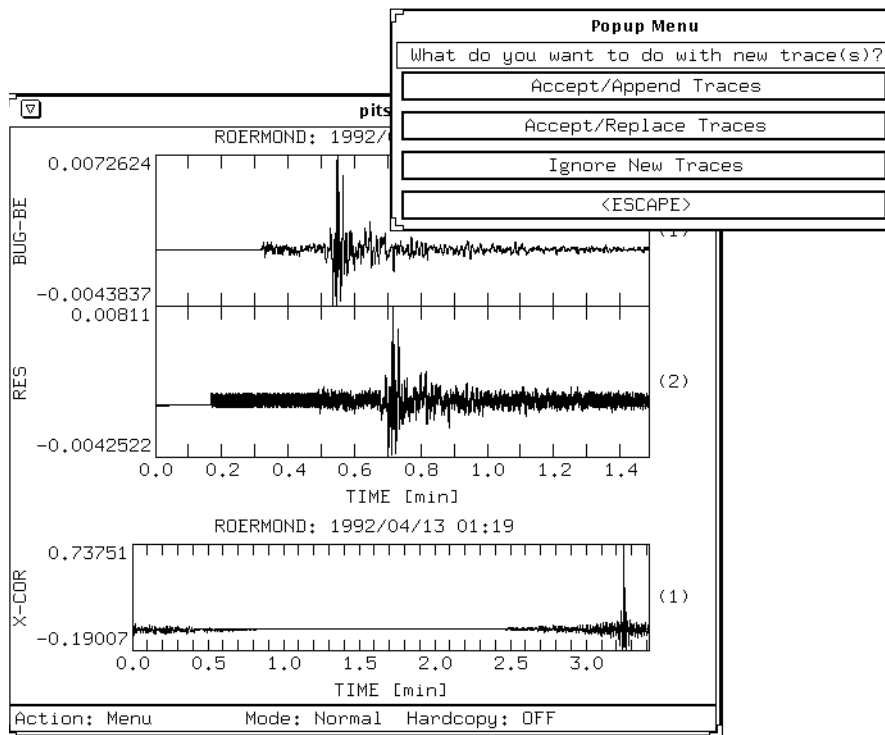
- Scaled (by individual autocorrelation functions)
- Unscaled

If the *Scaled* option is selected, the result is scaled by the individual autocorrelation functions. The autocorrelation function for a single trace will have an amplitude of 1 at zero lag.

The cross-correlation function will be displayed below the two input traces (Figure 7-32).

Figure 7-32. Cross-correlation result. Trace 2 is a copy of the first trace with a 10 second shift and noise added. The result of the cross-correlation is shown in the bottom trace (the trace indices were entered as 1, 2). The

peak on the right side of the xcorrelation trace indicates that the second trace lags the first.



Peaks in the cross-correlation trace in the left half of the trace correspond to times by which the second trace leads the first trace. Peaks in the right half of the trace, however, indicate times by which the second trace lags the first. The amount time lag is measured from the right end of the trace. In Figure 7-32, the second trace lagged the first by 10 seconds, and thus the peak in the cross-correlation trace is 10 seconds from the end of the trace. Another way to visualize this would be to cut out the right half of the trace and glue it on the left edge. Then times to the right of zero would indicate positive time shifts and times to the left would indicate negative time shifts.

7.4 (De)convolution

The *Advanced Tools* -> *(De)convolution* tool gives the user access to both convolution and deconvolution tools. Convolution can be performed in either the frequency domain or in the time domain. Deconvolution can only be performed in the frequency domain. Therefore the options available under this menu item are:

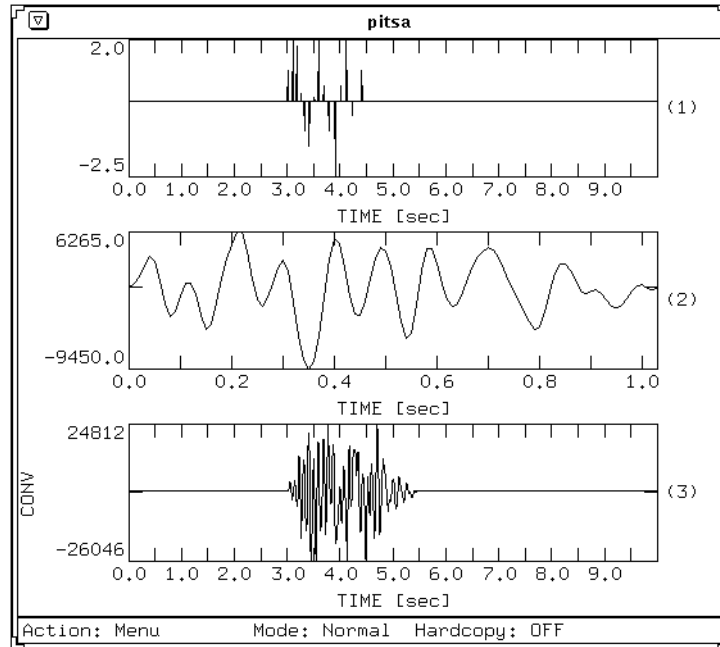
- Convolution (FREQ)
- Deconvolution (FREQ)
- Convolution (TIME)

7.4.1 Convolution (FREQ)

Select the option *Convolution (FREQ)* to perform convolution ($u(1) * u(2)$) in the frequency domain. The convolution is performed by multiplying the complex spectra of the two input traces and then performing an inverse FFT. PITSA will display

the input traces together with the trace resulting from the convolution (Figure 7-33).

Figure 7-33. Convolution in the frequency domain. Trace 2 was used for u(1) and trace 1 for u(2). The resulting convolution is shown in trace 3.



7.4.2 Convolution (TIME)

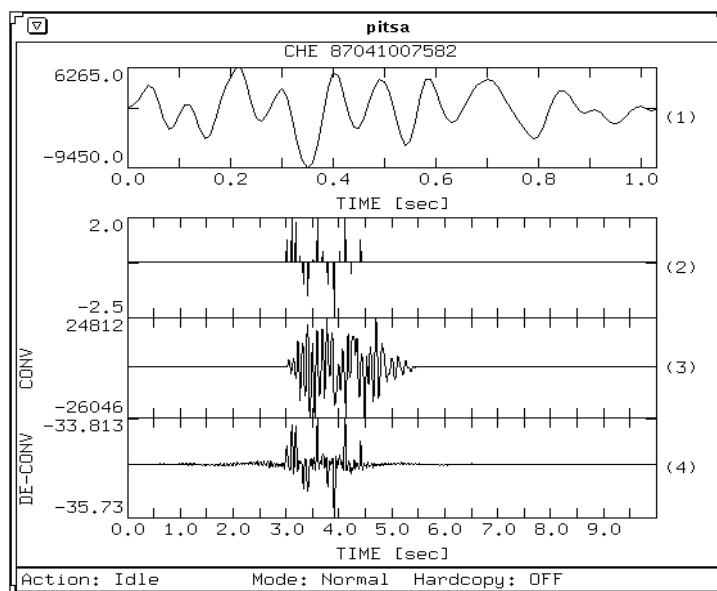
To perform convolution in the time domain, select the option *Convolution (TIME)*. PITSA will prompt for the trace indices to use for u(1) and u(2), where u(2) is the denominator trace for deconvolution. PITSA will display the two input traces and the resulting trace.

7.4.3 Deconvolution

To deconvolve in the frequency domain ($u(1)/u(2)$), select the option *Deconvolution (FREQ)*. To perform the deconvolution, PITSA will take the inverse FFT of the ratio of the complex spectra of the two input traces. In specifying the channel list, the first channel index corresponds to the numerator spectrum and the second channel index to the denominator spectrum.

Next, you will be prompted for the so-called waterlevel. In deconvolution, a small value in the denominator spectrum will cause an instability of the spectral division. All spectral values less than the waterlevel to be “filled up” to the waterlevel. In PITSA, the waterlevel is referenced to the maximum spectral value of the amplitude spectrum of the denominator spectrum in dB. For example, entering “20” for the waterlevel, will cause all spectral values in the denominator spectrum with an amplitude value below 0.1 of the maximum spectral value to be filled up to the waterlevel value. The phase values of the spectrum are unaffected by this process. A message window will display information about the number of spectral values found below the waterlevel. The “quality” of the deconvolution result may strongly depend on the selection of a proper waterlevel (see Figure 7-34)

Figure 7-34. The result of a deconvolution. The first three traces come from Figure 7-33. Traces 1 and 2 were convolved to produce trace 3 (indices were entered as 2,1). Then *Deconvolution (FREQ)* was selected and indices 3,1 were entered. The waterlevel for this example was purposely set to 20 dB (too small) to demonstrate the sensitivity to its proper selection.



7.5 Particle Motion

Particle motion analysis is an important tool for many seismological problems (e.g., phase identification, onset time determination, etc.). PITSA allows the user to interactively display particle motion diagrams (hodograms) in 2 and 3 dimensions for a moving window which can be shifted along the traces with a variable step size. Therefore, there are two options available for this tool:

- 2-D Snake
- 3-D Snake

PITSA will ask for the channels to use for the specified option (2 channels for 2D and 3 channels for 3D). PITSA will then have the user zoom in on the region of the traces that are to be plotted.

After the zoom is finished, PITSA will ask the user for the number of points to use for the particle motion 'snake'. This will be the maximum number of points that can be seen in the particle motion projection at any one time (the length of the moving window). For a sampling frequency of 100 Hz, entering "20" would result in a window of 0.2 seconds moving along the traces. The *Header Access -> Record* option can be used to find the sampling frequency (Section 8.3.3). The moving window is called a snake because as the user steps through the traces being projected, the particle motion line will move in a snake-like fashion with the tip of the snake pointing towards positive times. PITSA will also ask the user how many points to move in the traces for each step. If there are a large number of points in the zoom area, it can take a long time to step through each individual point and it is desirable to be able to move more than one point per step.

If the user selected a 3D snake, then additional input will be required. These input parameters are discussed in Section 7.5.2.

Much of the user's control for the particle motion displays is through the keyboard. Table 7-5 summarizes these controls.

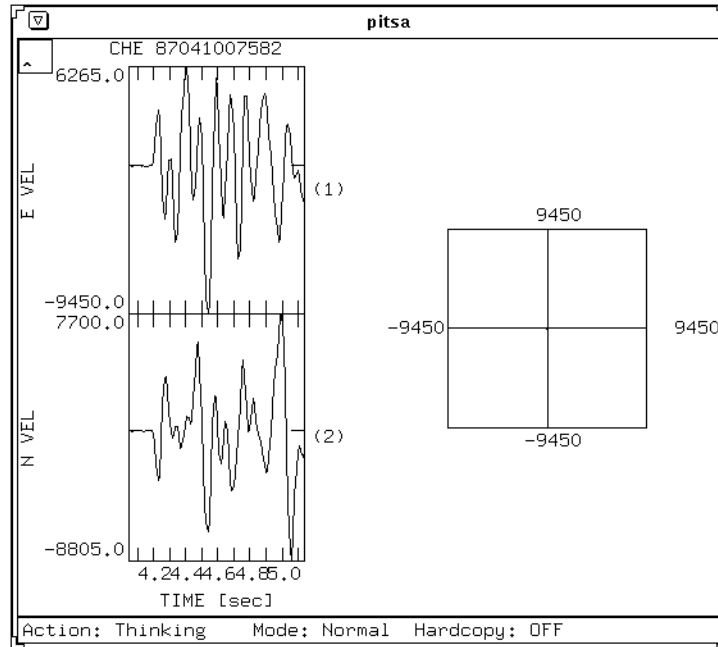
Table 7-5. Keyboard commands for particle motion analysis.

Key	2D Function	3D Function
R, ->	Moves the snake to the right, or forwards in time.	Same as 2D.
L, <-	Moves the snake to the left, or backwards in time.	Same as 2D.
CR	Moves the snake all the way to the right.	Same as 2D.
1,2,3	Lets the user pick a phase on either trace 1 or 2. The 3 key has no effect.	Same as 2D except that the 3 key picks a phase on the third trace.
A	Toggles the arrow mode from arrow only at the head; arrow at all points; and no arrows.	No function.
P, T, D	No function.	P increments the phi viewing angle and T the theta viewing angle. D changes the direction of the increment.
+, -	Increments (+) or decrements (-) the snake step size.	Same as 2D.
?	Displays information about the snake.	Displays information about the snake and the viewing angle.
S	No function.	Spins the snake about the Z (or 3) axis.

7.5.1 2D Snake

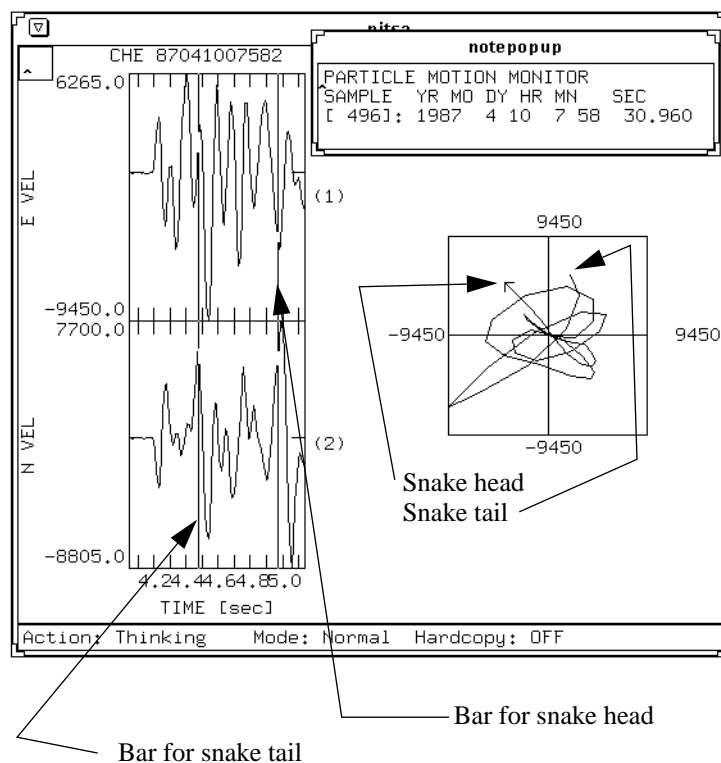
First enter the indices corresponding to the horizontal (x-axis) and vertical (y-axis) coordinate axes, respectively. Then PITSA will have the user zoom in on the section of data to project and ask for the length and step size of the snake. Figure 7-35 shows the 2D particle motion screen as it first appears. To the left, the zoomed up areas of the component traces are displayed while on the right, the 2D projection is displayed. When the screen first appears, only the first data point is displayed in the projection. To view the particle motion, the user steps through the traces by pressing either the R key or the right arrow (->) key.

Figure 7-35. The 2D particle motion screen as it first appears. To the right, the two traces used for the horizontal (x axis; top trace) and vertical (y axis; bottom trace) component are displayed. To the left is the partial motion display.



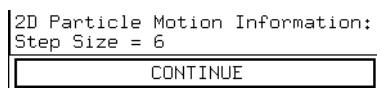
Each time the R or right arrow (\rightarrow) key is pressed, the snake will move forwards in time by a number of points determined by the snake step size. A vertical bar will appear at the beginning of the traces to indicate where the head of the snake is. As the snake moves forward in time, the head will eventually run off the end of the trace and another vertical bar will appear on the traces to indicate where the tail is. Figure 7-36 shows the particle motion screen after the snake has been moved to the right far enough so that both the head and tail vertical bars are on the screen. To move the snake backwards in time, press the L key or the left arrow (\leftarrow) key.

Figure 7-36. The 2D particle motion screen after the snake has been moved far enough to the right so that both the head and tail bars are shown on the left hand traces. In the upper righthand corner is the pop-up window which displays the time for the data point at the head of the snake.



The user can change the step size the snake takes by pressing either the + key (to increase the step size by 1) or the - key (to decrease the step size by 1). To view the step size, the user can hit the ? key. PITSA will display a message with information about the current snake parameters (Figure 7-37). A complete list of keyboard commands can be found in Table 7-5.

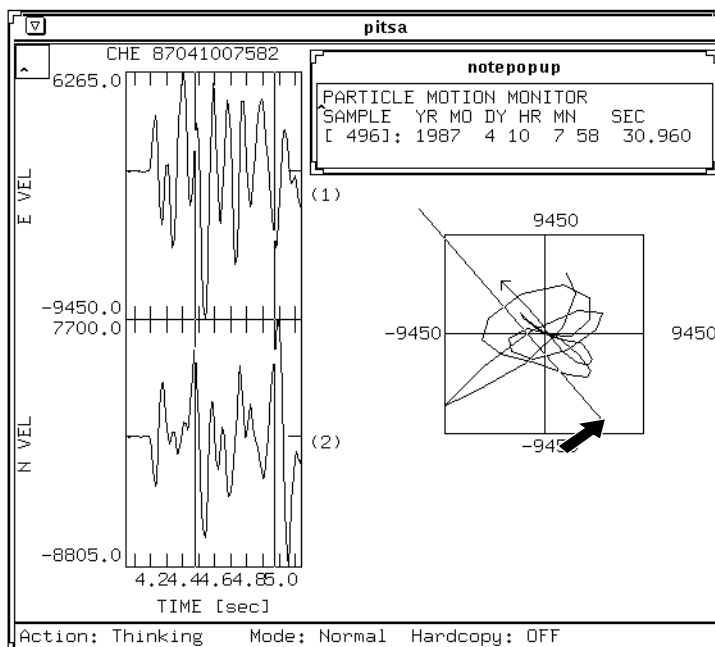
Figure 7-37. If the user hits the ? key, PITSA will display information about the particle motion snake.



7.5.1.1 Estimating polarization angles from the 2-D display

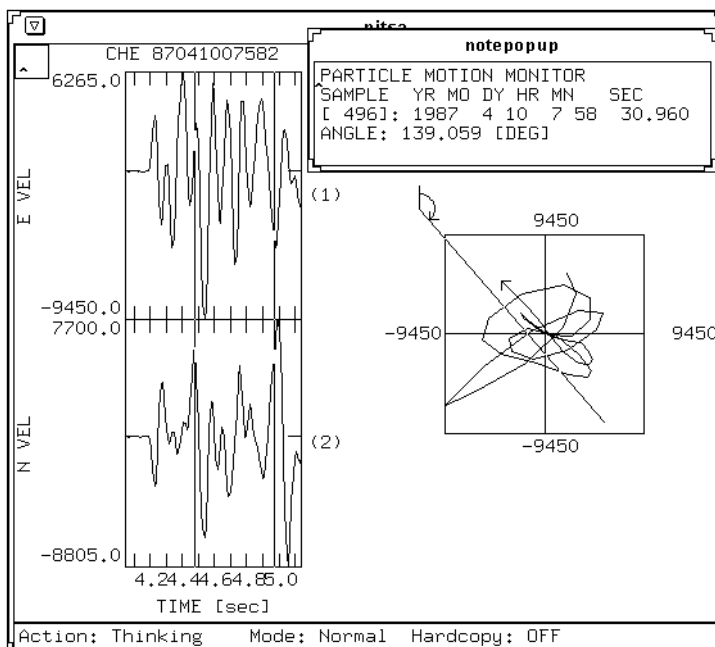
In addition to simply looking at the two dimensional projection of the time series traces, polarization angles of the particle motion can be measured. This is done by drawing a line on the plot. Move the mouse to a point where you wish to start drawing a line which will define the angle to be measured. Press the left mouse button and drag a line by moving the mouse cursor. Figure 7-38 shows the screen while a line is being drawn this way.

Figure 7-38. The particle motion screen while a line is being drawn to estimate angles.



When the mouse button is released, PITSA will draw a line between the two selected points and display the angle of this line in the pop-up window (Figure 7-39). The angle will be referenced to the start of the line drawn in map coordinates (toward the top of the screen is zero degrees). In order to avoid confusion, PITSA will draw a small vertical line and circular arrow at the start of the line (shown in Figure 7-39). When you move the snake again the angle tool will be removed.

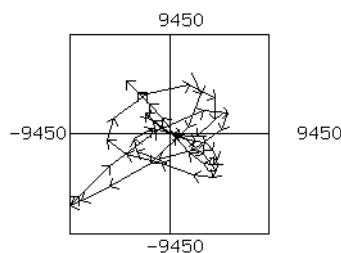
Figure 7-39. The 2D angle tool after a line has been completed.



7.5.1.2 Alternative displays of the particle motion snake

Initially, PITSA will display a single arrow at the head of the snake. The user can use the A key to toggle between this mode and two others. Pressing the A key once will cause PITSA to display an arrow at each point in the particle motion snake (Figure 7-40).

Figure 7-40. Displaying an arrow at all points in the particle motion snake.



Pressing the A key a second time will change the display so that no arrows are shown.

7.5.1.3 Phase picking from the particle motion plot

You can make a phase pick at the location of the snake head by pressing either the 1 key (make the pick on the first trace) or 2 key (make the pick on the second trace) key. PITSA will display the text input window shown in Figure 7-41. Up to 9 characters can be entered as the phase ID. The user can view and edit the picked phase by going into the *Routine Tools* -> *Phase Picking* tools (see Section 6.4).

Figure 7-41. By pressing either the 1 (for first trace) or 2 (for the second trace), the user can enter a phase pick at the location of the snakes head.

Enter phase id for trace 1
^
max 9 characters
To re-gain focus, put pointer in upper left box

Table 7-5 summarizes the key commands available.

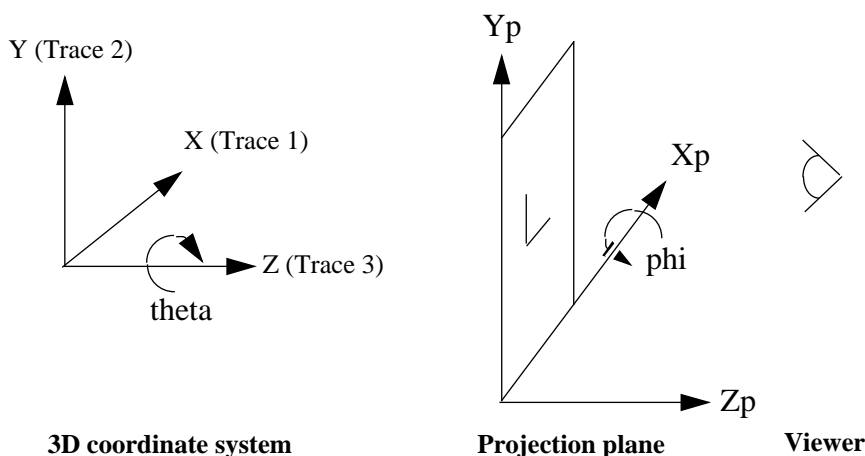
7.5.2 3D Snake

In addition to the trace indices, size of the particle motion snake and the number of points to step, the 3D option for the *Particle Motion* tool will require some additional information. The first three questions will have to do with how PITSA will project the 3D display onto the computer screen.

For the 3D display, PITSA uses a right hand 3 dimensional coordinate system as shown in Figure 7-42. PITSA projects all the 3D points onto an imaginary plane placed between a hypothetical viewer and the 3D coordinate system. Two angles determine the orientation of the 3D coordinate system with respect to the projection plane, *theta* and *phi*. *Theta* is the angle which the coordinate system is rotated about the Z axis. If *theta* is zero (assuming *phi* is zero), then the viewer's eye is

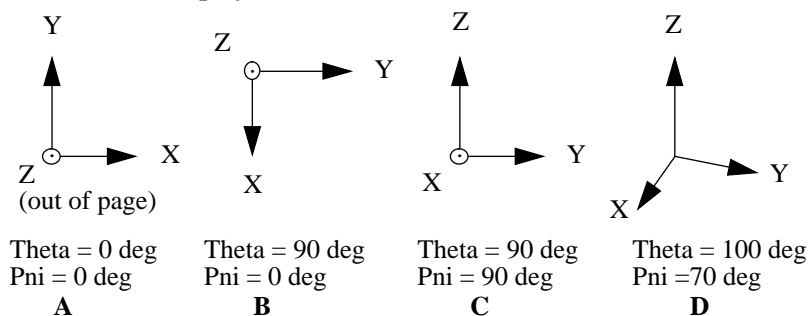
looking straight down the positive Z-axis and the Y-axis points up and the X-axis points to the right (Figure 7-42).

Figure 7-42. Projection of the 3D coordinate system onto an imaginary plane between the viewer and the coordinate system. Two angles, θ and ϕ , determine the orientation of the 3D coordinate system relative to the projection plane.



If θ is 90 degrees, then the Y-axis points to the right and the X-axis points down (Figure 7-43a). ϕ is the angle which the coordinate system is rotated about the X-axis. If ϕ is zero (assuming θ is zero), then the viewer is looking straight into the positive Z-axis and if ϕ is 90 degrees, then the viewer is looking straight into the negative Y-axis. Figure 7-42 shows the projection with both θ and ϕ set to zero. The order of these rotations is important: first the coordinate system is rotated about the X-axis ϕ degrees and then about the Z-axis of the already rotated frame by θ degrees. Thus visually, changing the angle ϕ will always rotate the coordinate system about the X-axis of the projection plane (X_p) and changing θ will rotate the coordinate system about the Z-axis of the 3D system. PITSA will supply the user with default values of -30 degrees for θ and 80 degrees for ϕ .

