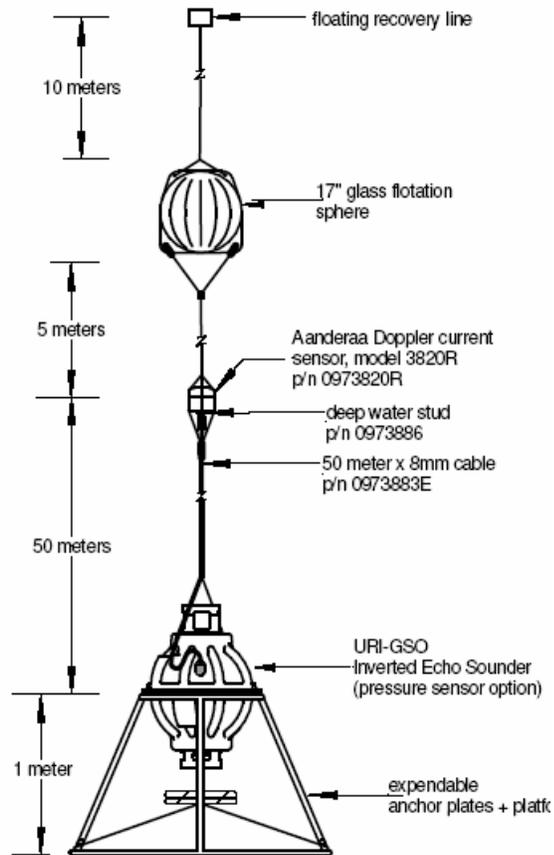


Graduate School of Oceanography
University of Rhode Island
Narragansett, Rhode Island

Inverted Echo Sounder Data Processing Manual

GSO Technical Report No. 2007- 02



By
Maureen Kennelly, Karen Tracey, and D. Randolph Watts

June 2007

Abstract

The Inverted Echo Sounder (IES) is an ocean bottom-moored instrument that measures the vertical acoustic travel time (VATT) round-trip from the sea floor to the sea surface and back. The VATT varies principally due to changes in the temperature profile of the water column, making the IES well-suited for monitoring changes in temperature structure and dynamic height (baroclinic signal). Currently, the Model 6.2, a combined IES, data-logger, and acoustic release, with optional measurements of bottom pressure, temperature and current speed and direction (with attached Aanderaa™ Doppler current sensor) is produced at URI/GSO. Data are processed in situ and are available (optional) remotely by an acoustic telemetry link or expendable, satellite-link data shuttle. In addition to the IES-measured baroclinic signals, barotropic near-bottom pressure variations may be measured with the optional pressure sensor. A report was written in 1991 describing IES data processing (Fields et al., 1991). Since that report, significant improvements have been made to both IES hardware and software, warranting an update of the IES data processing. This report will document standard processing steps currently carried out for IES Models 6.1 and 6.2 at URI/GSO. A separate document, Inverted Echo Sounder User's Manual, IES Model 6.2, describes the IES hardware and instrument configuration.

Table of Contents

Abstract..... 1

List of Figures..... 4

List of Tables 7

List of Acronyms..... 7

Chapter 1. Introduction..... 9

1.1 Prior to Processing - Setting up the Directory Structure..... 9

Chapter 2. Overview of IES Processing..... 12

Chapter 3. Step by Step Processing..... 16

3.1 Introduction..... 16

3.2 IES_GUIDAT2MAT..... 16

3.3 IES_GUIFREQ2MAT 22

3.4 InitGuiDriver.dat - Parameter File Setup 25

3.5 IES_GUIDRIVER..... 27

3.5.1 Step 1: Windowing and Hourly Values..... 29

3.5.3 Step 3: Initial Detiding..... 46

3.5.4 Step 4: Dedrifting..... 51

3.5.5 Step 5: Final Detiding 57

3.5.6 Step 6: Velocity Processing 59

3.5.7 Step 7: Lowpass filtering..... 66

3.5.8 Step 8: Plotting 69

Chapter 4. IES_GUIDRIVER Special Cases 72

4.1 No Travel Time Data 72

References..... 73

Acknowledgments 73

Appendix A Example plots generated by IES_GUIDAT2MAT for PIES..... 74

Appendix B Examples of additional plots generated by IES_GUIDAT2MAT for currents (CPIES only) 81

Appendix C Response Analysis of Tides..... 83

C.1 Theory 83

C.2 Tidal Prediction Options 85

List of Figures

Figure 1 Recommended directory tree structure.....	10
Figure 2 Default IES_GUIDAT2MAT window	12
Figure 3 Default IES_GUIDRIVER window	13
Figure 4 IES Data Processing Flowchart	15
Figure 5 Example IES_GUIDAT2MAT window- READY State.....	17
Figure 6 Example IES_GUIDAT2MAT window - FINISHED State	18
Figure 7 Example TXXX_#.mat file contents	19
Figure 8 Example PXXX_#.mat file contents	19
Figure 9 Example EXXX_#.mat file contents	19
Figure 10 Example FXXX_#.mat file contents	20
Figure 11 Example CXXX_#.mat file contents (CPIES only)	20
Figure 12 Default IES_GUIFREQ2MAT window	23
Figure 13 Example IES_GUIFREQ2MAT window after coefficients loaded	24
Figure 14 Example InitGuiDriver.dat file.....	26
Figure 15 Example IES_GUIDRIVER window after initialization for processing ALL steps.....	29
Figure 16 Example IES_GUIDRIVER window after initialization for processing Step 1 only	31
Figure 17 Figure window generated by Step 1 with full range of tau measurements	32
Figure 18 Example IES_GUIDAULIMITS window	32
Figure 19 Example IES_GUIDAULIMITS window with refined tau bounds	33
Figure 20 Example IES_GUIDAULIMITS window with reduced Y-axis range	34
Figure 21 Figure window generated by Step 1 with full range of pressure measurements (top) and expanded view about mean value (bottom).....	35
Figure 22 Figure window generated by Step 1 showing full pressure time series after initial windowing	36
Figure 23 Figure window generated by Step 1 with full range of temperature measurements (top) and expanded view about mean value (bottom).....	37
Figure 24 Figure window generated by Step1 showing full temperature time series after initial windowing	37
Figure 25 Figure window generated by Step 1 showing beginning of temperature time series with points to omit (12 hours after launch) shown in red	38
Figure 26 Figure window generated by Step 1 showing beginning of temperature time series with user selected points to omit shown in red.....	39
Figure 27 Example Site1.mat file contents after Step 1, windowing and hourly values ..	40
Figure 28 Example processing log after Step 1, windowing and hourly values.....	40
Figure 29 Example IES_GUIDRIVER window after initialization for processing Step 2 only	41
Figure 30 Example IES_GUIDESPIKE window showing data before despiking (top)...	42
Figure 31 Example IES_GUIDESPIKE window showing data before (top) and after despiking (bottom).....	43
Figure 32 Example IES_GUIDESPIKE window showing expanded view for hand editing	44
Figure 33 Example IES_GUIDESPIKE window showing data before (top) and after hand editing (bottom)	44

Figure 34 Example Site1.mat file contents after Step 2, despiking	45
Figure 35 Example processing log after Step 2, despiking.....	45
Figure 36 Example IES_GUIDRIVER window after initialization for processing Step 3, initial detiding only	48
Figure 37 Example Site1.mat file contents after Step 3, detiding	49
Figure 38 Example Site1.mat tidal phases and amplitudes.....	49
Figure 39 Example processing log after Step 3, initial detiding.....	50
Figure 40 Example IES_GUIDRIVER window after initialization for processing Step 4, dedrifting only.....	52
Figure 41 Example IES_GUIDEDRIFT2EXP window prior to fitting drift.....	53
Figure 42 Example IES_GUIDEDRIFT2EXP window showing data prior to fitting drift (left) and after drift removed (right)	54
Figure 43 Example IES_GUIDEDRIFT2EXP window showing expanded view of data prior to fitting drift (left) and after drift removed (right).....	55
Figure 44 Example Site1.mat file contents after Step 4, dedrifting.....	56
Figure 45 Example processing log after Step 4, dedrifting	56
Figure 46 Example IES_GUIDRIVER window after initialization for processing Step 5, final detiding only	57
Figure 47 Example Site1.mat file contents after Step 5, final detiding	58
Figure 48 Example processing log after Step 5, final detiding.....	58
Figure 49 Example IES_GUIDRIVER window after initialization for processing Step 6, Velocity Processing Only	59
Figure 50 Figure windows generated by velocity processing showing beginning of DCS temperature time series with points to omit shown in red	60
Figure 51 Example of IES_GUIDESPIKE CM window showing data before despiking (top).....	61
Figure 52 Example IES_GUIDESPIKE CM window showing U component data before (top) and after despiking (bottom)	61
Figure 53 Example IES_GUIDESPIKE window showing V component data before (top) and after despiking (bottom).....	62
Figure 54 Example of IES_GUIDESPIKE CM window showing expanded view for V component hand editing.....	63
Figure 55 Example of IES_GUIDESPIKE CM window showing DCS temperature data before despiking (top).....	64
Figure 56 Example IES_GUIDESPIKE CM window showing DCS temperature data before (top) and after despiking (bottom).....	64
Figure 57 Example of Site2.mat file contents after step 6, velocity processing.....	65
Figure 58 Example processing log after Step 6, velocity processing	65
Figure 59 Example IES_GUIDRIVER window after initialization for processing Step 7, lowpass filtering only.....	67
Figure 60 Example Site1lp.mat file contents for a PIES after Step 7, lowpass filtering ..	68
Figure 61 Example Site2lp.mat file for a CPIES after Step 7, lowpass filtering.....	68
Figure 62 Example processing log after Step 7, lowpass filtering	68
Figure 63 Example IES_GUIDRIVER window after initialization for processing Step 8, plotting only	69