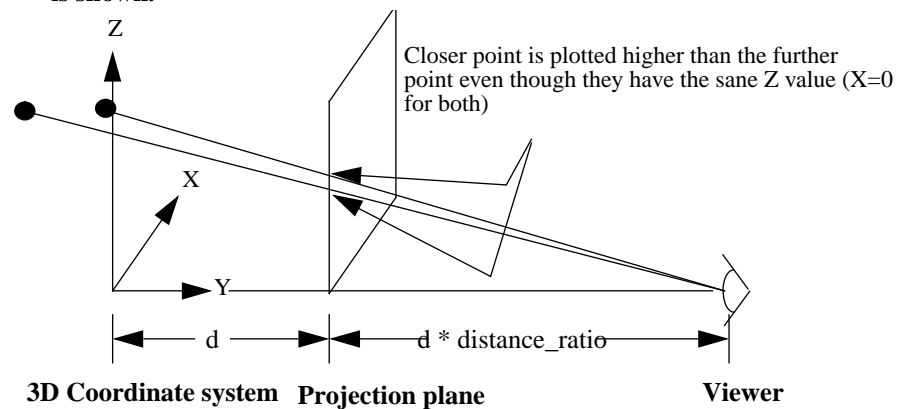
Figure 7-43. View of four different mappings from the 3D coordinate system to the projection plane. These figures show the location and direction of the 3 axis on the 2D display.



In addition to changing the orientation of the projection plane with respect to the 3D coordinate system, the user can apply foreshortening. Foreshortening makes objects that are closer to the projection plane appear larger and objects that are further away appear smaller. Typically, with particle motion plots, foreshortening tends to add more confusion than clarity.

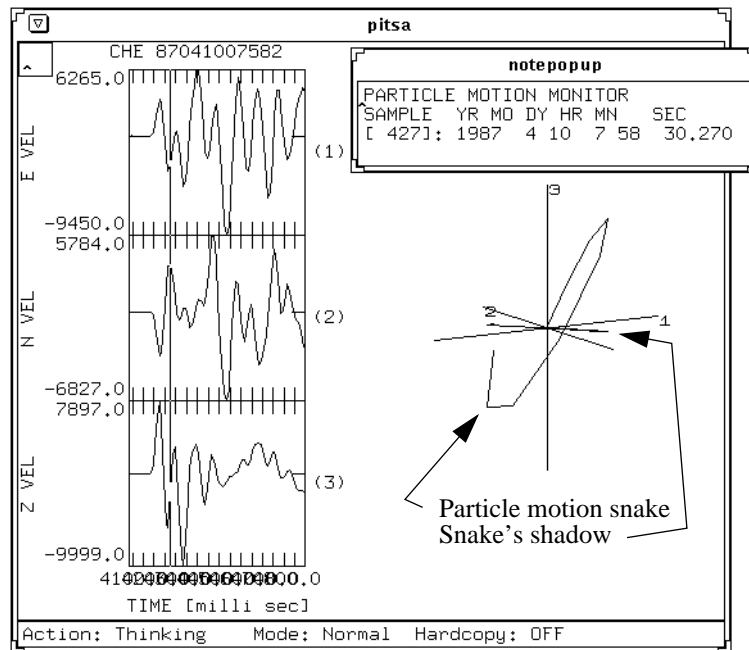
PITSA projects the 3D image onto a plane (always perpendicular to the line between the viewer's eye and the 3D coordinate system origin). The projection plane is placed at a distance d away from the 3D coordinate system, which is the maximum distance any of the particle motion points will be from the 3D origin. We then place the viewer's eye some factor of this distance away from the projection plane (called the distance ratio; see Figure 7-44). The higher this factor, the less effect foreshortening will have on the projected image. PITSA will provide the user with a default value of 50 for the distance ratio, which will result in little or no foreshortening.

Figure 7-44. An illustration of how foreshortening is performed in the 3D particle motion display. Two points are shown (both in the $X=0$ plane), one closer to the viewer than the other. The projection plane is placed a distance d from the origin of the 3D coordinate system and the viewer's eye is placed a factor of distance_ratio times d away from the projection plane. In order to map points from the 3D coordinate system, a line is drawn from the point to the viewer's eye and the intersection with the projection plane is painted. Thus the point that is closest appears to be taller than the point further away. In the figure, the first quadrant of the projection plane is shown.



Finally, PITSA will ask if you want to plot the snake shadow. It can be difficult to visualize what the particle motion actually looks like from the 2D projection. To assist in visualization, the projection of the particle motion onto the X-Y plane (or the shadow that would result from a light source at $Z = \text{infinity}$) can be plotted in the secondary drawing color (Figure 7-45).

Figure 7-45. The user can plot the projection of the particle motion snake onto the X-Y (or 1-2) plane. The shadow will be plotted in the secondary plotting color. In the above figure, it is hard to tell that the particle motion shown lies within a plane with out the plotting of the shadow.



The functioning of the 3D particle motion screen is nearly the same as with the 2D screen, with some additional functionality. Also, no angle estimation tool is available nor are arrows plotted. The user steps the snake through the source traces with the R or right arrow key (->) to move forward in time and the L or left arrow key (<-) to move backwards in time. Phase picks can be made at the location of the head of the snake with the 1 (to pick on first trace), 2 (to pick on the second trace) and 3 (to pick on the third trace) keys.

The viewing angle can be changed with the P (to change the phi angle) and T (to change the theta angle) keys. Pressing one of these keys will add a small incremental amount the appropriate angle. The user can toggle the direction of the angle increment with the D key. If the user types the ? key, PITSA will display the information shown in Figure 7-46. In addition, the user can spin the particle motion snake about the Z (or 3) axis by pressing the S key.

Figure 7-46. Information display in the 3D particle motion screen when the user hits the ? key.

```

3D Particle Motion Information:
Theta = -30.000001 deg
Phi   = 79.999998 deg
Step Size = 1
  
```

CONTINUE

Table 7-5 summarizes the key commands available.

7.6 Hilbert Transform (Envelope)

Waves propagating through an internal caustic within the Earth undergo a constant phase shift of -90° which corresponds to a Hilbert transform. PITSA provides a tool, *Advanced Tools* -> *Envelope / Hilbert Trans*, to simulate this effect numerically. The Hilbert transform can also be used to calculate the envelope of a signal. For a discussion of the theoretical background of this topic see Kanasewich (1981). There are two options for this tool:

- Keep Envelope
- Keep Hilbert Transform

For each selected trace, PITSA will calculate and display both the envelope and the Hilbert Transform (Figure 7-47); the choice of options only controls which one will be saved in the *Replace* or *Append* step. They can both be saved in a two-step process.

Figure 7-47. The result of an *Envelope / Hilbert Trans* calculation. The upper trace shows the original trace with the envelope calculation overlaid on top. The lower trace shows the Hilbert transform.

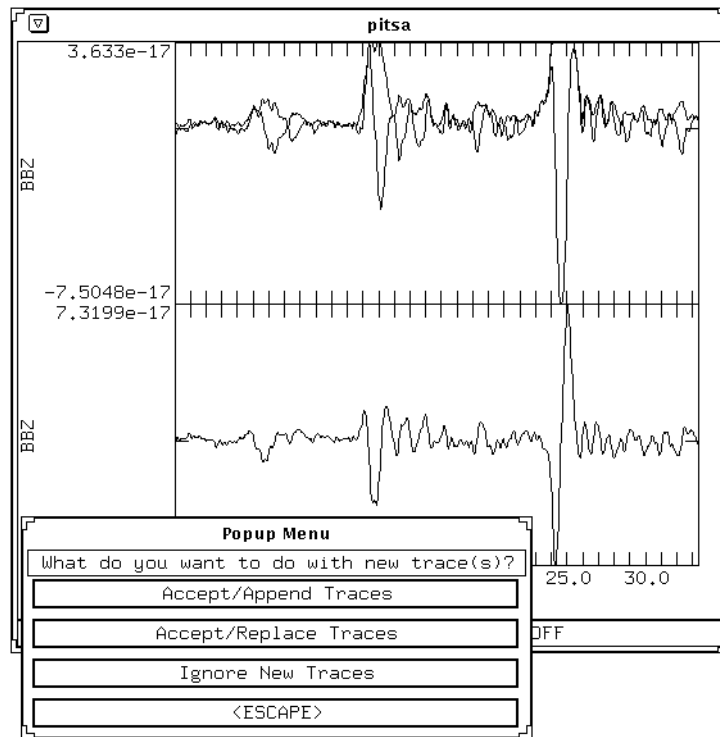
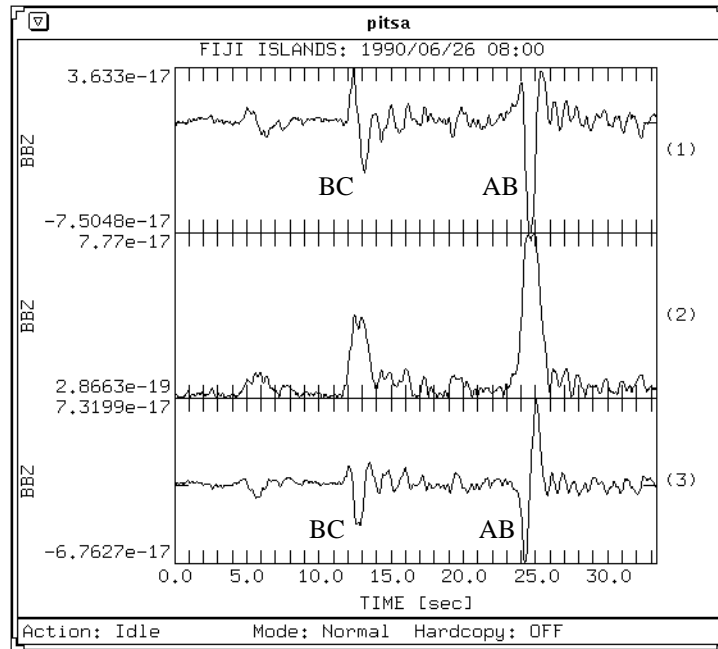


Figure 7-48 shows the display after an envelope and Hilbert calculation have been accepted and appended in separate steps. Trace 1 shows data collected at the GRF station C1 for the Fiji earthquake of June 26, 1990 (lat: 22.02 S; lon: 179.47W; Mb: 6.0; delta: 151.4 deg). The data shown correspond to the PKP wavegroup (vertical component) and show its three branches DF, BC, and AB. The AB branch has some interesting properties: It corresponds to a maximum traveltime path and since it touched an internal caustic, it has a constant phase shift of -90 degrees. This means that it was Hilbert transformed. This is why the AB wavelet of trace 1 (the original trace) and the BC wavelet of trace 3 (the Hilbert transformed trace) have a similar waveform.

Figure 7-48. PITSA after the envelope and Hilbert transform has been calculated for the first trace. Trace 2 has the envelope and trace 3 the Hilbert transform.



7.7 Rotate Components

PITSA's coordinate transformation tools, found under *Advanced Tools -> Rotate Components*, are used to rotate multi-component seismograms into ray-based coordinate systems. There are three options under this tool:

Table 7-6. Component rotation options

Menu Option	Description
Rotate N',E' -> N,E	2-D rotation of horizontal components, to correct non-standard component orientation
Rotate N,E -> R,T	2-D rotation of horizontal components to radial (R) and transverse (T) component.
Rotate Z,N,E -> L,Q,T	3-D rotation to longitudinally- (L), vertically- (Q), and transversely- (T) polarized components.

Obviously, an estimate of back-azimuth is required for any of these options, and the 3-D option also requires an estimate of incidence angle.

7.7.1 Back-azimuth

The back-azimuth is defined as the angle measured between the vector pointing from the station to the source and the vector pointing from the station to the north. If the data header contains an entry for the back-azimuth angle, it will be displayed as default (Section 8.3.2). The back-azimuth can also be determined from the particle motion analysis of the horizontal components of the P wave portion of a seismogram (Figure 7-49).

7.7.2 Incidence Angle

The incidence angle is defined as the angle measured between the ray vector at the station (from the source to the station) and the vector pointing from the station straight up. If the data headers contain a value for the incidence angle, it will be displayed as the default values. See the example in Section 7.7.4.1 for the steps to estimate the incidence angle from 3-component data.

7.7.3 Rotate N',E' -> N,E

For cases in which the horizontal component seismometers have been installed in non-standard orientations, the *Rotate N',E' -> N,E* option is used to rotate the recorded horizontal components to the standard north (N) and east (E) coordinate system.

7.7.4 Rotate N,E -> R,T

The *Rotate N,E -> R,T* option is used to rotate the standard horizontal components to the radial (R) and transverse (T) coordinate system.

This option may be used to estimate the apparent incidence angle (see example 7.7.4.1 below), which can be measured directly from a particle motion plot of the radial versus vertical particle motion components.

7.7.4.1 Determination of apparent incidence angle.

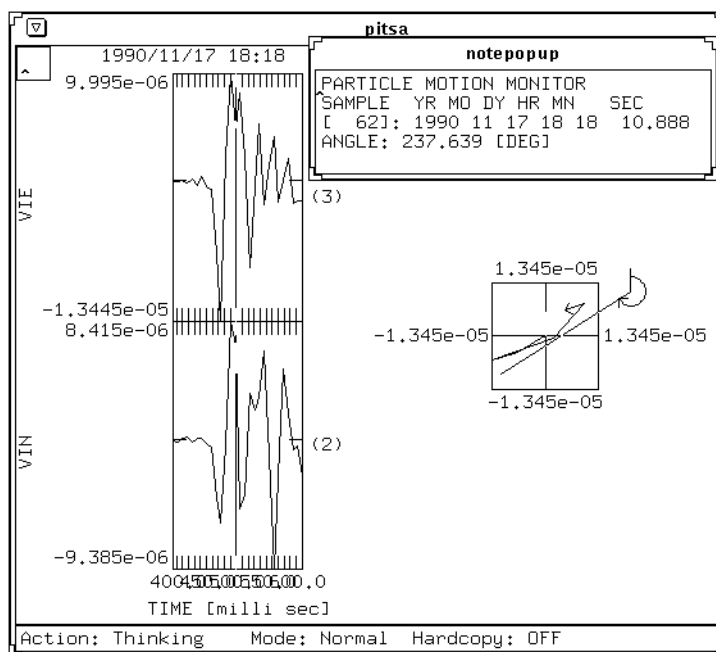
This example uses the *2-D Snake* tool (Section 7.5.1) and the *Component Rotation* tool to determine the apparent angle of incidence.

- Load a 3-component set of traces, assumed to be in the order Z, N, E.

The back-azimuth angle for rotation to the (R, T) coordinate system is determined from the particle motion plot of the horizontal components of the P wave portion of the seismogram.

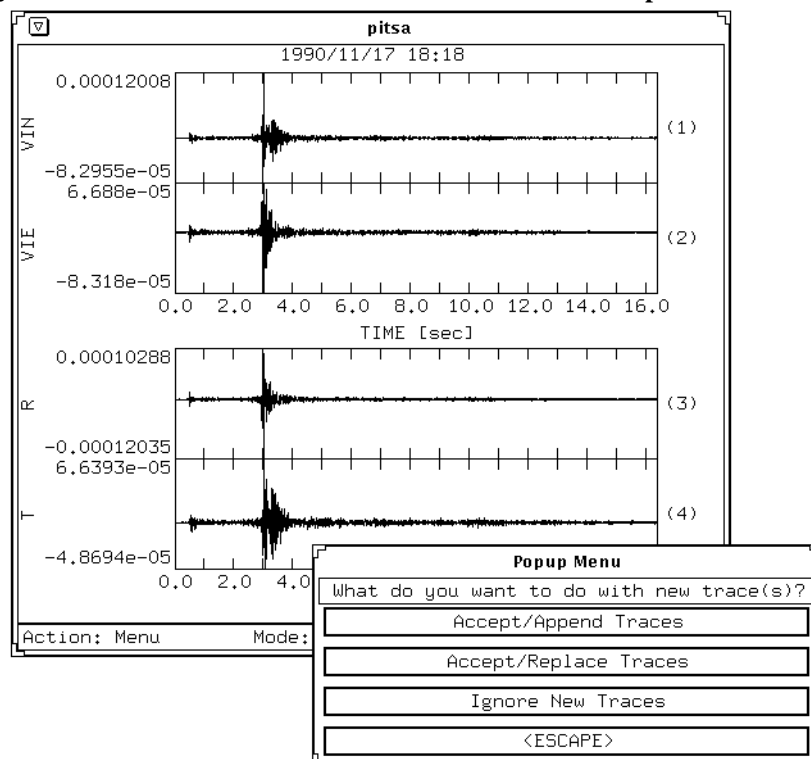
- Select *Advanced Tools -> Particle Motion -> 2D Snake*.
- Enter 3, 2 for the X and Y components.
- Use the Double Cursor method to zoom in on the part of traces containing the P wave.
- Accept the default for the number of points to use for the particle motion snake.
- Enter 1 for the snake step size.
- Step forward through the trace with the R key and use the angle tool to estimate the back-azimuth (see Figure 7-49).

Figure 7-49. Determination of the back-azimuth using the 2D particle motion.



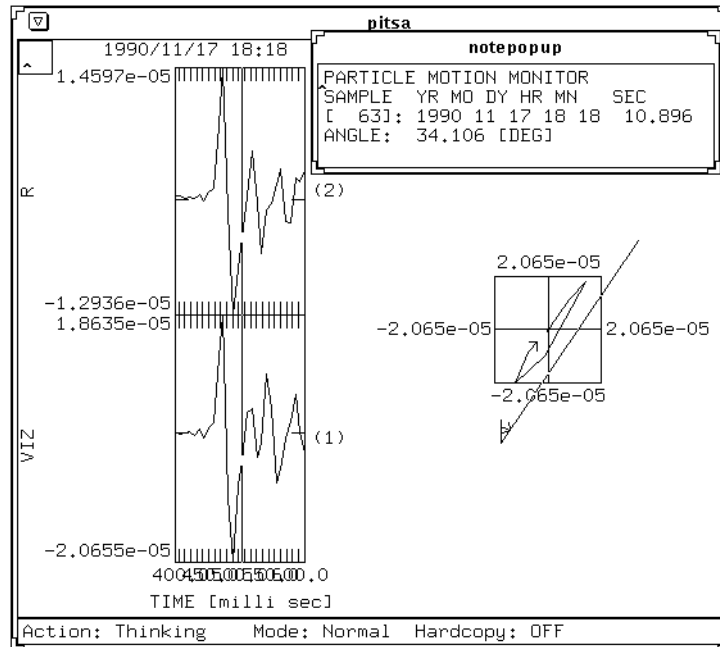
- Select *Advanced Tools -> Rotate Components -> Rotate N,E -> R,T*.
- Enter 2, 3 for the indices of the N and E traces.
- Enter the back-azimuth value you estimated.

Figure 7-50. The N and E traces rotated to the R and T components.



- Estimate the incidence angle by viewing the R and Z component in the 2D particle motion plot (Figure 7-51).

Figure 7-51. Incidence angle determination using the particle motion plot for the radial against the vertical components. The apparent incidence angle for this example was approximately 34°. Notice, that the apparent incidence angle is measured counterclockwise from the vertical (up) -> positive radial.

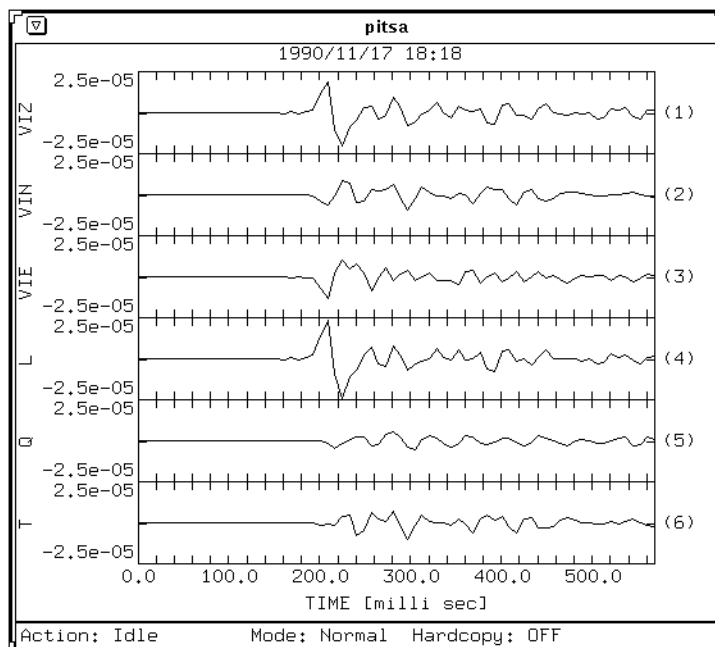


7.7.5 Rotate Z,N,E -> L,Q,T

The *Rotate Z,N,E -> L,Q,T* option is used to rotate components in 3-D, from the standard coordinate system into the longitudinal (L: along the path of propagation), vertically-polarized (Q: as in the SV phase), and transversely-polarized (T: as in the SH phase) coordinate system.

You will be asked to enter the back-azimuth and the incidence angle in degrees. For the determination of back azimuth and apparent incidence angle see Section 7.5 and the example in Section 7.7.4.1. An example is shown in Figure 7-52.

Figure 7-52. Original and rotated traces in (L,Q,T) coordinate system. Notice, that the viewports have been additionally modified to a common viewport.



7.8 Polarization Filter

In addition to the analysis of particle motion plots, with its *Polarization Filter* tool, PITSA offers a quantitative method to determine the polarization properties of seismic signals. Both a measure of the amount of linearity and the particle motion direction are calculated using the time-domain filter described by Montalbetti and Kanasewich (1970). A slightly different filter is based on the complex analytical signal presented by Vidale (1986).

7.8.1 Montalbetti and Kanasewich Filter

The method is based on the diagonalization of the covariance matrix for the set of components being used. The amount of linearity—which is called rectilinearity—is calculated from the ratio of the largest to the smallest eigenvalue. The particle motion direction is given by the direction of the eigenvector corresponding to the largest eigenvalue. Since the eigenvalues and eigenvectors must be determined for each time step, the method is fairly time-consuming. In order to speed up the calculation, the analysis window is shifted along the traces with a user-selectable step width. Finally, a set of time-varying operators is obtained which is used as gain control on the input time series to suppress unpolarized signal energy. Because the gain control operators are interpolated between samples for which the covariance matrix has actually been calculated, care must be taken with the choice of the step width of the sliding window. For the details on the theory of the polarization filter see Montalbetti and Kanasewich (1970) and Kanasewich (1980). For the following, the nomenclature of Kanasewich (1980) has been adopted.

Table 7-7. Options under the *Polarization Filter* tool:

Menu Option	Description
2 Components	Operates on the (N,E) components.
3 Components	Operates on the (Z,N,E) components.
Complex 3 Components	Operates on the analytical (Z,N,E) components

After entering the appropriate channels for the selected option, you will be asked for the window length in points and the number of points to shift per filter step to be used for the covariance matrix calculation. For a sampling frequency of 100 Hz, entering “20” as window length and “5” as step width would result in a window length of 0.2 seconds being moved along the traces with a step width of 0.05 seconds. If you enter “-1” for the window length, only the covariance matrix, the eigenvalues and the eigenvectors will be calculated, without performing any filtering. This provides a simple way to determine these parameters for a given set of components.

For each step, the covariance matrix is analyzed for the time window centered around the current sample. The window margin can be tapered (with the cosine taper function) with a user-selected tapering fraction. Entering “0.2” for the taper fraction will cause 10% on either margin of the window to be affected by the tapering.

After the eigenvalues λ_i and the eigenvectors e_i have been determined (i running from 1 to the number of components being used), the rectilinearity is calculated according to:

$$RL(t) = \left[1 - \left(\frac{\lambda_2}{\lambda_1} \right)^n \right]^j$$

Here λ_1 and λ_2 are the largest and the next-largest eigenvalues for the covariance matrix centered around time t . Furthermore, direction functions $D_i(t)$ with

$$D_i(t) = (e_i)^k$$

are calculated. You will enter values for the eigenvalue exponent n , the rectilinearity exponent j , and the eigenvector exponent k , respectively. The rectilinearity and the direction functions are smoothed in time using a moving average filter for which you must enter the half-width in number of points, resulting in new functions $\overline{RL}(t)$ and $\overline{D_i}(t)$, respectively. Finally, the filtered traces $s_i^{filt}(t)$ are obtained from the input signals $s_i(t)$ by

$$s_i^{filt}(t) = s_i(t) \overline{RL}(t) \overline{D_i}(t)$$

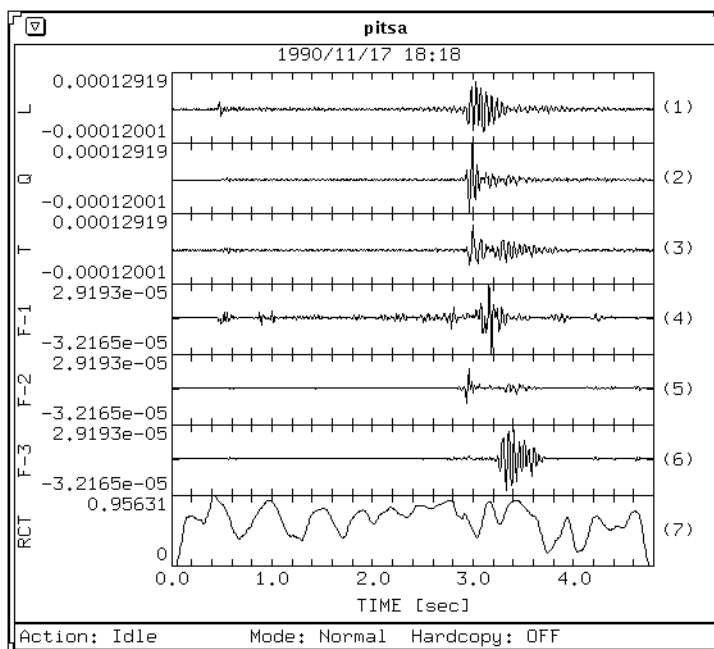
After all the calculations have been done, you will select a display option (Table 7-8).