Figure 64 Example Figure 1 window showing the hourly tau, pressure and temperature time series for a PIES.....	70
Figure 65 Same as Figure 64 with addition of u, v and DCS temperature for a CPIES ...	70
Figure 66 Example Figure 2 window showing the despiked pressure, tides, detided pressure with drift and dedrifted pressure time series	71
Figure 67 Example Figure 3 window showing the lowpass filtered, subsampled tau, pressure and temperature time series for a PIES	71
Figure 68 Same as Figure 67 with addition of u, v and DCS temperature for a CPIES ...	72
Figure 69 IES_GUIDAT2MAT default Figure 1, Travel Time (seconds)	74
Figure 70 IES_GUIDAT2MAT Figure 1, Travel Time (seconds) with reduced y-axis limits	74
Figure 71 IES_GUIDAT2MAT default Figure 2, Pressure (10kPa)	75
Figure 72 IES_GUIDAT2MAT Figure 2, Pressure (10kPa) with reduced y-axis limits..	75
Figure 73 IES_GUIDAT2MAT default Figure 3, Temperature (degrees C)	76
Figure 74 IES_GUIDAT2MAT Figure 3, Temperature (degrees C) with reduced y-axis limits	76
Figure 75 IES_GUIDAT2MAT default Figure 4, Engineering measurements	77
Figure 76 IES_GUIDAT2MAT default Figure 5, Engineering measurements	77
Figure 77 IES_GUIDAT2MAT default Figure 6, Engineering measurements	78
Figure 78 IES_GUIDAT2MAT default Figure 7, Temperature Frequency (Hz).....	78
Figure 79 IES_GUIDAT2MAT Figure 7, Temperature Frequency (Hz) with reduced y-axis limits	79
Figure 80 IES_GUIDAT2MAT default Figure 8, Pressure Frequency (Hz).....	79
Figure 81 IES_GUIDAT2MAT Figure 8, Pressure Frequency (Hz) with reduced y-axis limits	80
Figure 82 IES_GUIDAT2MAT default Figure 4 for CPIES	81
Figure 83 IES_GUIDAT2MAT Figure 4 with reduced y-axis limits for CPIES	82
Figure 84 IES_GUIDAT2MAT default Figure 5 for CPIES.....	82
Figure 85 Spherical harmonics corresponding to tidal amplitude functions	85
Figure 86 Description of contents of Tidal Prediction Option Files.....	87

List of Tables

Table 1 Raw data files downloaded from IES	11
Table 2 Data files generated by IES_GUIDAT2MAT	18
Table 3 Record Length for Tidal Prediction Options	47
Table 4 Tidal Constituents Provided by Default Processing	50

List of Acronyms

CPIES	IES with optional current meter and pressure sensor
DCS	Doppler Current Sensor
IES	Inverted Echo Sounder
GSO	Graduate School of Oceanography
PDT	Pulse Delay Telemetry
PIES	IES with optional pressure sensor
URI	University of Rhode Island
UT	Universal Time
VATT	Vertical Acoustic Travel Time

Chapter 1. Introduction

The Inverted Echo Sounder is an ocean bottom-moored instrument that measures the vertical acoustic travel time round-trip from the sea floor to the sea surface and back. Bottom pressure, temperature and current speed and direction are optional measured parameters. IESs are configured to emit twenty-four 12 KHz pings per hour, in programmable bursts of 4, 8, 12 or 24 pings. The time required for each ping to reach the surface and return is recorded within the instrument on removable (type ATA compact flash) memory cards (15-512 Mbyte capacity). Data are stored in ASCII files as described in the IES User's Manual.

This report documents the processing steps carried out after data have been downloaded from a recovered instrument. IES data can be processed on a variety of platforms running either Windows or UNIX operating systems with a processing package developed at URI/GSO. The current version of the processing software for Models 6.1 and 6.2 IES/PIES/CPIES instruments is IESpkg3 which uses MATLAB Version 7. The IES processing codes have been tested with Windows XP and Redhat Linux 9 operating systems. The MATLAB signal processing toolbox is required to run the processing programs.

1.1 Prior to Processing - Setting up the Directory Structure

To protect the integrity of raw data files, the following directory structure is recommended. Create a directory for the experiment or study. Then create subdirectories for raw data and processed data. Within the raw data directory, create subdirectories for each instrument. An example directory tree is shown in Figure 1 for an experiment called "GOM". Within directory "GOM" the subdirectories for raw and processed data are "raw_data" and "mat_data". In "raw_data" a subdirectory has been made for IES serial number "076" with site designator "P1" (*i.e.*, P1_076). Do not include spaces in the directory or site designator names. Spaces in directory names will cause MATLAB to fail. Site designators are usually assigned prior to launch by the Principal Investigator. The processing code will prepend the word "PIES" to any site designator entered as

numeric only. Serial numbers are assigned by URI/GSO at the time of manufacture. Data from PIES serial number 076 will be used throughout this manual to illustrate processing steps.

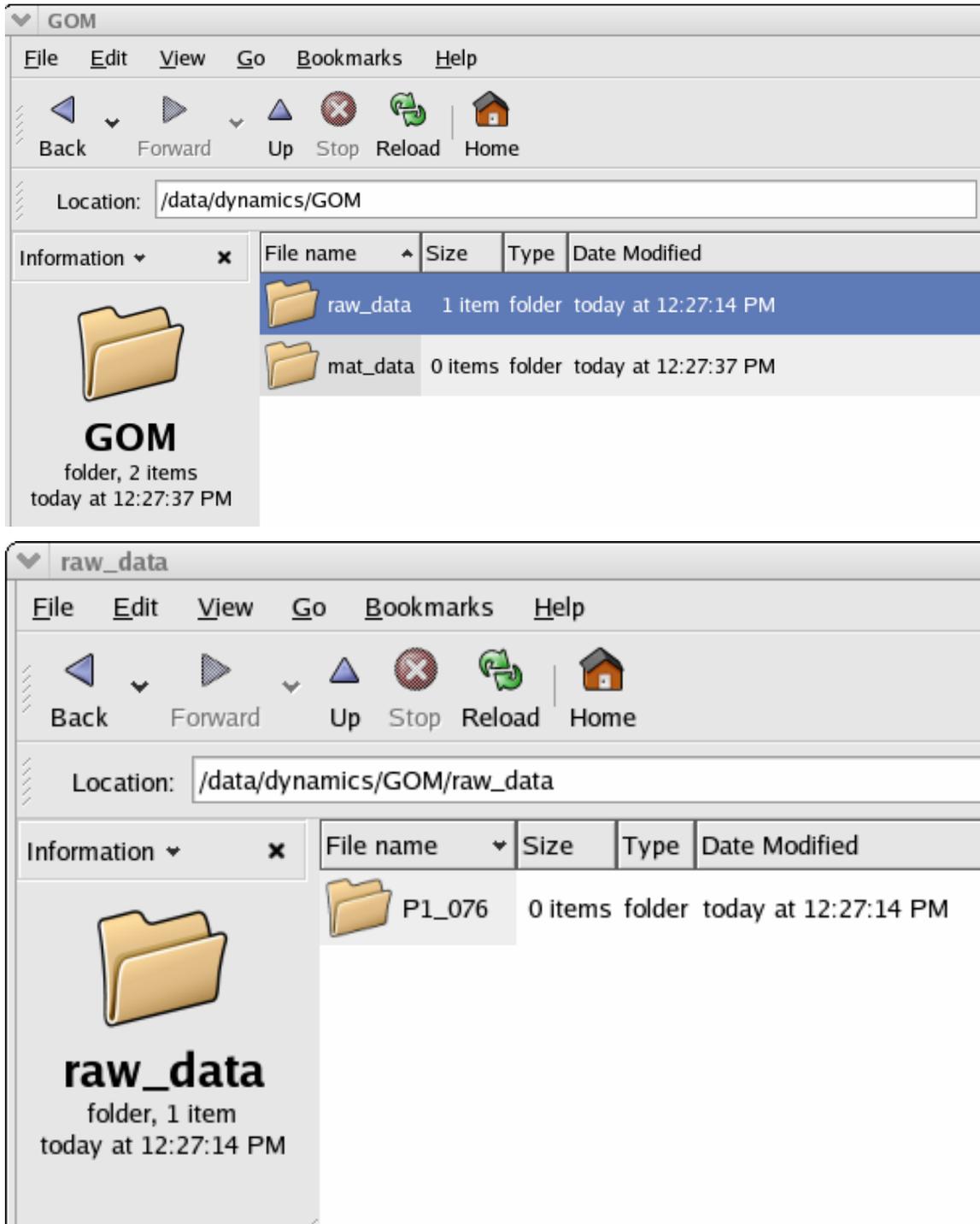


Figure 1 Recommended directory tree structure

After an IES is recovered at sea, raw data from the flash memory card are downloaded and should be put in the raw data subdirectory for that particular IES (e.g. Figure 1 GOM/raw_data/P1_076). Up to seven ASCII files are downloaded from each instrument and are listed in Table 1.

File	Contents
TXXX_#.DAT	Travel time data
PXXX_#.DAT	Pressure and temperature data
EXXX_#.DAT	Engineering data
FXXX_#.DAT	Unconverted pressure and temperature frequency data (in Hertz) From the Paroscientific pressure sensor
IESXXX.LOG	System log information
TELEM.DAT	Daily values to be telemetered
CXXX_#.DAT	Acoustic Doppler current meter data (CPIES only)

Table 1 Raw data files downloaded from IES

XXX is the instrument serial number (e.g. 076) and # is the mission configuration number (typically 1). The mission configuration number depends on the number of times the mission configuration has been modified. See the IES User's Manual for further information about mission configuration. The format and contents of the ASCII .DAT files are described in detail in the IES User's Manual.

The TELEM.DAT file contains daily, rather than hourly samples and is used with the Pulse Data Telemetry option from a ship. The IES processing package discussed in this report does not handle daily values and these data must therefore be processed separately if necessary. The TELEM.DAT file will not be discussed further in this manual. Consult the IES User's Manual for a description of the TELEM.DAT file.

On UNIX machines, permissions for the raw data files listed in Table 1 and the raw data directory (e.g. GOM/raw_data/P1_076) should be restrictive to prevent overwriting (*i.e.*, set read only permission for raw data directory and files).

Chapter 2. Overview of IES Processing

Run the two main processing programs in order:

```
IES_GUIDAT2MAT.m  
IES_GUIDRIVER.m
```

IES_GUIDAT2MAT.m is a GUI interface (Figure 2) to easily input several parameters to the program IES_DAT2MAT.m which performs the following operations:

- plots the raw data from the IES .DAT files listed in Table 1 and saves into .mat files
- gives an initial look at the data
- rescales the units to typical values

Each variable processed will be stored in a .mat file corresponding to the .DAT file name listed in Table 1.

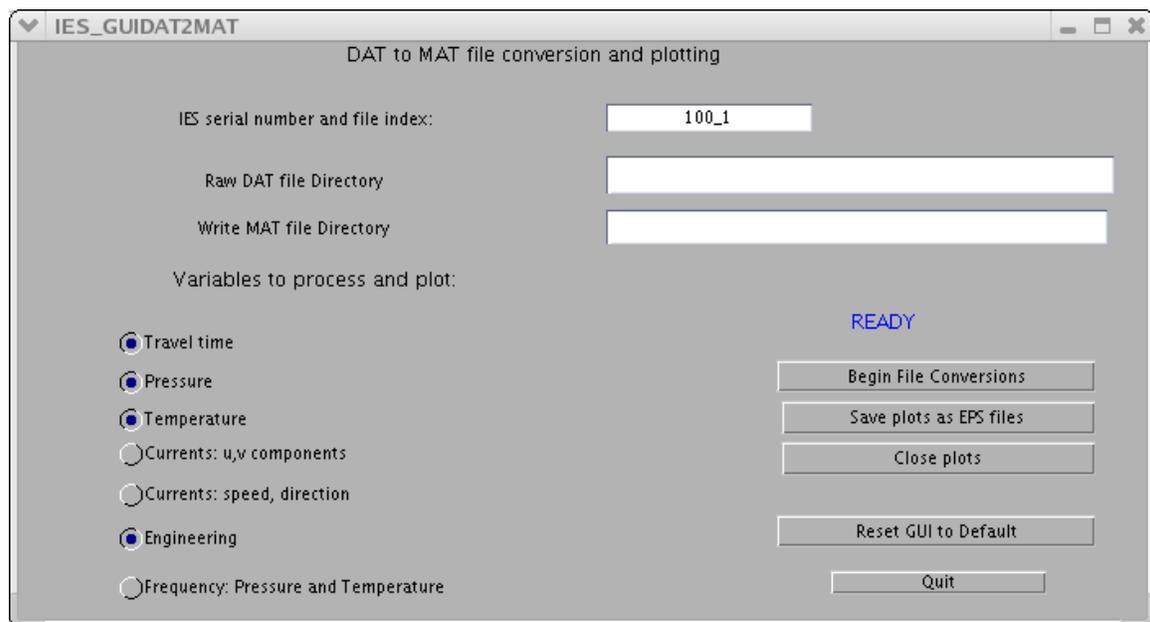


Figure 2 Default IES_GUIDAT2MAT window

IES_GUIDRIVER.m is a GUI interface (Figure 3) to easily input many parameters to IES_DRIVER.m which performs up to eight processing steps from initial windowing through lowpass filtering (see processing flowchart in Figure 4).

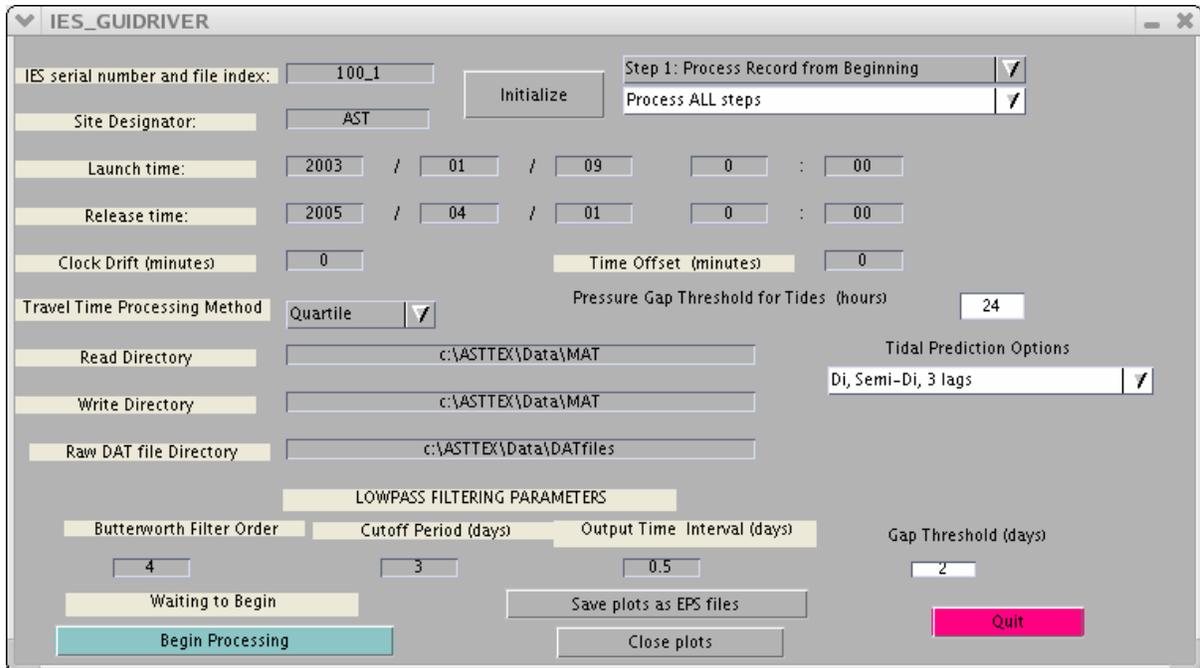


Figure 3 Default IES_GUIDRIVER window

Step 1) Windowing and Hourly values

- windows to remove large data spikes
- selects representative travel time for each hour using modified quartile (recommended), median or specialized methods
- calculates mean hourly values for pressure and temperature
- identifies on-bottom records (between launch and release command)
- eliminates pressure and temperature records prior to the internal temperature reaching equilibrium with surroundings
- assigns the mid-sample time for each variable in decimal days, relative to January 1 0000 UT of launch year
- checks for missing samples and fills the gaps with NaNs (e.g. during telemetry operations)