Table 7-8. Polarization Filter display options.

Display Option	Description
Eigenvector + Rectilinearity	Eigenvector components corresponding to the largest eigenvalue, together with the rectilinearity.
Eigenvector + Largest Eigenvector Proj	Eigenvector components corresponding to the largest eigenvalue, together with the projection of the particle motion vector on the direction of the dominant eigenvector.
Filtered Traces	Filtered traces $s_i^{filt}(t)$.

In Figure 7-53, an example is given for the application of the polarization filter on the traces from Figure 7-52.

Figure 7-53. Polarization filter. The three top traces correspond to the rotated traces (L, Q, T) from Figure 7-52 (more points are shown here). The four bottom traces show the filtered traces and the rectilinearity, respectively. Notice the clear separation of linear polarized energy on the individual components.



7.8.2 Vidale Filter

The filter introduced by Vidale (1986) is based on the same assumptions and techniques just described. The only difference is due to the complex valued signals used by the Vidale filter. The complex analytical signal is defined as:

$$s^{analy}(t) = s(t) + iH[s(t)]$$

Here $H[s(t)]$ is the Hilbert transformation of the input trace $s(t)$ and i is $\sqrt{-1}$. After converting each component to its analytical counterpart, we now compute the complex covariance matrix. After solving the calculation of the eigenvalue problem, the eigenvector according to the largest eigenvalue points in the direction of

the largest amount of polarization of the signal. Maximizing the following quantity by rotating the eigenvector $[x_0, y_0, z_0]$ according to the largest eigenvalue in the complex plane (searching over $\alpha = 0^\circ - 180^\circ$):

$$X = \sqrt{(Re[x_0 e^{i\alpha}])^2 + (Re[y_0 e^{i\alpha}])^2 + (Re[z_0 e^{i\alpha}])^2}$$

and computing

$$P_E = \frac{\sqrt{1 - (Max[X])^2}}{Max[X]},$$
 we reveal also a measure of the particle motions ellipticity. P_E is 1 for circular and 0 for linear polarized motion, respectively. This

quantity is especially useful if the signal consists of Rayleigh waves. Here $Re[\]$ represents the real part of a quantity. The azimuth of the direction of maximum polarization is simply:

$$\phi = \text{atan}\left(\frac{Re[y_0]}{Re[x_0]}\right).$$

The dip of the largest amount of polarization is given by:

$$\delta = \text{atan}\left(\frac{Re[z_0]}{\sqrt{(Re[x_0])^2 + (Re[y_0])^2}}\right)$$

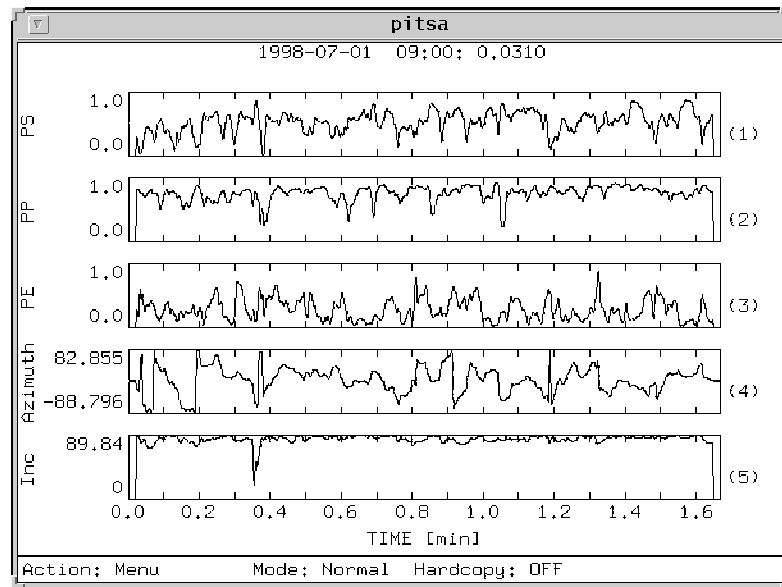
In order to rate the amount of linearity of the polarization, two additional quantities are introduced by Vidale (1986):

$$P_S = 1 - \frac{(\lambda_1 + \lambda_2)}{\lambda_0} \quad \text{and} \quad P_P = 1 - \frac{\lambda_2}{\lambda_1}.$$

If P_S is 1 if the signal is completely polarized in only one component. The latter quantity describes the planar polarization of the signal, which is 1 if the intermediate component is much bigger than the smallest component of polarization.

Because this computation is rather time consuming you are asked to enter for the window length in points and the number of points to shift the window per filter step. After calculations are done, you are asked for the number of points needed for smoothing the resulting traces. Finally all traces of quantities introduced above are displayed which can be accept and appended to the existing traces (Figure 7-54).

Figure 7-54. Complex Polarization filter. The first three traces show the linearity values P_S , P_P and the measure for ellipticity P_E , respectively. The last two traces display azimuth and dip of the eigenvector regarding to the largest eigenvalue of polarization.



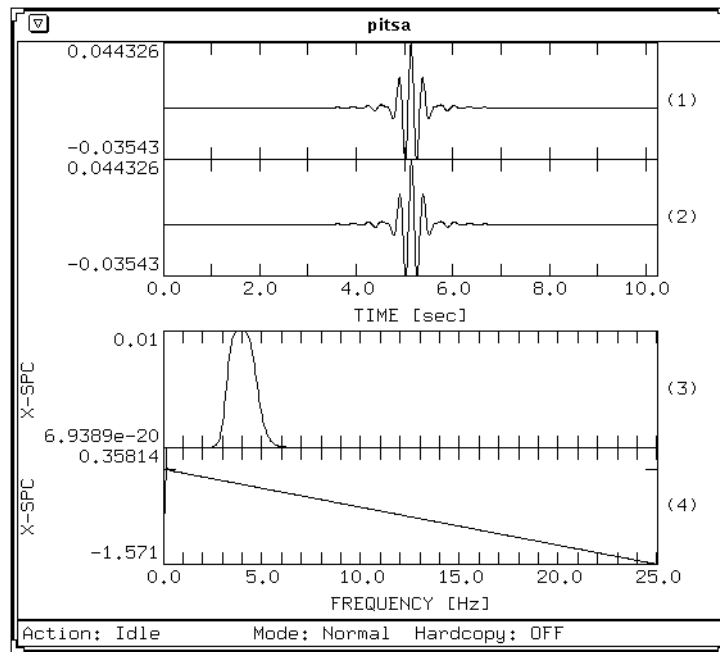
7.9 Cross Spectrum

The cross-spectrum is a measure of the joint spectral contents of two traces. If two traces are related by a simple linear time shift, the phase of the cross-spectrum is linear with slope proportional to the relative time difference. In the context of seismological signal analysis, this property has made the cross-spectrum an important tool to determine relative time shifts with accuracies better than the sampling interval (Poupinet et al., 1984; Ito, 1985). For a discussion of the influence of noise on the timing accuracy see Scherbaum and Wendler (1986).

After specifying two traces to use for the cross-spectrum calculation, you will be asked to enter the number of points for the FFT. A number equal to or larger than the default value (to avoid wrap around problems) should be entered. Then you will choose the type of spectral display and the axis type.

An example for the cross spectrum of two synthetic traces which are identical except for a one sample time shift is shown in Figure 7-55.

Figure 7-55. Cross spectrum. The second trace is a time delayed version (one sample) of the topmost trace. Below, the amplitude and phase of the corresponding cross spectrum are displayed. Notice the linear phase which indicates the pure time shift between the two uppermost traces.

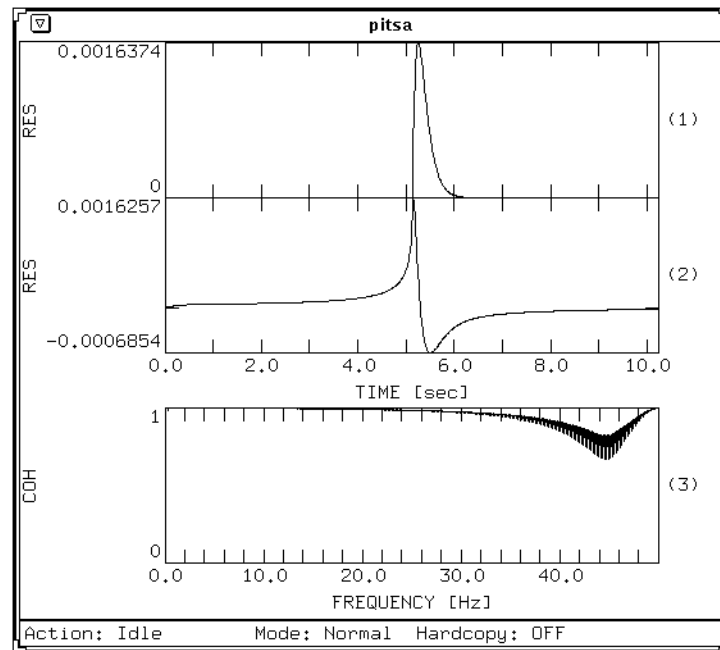


7.10 Coherence Spectrum

The coherence spectrum is a well-established measure of the association of two different traces which may be restricted to particular frequency bands. It is defined as the ratio of the smoothed cross-spectrum of two traces and the product of their smoothed amplitude spectra (Kanasewich, 1980).

After specifying the two traces to use for calculation of the coherence spectrum, you will be asked to enter the half width of the smoothing window for spectral smoothing, which must be an integer number greater than 0. Next the coherence spectrum will be displayed below the two input traces (Figure 7-56). The values of the coherence spectrum range between 0 (no coherence) and 1 (total coherence). If no smoothing is done, the coherence spectrum will be 1 regardless of the similarity of the traces (Kanasewich, 1980).

Figure 7-56. Coherence spectrum of a Brune source pulse (trace 1) and its Hilbert transform (trace 2). Notice that the coherence spectrum has a value of 1 within most of the frequency band although the waveforms are clearly different.



7.11 Response Spectrum

The response spectrum is a commonly used spectral measure in earthquake engineering which describes the maximum response of a single-degree-of-freedom oscillator to a given input motion (here the input trace). It is a function of the natural frequency of the oscillator and its damping ratio. After selecting the trace to be analyzed, the user has to provide the fraction of damping (in percent). A commonly used value is 5 %. The

- *Fraction of damping* of the oscillator. A commonly used value is 5 [%].
- *Display type*. Select either frequency or period. This controls if the values of the response spectrum are calculated and displayed with increasing frequency or period.
- Range and increment for period or frequency, respectively. Provide start value, end value, and increment value separated by commas.
- Select one of the three possible types of response spectrum to be displayed: displacement, velocity or acceleration.

Subsequently, the resulting spectrum will be calculated using the technique of Ehrenberg and Hernandez (1981).

7.12 Spectral Ratio

For two given spectra, option *Spectral Ratio* PITSA calculates the ratio of two selected input traces which must both correspond to complex spectra. In specifying the channel list, the first channel index corresponds to the numerator spectrum and

the second channel index to the denominator spectrum. The resulting spectrum is again a complex spectrum which can be treated accordingly (demultiplexed etc.).

After channel selection the user will be prompted for the so-called waterlevel. During spectral division, a small value in the denominator spectrum will cause an instability. Therefore, all spectral values less than the waterlevel are “filled up” to the waterlevel. In PITSA, the waterlevel is referenced to the maximum spectral value of the amplitude spectrum of the denominator spectrum in dB. For example, entering “20” for the waterlevel, will cause all spectral values in the denominator spectrum with an amplitude value below 0.1 of the maximum spectral value to be filled up to the waterlevel value. The phase values of the spectrum are unaffected by this process. A message window will display information about the number of spectral values found below the waterlevel. The “quality” of the result may strongly depend on the selection of a proper waterlevel.

This chapter describes how to use the various tools under the *Utilities* menu of PITSA. Table 8-1 outlines the classes of tools available.

Table 8-1. Classes of tools available under the *Utilities* menu.

Menu Selection	Description
Test Signals	Create a variety of test signals.
Trace Utilities	Combine and manipulate traces in many ways, change axis type.
Header Access	View and modify header information for station event, record, and output.
Phase Utilities	Use of slowness information, expected phase arrival times.
Resampling	Simulate the discretization of continuous data, gain-ranging A/D converter.

8.1 Test Signals

A variety of standard test signals can be created by selecting one of the options under the *Utilities* -> *Test Signals* menu:

Table 8-2. Test Signals.

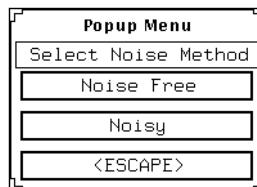
Test Signal	Description
Sine/Cosine	Sinusoids.
Spike(s)	Multiple spikes, positive and negative amplitudes.
Constant	Boring!
Brune source	A simple earthquake source model.
Mixed Signal	Superposition of sinusoidal, exponential, and linear functions.
Ricker Wavelet	Zero-phase wavelet.
Seismometer	Seismometer response.
Chirp Signal	Sinusoid with linearly increasing frequency from $f1$ to $f2$
RPN Equation	General function generator using Reverse Polish Notation (RPN).

The user can select the sampling rate, start time and the number of data points. In addition, noise can be added to the test signals.

8.1.1 Add Noise

During creation of test signals, you will be asked whether to add noise to the signal (Figure 8-1).

Figure 8-1. The Noise Method pop-up window.



If the option *Noisy* is selected, PITSA will prompt for the statistical properties of the noise to be added, which are described by the Noise Mode and the noise variance σ . There are two options, shown in Table 8-3

Table 8-3. Noise Modes.

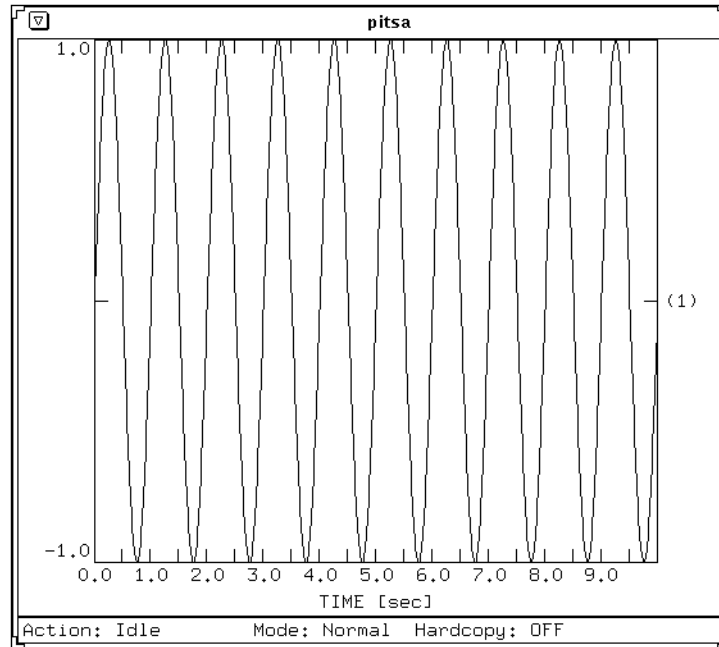
Mode	Mode Name	Noise Distribution
1	Equally distributed	Over the interval $[-\sqrt{6\sigma}, \sqrt{6\sigma}]$
2	Gaussian distribution	$(1/\sigma\sqrt{2\pi})\exp(-1/2(x/\sigma)^2)$

You will initialize the random number generator in PITSA by entering a negative integer value when asked for the random number seed value. The same seed value will produce the same noise sequence for a given sampling rate and test signal length and hence allows you to create repeatable noisy test signals.

8.1.2 Sine/Cosine

To create a sinusoidal signal ($f(t) = A \cdot \sin(2\pi ft + \phi)$), you will be asked to enter amplitude (A), frequency (f in Hz), and phase values (ϕ in degrees) for each trace to create. To create a cosine, enter a phase value of 90 degrees. Digitization frequency, time of the first sample, and the number of points for each trace are kept the same for multiple traces. Figure 8-2 shows a *Sine/Cosine* test signal.

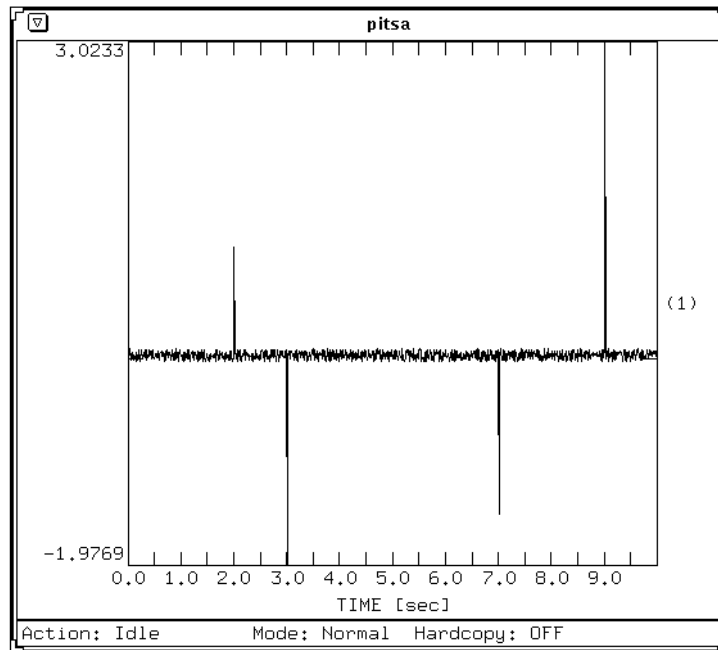
Figure 8-2. A Sine/Cosine test signal. This is a 100 Hz sampling rate, 1000 point sine wave with amplitude of 1, frequency of 1 Hz and phase angle of 0.



8.1.3 Spike(s)

Using this option, you can create spiky test signals consisting of multiple or single spikes of different amplitudes at different positions. PITSA prompts for the index of a spike position and subsequently for its amplitude. Amplitude values may be negative or positive. Legal spike indices are always positive, however, and PITSA continues asking for new spike positions until a negative value is read.

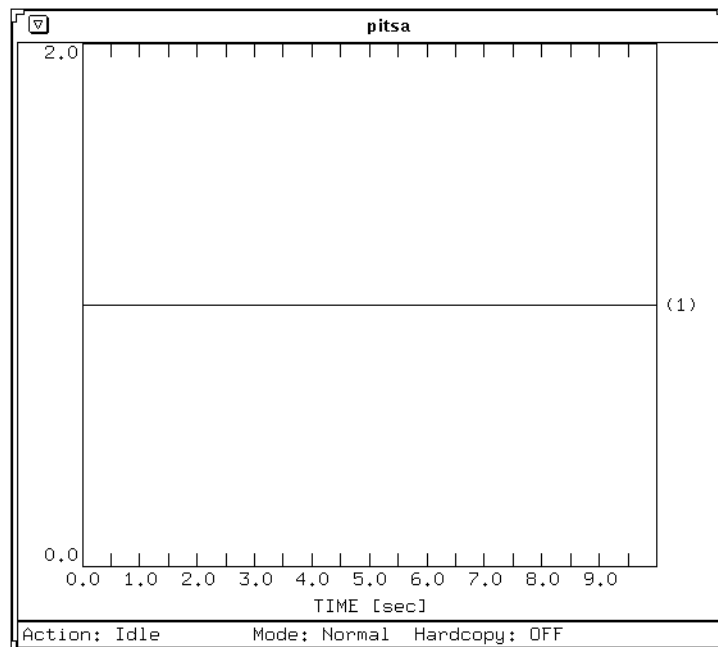
Figure 8-3. A spike test signal with noise added. The sampling rate is 100 Hz and there are 1000 points. Spikes have been added at index values 200 (value = 1.0), 300 (value = -2.0), 700 (value = -1.5) and 900 (value = -3.0).



8.1.4 Constant

This test signal is a simple single value function (Figure 8-4). The user will be asked for the value to apply to every point of the test signal.

Figure 8-4. A constant test signal (1000 points at 100 Hz sampling rate) with a value of 1.0. This signal did not have noise added to it.



8.1.5 Brune Source Signal

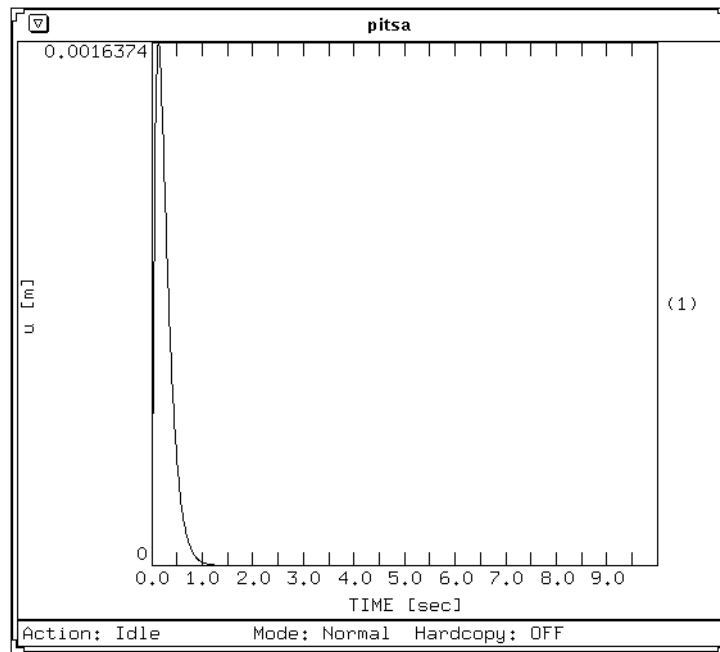
One can create a test signal corresponding to the displacement pulse $u(t)$ from a Brune earthquake source model (Brune, 1970) where:

$$u(t) = 2R(\theta, \phi) \frac{\sigma}{\mu} v_s \frac{r}{z} t e^{-2.34 \frac{v_s}{r} t}$$

with $R(\theta, \phi)$ being the radiation pattern, σ being the stress drop in bars, μ being the shear modulus in Pa, v_s the shear velocity in km/sec, and r and z being the source radius and the hypocentral distance, respectively, in km. In the formula above, the free surface is taken into account by a factor of 2 (SH waves).

PITSA prompts for input of values for the radiation pattern (rth), stress drop (sigma), shear modulus (mu), shear velocity (vs), source radius (r), and the hypocentral distance (z) separated by commas. An example for a Brune source pulse is shown in Figure 8-5.

Figure 8-5. Example of a noise free Brune source pulse created using the default input parameters (sampling rate = 100 Hz, number of points = 1000, rth = 0.625, sigma = 50.0, mu = 3×10^{10} , vs = 3.5, r = 10., z = 20.0).



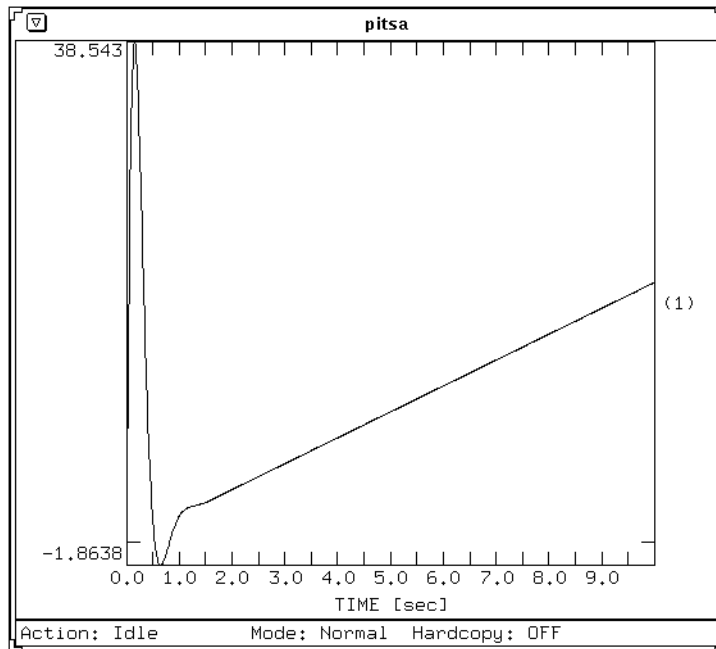
8.1.6 Mixed Signal

The mixed signal is a superposition of a sinusoidal, an exponential and a linear function in order to help the user create a very general type of test signal. It is calculated according to the following formula:

$$u(t) = A \sin(2\pi f + \phi) e^{-Bt} + Ct + D$$

PITSA prompts for the amplitude A , frequency f , phase term ϕ , and constants B , C , and D , all separated by commas.

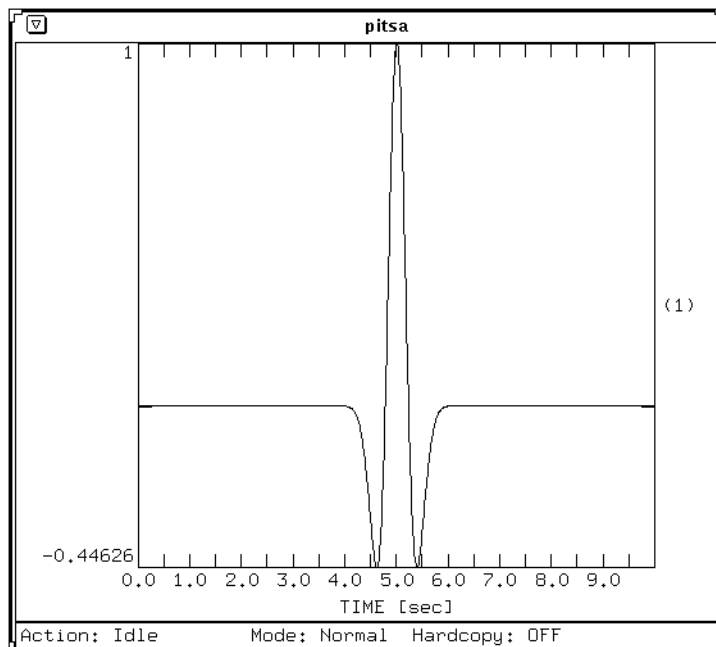
Figure 8-6. A 100 Hz 1000 point mixed signal with parameters of $A = 100.0$, $F = 1.0$, $\text{PHI} = 0.0$, $B = 5.0$, $C = 0.0$, $D = 2.0$.



8.1.7 Ricker Wavelet

The Ricker wavelet is a zero phase wavelet which is the second derivative of the error function. PITSA prompts for the dominant frequency in Hz. Figure 8-7 shows a Ricker wavelet with a dominant frequency of 1 Hz.

Figure 8-7. A Ricker wavelet with a dominant frequency of 1 Hz. The traces is 1000 points long sampled at 100 Hz.



8.1.8 Seismometer

Use this option to create test signals simulating the response of a seismometer to different input signals. PITSA will display a pop-up menu to select the type of signal:

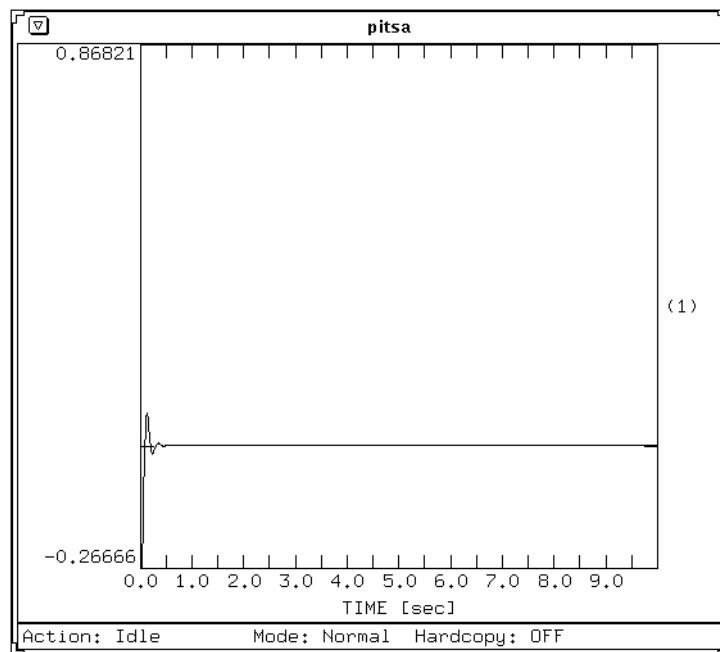
Table 8-4. Options for seismometer test signals

Signal Type	Description
<i>Displ. imp. -> displ. response</i>	Actual impulse response of a displacement seismometer.
<i>Accel. step. -> displ. response</i>	Theoretical response of displacement transducers to a step function in acceleration.
<i>Accel. step. -> veloc. response</i>	Theoretical response of velocity transducers to a step function in acceleration.

These last two kinds of input signal can be generated in seismometers with additional calibration coils by turning off (or on) a constant calibration current.

After selecting the signal type, you will be prompted for the eigenfrequency in Hz and the damping factor for the seismometer to simulate. The input signal is assumed to act at time zero.

Figure 8-8. Impulse displacement response seismometer test signal with an eigenfrequency of 5 Hz and damping factor of 0.4. The trace length is 1000 points and is sampled at 100 Hz.



8.1.9 Chirp Signal

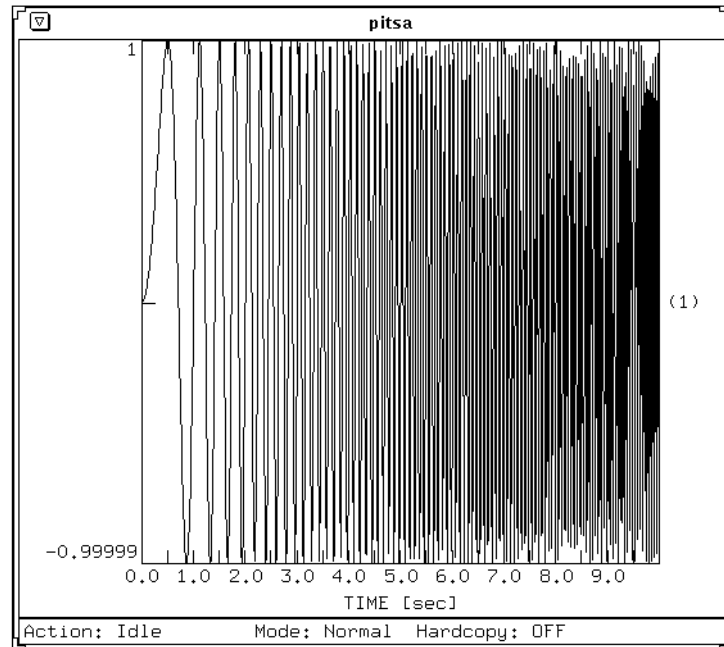
The chirp signal is produced by linearly increasing the frequency of a sinusoid between two frequencies f_1 and f_2 :

$$u(t) = A \sin(2\pi f(t)t + \phi)$$

$$f1 \leq f(t) \leq f2$$

PITSA prompts for the amplitude A , lower frequency $f1$, phase term ϕ , and the higher frequency $f2$, all separated by commas. Figure 8-9 shows an example of a chirp signal.

Figure 8-9. A 1000 point 100 Hz chirp signal with amplitude 1, $f1 = 0.01$ Hz, $f2 = 10.0$ Hz and a phase angle of 0.0.



8.1.10 RPN Equation

The *RPN equation* option is the implementation of a general purpose function generator using Reverse Polish Notation (RPN) for its input (the way many HP calculators work). The user will be asked to enter an RPN equation as input with X being the independent time variable. In order to create the sinusoidal signal:

$$u(t) = 3 \sin(2\pi ft)$$

with $f = 3$ Hz, you would enter

`X 6.28 * 3 * SIN 3 *`

Several functions and constants are available; they are enumerated in Table 8-5.

Table 8-5. Functions available in the RPN equation evaluator.

Function	Description
PI	The value of pi.
SIN	The trigonometric sine function (radians).
COS	The trigonometric cosine function (radians).
TAN	The trigonometric tangent function (radians).
ASIN	The trigonometric arc sine function (radians).
ACOS	The trigonometric arc cosine function (radians).
ATAN	The trigonometric arc tangent function (radians).
ABS	The absolute value function.
EXP	e raised to the power x ($3 \exp \rightarrow e^3$).
LN	Natural log function.
LOG	Log base 10.
POW	x to the power y ($x \ y \ \text{pow} \rightarrow x^y$).
SQRT	Square root function.
RECT	Rectangle function: $x \ \text{rect} = 1$ if $-1 \leq x \leq 1$; 0 other wise.
SINC	$x \ \text{sinc} \rightarrow \sin x / x$.
DELTA	Unit delta function.
UNIT_STEP	Unit step function.

8.2 Trace Utilities

The tools under *Trace Utilities* allow the user to perform simple trace manipulations like adding, shifting, or scaling traces, changing axis types, and stacking traces.

Table 8-6. Tools under the Utilities -> Trace Utilities menu.

Menu Item	Description
Single Channel Utilities	Utilities for manipulating single channels, including scaling, adding and subtracting offsets, power, logarithm, cyclic and linear shift, appending zeroes, time reversion and rectifying.
Double Channel Utilities	Utilities for manipulating two channels, including scaled addition, multiplication, stabilized ratio, euclidian distance calculus, arcus tangens and projection angles.
Triple Channel Utilities	Utilities for manipulating three channels, including euclidian distance calculus and projection angles calculation.
Scale Traces Together	Plot traces in common scale.
Re-Autoscale Traces	Reverting changed viewports to automatic scaling.
Change Axis Type	Choose different axis types for channels.
Stack Traces	Methods for stacking traces.
Unify Traces	Unify channels to common start time and length.
Normalize Traces	Normalize maximum in channel to given amplitude.

As their names indicate, some of these utilities require a specific number of input traces to act on.

8.2.1 Single Channel Utilities

The tools available under the Single Channel Utilities menu option are described in Table 8-7. They are all rather easy to understand and require one or two input parameters besides the channel list.

Table 8-7. Single Channel Utilities.

Tool	Description
$a * X + b$	Scale and offset a trace. X represents the trace to be processed. Input values a and b. Each point in the original trace is multiplied by a and then b is added to obtain a new trace.
$a * X ** b$	Raise all values of input trace (X) to a power b and scale by a factor a.
Log10(X)	Take Log10 of all values of the input trace X.
Cyclic Shift	Enter a value for number of points to shift; positive value shifts trace to the right, a negative value shifts to the left. All values shifted outside of the current window will re-appear on the other side.
Linear Shift	Enter a value for number of points to shift. A positive value will cause the trace to be shifted to the right, a negative value to the left. All values shifted outside of the current window are set to 0.
Append zeros	Zero padding. Input number of zeros to append.
Invert trace in time	Simple inversion in time. No additional input.
Rectify	Absolute value of every point in the input trace.

8.2.2 Double Channel Utilities

The tools available under the *Double Channel Utilities* menu option are described in Table 8-8. They mostly involve simple algebraic or trigonometric manipulations of two traces, point by point. Therefore the two traces used for input must contain the same number of points. The first trace of the channel list will be used for the X trace and the second trace will be used for Y.

Table 8-8. Double Channel Utilities.

Tool	Description
$a*X + b*Y$	Enter values for a and b.
$X*Y$	No further input.
X/Y	Stabilized ratio of X and Y. Enter threshold value c to prevent divide-by-zero errors. If $Y < c$, the denominator for the division will be set to $\pm c$.
$\text{Sqrt}(X*X + Y*Y)$	If input traces are viewed as components of a time-dependent 2-D vector, resulting trace corresponds to the length of this vector at any time.
$\text{atan2}(X,Y)$ [deg]	No further input.
Projection Angles	Resulting trace contains projection angle of the particle motion vector defined by X and Y, w.r.t. one of the selected coordinate axes. Calculated as inverse cosine of the scalar product of the particle motion vector and the selected coordinate axis.

8.2.3 Triple Channel Utilities

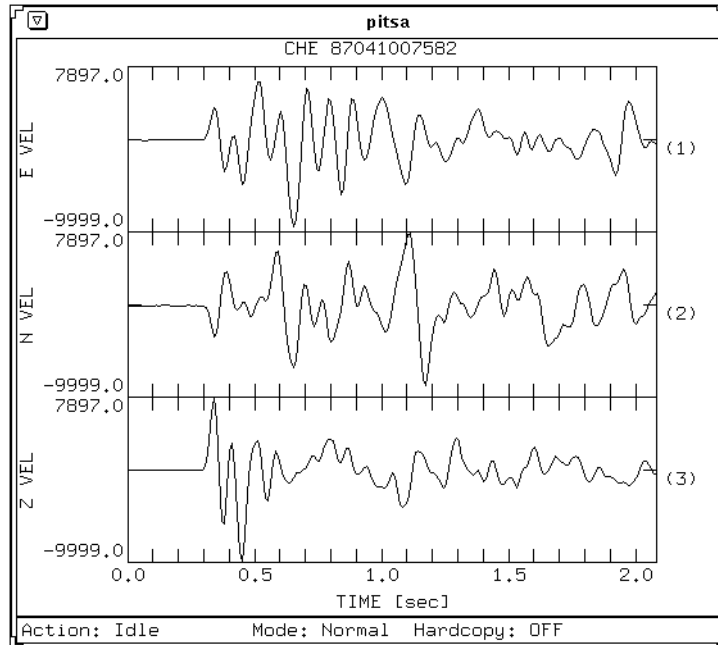
The tools available under the *Triple Channel Utilities* menu option are described in Table 8-9. They involve simple algebraic or trigonometric manipulations of three traces, point by point. Therefore the three traces used for input must contain the same number of points. The three traces of the channel list are interpreted in order as X, Y, and Z, respectively.

Table 8-9. Triple Channel Utilities.

Tool	Description
$\text{SQRT}(X*X + Y*Y + Z*Z)$	If input traces are viewed as components of a time-dependent 3-D vector, resulting trace corresponds to the length of this vector at any time.
Projection Angles	Resulting trace contains projection angle of the particle motion vector defined by X, Y, and Z w.r.t. one of the selected coordinate axes. Calculated as inverse cosine of the scalar product of the particle motion vector and the selected coordinate axis.

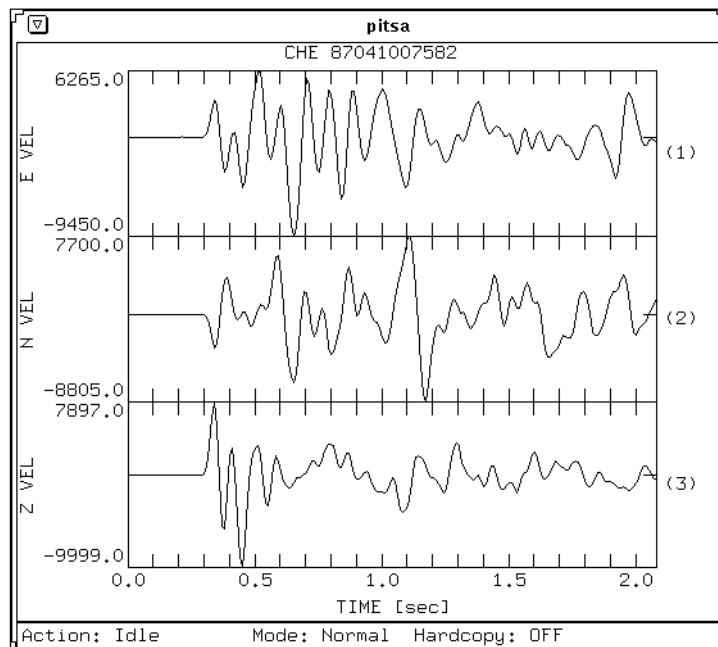
8.2.4 Scale Traces Together

In general, PITSA scales each trace automatically so that it is plotted at highest screen resolution. Especially with three component data, however, it is sometimes advantageous to preserve the real amplitude relationship between traces. The command *Utilities -> Trace Utilities -> Scale Traces Together* will cause the selected traces to be plotted at a common scale. Figure 8-10 shows three traces that have been scaled together. Figure 8-11 shows the same three traces after they have been re-autoscaled.

Figure 8-10. Three traces scaled together.

8.2.5 Re Autoscale Traces

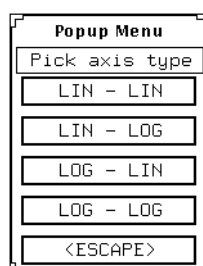
For traces which have been scaled together to a common viewport (Section 8.2.4), or have had their viewport changed manually (Table 8-12), automatic scaling for selected traces can be re-invoked using the *Utilities -> Trace Utilities -> Re Autoscale Traces* option. Figure 8-11 shows the traces in Figure 8-10 after they have been re-autoscaled.

Figure 8-11. The traces of Figure 8-10 after they have been re-autoscaled.

8.2.6 Change Axis Type

Axis display types can be changed using into any combination of linear and logarithmic scales, using the option *Utilities -> Trace Utilities -> Change Axis Type*. PITSA displays the menu shown in Figure 8-12, from which the selection is made.

Figure 8-12. Axis Type pop-up menu.



8.2.7 Stack Traces

An arbitrary number of channels can be stacked using the menu selection *Utilities -> Trace Utilities -> Stack Traces*. There are three options:

- Receiver Beam
- Source Beam
- Plain Sum

Either of the *Beam* options will calculate a single-beam sum for a given back-azimuth (or azimuth, in the case of *Source Beam*) and slowness while *Plain Sum* will simply sum up the selected list of channels on a point by point basis.

The Y-axis plot label of the resulting trace is set to “RES” for all these utilities. See Table 8-12 on page 8-18 for information on how to change this.

8.2.7.1 Receiver Beam

The selected traces are considered to represent recordings at a group of seismic stations from a single event. Therefore the “station” location information in the data headers must be set properly before this tool can be used. Station coordinates are required to be specified in degrees (geographic coordinates).

PITSA prompts for the back-azimuth and slowness, and for the index of a reference channel. For each trace, PITSA calculates the time delay with respect to the reference channel for a signal propagating at the given slowness and azimuth. The resulting trace consists of the stacked delayed traces (“delay and sum” beam).

8.2.7.2 Source Beam

In this variation of beam-forming, the selected traces are considered to represent recordings at a single station of waveforms from a cluster of earthquakes (a “source array”). Therefore the “event” location information in the data headers must be set properly before this tool can be used. Event coordinates are required to be specified in degrees (geographic coordinates).

PITSA prompts for the azimuth (leaving the source area) and slowness, and for the index of a reference channel. For each trace, PITSA calculates the time delay with respect to the reference channel for a signal propagating at the given slowness and azimuth. The resulting trace consists of the stacked delayed traces (“delay and sum” beam).

8.2.7.3 Plain Sum

No further input is required. The resulting trace will consist of the stacked trace of all the traces in the selected channel list.

8.2.8 Unify Traces

PITSA works with a group of traces most easily if they all start at the same point in time and have the same number of points. Selecting *Utilities -> Trace Utilities -> Unify Traces* will allow the user to enter a list of traces that are to be unified in this manner. PITSA will find the earliest start time and latest end time of all traces entered and then pad the beginning and end of each trace with zeros such that all the traces will start and end at the same time.

8.2.9 Normalize Traces

Utilities -> Trace Utilities -> Normalize Traces simply normalizes the traces given in the input list to the user specified value.

8.3 Header Access

Besides the actual trace data values, PITSA keeps additional information on each trace in what are called data headers. Some header information is always required for PITSA’s internal processing; PITSA usually maintains these entries automatically. Other header entries are only used by certain tools, however, and are therefore optional. This chapter describes how to gain access to the header entries and modify them. Some of the header information can also be written out to a plain ASCII file. The accessible header entries are grouped in five categories:

- Station
- Event
- Record
- Plot
- Output

Any changes made in any of the header entries are only stored in memory. You must re-save the traces in ISAM format if you want to make the change permanent.

8.3.1 Station

Station-related header entries which can be accessed are listed in Table 8-10.

Table 8-10. Station-related Header Entries.

Header Entry	Description
Station Code	Maximum 6 characters (e.g., STU).
Channel ID	Maximum 6 characters (e.g., LPZ).
Instrument Type	Maximum 8 characters (e.g., WWSSN).
Coordinates	Entered as X, Y, Z triplet plus coordinate system type code: 0 = degrees/degrees/km 1 = km/km/km 2 = m/m/m

8.3.2 Event

Event-related header entries which can be accessed are listed in Table 8-11.

Table 8-11. Event-related Header Entries.

Header Entry	Description
Origin Time	The pattern is: “year month day hour minute second”, separated by blanks.
Azimuth, Back-Azimuth	Azimuth is defined as the angle measured between the vector pointing from the source to the station and the vector pointing from the source to the north. Back-azimuth is defined as the angle measured between the vector pointing from the station to the source and the vector pointing from the station to the north. Can also be calculated from the event and station coordinates (see “Recalculate Parameters” option below).
Epi-, Hypo Distance	Epicentral and hypocentral distances (km). Can also be calculated from the event and station coordinates (see “Recalculate Parameters” option below).
Coordinates	Entered as X, Y, Z triplet, separated by blanks, plus coordinate system type code: 0 = degrees/degrees/km 1 = km/km/km 2 = m/m/m Event depth must be entered as a negative number.
Magnitude	Enter values for local magnitude (M_L), body wave magnitude (m_b), and surface wave magnitude (M_S) separated by blanks.
Recalculate Parameters	After event coordinates are entered, use this option to calculate azimuth, back azimuth, epicentral and hypocentral distances and make the corresponding header entries automatically for the selected list of channels.
Coordinates correction	Recalculates event coordinates according to a given reference point and azimuth.

8.3.3 Record

This group of entries contains the time of the first sample and the sampling rate (in Hz), which are entered together. Time entries in PITSA have to be entered as year, month, day, hour, minute, second, separated by blanks.

8.3.4 Plot

This selection gives the user access to trace header parameters that effect the way traces are plotted on the screen. The menu options are summarized in Table 8-12.

Table 8-12. Plot-related Header Entries.

Menu Option	Description
Viewport	Define plotting margins in user units. Parameters <i>xmin</i> and <i>xmax</i> define minimum and maximum value for horizontal axis (normally time), while <i>ymin</i> and <i>ymax</i> define minimum and maximum values for vertical axis (normally amplitude). Setting viewport manually turns off autoscaling and forces all traces to be plotted at same scale. Enter values for <i>xmin</i> , <i>xmax</i> , <i>ymin</i> and <i>ymax</i> separated by blanks.
Bottom Label	Label for bottom of entire plotting area. Only the bottom label stored in the header of the first trace on the screen will be displayed.
Top Label	Label for top of entire plotting area. Only the top label stored in the header of the first trace on the screen will be displayed.
X Axis	x-axis label (e.g., TIME) plotted at the bottom of each trace.
Y Axis	y-axis label (e.g., VEL N) plotted at the left side of each trace

8.3.5 Output

Some of the header information, such as phase picks and magnitudes, can be written out to a plain ASCII file for documentation and/or further processing. After entering the name of the output file, you will be prompted for a comment line which will be the first line of the output file (see example below). The remaining lines will contain the magnitudes, station coordinates, and all the picked phases for each of the selected channels:

```
comment: It's a sad but beautiful world
magnitudes sta: FA  Z   ML:  0 MB:  0 MS:  0
sta chan sta x   sta y   sta z   coord  phase  id
yr  mo  dy  hr  sec
FA  Z   12.2250  49.8610  0.4650  DGDGKM  EPU2   90
11  17  18   18  15.088
FA  Z   12.2250  49.8610  0.4650  DGDGKM  ESD0   90
11  17  18   18  20.592
```

8.4 Phase Utilities

The tools available under the *Utilities -> Phase Utilities* menu are used for specialized analysis of seismic phase.

8.4.1 Mark slowness window at receivers

This option enables the user to calculate artificial phase picks for the selected traces marking the begin and end time of a particular slowness window. In addition to the channel list to be processed the user has to enter the following parameters:

- Index of reference channel
- Reference phase ID. The corresponding phase has to exist on the reference channel.
- Delay time between the reference phase and the first phase marking the slowness.

- Time duration of the slowness window.
- Phase IDs for the begin and the end of the slowness window.
- Back azimuth (degrees) and slowness (seconds/degree) for the calculation of the time delays for each trace.

For each trace two artificial phase separated in time by the time duration of the slowness window are calculated and entered into the trace headers. On the reference trace the phase marking the begin of the slowness window will appear at the selected delay time. The reference phase must be defined on the reference channel while on the other channels it is optional.

8.4.2 Align phases for receiver slowness

Traces with a particular phase defined (e.g. P) can be rearranged such that the selected phase appears under a particular slowness at the receivers. This can be desirable for example to correct for site effects on the slowness properties. In addition to the trace list, the user has to enter the following parameters:

- Index of reference channel.
- Reference phase ID. The corresponding phase has to exist **on all traces** to be processed.
- Number of pre-phase samples. The resulting traces will be cut such that the selected number of pre-phase samples is present on the reference channel.
- Back azimuth (degrees) and slowness (seconds/degree) for the calculation of the time delays for each trace.
- Equalize traces to n samples (negative to ignore). In case a positive number is given, the resulting trace lengths are adjusted to an overall trace length of n samples.

8.4.3 Cut phase windows

For the selected traces, this option sets all amplitude values to zero which do not fall within the time window defined by the two selected phase IDs.

8.4.4 Relative onset time determination

This tool is used for determining high precision relative onset times according to the multiplet analysis introduced by Poupinet et al. (1984) and Scherbaum and Wendler (1986). While using this technique two basic assumptions are made. We are dealing with a cluster of events, and the velocity structure of the source region is constant in a volume containing all events of that cluster. According to the definition of an earthquake cluster, the waveforms of different events are coherent at a given receiver station. PITSA will calculate phase shifts of the slave event which are accurate down to sub sample timing errors if certain thresholds of the waveform coherence and the cross spectral amplitude are exceeded.

After input the trace list of one station for different events and selecting the reference trace containing the master event, PITSA prompts for the phase ID used for the calculations. Now you are asked to input a time window for the different steps of computation (Figure 8-13).

Figure 8-13. Input panel for cross correlation and cross spectrum window size.

Enter window for cross correlation

Separate items by comma (,)

notepopup

25<-----> 75

50<-----phase----->100

First, PITSA prompts for the time window used for the computation of the cross correlation. In addition the user has to specify also the window used for computing the cross and coherence spectrum, respectively. For estimating a meaningful coherence spectrum, you have to input the half frequency window size which is used for averaging the coherence values (Figure 8-14).