Step 2) Despiking

- identifies data spikes using Slope or Tukey methods and replaces them with interpolated values
- slope method identifies measurements that change more rapidly from the preceding value by more than the specified rate (tolerance)
- Tukey method performs median filters on the time series and identifies values that differ from the filtered record by the specified tolerance as outliers (requires the MATLAB signal processing toolbox)
- travel times can only be despiked using the slope method, either method will work on pressure and temperature
- hand editing of spikes can also be performed

Step 3) Initial Detiding

- calculates and removes tides from the pressure record (despiked but still containing drift) using a FORTRAN program called RESPO.for

Step 4) Dedrifting

- fits and removes drift from the pressure record (uses MATLAB optimization toolbox if available)

Step 5) Final Detiding

- adds the tides to the dedrifted pressure record and then recalculates the tides.
- stores the amplitude and phase of the user requested major tidal constituents

Step 6) Velocity Processing (if CPIES)

- calculates hourly averages of the velocity components and temperature

Step 7) Lowpass Filtering

- applies a Butterworth filter to all variables using user specified parameters (requires MATLAB signal processing toolbox)
- chops beginning and end of records to remove startup transients
- interpolates output time series to be “on the hour”

Step 8) Plotting

- produces 3 standard plots of final products

PIES/CPIES Processing Flowchart

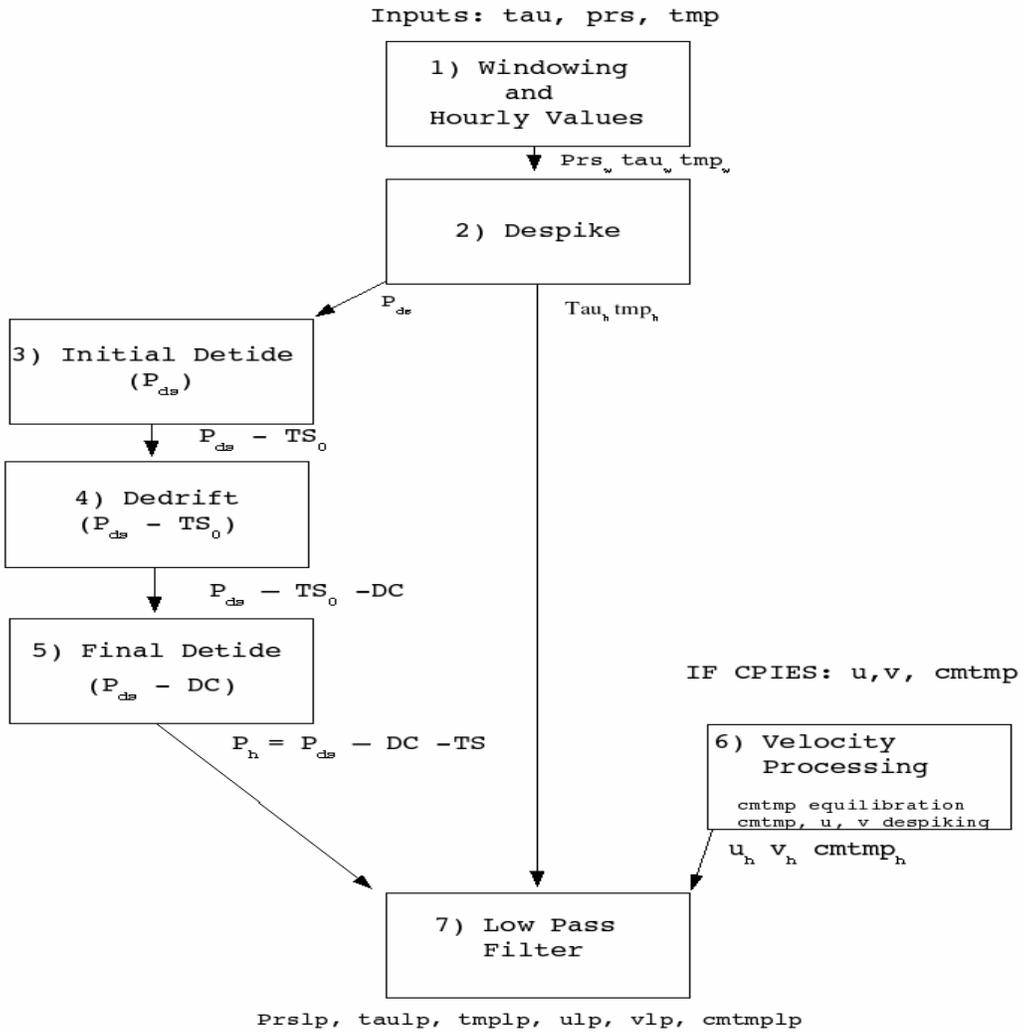


Figure 4 IES Data Processing Flowchart

Chapter 3. Step by Step Processing

3.1 Introduction

The two main IES processing programs are:

- 1) IES_GUIDAT2MAT.m
- 2) IES_GUIDRIVER.m

IES_GUIDAT2MAT must be run first to create .mat files from the raw (.DAT) data files listed in Table 1. IES_GUIDRIVER, the main processing program for IES Models 6.1 and 6.2, is then run. The programs are designed to process travel time, pressure and temperature data. However, the codes will run if any of those measurements are missing. The current meter data of a CPIES can also be processed by these two programs. Two final data products are produced:

- 1) hourly file containing
 - despiked travel time and temperature
 - despiked, detided, and dedrifted pressure
 - despiked currents and current meter temperature (CPIES only)
- 2) lowpass filtered, subsampled file at user-specified interval in days

In the following sections, processing for a PIES instrument will be described first, followed by modifications for a CPIES.

3.2 IES_GUIDAT2MAT

IES_GUIDAT2MAT.m is the GUI interface to easily input several parameters to IES_DAT2MAT.m. The main purposes of this program are:

- to plot the raw data contained in the ASCII .DAT files to provide an initial assessment of the data quality
- to convert the .DAT files listed in Table 1 into .mat files containing standard scientific units for subsequent processing.

Each variable processed by this program will be output to a .mat file corresponding to the original .DAT file name listed in Table 1 (e.g. the output file for GOM/raw_data/P1_076/T076_1.DAT will be GOM/mat_data/T076_1.mat).

IES_DAT2MAT can be executed without the GUI interface. In MATLAB run help on IES_DAT2MAT for usage instructions.

Start MATLAB (typically, MATLAB is started from the directory where you will store the .mat files, e.g. GOM/mat_data/). At the MATLAB prompt type

>>IES_GUIDAT2MAT

The default IES_GUIDAT2MAT window will appear as in Figure 2. Carry out the following steps which result in the display shown in Figure 5:

- Replace “100_1” with the 3-digit IES serial number and file index which corresponds to the mission configuration (e.g. 076_1).
- Enter the Raw DAT file directory
- Enter Write MAT file directory
- Note the pre-selected radio buttons in the lower left: Travel time, Pressure, Temperature and Engineering.
- For CPIES, toggle the appropriate Currents radio button (either u, v components or speed, direction) depending on how your instrument was configured prior to deployment (*not your post processing preference!*). Note however, the output .mat file will contain both types of data, u, v components and speed, direction. If you are uncertain of the CPIES setup, most instruments manufactured after January 2004 have been configured for the u, v components option.
- Toggle the Frequency radio button: Pressure and Temperature radio button in the lower left (optional). Do this if you think you may need raw frequency data (in Hertz) at a later time
- Press the button on the right side of the GUI window **Begin File Conversions**

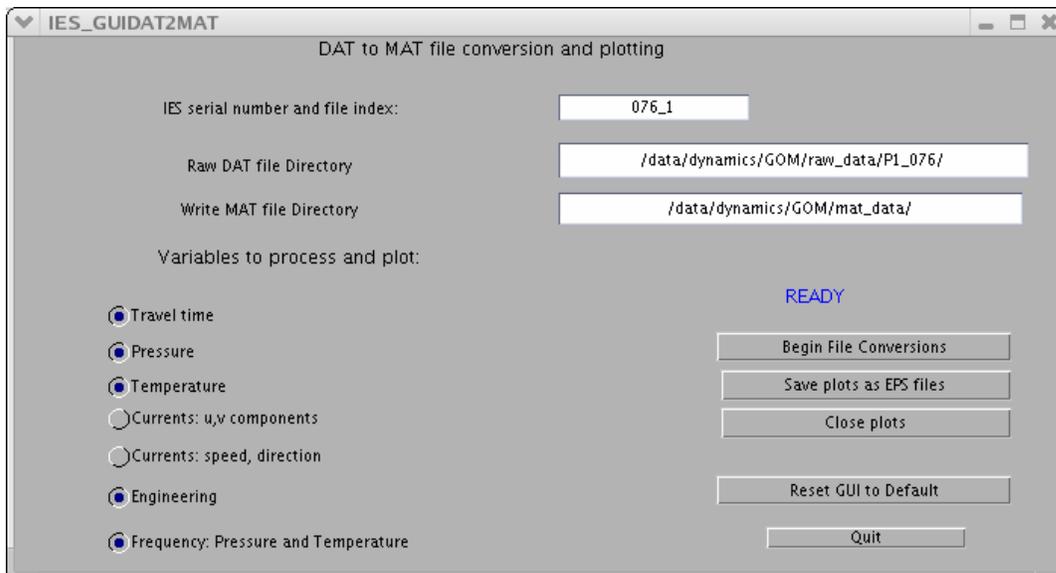


Figure 5 Example IES_GUIDAT2MAT window- READY State

MATLAB will then create figures and .mat files according to the selected variables. A maximum of 8 (10 for CPIES) Figure windows and 4 (5 for CPIES) .mat files (listed in Table 2) may be created.