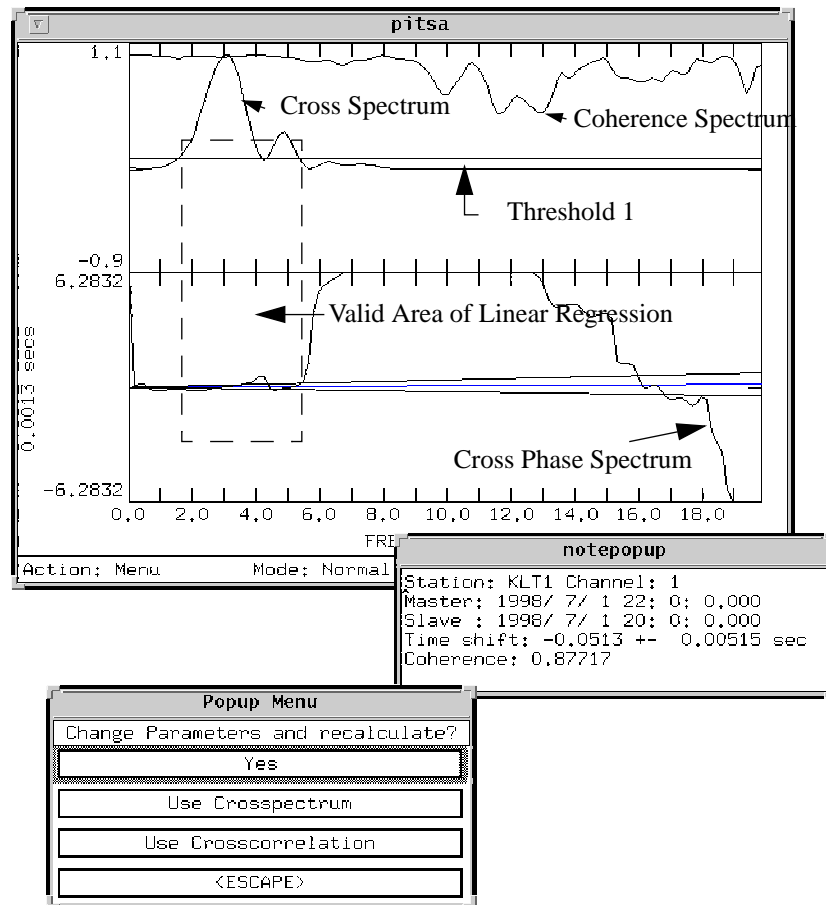
Figure 8-14. Input panel for smoothing spectral values of coherence spectrum.

Enter half width m for spectral smoothing

total length of smoothing window: $2\ m + 1$ points

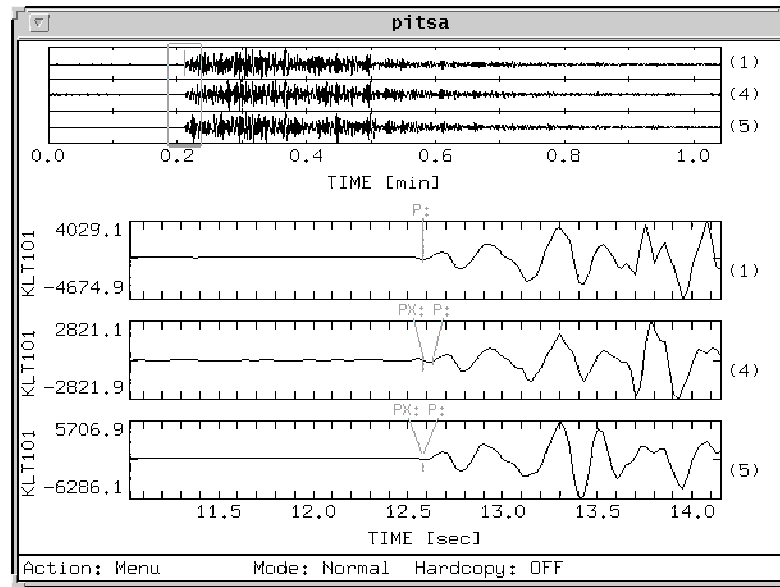
Now PITSA prompts for two thresholds. You are asked to input the threshold for the normalized cross spectrum (*weighting factor*). Then PITSA prompts for the threshold for the coherence. In order to get a timing accuracy smaller than a sample, a linear regression of the cross phase spectrum is performed only in the signal part exceeding this two thresholds. The result is shown in Figure 8-15.

Figure 8-15. Result of relative onset time determination. The upper panel shows the coherence spectrum, the cross spectrum and the threshold (*weighting factor*) for the cross spectrum. In the lower panel the cross spectral phase and the linear regression line and error bounds are shown. In the note popup the estimated time shift is displayed.



In the upper panel of the PITSA screen appears the computed coherence spectrum, the cross spectrum and the threshold for the cross spectrum. In the lower panel the cross spectral phase and the result of the linear regression as well as the error regions are shown as blue and black line, respectively. The regression algorithm assumes zero shift at zero frequency. If the computed phase shift is abnormal or the bandwidth of the valid frequency band is too small, you can use the time estimation by accepting *Cross correlation*. In order to change the settings used for the computation, you can select *Yes* in the *Change Parameters and recalculate* popup menu. Finally PITSA includes a new phase ID labeled as *PX* to the preexisting phase list (Figure 8-16).

Figure 8-16. Phase entries labeled *PX* are added to the phase list of the slave event.



The relative onset time of the “slave events” according to the master is simply:

$$\Delta t_{MS}^i = t_{Master}^i - t_{PX}^i$$

Where t_{Master}^i is the onset time of the master event, t_{PX}^i the onset time of the just computed Phase *PX* and Δt_{MS}^i the relative onset time of the slave event according to the master event.

8.5 Resampling

The tools under the *Utilities -> Resampling* menu are used for demonstrating the effects which can occur during sampling and A/D conversion of seismic signals—they are primarily used for teaching. For example, an input signal (e.g., a test signal with well known properties) can be re-discretized (setting the values between every *n*-th sample to zero) to demonstrate the effects of analog to digital conversion. In addition, the effects of digital to analog conversion can be demonstrated on re-discretized signals using Whittaker's reconstruction technique. You can also simulate the effects of downsampling (keeping only every *n*-th value) an input signal of comparatively high dynamic range using a gain-ranging A/D converter or down-sample at full floating point precision. Many examples of the use of these tools are contained in the book, *Short Course on First Principles of Digital Signal Processing for Seismologists* by Frank Scherbaum (1993).

There are five options under the *Utilities -> Resampling* menu:

- Simulate discretization in time
- Reconstruction of discretized signals

- Integer decimation
- Resampling of interpolation polynomial (4-deg)
- Simulate A/D Converter

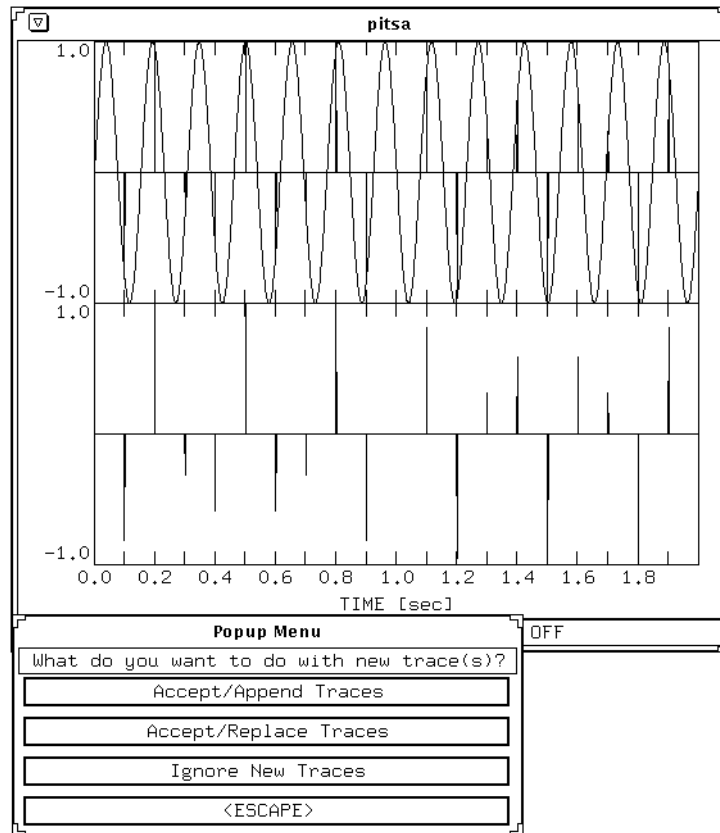
8.5.1 Simulate discretization in time

All trace data in PITSA are already discretized. Using this tool, however, the input trace is treated as a pseudo-continuous trace from which discrete samples are taken every $1/f_{\text{DIS}}$ samples, where f_{DIS} is the discretization frequency. PITSA prompts for the discretization frequency (f_{DIS}) in Hz. The resulting trace keeps the original sampling rate, but trace values are set to zero at points between the “resampled” points. For this option to work properly, the specified discretization frequency f_{DIS} must be smaller than the real digitization frequency f_{DIG} .

In the top trace of the display for this tool, the discretized signal is superimposed on top of the original trace while in the bottom trace it is displayed alone. It can be kept for later processing, e. g., for demonstrating pseudo D/A conversion using Whittaker reconstruction.

Figure 8-17 demonstrates the discretization of a sinusoidal signal of 6.5 Hz with a discretization frequency of 10 Hz, a nice demonstration of aliasing.

Figure 8-17. Discretization of a 6.5 Hz sinusoid with $f_{\text{DIS}} = 10$ Hz. The original sampling frequency was 100 Hz.



8.5.2 Reconstruction of discretized signals

A method called “Whittaker reconstruction” is used in this tool to simulate the process of digital to analog conversion for traces which have been obtained by using the re-discretization tool described in Section 8.5.1. It uses a simple time-domain lowpass filter technique

PITSA will prompt for the frequency used for the re-discretization (in Hz). This is not the actual digitization frequency! See Section 8.5.1 for the differences.

The top panel of the display will contain the reconstructed signal, superimposed on top of the corresponding discretized trace. The lower panel shows the reconstructed signal by itself (Figure 8-18).

Figure 8-18. The result of a Whittaker reconstruction. In the upper trace, the original discretized trace and the reconstructed trace are displayed together while in the lower trace, only the reconstructed trace is displayed.

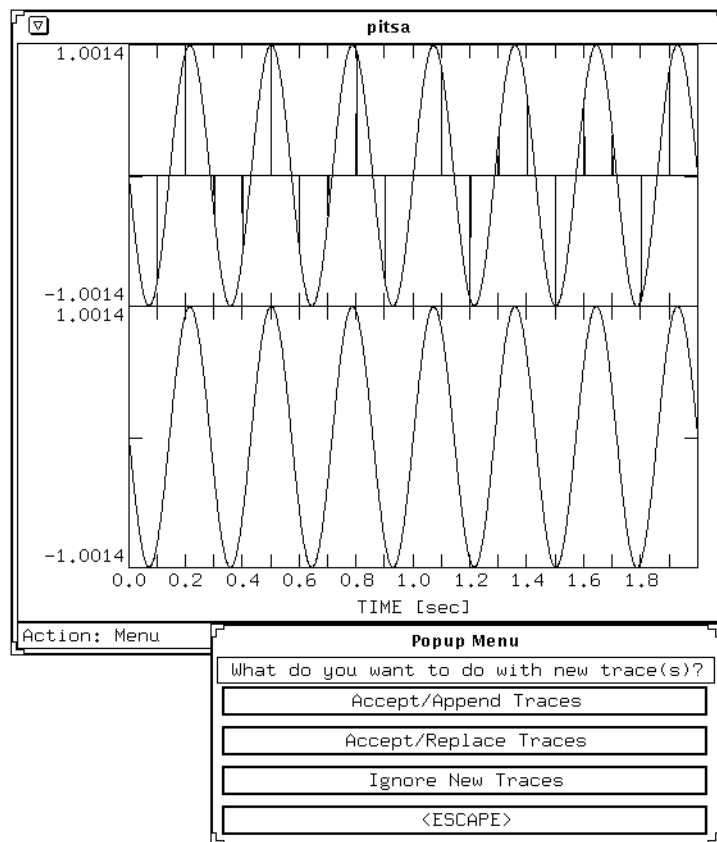
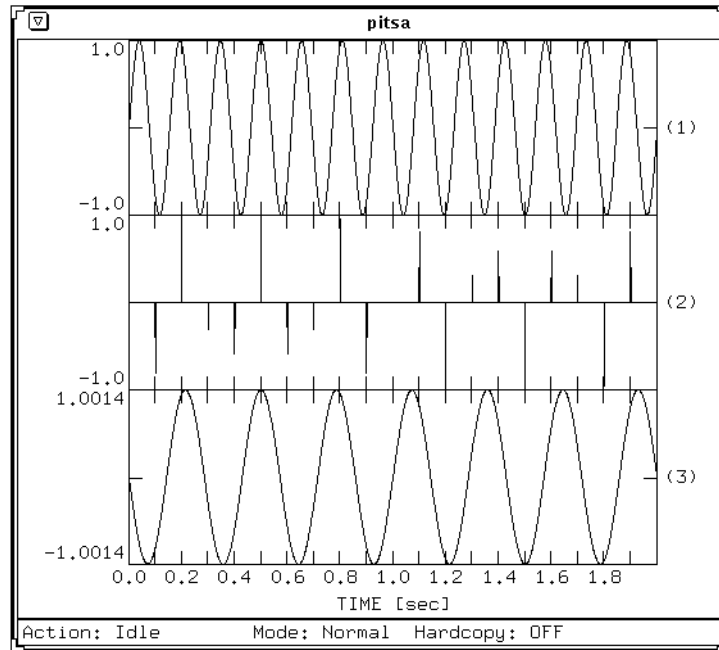


Figure 8-19 shows the display after the reconstructed trace in Figure 8-18 was appended to the traces in memory.

Figure 8-19. Whittaker reconstruction applied to the discretized trace of Figure 8-17. The top trace shows the input sinusoid (signal frequency 6.5 Hz), the middle trace the discretized signal ($f_{\text{DIG}} = 10 \text{ Hz}$), and the bottom trace the result of reconstructing trace 2. Notice that the reconstructed trace has a lower signal frequency (3.5 Hz) than the trace which was discretized which demonstrates the aliasing effect.



8.5.3 Integer decimation

Integer decimation allows the selection of every n -th value in the input trace (starting with the first one). The user has to provide the decimation ratio as an integer value. E.g. giving an input value of 2 will result in an output trace containing the first, third, fifth, and so forth values of the input trace.

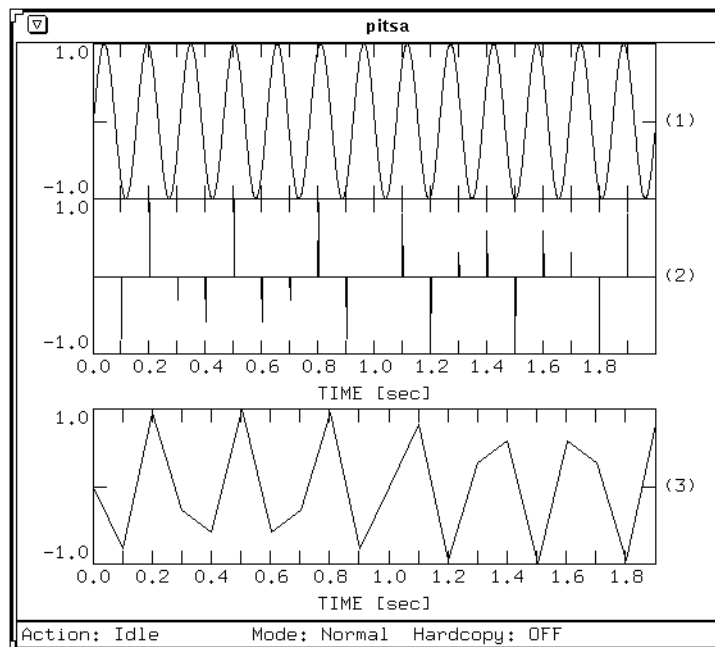
8.5.4 Resampling of interpolation polynomial (4-deg)

In contrast to the simulation of discretization discussed in Section 8.5.2, where all the samples between the discretization interval were set to zero but kept in the trace, the *Resampling...* tool described in this section actually downsamples the input trace to a new digitization frequency. Hence, the number of samples within the new trace will be smaller by a factor of $f_{\text{DIG_NEW}} / f_{\text{DIG_OLD}}$ in comparison to the input trace. Here, $f_{\text{DIG_NEW}}$ and $f_{\text{DIG_OLD}}$ denote the original digitization frequency of the input trace and the resampling digitization frequency, respectively.

PITSA will prompt for the new digitization frequency in Hz (the original digitization frequency will be displayed in the text input window). The resulting trace will contain samples at time intervals taken from the input trace at time intervals $1/f_{\text{DIG_NEW}}$. The new data values are interpolated from the neighboring points of the original data series using a 4th degree polynomial.

Figure 8-20 shows an example of a digitally resampled trace versus a discretized trace.

Figure 8-20. Traces 1 and 2 correspond to traces 1 and 2 in Figure 8-19. Trace 1 is the input trace that was first discretized (trace 2) and then resampled (trace 3).



8.5.5 Simulate A/D Converter

The option *Utilities -> Resampling -> Simulate A/D Converter* simulates the behavior of a gain-ranging A/D converter (ADC). It is a simulation because we cannot use a true analog signal for input. Instead we use the method discussed in Section 8.5.1 to create our input signal.

Gain-ranging ADCs were common in seismology in the days before high-performance ADCs with up to 24 bit resolution, but they can still be found in use and it is useful to understand their deficiencies. The gain-ranging ADC consists of a fixed-gain pre-amplifier, a programmable gain-ranging amplifier (PGA) with a variable number of gain states, and a plain ADC for which the resolution and the least significant bit values can be assigned. The pre-amplifier is used to attenuate the input signal so that it does not exceed the limited scale range of the plain ADC.

For the ADC simulation, the user has the choice of four output signals, listed in Table 8-13.

Table 8-13. Output Signal Types for ADC Simulation.

Type	Description
Quantized Value	Actual output signal from the gain-ranging ADC.
Mantissa	Output of the plain ADC.
Exponent (-gain)	Gain setting of the PGA
Error Signal	Difference between the quantized value and the input trace.

If gain ranging is turned on, the actual output signal is combined from the output of the plain ADC (“mantissa”) and the gain setting (“exponent”) of the programmable

gain amplifier (PGA). In this case, the internal representation is similar to a scientific (mantissa, exponent) notation.

For the simulation of a gain-ranging ADC, you should start with a test signal generated at a very high sampling rate (e.g., 1000 Hz) which will be treated as an “analog” signal. The sampling of this signal is simulated by a simple down-sampling process which uses only linear interpolation (unlike the tool in Section 8.5.4)—but it’s OK because the original sampling rate is so high. To begin the simulation PITSA prompts for the digitization frequency (Hz) of our hypothetical recording system. A frequency of 20-80 Hz might be appropriate. Before discussing the other input parameters which are needed to describe the behavior of the gain-ranging ADC, it is helpful to understand how gain-ranging works.

8.5.5.1 How Gain-Ranging Works

The incoming sample is first multiplied by the scaling factor of the fixed-gain pre-amplifier. Next it is verified that the signal will fit within the lowermost 25% scaling range of the plain ADC. If it does not, the PGA takes over and attenuates the sample amplitude in steps of 2 until it is small enough to meet this condition. At the same time, the gain state (which is a number between 0 and the maximum number of gain states possible - 1) of the PGA is stored. A difference of 1 in gain states corresponds to a factor of 2 in attenuation.

Next, the actual analog to digital conversion, simulating an n-bit ADC, is done. It consists of rounding the output value of the PGA to the next lowest value an ADC could represent with the given value for the least significant bit (LSB) and the given resolution. Given n bits of resolution and a value for LSB, 2^n values ($0 \cdot \text{LSB}$ - $[2^n - 1] \cdot \text{LSB}$) can be represented before saturation occurs. The quantized value is finally calculated by multiplying the output value of the plain ADC by 2^{gain} , with *gain* being the gain state of the PGA.

8.5.5.2 Input Parameters

After the digitization frequency for resampling has been entered, PITSA will prompt for the additional parameters for the gain-ranging ADC.

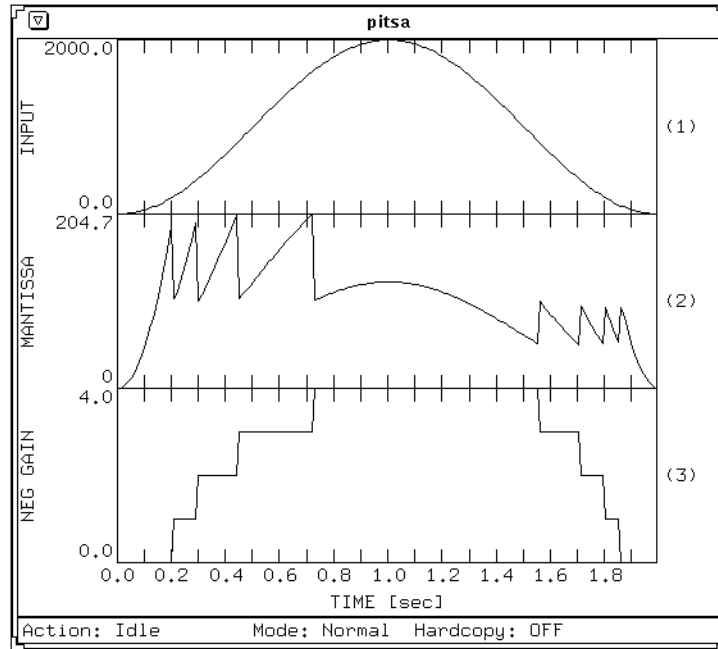
Table 8-14. Parameters for a Gain-Ranging ADC.

Parameter	Description
Pre-amplifier gain	Scaling factor (e.g., 1.0) which simulates a fixed gain pre-amplifier.
Resolution	Number of bits (e.g., 12) for the plain ADC.
LSB value	Value of least significant bit (e.g., 0.1).
Number of PGA gain states	How many times (e.g., 10) the PGA can decrease the amplitude of the input signal for the plain ADC.

During the conversion, PITSA will display a note window with the actual values for the current index, input value, output value, gain state, mantissa, and the error signal.

Figure 8-21 shows an example for the simulation of the mantissa and gain output of a gain ranging A/D converter operating on a cosine bell.

Figure 8-21. Gain ranging A/D converter on a cosine bell. The top trace shows an input signal while the middle and bottom traces simulate the mantissa and gain function of a 12 bit gain ranging A/D converter with 11 PGA gain states.



The tools under the *Special Plots* menu are used for specialized displays in various kinds of seismological analysis. These tools are summarized in Table 9-1.

Table 9-1. Special Plots menu items.

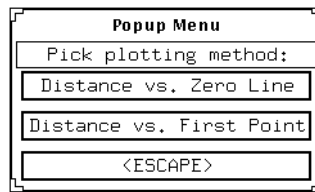
Menu Item	Description
Epi-Distance Plot	Plot seismic traces as a function of epicentral distance.
Hypo-Distance Plot	Plot seismic traces as a function of hypocentral distance.
Receiver Array Vespagram	Calculates and plots vespagram (Davies et. al., 1971).
Receiver Array Slowness Power Spectrum	Calculates and plots the slowness power spectrum of a receiver beam regarding to the delay and sum technique.
Superimpose Traces	Plot two or more seismic traces with equal sampling rate and length together in one box.

9.1 Epi-Distance Plot

This option is used to plot seismic traces in order of increasing epicentral distance. Epicentral distances must be set correctly within the data headers of the traces to be used.

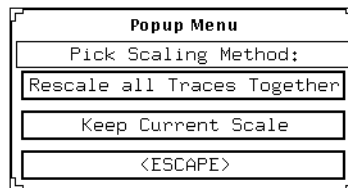
PITSA will prompt for the plotting method (Figure 9-1). The *Distance vs. Zero Line* option will place the imaginary zero line of each trace at the corresponding distance while *Distance vs. First Point* will place the first point of the data trace at the corresponding distance.

Figure 9-1. Plotting methods pop-up menu.



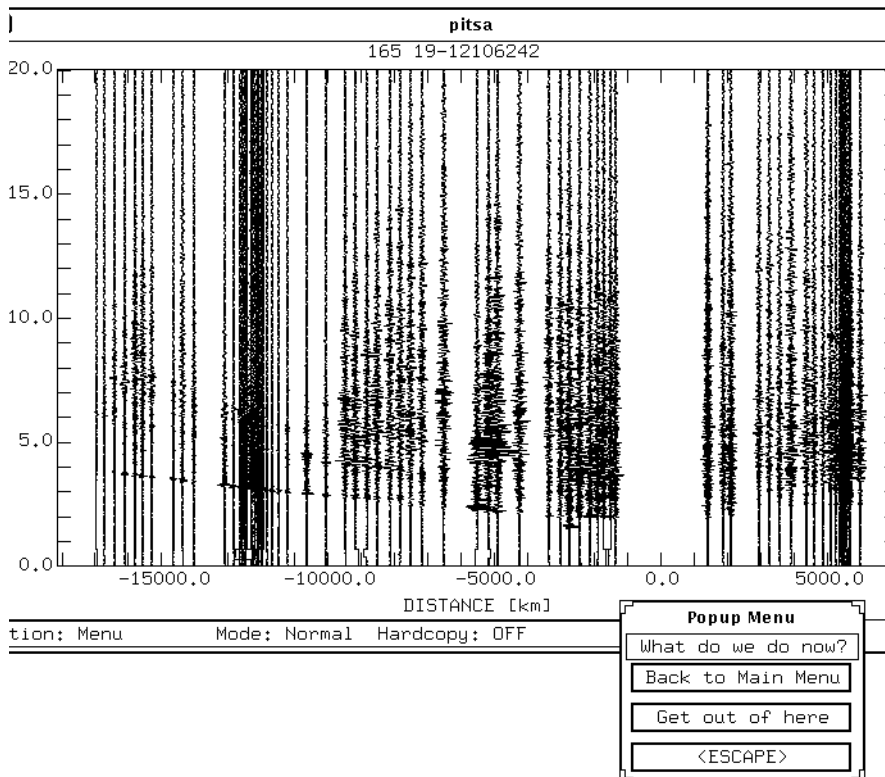
Next PITSA will ask for the scaling method (Figure 9-2). The user can select either *Rescale all Traces Together* to plot all traces at the same scale or select *Keep Current Scale*.