File	Contents
TXXX_#.mat	Travel time data
PXXX_#.mat	Pressure and temperature data
EXXX_#.mat	Engineering data
FXXX_#.mat	Unconverted pressure and temperature frequency data (in Hertz)
CXXX_#.mat	Acoustic Doppler current meter data (CPIES only)

Table 2 Data files generated by IES_GUIDAT2MAT

On the right side of the GUI window **FINISHED** now replaces **READY** (Figure 6).

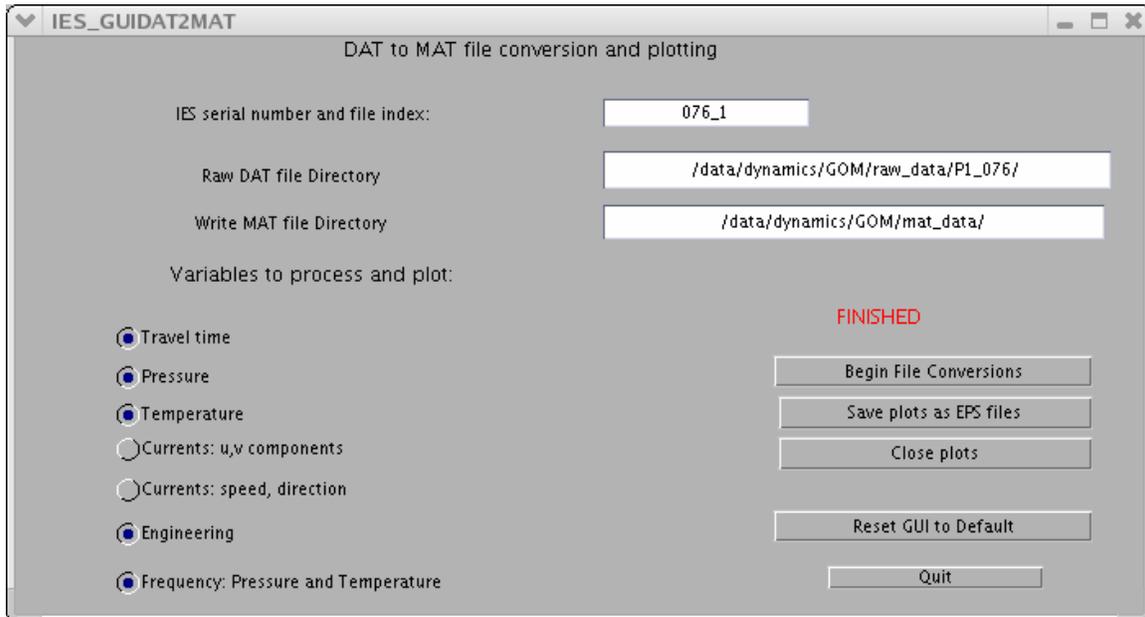


Figure 6 Example IES_GUIDAT2MAT window - FINISHED State

The variables contained in the output .mat files listed in Table 2 are shown in

- Figure 7: T076_1.mat travel time variables
- Figure 8: P076_1.mat pressure and temperature variables
- Figure 9: E076_1.mat engineering variables
- Figure 10: F076_1.mat frequency variables (pressure and temperature)
- Figure 11: C102_1.mat current meter variables (CPIES only)

```
>>
>>
>> load T076_1.mat
>> whos
Name              Size              Bytes  Class

Tau               230664x1          1845312 double array
theadr            5x10              100     char array
tieshrs          230664x1          1845312 double array
```

Figure 7 Example TXXX_#.mat file contents

```
>>
>> load P076_1.mat
>> whos
Name              Size              Bytes  Class

P                 57666x1           461328 double array
Tmp               57666x1           461328 double array
pheader          22x10              440     char array
pieshrs          57666x1           461328 double array
```

Figure 8 Example PXXX_#.mat file contents

```
>> load E076_1.mat
>> whos
Name              Size              Bytes  Class

Efrq              401x13            41704 double array
eheader           4x10               80     char array
eieshrs           401x1              3208 double array
```

Figure 9 Example EXXX_#.mat file contents

```

>>
>> load F076_1.mat
>> whos
Name                Size                Bytes  Class

fP                  57666x1            461328  double array
fTmp                57666x1            461328  double array
fheader             22x10              440     char array
fieshrs             57666x1            461328  double array

```

Figure 10 Example FXXX_#.mat file contents

```

>>
>> load C102_1.mat
>> whos
Name                Size                Bytes  Class

cieshrs             54168x1            433344  double array
compass_direction   54168x1            433344  double array
direction           54168x1            433344  double array
ping_count          54168x1            433344  double array
signal_strength     54168x1            433344  double array
speed               54168x1            433344  double array
temperature         54168x1            433344  double array
tilt_east           54168x1            433344  double array
tilt_north          54168x1            433344  double array
u                   54168x1            433344  double array
v                   54168x1            433344  double array

```

Figure 11 Example CXXX_#.mat file contents (CPIES only)

NOTE: If a .DAT file does not exist for a parameter, the remaining data can still be processed. However, a warning message will be displayed in the MATLAB window. For example, if there is no TXXX_#.DAT file, the message “*** WARNING: Could not open Travel Time” will be displayed. The rest of the files will be converted to .mat files and figures for the converted parameters will be generated.

It is recommended that you save and print the figures created by IES_GUIDAT2MAT by pressing the button **Save plots as EPS files**. IES_GUIDAT2MAT will create a “figs” subdirectory (e.g. GOM/mat_data/figs) in which to store the .eps files. The saved filenames will have the syntax PIESXXX_#_fig1.eps. Examples of the MATLAB figure windows are included in Appendix A for a PIES with the two additional plots generated for a CPIES in Appendix B.

To assess data quality, examine the figures windows generated by IES_GUIDAT2MAT.

Figure 1: Travel time (seconds). The full scale is 0-10 seconds but most of the data will be concentrated in a narrow range of approximately 0.05 seconds (e.g. Appendix A, Figure 69). Check the travel time quality by zooming in using the toolbar in the figure window (e.g. Appendix A, Figure 70)

- an instrument moored between 2000 and 5000 m depth records travel time data in the range 2.6 to 6.6 seconds
- values near 10 seconds are bad and indicate no echoes
- good quality data have small scatter

Figure 2: Pressure (dbar). The default plot will show the full range of pressure (e.g. Appendix A, Figure 71). The signal typically looks linear because it includes a wide range of pressure values in going from air (while on the deck) to the ocean bottom. Zooming in will show the bottom pressure which is typically dominated by the tidal signal (Appendix A, Figure 72).

Figure 3: Temperature (deg C). The default plot will show a large range (e.g. Appendix A, Figure 73), going from air temperature values followed by a rapid decrease as the instrument equilibrates to the value at the seafloor with a linear temperature for the remainder of the time series. Zoom in on the linear portion to see the temperature stabilize approximately 24 hours after deployment (e.g. Appendix A, Figure 74).

If the instrument is a CPIES, Figure 4 will contain current speed, current direction, a vector stick plot and current meter temperature (e.g. Appendix B, Figure 82). The large y axis ranges shown by the default figure for the current speed and stick plot time series indicate that there may be bad data at the beginning of the record (e.g. Appendix B, Figure 83). This bad data may be removed from the time series during subsequent processing (see Section 3.5.6 and Figure 54). Figure 5 will contain signal strength, compass direction, north and east tilt (deg) and ping count (Appendix B, Figure 84).

Figures 4-6 (PIES) or 6-8 (CPIES). Engineering plots. These are primarily used to assess instrument engineering performance (e.g. Appendix A, Figures 75-77).

Figure 7 (PIES) or 9 (CPIES). Temperature Frequency in Hertz (e.g. Appendix A, Figure 78). Reducing the y-axis range (e.g. Appendix A, Figure 79) should show a time series like that in Appendix A, Figure 74.

Figure 8 (PIES) or 10 (CPIES). Pressure Frequency in Hertz (e.g. Appendix A, Figure 80). Reducing the y-axis range (e.g. Appendix A, Figure 81) should show a time series like that in Appendix A, Figure 72.

NOTE: Each time the **Save plots as EPS files** button is pressed the EPS files are overwritten with the current figures. Axis limits will reflect whatever appears in the Figure window at the time the button is pressed.

To complete this step, press the **Quit** button in the lower right corner of the IES_GUIDAT2MAT GUI window or **X** in the upper right corner (Figure 6).

3.3 IES_GUIFREQ2MAT

IES_GUIFREQ2MAT is the GUI interface to easily input parameters to IES_FREQ2MAT.m. IES_FREQ2MAT is provided to calculate pressure in dbar and temperature in degrees C from the recorded frequencies in Hertz using the conversion coefficients supplied by Paroscientific, Inc. IES_GUIFREQ2MAT will not normally be needed in routine processing since IESs are configured to internally convert pressure and temperature frequencies to scientific units which are then downloaded to the PXXX_#.DAT file listed in Table 1. However there may be circumstances where you need to run this program:

- the wrong Paroscientific calibration coefficients were loaded into the IES when the instrument was set up
- the temperature sensor fails and you want to recover pressure data.

To execute this program, type the following at the MATLAB prompt:

```
>> IES_GUIFREQ2MAT
```

The default IES_GUIFREQ2MAT window will appear as in Figure 12.

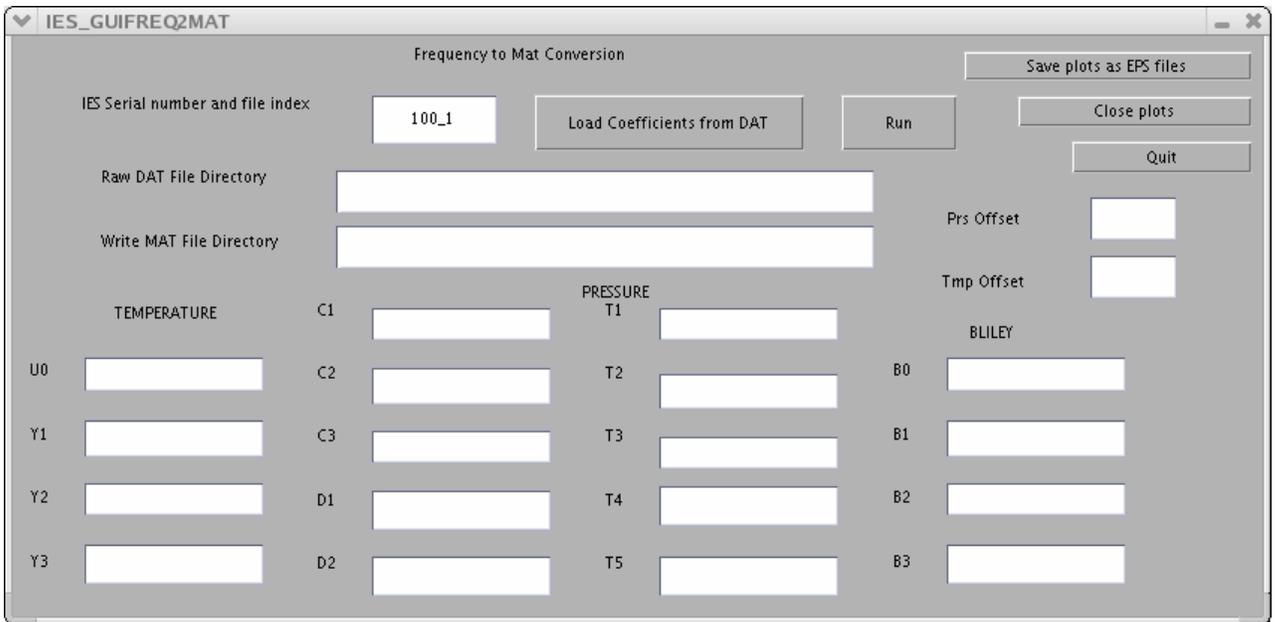


Figure 12 Default IES_GUIFREQ2MAT window

Enter:

- IES Serial number and file index
 - Raw DAT file Directory
 - Write MAT file Directory
 - Bliley coefficients B0, B1, B2 and B3 (obtained from the IES engineer).
- NOTE:** the Bliley coefficients must be entered manually.

To load the Paroscientific conversion coefficients entered into the IES at set-up (*i.e.*, from the FXXX_#.DAT file) press the button

- **Load Coefficients from DAT**

This will cause the coefficients U0, Y1, Y2, Y3, C1, C2, C3, D1, D2, T1, T2, T3, T4, T5, and Prs and Tmp Offset to be read from the FXXX_#.DAT file. Values can then be verified against the coefficients provided by Paroscientific and edited if necessary in the GUI window. Otherwise manually enter these conversion coefficients. Example values for 076_1 are shown in Figure 13.

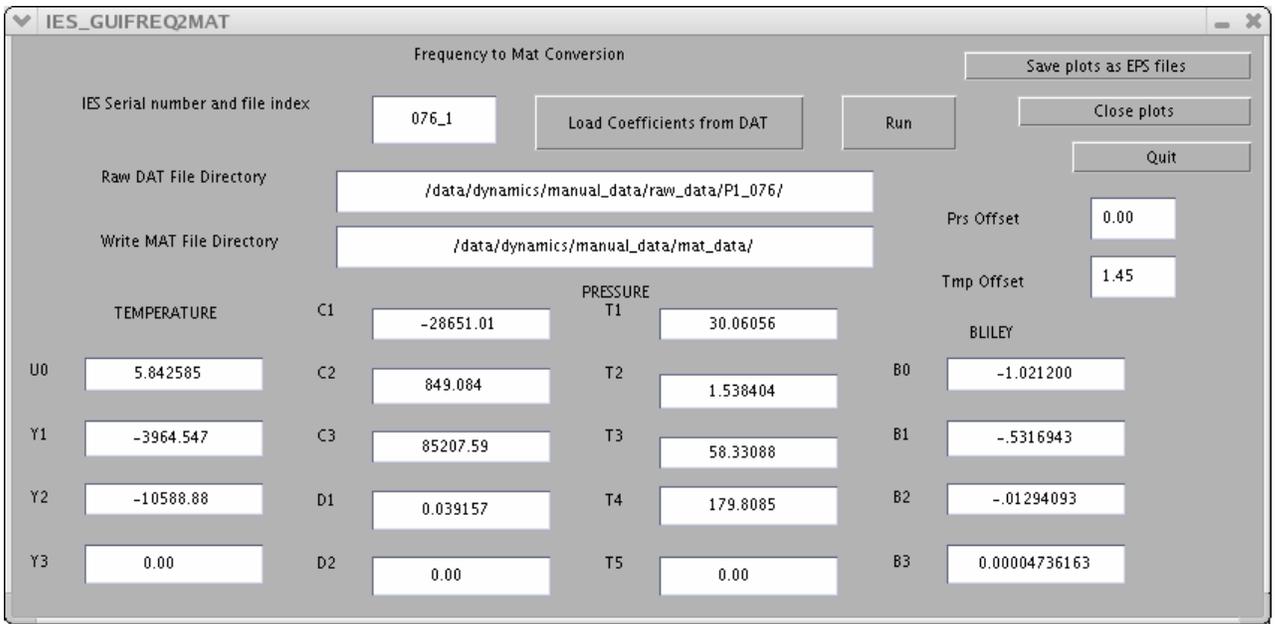


Figure 13 Example IES_GUIFREQ2MAT window after coefficients loaded

- Press **Run**

The program plots the converted temperature and pressure data, and then saves the data to a file called PXXX_#.mat which can be used by IES_GUIDRIVER. Save the plots by pressing the **Save Plots as EPS files** button. The plots will be equivalent to Figures 2 (pressure) and 3 (temperature) output by IES_GUIDAT2MAT and shown in Appendix A (Figure 71 and Figure 73). To terminate this program, press the **Quit** button in the upper right of the IES_GUIFREQ2MAT GUI window or **X** in the upper right corner.

3.4 InitGuiDriver.dat - Parameter File Setup

For the user's convenience, prior to running IES_GUIDRIVER, a parameter file called **InitGuiDriver.dat** may be created. This parameter file can be considered a lookup table, based on IES serial number, containing information relevant to a particular instrument and study. The parameter file is used to initialize the GUI with values so that the user does not have to enter them manually. A single parameter file can be made for all instruments in a study using your preferred editor (e.g., WordPad, vi, emacs). Default values for all instruments are listed first, followed by instrument specific entries. The user will need to consult the log sheets from the deployment/recovery cruises for some values (e.g., timeoffset, clockdrift, launch and release times). **NOTE:** All times should be entered as UT.