Figure 9-2. Scaling methods pop-up menu.



Finally, the user will be asked to enter a value for the width of the plotting window going to be reserved for each trace. PITSA expects this value to be given in percent of the total window width. Figure 9-3 shows an example if an epicentral distance plot.

Figure 9-3. *Special Plots -> Epi Distance Plot*. Files were scaled together and plotted vs. the zero line. The width of traces is 5%. These data are from aftershock recordings of the Chalfant Valley earthquake of 1986 (Luzitano, 1988) which is in the ISAM file system *isam11* in the *demodata* directory.



After PITSA displays the plot, a menu will appear (Figure 9-3) with two options. Selecting either of these options will bring PITSA back to the top level.

9.2 Hypo-Distance Plot

This plotting option is used to plot seismic traces in order of hypocentral distance. It works exactly as the *Epi-Distance* option described above. Hypocentral distances must be set correctly in the data headers for this option to be used.

9.3 Receiver Array Vespagram

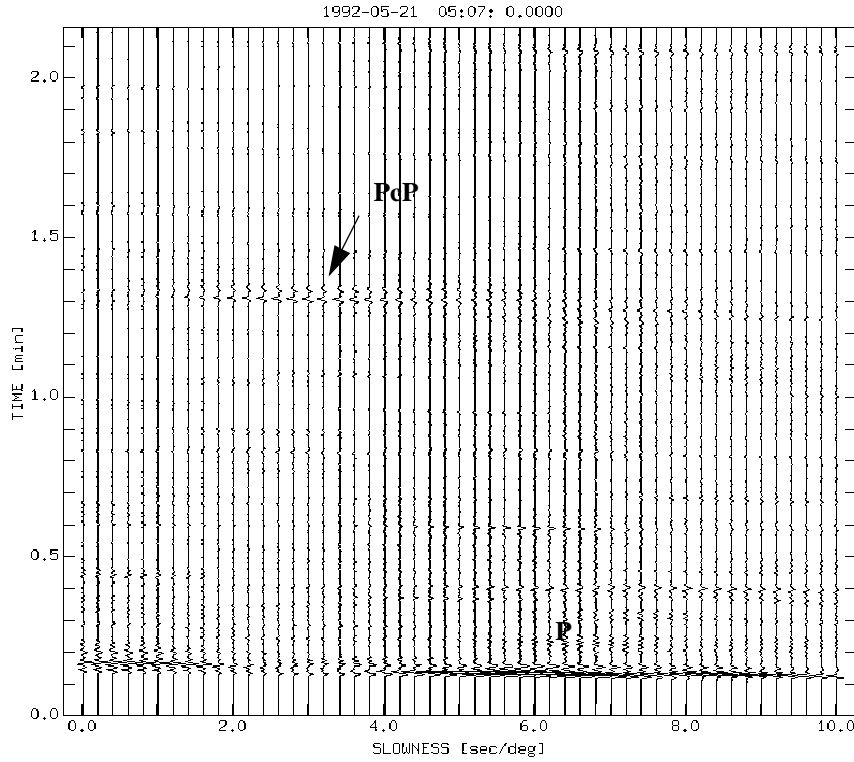
Vespagram analysis (Davies et. al., 1971) is a simple but powerful technique to identify individual phases based on their slowness properties. In a nutshell, a vespagram is a set of array beams (here receiver array beams) formed for different slowness values and a fixed azimuth. To improve its performance it is implemented here in conjunction with the Nth root technique proposed by Muirhead and Datt (1976).

The user must provide the input for the following parameters:

- Channel list for the traces to be used.
- Index of reference channel. The time scale of the reference channel will remain unshifted. Depending on the slowness and back-azimuth, during beam forming, all other traces will be shifted with respect to the selected reference channel.
- Slowness range and slowness increment to be used for the beam forming. The sequence of input values is: smallest slowness value, largest slowness value, slowness value increment. All slowness values in sec/deg.
- Parameter N for Nth root processing.

Subsequently, the user will be asked to enter a value for the width of the plotting window going to be reserved for each trace. PITSA expects this value to be given in percent of the total window width. Finally the user has to indicate if the traces should all be scaled to the same maximum amplitude value (normalized) before the beam forming. Finally, the user will be asked to enter a value for the width of the plotting window going to be reserved for each trace. PITSA expects this value to be given in percent of the total window width. Figure 9-3 shows an example if an epicentral distance plot. An example for a vespagram plot is shown in Figure 9-4.

Figure 9-4. Special Plots -> Receiver array vespagram Vespagram section calculated for the Lop Nor nuclear explosion of 1992/05/21 recorded at the GRF array.



9.4 Receiver Array Slowness Power Spectrum

Warning: this tool is for advanced users only who are familiar with the principles of an array of receivers and digital beamforming techniques! Computing the slowness power spectrum is only useful if all station coordinates are known and included in the data header. No correction to station elevation is made while computing the beam power and the plane wave assumption must be valid for the data.

This option is used to perform a delay and sum beamforming of a set of traces using the simple formula:

$$b(t) = s_{ref}(t) + \sum_{i=1}^{N-1} s_i(t + \vec{r}_i \hat{u})$$

Where $b(t)$ is the beam trace, $s_{ref}(t)$ represents the reference seismogram, s_i are the time shifted traces of the remaining array stations, r_i are the distances of the array stations regarding to the reference station, N is the total number of stations and u the horizontal slowness which is used to estimate the time shifts regarding to the reference station.

The beam power is computed by simply summing up the resulting beam trace per input time window and normalize it with respect to the number of array stations. Regarding to a grid search of different horizontal slowness values the result is plotted as a density map and the maximum slowness vector as well as the backazimuth of the beam are displayed.

After entering the channel list of array stations, the user is asked to input the reference channel. This action is equivalent to announce the reference station. Now the slowness grid in East-West and North-South direction must be specified. You are forced to input minimum, maximum and step size of the slowness grid in sec/deg (Figure 9-5).

Figure 9-5. Input panel for the slowness grid used for the delay and sum beamforming

```

Enter E-W slowmin,slowmax and slowstep in sec/deg:
..
Signal propagating to E -> positive slowness
Seperate items by commas!

```

Next PITSA will prompt for *n*th root processing the beamtrace according to the work of Muirhead and Datt (1976). The user can select either $n = 1$ (*unchanged*) up to $n = 4$ (Figure 9-6). This feature is useful to enhance small coherent signals in a seismogram with a strong and dominant signal (e.g. PcP and P phases in teleseismic signals).

Figure 9-6. *n*th root processing popup menu

Popup Menu
n-th root processing: Choose n:
n = 1 (unchanged)
n = 2
n = 3
n = 4
<ESCAPE>

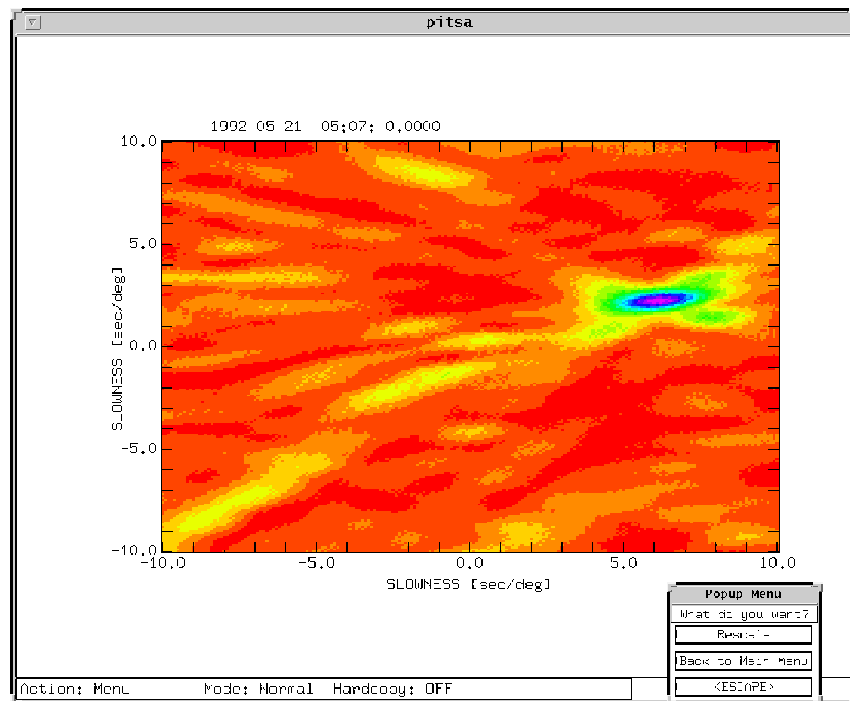
PITSA will output two ASCII files containing the slowness power values for the total slowness grid as well as the position of the maximum value and the backazimuth. The user has to specify the first file name in the input panel appearing next. The latter file name is formed automatically by adding the extension *max*.

PITSA will prompt now for *Normalize traces before beamforming*. Answering y (yes) will normalize all traces according to 1.0.

Before the resulting beam power distribution is displayed, the user can specify the minimum and maximum value for scaling (Figure 9-7). The beam power is plotted as density map either in a color or an gray scale plot according to the *COLOR_MODE* entry in the pitsa.cfg file (Figure 9-8).

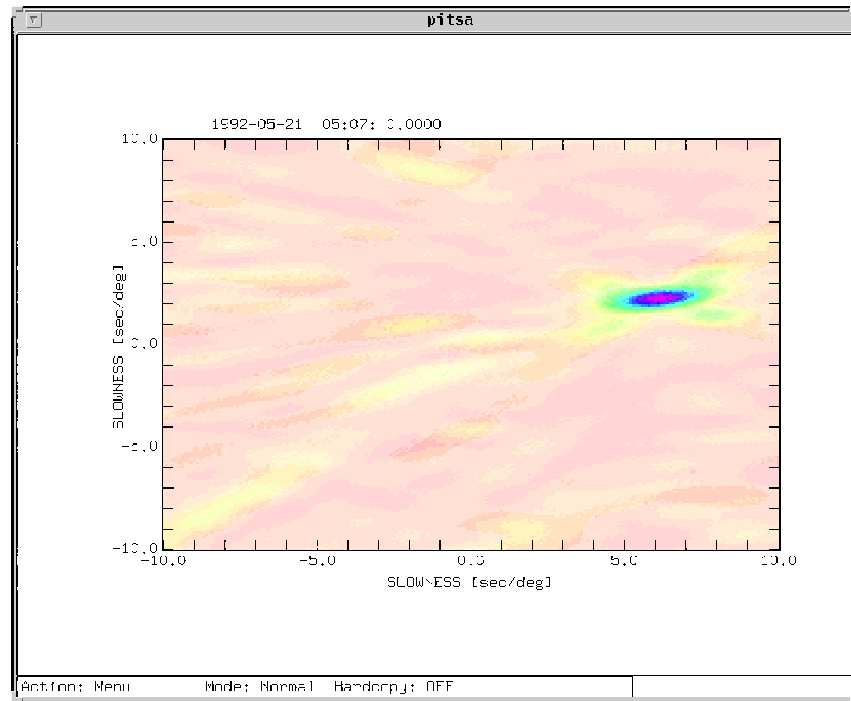
Figure 9-7. Input for scaling the result of slowness power spectrum

Enter Minimum and Maximum of the Scale;	
1.11533	4.74974
Trace_min: 1.11533 Trace_max: 4.74974	

Figure 9-8. Slowness power spectrum with respect to the given horizontal slowness grid. Also the *Rescale popup menu* is visible.

The *Rescale popup menu* enables you to change the minimum - maximum values for the beam power plot. If you are choosing *Back to Main Menu* and the *COLOR_MODE* in the *pitsa.cfg* file is set to *COLOR_SCALE*, PITSA will prompt next for *superimpose array response*. Here the user can either type *y* (yes) or *n* (no). The array response is calculated regarding to the horizontal slowness of maximum beam power. The reference trace is duplicated to the number of array stations. For each individual station the time delay of the location of the maximum beam power is computed and the traces are successively shifted. Using this synthetic seismograms the beam power distribution for the used slowness grid is computed. The result is overlaid as brightness of colors on the original beam power distribution (Figure 9-9).

Figure 9-9. Array response according to the horizontal slowness of the maximum beampower overlaid on the true power distribution



Finally the user is asked to *Continue (y/n)*. Typing *y* (yes) returns to the main screen of PITSA while *n* (no) just keeps the current screen. The position of the maximum beam power and the back azimuth are displayed in the lower most frame (Figure 9-10).

Figure 9-10. Continue to main window. Also the position of the maximum beam power, the absolute value of the slowness vector and the backazimuth are displayed

Continue y/n?						
y						
MAXIMUM:	slow_x	-6.1000	slow_y	-2.2000	slow	6.4846
	bazim	70.1679	power	13.0668		

Please be aware that the slowness distribution shown in the beam power plot is negative regarding to the true slowness vector announced in Figure 9-10!

9.5 Superimpose Traces

This plotting option can be used to compare two or more seismic traces. First the user will be asked for a channel list of traces to be plotted together in one box. After specifying the channel list, the user is asked to specify one trace to be plotted with drawing color 2 (see Table 4-1). The popup menu for specifying the trace is shown in (Figure 9-11).

Figure 9-11. Popup menu for specifying the colored trace

Enter index of trace to be plotted red:
~

The plotting option can only be performed on traces with equal length and equal sampling rate. However, spectra can also be plotted together in one box, if they have equal frequency resolution $\delta\omega$. As an example Figure 9-12 shows two noisy synthetic Ricker wavelets with different frequency content in one box and in Figure 9-13 the resulting amplitude spectra of these wavelets are plotted together.

Figure 9-12. Two seismic traces plotted in one box with ‘superimpose traces’.

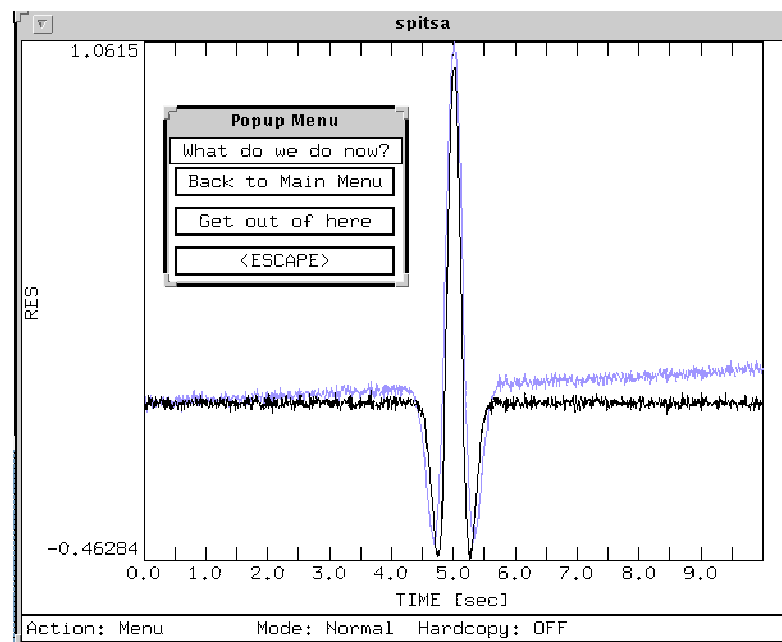
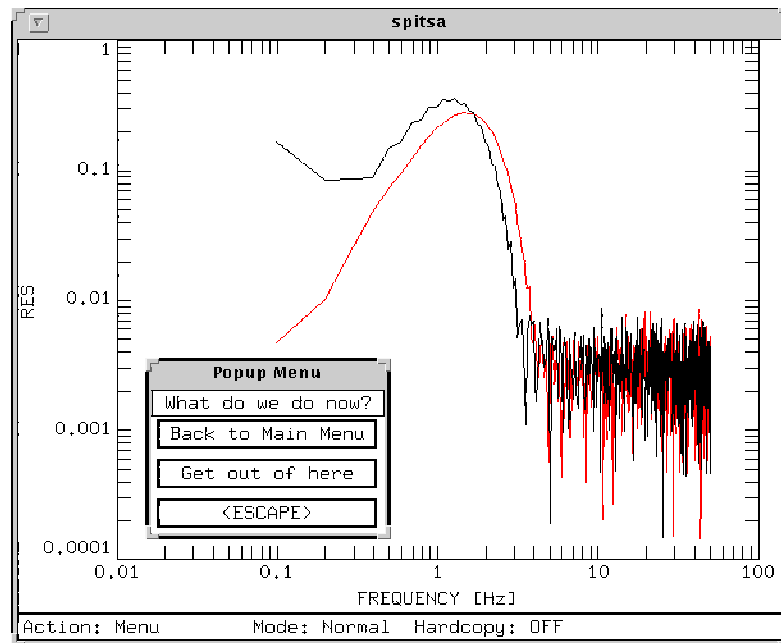


Figure 9-13. Superimposing amplitude spectra in one box.



-
- Backus, M. M., Water reverberations - their nature and elimination, *Geophysics*, **24**, 233-262, 1959.
- Baer, M. and U. Kradolfer, An automatic phase picker for local and teleseismic events, *Bull. Seismol. Soc. Am.*, **77**, 1437-1445, 1987.
- Bakun, W. H. and W. B. Joyner, The M_L scale in central California, *Bull. Seismol. Soc. Am.*, **74**, 1827-1843, 1984.
- Banfill, R., Description of SUDS format (exact title not known yet), 1992.
- Bratt, S. R. and T. C. Bache, Locating events with a sparse network of regional arrays, *Bull. Seismol. Soc. Am.*, **78**, 780-798, 1988.
- Brune, J. N., Tectonic stress and the spectra of seismic shear waves from earthquakes, *J. Geophys. Res.*, **75**, 4997-5009, 1970.
- Chui, C. K., An introduction to wavelets, *Wavelet analysis and its applications*, **1**, Academic Press Inc., Boston, San Diego, New York, Sydney, Tokyo, Toronto, 1992.
- Davies, D., E. J. Kelly, and J. R. Filson, Vespa process for analysis of seismic signals, *Nature Phys. Science*, 232, 8-13, 1971.
- Ehrenberg, J.E., and E. N. Hernandez, Covariance invariant digital filtering - A better digital processing technique for ground motion studies, *Bull. Seism. Soc. Am.*, **71**, 1361-1367, 1981.
- Gibson, Jerry D., *Principles of Digital and Analog Communications*, Second Edition, Macmillan Publishing Company, New York, 1993.
- Hamming, R. W., *Digital filters*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 257 pp., 1983.
- Ito, A., High resolution relative hypocenters of similar earthquakes by cross-spectral analysis method. *J. Phys. Earth*, **33**, 279-294, 1985.
- Kanamori, H., The energy release in great earthquakes, *J. Geophys. Res.*, **89**, 5719 - 5731, 1984.

- Kanasewich, E. R., Time sequence analysis in geophysics, The University of Alberta Press, Edmonton, 480 pp., 1981.
- Lee, W. H. K., and J. C. Lahr, HYPO71: A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes, U. S. Geol. Surv. Open-File Report, 100 pp., 1972.
- Lee, W. H. K., and J. C. Lahr, HYPO71 (REVISED): A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes, U. S. Geol. Surv. Open-File Report 75-311, 114 pp., 1975.
- Lee, W. H. K., and C. M. Valdés, HYPO71PC: IASPEI Software Library Vol. I (Ed. W. H. K. Lee), 203-236, 1989.
- Luzitano, R.D., Data report for dense array recordings from nine aftershocks of the July 21, 1986 earthquake in Chalfant Valley, California, U.S. Geological Survey Open-File Report 88-71, 50 pp., 1988.
- Montalbetti, J. F., and E. R. Kanasewich, Enhancement of teleseismic body wave phases with a polarization filter, *Geophys. J. R. astr. Soc.*, **21**, 119-129, 1970.
- Muirhead, K. J. and R. Datt, The n-th root process applied to seismic data, *Geophys. J. R. astr. Soc.*, **47**, 197-210, 1976.
- Poupinet, G., Ellsworth, W. L., Frechet, J., Monitoring velocity variations in the crust using earthquake doublets: An application to the Calaveras Fault, California, *J. Geophys. Res.*, **89**, 5719-5731, 1984.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 735 pp., 1988.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1992.
- Poupinet, G., W. L. Ellsworth and J. Frechet, Monitoring velocity variations in the crust using earthquake doublets: an application to the Calaveras fault, California, *J. Geophys. Res.*, **82**, 2981 - 2987, 1977.
- Richter, C. F., An instrumental earthquake magnitude scale, *Bull. Seismol. Soc. Am.*, **25**, 1-32, 1935.
- Seidl, D., The simulation problem for broad-band seismograms, *J. Geophys.*, **48**, 84-93, 1980.
- Scherbaum, F., Combined inversion for the three-dimensional Q structure and source parameters using microearthquake spectra, *J. Geophys. Res.*, **95**, 12,423-12,438, 1990.
- Scherbaum, F., Short Course on First Principles of Digital Signal Processing for Seismologists, in IASPEI Software library Vol 5, 1993.
- Scherbaum, F., and J. Wendler, Cross spectral analysis of Swabian Jura (SW Germany) three-component microearthquake recordings, *J. Geophys.*, **60**, 157-166, 1986.

-
- Scherbaum, F., and M. Wyss, Distribution of attenuation in the Kaoiki, Hawaii, source volume estimated by inversion of P wave spectra, *J. Geophys. Res.*, **95**, **12**, 439-12,448, 1990.
- Scherbaum, F. and H. Sato, Inversion of full seismogram envelopes based on the parabolic approximation: Estimation of Randomness and attenuation in southeast Honshu, Japan, *J. Geophys. Res.*, **96**, 2223-2232, 1991.
- Stearns, S. D., *Digital Signal Analysis*, R. Oldenbourg, München Wien, pp. 437, 1984.
- Stearns, S. D., and R. A David, *Signal processing algorithms*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1988.
- Tottingham, D. M., W. H. K. Lee, and J. A. Rogers, MDETECT: IASPEI Software Library Vol. I (Ed. W. H. K. Lee), 49 - 88, 1989.
- Tottingham, D. M., and W. H. K. Lee, XDETECT: : IASPEI Software Library Vol. I (Ed. W. H. K. Lee), 49 - 88, 1989.
- Vidale, J. E., Complex polarization analysis of particle motion, *B.S.S.A.*, **76**, 1393 - 1405, 1986.
- Ward, P. L., SUDS: Seismic Unified Data System, USGS Open-File Report 89-188, 1989.
- Wlech, P. D., The use of Fast Fourier Transform for the estimation of power spectra: a method based on time averaging over short modified periodograms, *IEEE Trans. Audio Electro - acoust.*, **AU-15**, 70 - 73, 1967.
- Willmore, P. L. (Ed.), *Manual of seismological observatory practice*, publ. by World Data Center A for Solid Earth Geophysics, U. S. Dept. of Commerce, Willmore, Boulder, CO 80303, USA, 1979.

The PITSA Configuration File

When PITSA is first started, it will look at the environment variable `PITSA_CONFIG_PATH_ENV` and use this as the path to find a special configuration file named `pitsa.cfg`. If the environment variable is not set, then PITSA will look in the directory where it was started for the configuration file.

The configuration file is a plain ASCII file that the user can edit to set preferences. Each entry in the file consists of two lines. The first line is the name of the parameter that is to be set and the second line is the value that it is being set to. Here is an example `pitsa.cfg` file:

```
SET_FOCUS
true
SET_FOCUS_LOOP
100
KILL_NOTE
false
MENU_POST
<Btn3Up>
MENU_SLCT
<Btn1Up>
TEXT_MARG
4
CANVAS_WIDTH
900
CANVAS_HEIGHT
550
DRAW_1_COLOR
black
DRAW_2_COLOR
red
BACKGROUND_COLOR
white
COLOR_MODE
COLOR_SCALE
MAX_NOCH
30
MAX_TRLEN
8000000
```

```
VELOCITY
6000.
DENSITY
2800
RADIATION_PATTERN
1
SURFACE_CORRECTION
2
SPECTRAL_ROLL_OFF_FACTOR
2.0
DEFAULT_CORNER_F
10
DEFAULT_Q
50000
RESONANCE_AMP
1.0
RESONANCE_FREQ
1.0
```

The parameters that can be set in the configuration file are as follows. The possible values for the parameter are shown in parenthesis after the parameter name.

B.1 General Parameters:

B.1.1 BACKGROUND_COLOR

(black, red, yellow, white or any valid X color)

This is the color that the PITSA window will use as the background color.

B.1.2 CANVAS_HEIGHT

(integer number 0-5000)

This is the height the PITSA window will have in units of pixels.

B.1.3 CANVAS_WIDTH

(integer number 0-5000)

This is the width the PITSA window will have in units of pixels.

B.1.4 DRAW_1_COLOR

(black, red, yellow, white or any valid X color)

This is the color that PITSA will use for drawing the main graphics and traces.

B.1.5 DRAW_2_COLOR

(black, red, yellow, white or any valid X color)

This is the color that PITSA will use for drawing highlighted graphics.

B.1.6 COLOR_MODE

(COLOR_SCALE, GRAY_SCALE)

This parameter controls whether the output of some routines is drawn in color or in grayscale (default). When setting to color mode the output is restricted to 156 colors only.