An example InitGuiDriver.dat file is shown in Figure 14. Descriptions of possible parameters to specify are listed in the comments section of InitGuiDriver.dat. (**NOTE:** Items for GUI window pull down menus cannot be entered into the InitGuiDriver.dat file). Edit InitGuiDriver.dat. After the comments section type the word "array" followed by a descriptor for your study (e.g. EGOM). On the lines following "array" type any parameters common to all instruments (e.g. readdir, writedir, lpgap) or parameters for which you wish to specify a default value (e.g. time offset). Next, type in the parameters specific to each instrument (e.g. launch, release, clockdrift, rawdir). The keyword "psn" begins the list for a particular instrument. There should be no spaces in the value entered for siteid. Spaces will be replaced with the underscore character. Consult the deployment/recovery cruise log sheets for the values of launch, release, and clockdrift. Clock drift should be entered in minutes, however equations are allowed (e.g., for 076_1, from the recovery log sheet, a clock drift of -14 seconds can be entered as -14/60 minutes, see Figure 14). TimeOffset is typically zero and it is recommended that the default value zero be entered with the "array" parameters (Figure 14). However, if there are multiple instruments at the same site, one will have a time offset with the offset value listed on the cruise log sheets. Enter the TimeOffset for the second instrument at a site with its parameters (see timeoffset entry for psn 090_1 in Figure 14). Including "timeoffset 0" with the "array" entries prevents a non zero value being inadvertently applied in a case of no time offset.

```

%Possible parameters to specify:
% PSN - IES serial number and file index
% SITEID - is an alternate name to use for the final output .mat files.
%           If not specified, then the unfiltered records will be stored
%           in PIESPSN.mat and the filtered records in PIESPSNlp.mat
% launch - is time when instrument was launched and specified as
%           [year month day hour minute]. This information can be
%           found the log sheet from the launch cruise.
% release - is the time when the release command was sent on the recovery
%           cruise. It is specified as [year month day hour minute].
% ClockDrift - is the amount of time in minutes that the internal
%           clock drifted during the deployment period.
%           If drift exceeds 1 minute per year, the 1 hour
%           sampling interval is adjusted. Otherwise, the start
%           time is shifted. Default = 0.
% ReadDir - is the directory containing the input .mat files
% WriteDir - is the directory where intermediate and final .mat
%           files are saved
% RawDir - is the directory containing the raw .DAT files Default = pwd
% LPGap - is the maximum allowable pressure data gap size that can be
%           linearly interpolated before lowpass filtering. Records
%           containing larger than this threshold will be separated into
%           segments and the filtering will be run on each portion
%           separately. Default = 2 days.
% TideGap - is the maximum allowable data gap size that can be linearly
%           interpolated before performing the tidal analysis. Records
%           containing gaps larger than this threshold will be
%           separated into segments and the tidal analysis run on each
%           portion separately. Default = 24 hours.
% sampT - is the sample interval for filtered output (in days).
%           Default=0.5 days
% filtT - is the filter cutoff period. Default=3 days
% Order - the order of the butterworth lowpass filter. Default=4
% TimeOffset - is the programmed offset in minutes from on-the hour
%           sampling. Its value is taken from the deployment log
%           sheet. Positive offset means IES samples earlier than
%           expected. Negative offset means it samples later than
%           expected. Default = 0.

array EGOM
readdir /data/dynamics/GOM/mat_data/
writedir /data/dynamics/GOM/mat_data/
lpgap 1
timeoffset 0

psn 076_1
siteid Site1
launch 2004 12 19 15 32
release 2006 01 24 01 23
clockdrift -14/60
rawdirt /data/dynamics/GOM/raw_data/P1_076/

psn 090_1
siteid Site4_2
launch 2005 8 24 05 01
release 2006 01 25 10 18
clockdrift 0
timeoffset -5
rawdirt /data/dynamics/GOM/raw_data/P1_076/

```

Figure 14 Example InitGuiDriver.dat file

Not all values need to be specified in InitGuiDriver.dat and when values are not specified, the defaults of Figure 3 (e.g., Lowpass Filtering Parameters) will be used unless the user subsequently changes the values directly in the GUI. **IMPORTANT:** InitGuiDriver.dat needs to be located in the current working directory (e.g. GOM/mat_data/) or in the Read Directory specified in the GUI (prior to pressing **Initialize**), otherwise MATLAB will prompt the user for the directory location.

3.5 IES_GUIDRIVER

This program performs the following steps:

- Step 1: windowing and hourly values
- Step 2: despiking
- Step 3: initial detiding
- Step 4: dedrifting
- Step 5: final detiding
- Step 6: velocity processing (if CPIES)
- Step 7: lowpass filtering
- Step 8: plotting

IES_GUIDRIVER creates two subdirectories “wrkspc” and “figs” (if not already created by IES_GUIDAT2MAT) in the “Write Directory” (e.g. GOM/mat_data/wrkspc/ and GOM/mat_data/figs/). The “wrkspc” subdirectory holds intermediate MATLAB workspaces created by IES_GUIDRIVER as well as a tide log generated during detiding. The “figs” subdirectory holds the final figures created during plotting, Step 8.

The .mat files produced by IES_GUIDAT2MAT listed in Table 2 are the input data sets for IES_GUIDRIVER. At various steps, the MATLAB workspace is saved (e.g. in the wrkspc subdirectory: WindowedXXX_#.mat is saved after the windowing step, DespikedXXX_#.mat is saved after the despiking step, etc). In general, you will not need to use these intermediate MATLAB workspaces (WindowedXXX_#.mat, DespikedXXX_#.mat). However, do not delete them since they allow you to process the data starting at any step, not just step 1. For example, it is common to iterate and improve drift removal after all the IES data sets have been processed.

IES_GUIDRIVER creates a log file documenting which steps were carried out and the processing parameters specified. Information is appended to this log file after

each processing step. **NOTE:** If you start over at an intermediate step, log information, workspaces and data files associated with subsequent steps will be overwritten.

The final data sets will be named based on what is specified as the “site designator” which can be entered into the InitGuiDriver.dat parameter file as siteid (Figure 14). For example, if you specify “Site1” as the site designator then Site1.mat and Site1lp.mat will be the final data sets created for the hourly and lowpass filtered data sets, respectively. Default names PIESXXX_#.mat and PIESXXX_#lp.mat are used if a site designator is not specified. We recommend using short site designators since they are also used to name the structure arrays in the final data sets. IES_DRIVER can be executed without the GUI interface. Details about IES_DRIVER options can be obtained by running help on IES_DRIVER in MATLAB.

At the MATLAB prompt, type:

```
>>IES_GUIDRIVER
```

The default IES_GUIDRIVER window will appear as in Figure 3. The GUI parameters can be initialized manually or from a previously created parameter file (see Section 3.4 InitGuiDriver.dat). To initialize with the values entered in an InitGuiDriver.dat file, that file must be in your current working directory or specified in the Read Directory entry in the GUI window. Enter the IES serial number and file index (e.g. **076_1**) and press the **Initialize** button in the upper middle of the GUI. The defaults of Figure 3 will be replaced with the values associated with **076_1** from the InitGuiDriver.dat file (listed in Figure 14) as shown in Figure 15. Parameters not specified in InitGuiDriver.dat will keep the default values of Figure 3. **NOTE:** Once initialized and the first step has been run, any value filled in (as in Figure 15) will be retained for all subsequent processing steps and need not be specified again unless changes are desired.

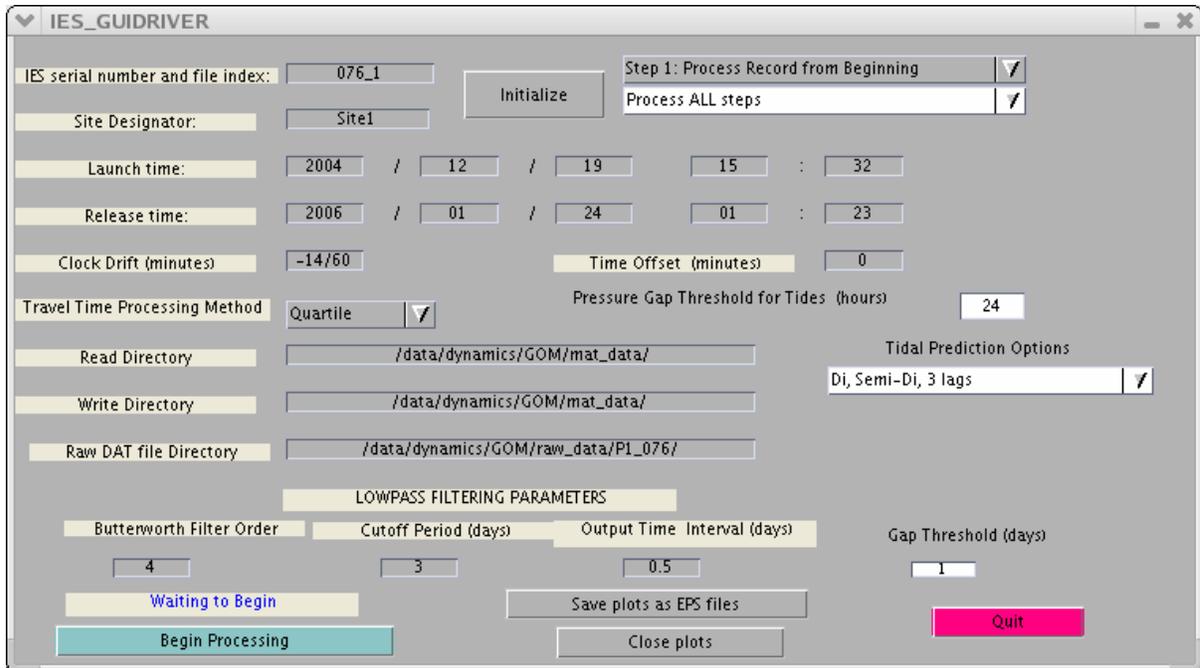


Figure 15 Example IES_GUIDRIVER window after initialization for processing ALL steps