B.1.7 KILL_NOTE

(true or false)

This parameter controls whether the note window (the window that pops up during zooming, for example) is completely destroyed or just popped down. This value is normally false because it is faster to pop up and down the note window rather than destroy and re-create the note window each time it is needed. However, on some systems the note tends to pop up at unpredictable positions if it is just popped down. If this is happening and undesirable, then this value should be set to true.

B.1.8 MAX_NOCH

(integer number 10 or greater)

This is the maximum number of channels that PITSA will allow to be loaded at once. This can be set as high as desired.

B.1.9 MAX_TRLEN

(integer number 100 or greater)

This is the maximum number of points that PTISA will allow in a single trace. This can be set as high as desired.

B.1.10 MENU_POST

(<Btn1Down>, <Btn2Down> or <Btn3Down>)

This controls which mouse button will cause the main menu to be posted (popped up on the screen). This is normally set to <Btn2Down>, which means that pressing the second (or middle) mouse button will pop up the main menu.

B.1.11 MENU_SLCT

(<Btn1Down>, <Btn1Up>, <Btn2Down>, <Btn2Up>, <Btn2Down> or <Btn2UP>)

This controls which mouse button will cause a menu item to be selected. Normally MENU_POST is set to <Btn2Down> and MENU_SLCT is set to <Btn1Down>. This way, clicking the middle mouse button causes the main menu to popup. The user can then take their finger off of the mouse button and move the mouse pointer through the menu tree. Then, when the desired menu item is located, the user can

press the left menu button to make the selection. Another popular configuration is to have the menu pop when the left mouse button is pressed and the menu selection made when the left mouse button is released. This requires that the user keep the mouse button pressed while moving through the menu tree. To set up this configuration, `MENU_POST` would be set to `<Btn1Down>` and `MENU_SLCT` would be set to `<Btn1Up>`.

B.1.12 SET_FOCUS

(true or false)

This parameter tells PITSA whether or not it should try and set the keyboard focus on the text input window after it is popped up. On some systems, PITSA crashes intermittently when this is set to true. When PITSA is running on a SUN4 under OpenWindows, there is no problem and this value should be set to true.

B.1.13 SET_FOCUS_LOOP

(integer number 0-1000)

When PITSA is setting the focus on the text input window (meaning that key strokes will go to the text input window), it sometimes misses the window (the window appears after the focus is set). This value tells PITSA how long it should wait before trying to set the focus. Typically, a value of 100 works well. The larger the number, the longer PITSA will wait before setting the focus.

B.1.14 TEXT_MARG

(integer number 0-10)

This value determines how much space in pixels is allocated between text and box edges. On some systems fonts can act a little differently and not fit in the menu and text windows correctly. If the text looks crowded or if there is too much space between the text and box edges, this value can be adjusted. Normally this value is 4.

B.1.15 PICK_ADJ_WINDOW

(integer number)

This value determines the default width of the zooming box (in pixels) which is used for editing phase picks.

B.2 Automatic Phase Picker Configuration ***Parameters:***

B.2.1 P_DUR

(floating point value)

The maximum P wave amplitude is evaluated within a time window of P_DUR seconds following the determined P phase onset. 5 . 0 is the default value.

B.2.2 PRESETDUR

(floating point value)

$S(t)$, the variance of $SF(t)$, is evaluated within the first PRESETDUR seconds of the trace. It will, however, be updated as long as $CF(t)$ stays below THRSHL2 and the current time is less than 2*PRESETDUR. 0 . 5 is the default value.

B.2.3 TDOWNMAX

(floating point value)

$CF(t)$ may drop below THRSHL1 for less than TDOWNMAX seconds without causing the pick flag to be cleared. 0 . 1 is the default value.

B.2.4 TUPEVENT

(floating point value)

$CF(t)$ has to stay above THRSHL1 for at least TUPEVENT seconds for the pick to be accepted.

B.2.5 THRSHL1, THRSHL2

(floating point number)

These values determine the evaluation thresholds for $CF(t)$ and corresponds to the parameters S1 and S2 in Baer and Kradolfer (1987). 10 . 0 and 20 . 0 are the default values.

B.3 Configuration Parameters for GSE writing

B.3.1 GSE_1OR2

(GSE1, GSE2)

The valid entries for GSE_1OR2 are GSE1 or GSE2, indicating which main format of GSE is used for file writing.

B.3.2 GSE_FORM

(CMP6, INTV)

The entry GSE_FORM keeps the desired subformat for writing GSE datafiles and can be one of CMP6 or INTV for both GSE1 and GSE2. For the GSE2 main file format CMP6 is changed to CM6, and INTV to INT internally.

B.3.3 GSE_NDIFF

(integer value 0, 1, 2)

This entry has only effect when writing GSE1 datafiles in CMP6 subformat. It is an integer value 0 , 1 , or 2 and gives the number of differences that will be used for writing.

B.3.4 GSE_RESOL

(integer value 16, 32)

The entry GSE_RESOL specifies the resolution in bit which is used for converting float values into long value representation, which will be used when writing GSE datafiles. A typical value would be 16 . The internal float values in a trace are divided by the value $(\text{maxamp} - \text{minamp}) / (2^{\text{GSE_RESOL}})$ to get long values that can be written to a GSE datafile. As you can see this value is data dependent and for a low dynamic range in the data, the value will be very small. As a result please keep in mind, that under unfavorable circumstances an information loss can occur. This is due to the limited accuracy when writing this value to the appropriate entry in the GSE header!

Printer Configuration File

When the Hardcopy Mode is turned on in PITSA, the program will look in the directory defined by the environment variable `PITSA_PRINTDEF_PATH_ENV` for the file defined by `PITSA_PRINTDEF_NAME_ENV` to use for the printer definition file. The user will then be prompted for the printer definition file to use with the `PITSA_PRINTDEF_NAME_ENV` file name as the default.

The printer definition file tells PITSA how to generate output. The standard definition file that comes with PITSA is named `8X11_landscape.PS` and gives definitions for printing to PostScript printer on 8.5" by 11" paper in landscape mode. If the user desires to send output to another plotting device, such as a pen plotter, a definition file will have to be created. Below is the file `8X11_landscape.PS` with explanations of what each parameter controls:

C.1 Printer Configuration Parameters

C.1.1 #UPPER_LEFT_CORNER

`36,72`

This is the location of the upper left corner of the output in device units.

C.1.2 #UPPER_RIGHT_CORNER

`36,792`

This is the location of the upper right corner of the output in device units.

C.1.3 #LOWER_LEFT_CORNER

`576,72`

This is the location of the lower left corner of the output in device units. With these three parameters, the output can be placed at any location on the output device.

C.1.4 #PLOT_HEAD

```
%%!\n/m {moveto} def\n/l {lineto} def\n0 setgray\n0.1 setlinewidth\n/Courier findfont\n16 scalefont\nsetfont\nnewpath\n
```

This defines text that will be printed at the very start of the plot output. \n's are expanded to carriage returns in the output file. This line is written out to the plot output file with the C command:

```
fprintf(fp_plotfile,PLOT_HEAD);
```

C.1.5 #PLOT_TAIL

```
stroke showpage\n
```

This defines text that will be placed at the very end of the plot output file. Again, \n's will be expanded to carriage returns. The line is written out to the plot output file with the C command:

```
fprintf(fp_plotfile,PLOT_TAIL);
```

C.1.6 #PLOT_MOVETO

```
%5.1f %5.1f m\n
```

This is the format specifier that will be used to write out moveto commands. A moveto command instructs the output device to lift the pen and move it to the x y coordinates specified. The moveto command is sent to the plot output file with the C command:

```
fprintf(fp_plotfile,PLOT_MOVETO,x,y);
```

C.1.7 #PLOT_LINETO

```
%5.1f %5.1f l\n
```

This is the format specifier that will be used to write out lineto commands. A lineto command instructs the output device to move the pen (while down) to the x y coordinates specified. The lineto command is sent to the plot output file with the C command:

```
fprintf(fp_plotfile,PLOT_LINETO,x,y);
```

C.1.8 #PLOT_STROKE

```
stroke newpath\n
```

This is the format specifier that will be used when a stroke command is sent the output file. A stroke command instructs the plotting device to actually put ink on the paper for the pen path described by moveto and lineto commands. Some plotting devices will put ink on the paper while the path is being defined and this command is not needed (i.e. real plotters). The C command is:

```
fprintf(fp_plotfile,PLOT_STROKE);
```

C.1.9 #PLOT_MAX_LINE

```
1000
```

This value determines the maximum number of `lineto` commands allowed before a stroke command is given.

C.1.10 #PLOT_START_AXIS

```
stroke newpath 1 setlinewidth\n
```

This defines the text that will be sent to the plot output file just before plotting of the axis is started. This allows special values to be set such as line thickness or color that will only be used for the plotting of the axis. The C command is:

```
fprintf(fp_plotfile,PLOT_START_AXIS);
```

C.1.11 #PLOT_END_AXIS

```
stroke 0.1 setlinewidth newpath\n
```

This defines the text that will be sent to the plot output file just after plotting of the axis is ended so that any special effects installed by `PLOT_START_AXIS` can be turned off.

C.1.12 #PLOT_START_CHAR

```
stroke newpath 0.2 setlinewidth\n
```

This defines text that will be sent to the plot output file before character data is sent to the plot output file. Character data is only sent to the output file when a screen capture is made while the note is on the screen. The C command is:

```
fprintf(fp_plotfile,PLOT_START_CHAR);
```

C.1.13 #PLOT_END_CHAR

```
stroke 0.1 setlinewidth newpath\n
```

This defines text that will be sent to the plot output after character data has been sent to the output file. The C command is:

```
fprintf(fp_plotfile,PLOT_END_CHAR);
```

C.1.14 #PLOT_START_STRING

```
gsave\ncurrentpoint translate\n90 rotate\n(
```

This defines the text that will be sent to the plot output file just before a character string to be plotted is sent out. The C command is:

```
fprintf(fp_plotfile,PLOT_START_STRING);
```

C.1.15 #PLOT_END_STRING

```
)show\n grestore\n
```

This defines the text that will be sent to the plot output file just after a character string is sent to the plot output file. The C command is;

```
fprintf(fp_plotfile,PLOT_END_STRING);
```

C.1.16 #PLOT_STRING_HEIGHT_X

18

This defines the height in plotter units that character strings will have in the X direction.

C.1.17 #PLOT_STRING_HEIGHT_Y

0

This defines the height in plotter units that character strings will have in the Y direction.

C.1.18 #PLOT_COMMENT_START

%

Text to send to the plot output file at the start of a comment.

C.1.19 #PLOT_COMMENT_END

\n

Text to send to the plot output file at the end of a comment.

C.1.20 #PLOT_COLOR1

Text to send to the plot output file to select drawing color 1.

C.1.21 #PLOT_COLOR2

Text to send to the plot output file to select drawing color 2.

C.1.22 #PLOT_OUTPUT_FILE_NAME

plotout

The name to use for the plot output file. The user will have the option of changing this file name.

C.1.23 #PLOT_PRINT_COMMAND

```
cat %s %s %s | lpr -Pfastlw -h\n
```

Format for the system command to issue to send an output file directly to the printer. The C command is:

```
sprintf(command_str,PLOT_PRINT_COMMAND,temp1,temp2,temp3);  
system(command_str);
```

C.1.24 #PLOT_RM_COMMAND

```
rm %s\n
```

Format for the system command to remove a file. The C command is:

```
sprintf(command_str,PLOT_RM_COMMAND,file_name);  
system(command_str);
```

C.1.25 #PLOT_CAT_COMMAND

```
cat %s %s %s > %s\n
```

Format for the system command to concatenate three files together to generate the final output file. The C command is:

```
sprintf(command_str,PLOT_CAT_COMMAND,temp1,temp2,temp  
p3,out_file);  
system(command_str);
```

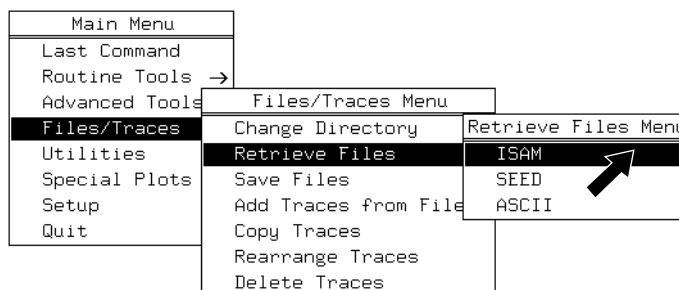

When the program mode is turned on, PITSA will store all commands to an output file. This command file will consist of a simple ASCII text file that can be edited by the user. Here is an example command file created by PTISA:

```
MAINMENU Retrieve_Files_Menu 0
GETSTRING datsun4
GETSTRING ;
SPECIAL Prog_Msg 0
ARROW 0.124444 0.314545 0.211111 0.201818
ARROW_END 0.0 0.0 0.0 0.0
A program message with an arrow.

ENDMSG
MAINMENU Integration_Type 2
GETSTRING ;
BOBMENU Accept/Append Traces
BOBMENU Use selection(s) for remaining traces
MAINMENU Setup_Menu 3
```

There are four different types of commands stored in these files. The first type is a main menu selection. Selections from the main menu are written out on one line starting with the key word MAINMENU followed by the name of the menu (with underscores replacing blanks) and the menu item selected where 0 is the first item in the menu panel. In the above example, the first command stored is from the main menu selection *Files/Traces -> Retrieve Files -> ISAM* as shown in Figure D-1.

Figure D-1. The main menu selection (*Files/Traces -> Retrieve Files -> ISAM*) that corresponds to the first line in the above example of a command file.



The second type of command stored is a pop-up menu selection. These selections are stored on one line starting with the key word `BOBMENU` followed by the name of the menu selection made. The third type is the user response to a text input window, which is written out as a single line starting with the keyword `GETSTRING` followed by the string the user entered.

Finally, PITSA also stores special actions such as pressing the <F5> key to pop up a message. For the menu message, several lines are written out. The first line is `SPECIAL Prog_Msg 0`. Next comes the location of any arrows. These lines start with the key word `ARROW` and have four numbers following them. The four numbers are the x, y start and end points of the arrow. The units for these numbers are fractions of the screen width and height so that if the window is resized, the arrows will appear in roughly the same location. Then comes the line that starts with the key word `ARROW_END` to indicate there are no more arrows. Then between this line and the line that starts with the key word `ENDMSG` the text for the message window appears.

With enough knowledge of the flow of PITSA, the user can fine tune command files without having to re-create them from scratch.

The GSE format for waveforms is an ascii file, consisting of a header section, a data section and an end mark and checksum value for verification purposes.

For GSE1.0 the header section consists of 2 lines and starts with keyword WID1, while for GSE2.x the header is only one line starting with keyword WID2. The header entries and their formatted (strict!) position in the appropriate lines are given in Table E-1 and Table E-2.

The data section start with the keyword DAT1 for GSE1.x and DAT2 for GSE2.x, respectively. According to the subformat specified in the header, the data is stored in 80 character long ascii lines for GSE1.x and in 132 character long ascii lines for GSE2.x. An example for each supported format is given in Figure E-1 to Figure E-4.

The end of the data section is indicated by the CHK1 or CHK2 label (GSE1.x / GSE2.x) followed by an long integer value containing the checksum as calculated by the GSE checksum algorithm.

The most detailed information about GSE format can be found online following the index link to GSE/CRP/243 at the Prototype International Data Center PIDC. the URL is: <http://www.cmr.gov> and is a online version of the Group of Scientific Experts (GSE) Third Technical Test (GSETT-3) Conference Room Paper 243 of 1995.

Table E-1. WID1 section of GSE1.x waveform files.

Position	Name	Format	Description
LINE 1:			
1-4	“WID1”	a4	Must be “WID1”
6-13	segment start date	i8	yyyddd e.g., “1984045”, year + day of year
15-16	start hours	i2	hh
18-19	start minutes	i2	mm
21-22	start seconds	i2	ss
24-26	start milliseconds	i3	nnn
28-35	number of samples	i8	nnnnnnnn
37-42	station	a6	use ISC code
44-51	channel id	a8	
53-54	channel	a2	e.g., sz, lz, ..., etc., null = “-”
56-66	sample rate	f11.7	samples per second
68-73	system type	a6	e.g., SRO, GS-13, etc., null = “-”
75-78	data format type	a4	“INTX”, “FLTX”, or “CMPZ” X is width of the data field or “V” for variable. “CMPZ” denotes compressed data utilizing Z bits per byte.
80	differencing flag	i1	0 or “”, 1, 2 for nil, first or second differences respectively.
LINE 2:			
81-89	calibration (gain)	f9.6	ground motion per digital count at calibration period, null = “0”
90	units of motion	a1	0 or blank: nm (displacement) 1: nm/s (velocity) 2: nm/s**2 (acceleration)
91-97	calibration period	f7.4	in seconds, null = “-1”
99-107	station latitude	f9.4	decimal degrees, north > 0, null = “-999”
109-117	station longitude	f9.4	decimal degrees, east > 0, null = “-999”
119-127	station elevation	f9.4	m above sea level > 0, null = “-999”
129-137	depth of sensor beneath surface	f9.4	m > 0, null = “-999”
139-145	beam azimuth	f7.2	degrees measured clockwise from north, null = “-1”
147-153	beam slowness	f7.2	seconds per degree, null = “-1”
155-160	horizontal orientation	f6.1	orientation of horizontal sensors, measured clockwise from north, null = “-1”

Table E-2. WID2 section of GSE2.x waveform format.

Position	Name	Format	Description
1-4	“WID2”	a4	Must be “WID2”
6-15	Date	i4,a1,i2,a1,i2	Date of the first sample (yyyy/mm/dd)
17-28	Time	i2,a1,i2,a1,f6.3	Time of the first sample (hh:mm:ss.sss)
30-34	Station	a5	Station Code
36-38	Channel	a3	FDSN channel code
40-43	Auxid	a3	Auxiliary identification code
45-47	Sub_format	a3	“INT”, “CMn”, or “AUx” “INT” is free format integers as ASCII characters; “CM” denotes compressed data, and n is either 6 (6-bit compression), or 8 (8-bit binary compression) “AU” signifies authentication and x is T (uncompressed binary integers), 6 (6-bit compression), or 8 (8-bit binary compression).
49-56	Samps	i8	Number of samples
58-68	Samprate	f11.6	Data sampling rate (Hz)
70-79	Calib	e10.2	Calibration factor; i.e. the ground motion in nanometers per digital count at calibration period (calper).
81-87	Calper	f7.3	Calibration reference period; i.e., the period in seconds at which calib is valid; calper should be near the flat part of the response curve (in most cases 1 sec).
89-94	Instype	a6	Instrument type
96-100	Hang	f5.1	Horizontal orientation of sensor, measured in positive degrees clockwise from North (-1.0 if vertical)
102-105	Vang	f4.1	Vertical orientation of sensor, measured in degrees from vertical (90.0 if horizontal).

Figure E-1. Example for GSE2.x waveform file. The subformat used was the CM6 (6-bit compressed).

```

WID2 1997/04/02 23:30:00.024 PD01 SHZ CM6 12000 40.000000 9.00e-03 1.0000 23900 0.0
DAT2
VU9IUGV3m6mPZ9X9II0RI8kR2kNUPUTTBGRPFV-HkE7N20kKkISURUF3UFkG-K1UEPV7mFn3YEWHIK1
6J-TVE3IEBUI0RQ9QS19UNkPI5UL+C6AU0I1A5QUHP0kILm6m4Z0X9IGiBkNUFkLkOVHHkEUIRLGIG1U
H-L8kFI2KBQHV .....
...
...
..... -IQkJUJ9TUIJV601UFB74N8V1JUGUMI1DGJUOm2mKY5VMm5l-kS2G15UGUSkHUM+l2UN1Q
D8+BUJ4+LHUH0kKUHLkMB3
CHK2 9977338

```

Figure E-2. Example for GSE2.x waveform file. the subformat used was the INT format.

[illegible]

Figure E-3. Example for GSE1.x waveform file The subformat used was the INTV (integer variable).

[illegible]

Figure E-4. Example for GSE1.x waveform file. The subformat used was the CMP6 (6-bit compression).

WID1 1998182 22 00 00 031 144000 RT7651 REFTEK32 0 40.0000000 NOTYPE CMP6 2
 1.0000001 1.0000 0.0000 0.0000 0.0000 -999.0000 -1.00 -1.00 -1.0
 DAT1
 jLtSkFrFJVEVFVEIEWGTl3o-m1cBlRnEVJSXRMMmDV3lL9kJmTXGR3aBn0IHVEp77W+ICUIIQWBYMmI2n
 3nEV7kIbCkPpRaSV+n+kQn6kQW9V6W-VSnAVEMs3OW4WLX3IEZOWlo+n7qD1UIJV5c0MR19r-VKUIOX6k
 RVr6o4UIWsn+kRY-n6VDX3mTkQlIWNl3QYnNpI1KIOXFo5V7a1pBVOV9p-Z8UlnKUIQICW2V7nRVGW1PI
 ...
 ...
 ...
 nyQXVhqlUwDU56lyTWrpNgRVfVdMmUCVdOw2UmIlXOXKVZRkr3VWHko2kyTIdkpGUrMUqKlB6UwCl-
 ktAUl-m5gPyBW+nPm-o2Up6pGla6Xg2miHlnHXaSmkEeGVXlOeUkHlmbUS6nct9VjKmoDVpAWY1npLUs
 KVqPIY0kxEWaNIZ9li3WoAlo2Up4wCtKVy7mrCUUsUz2le3VV-mBlLn-kkTm3e4VXJsOvFqkLrLuoBw9
 ...
 ...
 ...
 CHK1 127428090

GSE Instrument Response File

The format of the instrument response files to be used in connection with the Instrument Simulation tool in PITSA Section 6.3 is based on the format of the GSE calibration section:

```

LINE 1:
0 - 3:   a4 header ID (must be: CAL1)
5 - 10:  a6 station or array name
          (ISC code e.g. NB2)
12 - 19: a8 channel id (unique channel code)
21 - 22: a2 channel (sz, lz, ln, etc., null="-")
24 - 29: a6 system type
31 - 33: a3 response type "PAZ", "FAP", "FIR",
          or "FPZ", null="-"
35 - 40: 3i2 YYMMDD start date of validity
          for calibration
42 - 45: 2i2 HHMM start time
47 - 52: 3i2 YYMMDD end date of validity for
          calibration, blank if not known
54 - 57: 2i2 HHMM end time
58-79: a32 blanks reserved
if RESPONSE TYPE PAZ: response is given in poles and
zeroes
{
    LINE 2:
    0 - 7:   i8   number of poles (npole)
    LINE 3 - N: (N = npole + 2)
    0 - 7:   f,e  real part of pole
    8 - 15:  f,e  imaginary part of pole
    LINE N+1:
    0 - 7:   i8   number of zeroes (nzero)
    LINE N+2 - M: (M = N+1 + nzero)
    0 - 7:   f,e  real part of pole
    8 - 15:  f,e  imaginary part of pole
    LINE M+1:
    0 - 19: f,e  scale factor C to normalize to
              ground displacement in nm
    LINE > M+1: explanatory information as needed
}

```

```

else if RESPONSE TYPE FAP: response is given in frequency - amplitude - phase
{
    LINE 2:
    0 - 7:   i8    number of triplets (ntrip)
    LINE 3 - N: (N = ntrip + 2)
                3f, or e    triplets frequency,
                        amplitude, phase
    LINE > N: an explanatory information as needed
                (variable length)
}
else if RESPONSE TYPE FPZ: response is given using
    a combination of Finite Impulse Response (FIR)
    and Infinite Response (in PAZ) format
{
    LINE 2:
    0 - 7:   i8    number of filters (nfir)
    LINE 3:
    0 - 3:   a4    filter header
                        (must be of form FIR1 or PAZ)

    if FIR1:
    {
        5 - 15: f11.7 sampling rate
                        (input samples per second)
        17 - 24: i8    number of coefficients
        26 - 33: a8    filter ID, filter designator
                        code (e.g. GSD 240)
        LINE 4 - N: (N = ncoef + 3)
        0 - 15:  f,e filter coefficients
        J = N
    }
    else if PAZ:
    {
        LINE 4:
        0 - 7:   i8    number of poles (npole)
        LINE 5 - N: (N = npole + 4)
        0 - 7:   f,e real part of pole
        8 - 15:  f,e imaginary part of pole
        LINE N+1:
        0 - 7:   i8    number of zeroes (nzero)
        LINE N+2 - M: (M = N + 1 + nzero)
        0 - 15:  f,e real part of zero
        17 - 32: f,e imaginary part of zero
        LINE M+1:
        0 - 7:   f,e scale factor C to normalize
                        to ground displacement (nm/ct)
        J = M+1
    }
    LINE J+1:
    0 - 3:   a4    filter header
                        (must be of form FIR1 or PAZ1)
    if FIR1:
    {
        5 - 15: f11.7 sampling rate
                        (input samples per second)
        .... same as above ...
    }
}

```

```

    }
    else if PAZ:
    {
        LINE J+2:
        0 - 7:  i8  number of poles (npole)
        .... same as above ...
        additional explanatory information
        as needed
    }
}
else if RESPONSE TYPE FIR: response is given using
    Finite Impulse Response (FIR) filter
    coefficients format
{
    LINE 2:
    0 - 7:  i8  number of filters (nfir)
    LINE 3:
    0 - 3:  a4  filter header (must be FIR1)
    5 - 15: f11.7 sampling rate
                (input samples per second)
    17 - 24: i8  number of coefficients (ncoef)
    26 - 33: a8  filter ID, filter designator code
                (e.g. GSD 240)
    LINE 4 - N: (N = ncoef + 3)
    0 - 15:  f,e filter coefficients
    LINE N+1:
    0 - 3:  a4  filter header
                (must be of form FIR1 or PAZ1)
    5 - 15: f11.7 sampling rate
                (input samples per second)
    17 - 24: i8  number of coefficients
    26 - 33: a8  filter ID, filter designator code
                (e.g. GSD 240)
    LINES N + 2 - M: (M = ncoef + N + 1)
    0 - 15:  f,e filter coefficients
    LINES M = 1 - L:  additional filters as
                      required
    LINE L+1:  additional explanatory information
                as needed
}

```


Automatic Phase Picking

Automatic phase picking is done in PITSA using the algorithm of Baer and Kradolfer (1987).

This technique is based on a statistical evaluation of a function defined as $SF(t)$. This function is calculated from the envelope function of each trace which has been raised to the 4th power. For each trace, the variance $S(t)$ of $SF(t)$ is calculated within a certain time window of given length (PRESETDUR) which starts at the beginning of the trace. The function which is actually evaluated is:

$$CF(t) = (SF(t) - \overline{SF(t)}) / (S(t)) \quad (\text{equation (8) Baer and Kradolfer, 1987}).$$

A pick flag is set if $CF(t)$ increases above a given threshold (THRSHL1). However, as long as $CF(t)$ stays below a second threshold (THRSHL2) and the current time is less than 2*PRESETDUR, $S(t)$ and the variance of $SF(t)$ is updated.

A pick is accepted only if the pick flag is set for a certain time interval (TUPEVENT). However, since $CF(t)$ may not necessarily be a smooth function, it may drop below the given threshold (THRSHL1) for less than a certain duration (TDOWNMAX) without causing the pick flag to be cleared.

After a pick has been accepted, the maximum P wave amplitude (p_{amp}) is evaluated within a certain time period following the pick (P_DUR). The largest amplitude before the pick flag in the trace was set for the first time is considered to correspond to the noise amplitude (n_{amp}). The ratio $x_r = p_{amp}/n_{amp}$ is evaluated as a measure of reliability of the pick. On the basis of x_r , the HYPO71 phase descriptions are determined for each phase. The quality factor is taken to be 0 for $x_r > 8.0$, 1 for $x_r > 6.0$, 2 for $x_r > 4.0$, 3 for $x_r > 1.5$, and 4 for $x_r \leq 1.5$. Quality factors of 0 or 1 are labeled impulsive (I), while the others are labeled as emergent (E) onsets.

The first motion direction for each phase is also determined automatically and added to the phase description.

The performance of the phase picker depends strongly on the values of the tuning parameters PRESETDUR, TDOWNMAX, TUPEVENT, THRSHL1, THRSHL2, and P_DUR (Table G-1). These can be altered within the configuration file `pitsa.cfg` (see Appendix B).

Table G-1. Tuning parameters used by the automatic P-phase picker.

P_DUR	On default set to 5.0. This means the maximum P wave amplitude is evaluated within a time window of 5 seconds following the determined P phase onset.
PRESETDUR	On default set to .5. This means that $S(t)$, the variance of $SF(t)$ is evaluated within the first .5 seconds of the trace. However, it will be updated as long as $CF(t)$ stays below THRSHL2 and the current time is less than $2*PRESETDUR$.
TDOWNMAX	On default set to 0.1. This means that $CF(t)$ may drop below THRSHL1 for less than 0.1 seconds without causing the pick flag to be cleared.
TUPEVENT	On default set to 0.3. This means that $CF(t)$ has to stay above THRSHL1 for at least 0.3 seconds for the pick to be accepted.
THRSHL1, THRSHL2	These values determine the two evaluation thresholds for $CF(t)$. They correspond to the parameters S1 and S2 in Baer and Kradolfer (1987), and are set on default to 10 and 20, respectively.

Problem-Solving with HYPO71

It may be the case that HYPO71 will not run correctly. In this case, the user will need to understand a little bit about how PITSA sets up the HYPO71 run in order to find the problem. Problems can occur if a station that a phase has been picked on is not in the station file, for example. This section will outline the files that PITSA creates and the system command it issues to run HYPO71 so that the user can run HYPO71 by hand.

First PITSA generates a file named `input` in PITSA's current directory (the same directory that it looks for the station and phase file in). If the user does not have write permission for this directory, PITSA will not be able to run HYPO71. This small file will contain three lines, for example:

```
pitsahyp.inp
hypo_out.prt
hypo_out.pun
```

The first line is the name of the input file that HYPO71 will use which is also generated by PITSA in PITSA's current directory. This is a hard wired name and must be `pitsahyp.inp`. The second line is the name of the output file HYPO71 will create and is the HYPO71 output file name that the user provides PITSA with `.prt` appended. This is the file that HYPO71 will create if it runs successfully and is the file PITSA reads the new event information from. It will be created in PITSA's current directory. The third line is a file name that is needed by HYPO71 but is not used by PITSA. It is generated the same way the `.prt` file is generated except that `.pun` is appended instead of `.prt`. The `pitsahyp.inp` file is generated by appending together the control file, velocity model, station file, phase file and other required lines together. The first two lines are always:

```
HEAD                                HYPO71PC SETUP FOR PITSA VER 3.3
RESET TEST(06)=1.
```

Then a blank line followed by the station file (this is the station file that is generated in example):

```
CRPV3754.75N12154.36W331000
HPRV3657.19N12141.70W940000
JPLV3658.62N12149.92W158000
HG WV3701.02N12139.18W133000
JECV3703.04N12148.54W438000
```

```
HCRV3657.46N12135.04W241000
HAZV3653.08N12135.46W122000
JHLV3706.54N12149.98W908000
HJGV3647.88N12134.44W171000
HCAV3701.52N12129.04W332000
JALV3709.50N12150.82W244000
CADV3709.83N12137.56W207000
```

Followed by another blank line and the velocity model file:

```
4.00    0.0
5.90    3.5
6.80   15.0
8.05   25.0
```

Then another blank line and the control header file:

```
5. 25. 50. 1.78 2 1 18 1 1 11
```

And then the phase file (this is the phase file that is generated in example):

```
CRPVIPD0 891023002935.76
HPRVEPD0 891023002932.25
JPLVIPU0 891023002932.78
HGWVIPU0 891023002932.35
JECVEPD1 891023002932.78
HCRVEPU0 891023002933.01
HAZVIPD0 891023002933.04
JHLVIPD0 891023002933.44
HJGV?2 891023002933.40
HCAVIPU0 891023002934.43
JALVIPD0 891023002934.09
CADVEPU0 891023002934.47
```

After the input and pitsahyp.inp files have been created, PITSA then issues the following system command:

```
cd <PITSA's current path>; hypo71pc < input > output
```

If hypo71pc is not in the users path, then PITSA will not be able to run the HYPO71 program. The output of the PITSA program is redirected to the file output, which will contain any error messages HYPO71 generates. If HYPO71 does not run, then the user should take a look at this file. The user can run HYPO71 by hand by changing into the directory that contains the station and phase file and then typing hypo71pc. HYPO71 will ask the user for the three file names that are normally in the input file and then attempt to locate the event.

Example: Running HYPO71 from PITSA.

- First load in the isam1 file system:
- Change to the PITSA demodata directory and start PITSA.
- Select *Files/Traces* -> *Retrieve Files* -> *ISAM* and enter isam1 as the file name.
- Select all channels to be loaded in by entering ; as the channel list to load.

PITSA will now display 12 channels. These channels have already been picked, so all we have to do is create a station and phase file:

- Select *Routine Tools -> Phase Picking*.
- Select *Phase Picking Menu -> Output phases(s)*.
- Select *HYPO71* as the phase output format.
- Select *;* as the channel list.
- Enter *isam1.pha* for the name of the phase file.

After PITSA creates the phase output file, you should see the message in Figure 9-14 appear in the upper left hand corner of the screen. You must click in the button before you can continue on.

Figure 9-14. The message that appears after the phase file is created in example .

Total number of phases in new file =	12
Number of new phases used =	12
Number of new phases ignored =	0
Number of old phases used =	0
Number of old phases ignored =	0
Number of warnings =	0

CONTINUE

Now create a station file:

- Select *Phase Picking Menu -> Output station(s)*.
- Select *HYPO71* as the station output format.
- Enter *;* for the channel list.
- Enter *isam1.sta* for the name of the station output file.

Now you are ready to run HYPO71:

- Select *Phase Picking Menu -> Run location program*.
- Select *HYPO71* as the location program to run.
- Enter *velmod.hdr* for the velocity model (it will be the default).
- Enter *control.hdr* for the control header (it will also be the default).
- Enter *isam1.sta* for the station file.
- Enter *isam1.pha* for the phase file.
- Enter *hypo_out* for the output file (it will be the default).
- When HYPO71 finishes, select *Load in results*.

After the location results are read in, you will see the message in Figure 9-15 appear in the upper left hand corner of the screen. In order to continue on, you must click in the *CONTINUE* button. PITSA has now read in the new event location and created 12 synthetic phases that show the phase residuals. We can view the residuals by selecting *Phase Picking Menu -> Adjust phase(s)*:

Figure 9-15. The message PITSA displays after HYPO71 result are read in example .

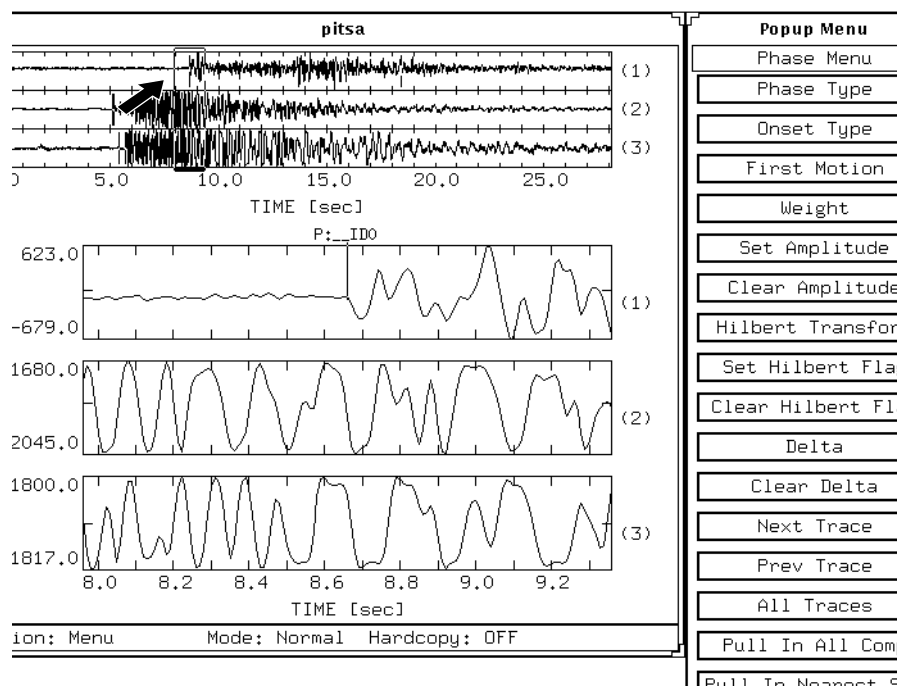
New Location:	36.95 [lat deg]	-121.74 [lon deg]	18.29 [depth km]
P residuals:	0.978 [ave sec]	9.760 [max sec]	12 [# residuals]
S residuals:	0.000 [ave sec]	0.000 [max sec]	0 [# residuals]
12 synthetic phases created			

CONTINUE

- Select *Phase Picking Menu* -> *Adjust phase(s)*.
- Enter ; for the channel list.
- Enter 3 as the number of channels to look at each time.
- Enter 3 as the number of channels to increment each pass.

Figure 9-16 shows PITSA on its first pass through the phase picking screen. Notice that the zoom window is centered around the first phase pick in the top trace, which is (1) in this case. This is how PITSA always aligns the zoom window when this screen first comes up. For trace (1), we only see the phase that was previously picked and we don't see any phase picks for the other two traces. This is because the residual phase for trace (1) is outside of the zoom box as well as the phases for traces (2) and (3). To see the phase picks and residuals of the other two traces, move the left edge of the zoom box by pressing the left mouse button while the mouse pointer is just outside and to the left of the zoom box. The pointer in figure Figure 9-16 shows where your mouse pointer should be when you first press down on the button. Then, while the mouse button is held down, move the mouse pointer to the left. As you do, the left edge of the zoom box will follow the cursor. When the left edge of the zoom box is to the left of the picks in trace (2), release the mouse button. When the mouse button is released, the traces in the lower part of the screen will be redrawn to reflect the new zoom box. Figure 9-17 shows the screen after the left edge of the zoom box has been moved.

Figure 9-16. The PITSA phase picking screen on the first pass in example .



We can see the synthetic phase for trace (1) by moving the right edge of the zoom box using the same method as before, except that we put the mouse pointer just outside on the right side of the zoom box; the pointer in Figure 9-17 shows where the mouse pointer should be before you press down the left mouse button. Figure 9-18 shows the screen after the right edge of the zoom box has been moved

Figure 9-17. The PITSA phase picking screen after the left edge of the zoom box has been adjusted in example .

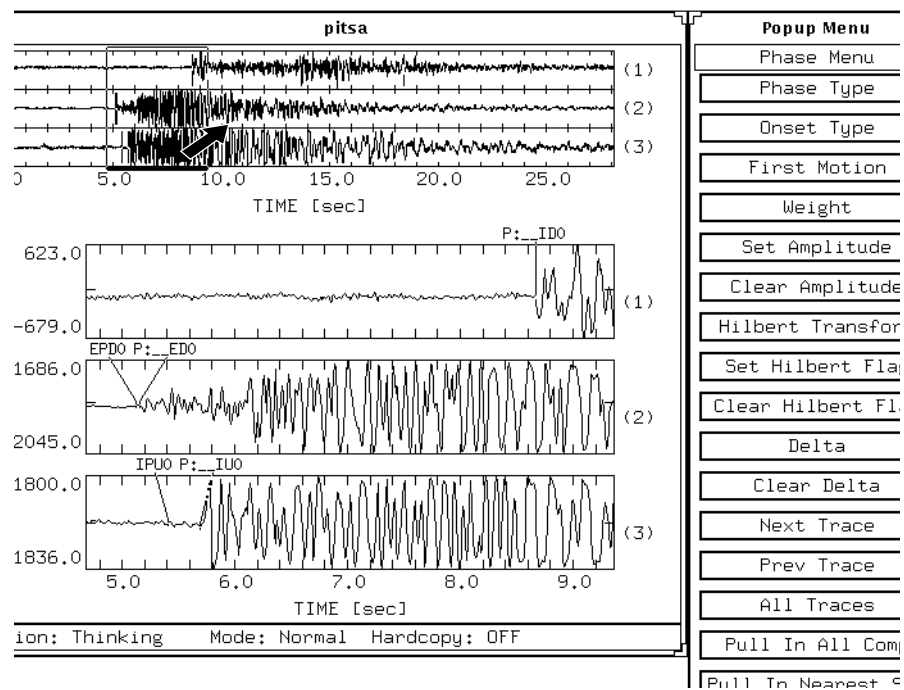
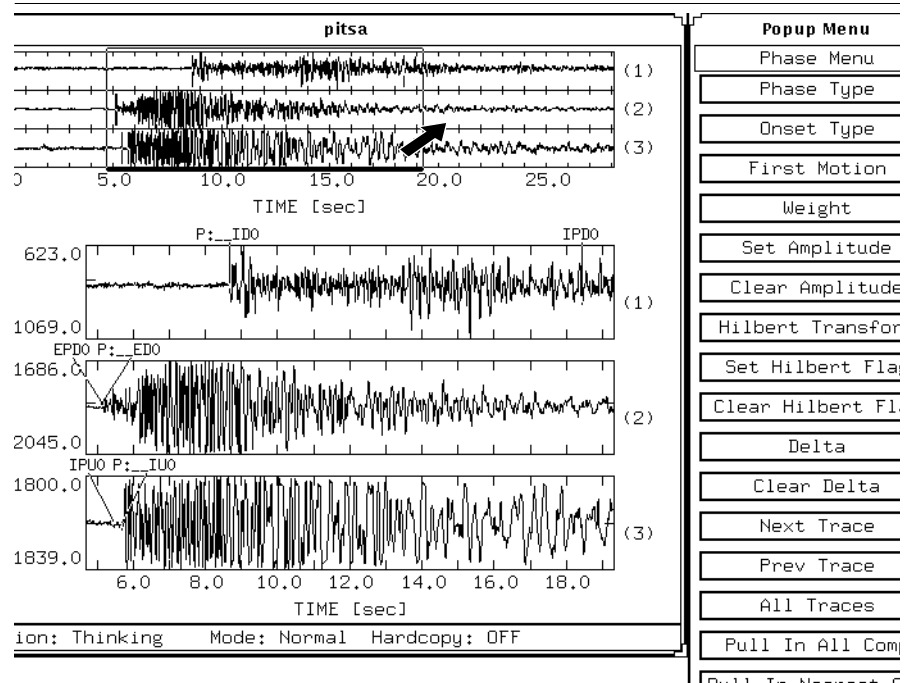


Figure 9-18. The PITSA phase picking screen after the right edge of the zoom box has been adjusted in example .



In Figure 9-18 we can see the phase and residual phases for all three traces. In trace (1), the phase that was used for the location (the previously picked phase) is P: __ID0 and the residual phase (a synthetic phase because it has no colon) is IPD0. Notice that the phase id for the synthetic phases is different from the phase id of the picked phase. This is because the residual phase uses the HYPO71 naming convention which places Impulsive or Emergent first, followed by either a P or S, then the first motion, and then the weight. The residual phases can be used as an aid

in adjusting the phases. Notice that you will not be able to adjust or edit the residual phases.

The residual for trace (1) is quite large indicating that there is something wrong with that station, either the station location or timing is off. It would probably be best to re-locate this event leaving out this station. To do this, you can just delete the phase pick, create a new phase file (entering 2 : 12 as the channel list) and then re-run HYPO71.

- To view the remaining traces, select the *Phase Menu* -> *DONE* selection. This will bring up the next three traces.
- Continue selecting *Phase Menu* -> *DONE* until all the phases have been viewed. This ends the example

Index

Symbols

1

A

A/D converter 1, 26, 28
Accepting new or processed traces 8
Acknowledgements 3
Adjust phases 14
 controlling zoom 14
 phase menu 22
 all traces 28
 clear amplitude 25
 clear delta 26
 clear Hilbert flag 27
 clear pulled in 28
 delta 25
 done 29
 first motion 22
 Hilbert transform 26
 next trace 28
 onset type 22
 phase type 22
 prev trace 28
 pull in all comp 28
 pull in nearest sta 28
 refresh 28
 set amplitude 23
 set Hilbert flag 27
 weight 23
 picking phases 16
Advanced tools
 coherence spectrum 47
 convolution 26
 correlation 25
 cross spectrum 46
 envelope 37
 FFT 12

filter 2
Hilbert transform 37
inverse FFT 15
maximum entropy spectrum 17
particle motion 28
polarization filter 42
rotate components 38
spectrum 11
and 9
AR process 17
ARMA coefficients 9
ASCII format 2, 2
 header line 7
 retrieve files 5
 save files 9
Automatic processing 4
Autoscale traces 13
Average 43
Axis type 14, 16, 46, 9, 14
Azimuth 39, 41, 14, 17

B

Band pass filter 2, 6, 7
Baseline correction 42
 linear regression 44
 linear trend 43
 offset 43
 running average 43
Beam forming 15
Body wave magnitude 34
Brune source 5
Butterworth filter
 Band Pass 2
 High Pass 5
 Low Pass 4

C

Changing directories 3
Chirp signal 7
Clear
 all phases 32
 amplitude 25
 phases 32
 select phases 32
Coherence spectrum 47

- Color settings 1
- Command files 3
- Constant functions 4
- Constants 4
- Convolution 26, 26, 27
- Copy traces 10
- Correlation 25
- Cross spectrum 46
- .cshrc 1

D

- Data trace 1
- Deconvolution 26, 27
- Delete traces 12
- Delta phase 25
- De-ringing filter 5
- Dialog boxes
 - clearing quickly 1
 - keyboard focus 6
 - text input 5
- Dialog menus 7
- Difference
 - equation 2
 - equation filter 8
- Differentiate 41
- Digital resampling 25
- Digitization 22
- Discretize signal 23
- Distance plot 3
- Double channel utilities 11

E

- Edit 6
 - despiking 9
 - glitch editing 8
 - interpolate 6, 8
 - point editing 6
- Envelope 37
- Environment variables
 - PITSA_CONFIG_PATH_ENV 2
 - PITSA_HYPO71_PATH_ENV 2, 31
 - PITSA_PRINTDEF_NAME_ENV 2, 7
 - PITSA_PRINTDEF_PATH_ENV 2, 7
- Adjust phases
 - phase menu 29
- Event
 - access 16
- Exit 9

F

- FAP 10
- FFT 12
 - demultiplex from FFT format 17
 - inverse 15
 - multiplex to FFT format 16
- File I/O 1
- File types 3
- Files and traces
 - retrieve files 3
 - save files 8
- Filter 2
 - band pass 2, 6
 - de-ringing 5
 - difference equation 8
 - gaussian band pass 6

- high pass 5
- low pass 4
- notch 7

- FIR 10

- First motion 22

- Flags
 - plotall 10

- Font size 2

G

- Generic Wood Anderson magnitude 34
- GSE 2
- GSE format 10
- GSE Reading 8

H

- Hardcopy
 - screendump files 7
 - status line 9
- Header access 1, 15
 - Event header 16
 - output information 18
 - Plot header 17
 - Record header 17
 - Station header 15
- High pass filter 5
- Hilbert
 - flag 27
 - transform 26
- Hilbert transform 37
- History function 9
- HYP071 1
 - control.hdr 31
 - debugging 32
 - example 2
 - phase labels 20
 - phase output status message 30
 - processing 30
 - running 31
 - synthetic phases 31
 - weight 23
- HYP071 phase id convention 5

I

- Incidence angle 41
- Installation
 - user 1
- Instrument correction 9
 - GSE format 10
 - response files 10
- Integrate
 - Tick's rule 40
 - trapezoidal rule 40
- Inverse FFT 15
- IRIS 4
- ISAM format 2, 2
 - example 4
 - retrieve files 3
 - save files 9

L

- Last command 9
- Linear
 - regression 44
 - regression removal 44
 - trend removal 43
- Local magnitude 34

Location
 running HYPO71 32
Location programs 31

M

Magnitudes 34, 34
Maximum entropy spectrum 17
Menu
 main menu item selection 4
Mixed signal 5
Moment magnitude 34, 37, 40
Multiplex format 16, 17

N

Next trace 28
Noisy traces 2
Notch filter 2, 7
Numbered postscript files 7

O

Offset removal 43
Onset type 22
Output phases 29

P

Particle motion 28
 2D snake 29
 3D snake 33
Path to data files 3
PAZ format 10
Personal note
 by Frank Scherbaum 4
 by James Johnson 4
Phase error 25
Phase id's 22
Phase labels 20
Phase picking 12
 adjusting and editing phases 14
 auto picker parameters 1
 automatic phase picking 13
 clear phases 32
 clearing all phases 32
 clearing select phases 32
 DONE 33
 HYPO71 30
 output phases 29
 output stations 31
 particle motion 33, 36
 phase labels 20
 plot all 33
 run location program 31
 sort by distance 29
PITSA.CFG 2
 P Picker 1
 phase picking 1
Plain ASCII files 5
Plain Sum 15
Plot access 17
Plot all 33
Plot all flag 10
Polarization filter 42
Pole zero file
 filter 8
Popup menus 7
Prev trace 28
Processed 8
Program mode 3

 accepting default 6
 input files 4
 input modifiers 6

Q

Quit 9

R

Reading data
 ASCII files 5
 ISAM format 3
Rearrange traces 11
Record access 17
Rectilinearity 43
Resampling 1, 22, 22, 25
Resizing PITSA 9
Retrieve file 3
Ricker wavelet 6
Rotate components 38
Routine tools
 adjust phase(s) 14
 baseline corrections 42
 differentiate 41
 edit 6
 integrate 40
 magnitudes 34
 phase picking 12
RPN equation 8
Running average 43

S

Sampling rate 2, 17
Save files 8
 ASCII format 9
 ISAM format 9
Scale traces together 12
Section plots 2
Seismometer signal 7
Set amplitude 23
Setup
 hardcopy mode 2
 program/track mode 2
Simulate A/D Converter 26
Sine/cos 3
Single channel utilities 10
Sinusoidal signals 3
Snake
 Particle motion 28
Sort traces by distance 29
Spectrum 11
Spikes 3
Stack traces 14
Station
 access 15
Statistics 43
Status line 8
Surface wave magnitude 34

T

Taper functions 12
Test signals 1
 noisy traces 2
Text 5
Text input 5
trace 7
Traces
 accepting new 8

- adding traces from ISAM file
 - systems 9
- autoscale traces 13
- copy traces 10
- delete traces 12
- input for channel lists 7
- processing large data sets 3
- rearrange traces 11
- scale traces together 12
- scaling traces 9
- utilities 1, 9
- Track mode 3
 - accepting default 6
 - input files 4
 - input modifiers 6
- Triple channel utilities 12
- U**
 - Unify Traces 15
 - University NAVSTAR Consortium 4
- W**
 - Waterlevel 27
 - Whittaker reconstruction 24
- Z**
 - Zoom 2
 - accept zoom window 4
 - adjust zoom window 4
 - index selection 5
 - tapered 4
 - Zoom box 14