Note, that the default processing assumes that all steps will be carried out from beginning to end (*i.e.*, Figure 15 top right pull down menu shows **Step 1: Process Record from Beginning** and second pull down menu shows **Process ALL Steps**). The user may choose to start and stop at other steps. However, the steps must be run sequentially, they cannot be executed out of order. For this discussion, processing will be carried out one step at a time and the results of each step will be described before proceeding.

3.5.1 Step 1: Windowing and Hourly Values

This first step performs several tasks. Based on the launch and release times specified in the IES_GUIDRIVER window, it identifies which records were collected while the instrument was on the bottom. It assigns a time base for each variable in decimal days, relative to January 1 0000 UT of the launch year. It checks for missing samples (during telemetry operations) and fills the gaps with NaN values. It selects a representative travel time for each hour using modified quartile, median, or specialized methods. It calculates mean hourly values for pressure and temperature. With user

interaction, it removes large spikes and eliminates pressure and temperature records prior to the internal temperature reaching equilibrium with surroundings.

After initialization, keep **Step 1: Process Record from Beginning** for the upper pull down menu, but change the second pull down menu to **Stop before Despiking** (Figure 16). Specify the **Travel Time Processing Method** (Quartile (default), median, specialized, or do not reprocess) using the pull down menu in the middle of the IES_GUIDRIVER window. We recommend using the “Quartile” method which tracks the clean, sharp leading edge of the raw travel time measurements well. Using this method, each hourly burst of tau measurements are passed through two stages of windowing to eliminate outliers and to reduce the noise in the hourly values. In the first stage, the measurements in each burst are sorted and the first quartile value is calculated. These hourly values are then smoothed with a 4 hour median filter. Subsequently, the travel time measurements within each hourly burst are windowed with an asymmetric window about these fitted values. Because a histogram of the travel time measurements during a typical hourly sample is skewed with more spread to the right of the mean (*i.e.*, toward longer travel times) an asymmetric window (from minus 100 counts (~5 ms) to plus 50 counts (~2.5 ms)) is used to exclude more long taus than short taus from the data window. The n measurements which fall within this asymmetric window are passed to the second stage of processing. In the second stage, the first quartile is found for the n travel time measurements in each hourly burst. Next the $n/6$ travel times nearest the quartile value are averaged to obtain the final travel time for each hour. For example, if $n=24$ for hour h (*i.e.*, all samples in the hour are acceptable), the hourly travel time will be an average of four measurements. This averaging reduces the noisiness of the hourly values.

The “median” option is supplied because it is an easily understood statistical method. However, the median method will shift the hourly values to longer travel times and may create some false offsets.

The “specialized” option is provided to allow users the flexibility of creating their own method for selecting the appropriate hourly tau value. The user should edit and modify IES_TTSPECIAL.m for this purpose. One example of another method is IES_TT3RD.m which selects the third earliest ping for the hourly tau value.

Press the button **Begin Processing** in the lower left of the IES_GUIDRIVER window (Figure 16).

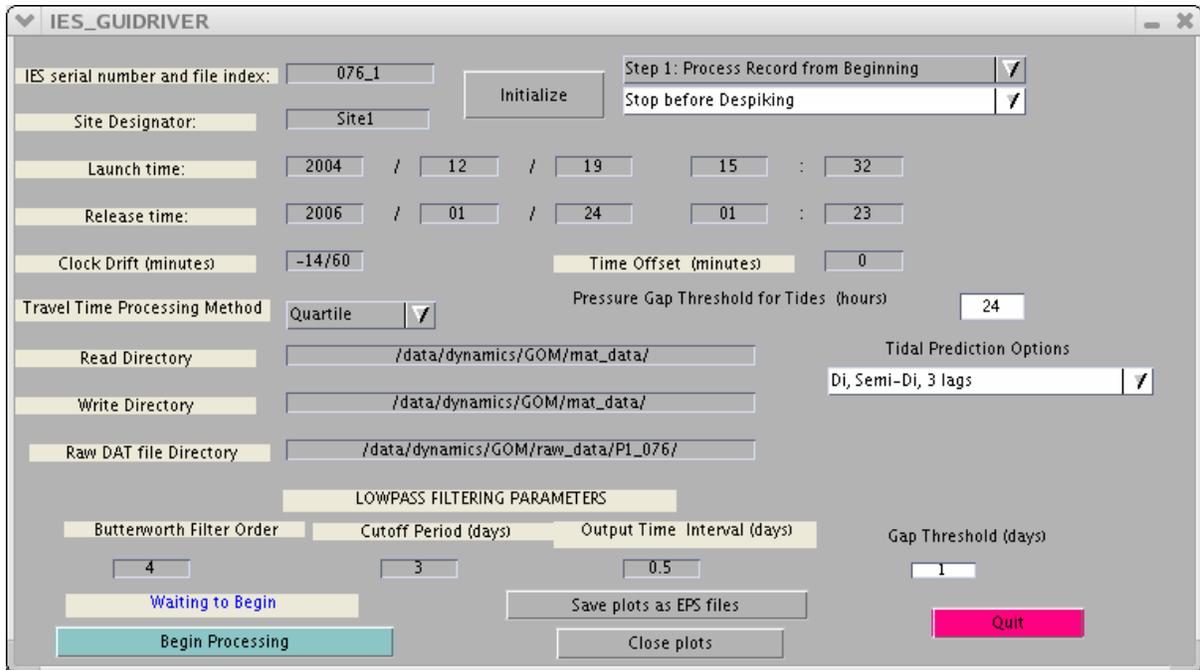


Figure 16 Example IES_GUIDRIVER window after initialization for processing Step 1 only

The status button in the lower left of the IES_GUIDRIVER window will change from **Waiting to Begin** to **Running**. A figure window will appear (Figure 17) showing a plot of the full range of tau measurements (in seconds) as a function of record number. The first step is to specify an acceptable range of good tau counts to eliminate early and/or late echo returns from being used during subsequent processing steps. Position the crosshairs below the good tau counts (near 3.2 seconds) and click the mouse to select the lower bound. Then reposition the crosshairs above the good tau counts and click the mouse again to select the upper bound. Next, THE IES_GUITAULIMITS window opens (Figure 18).

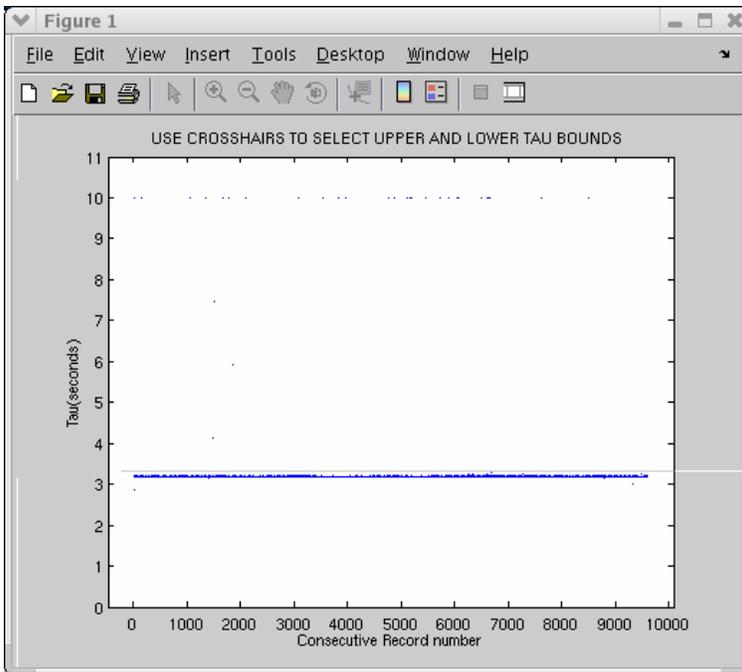


Figure 17 Figure window generated by Step 1 with full range of tau measurements

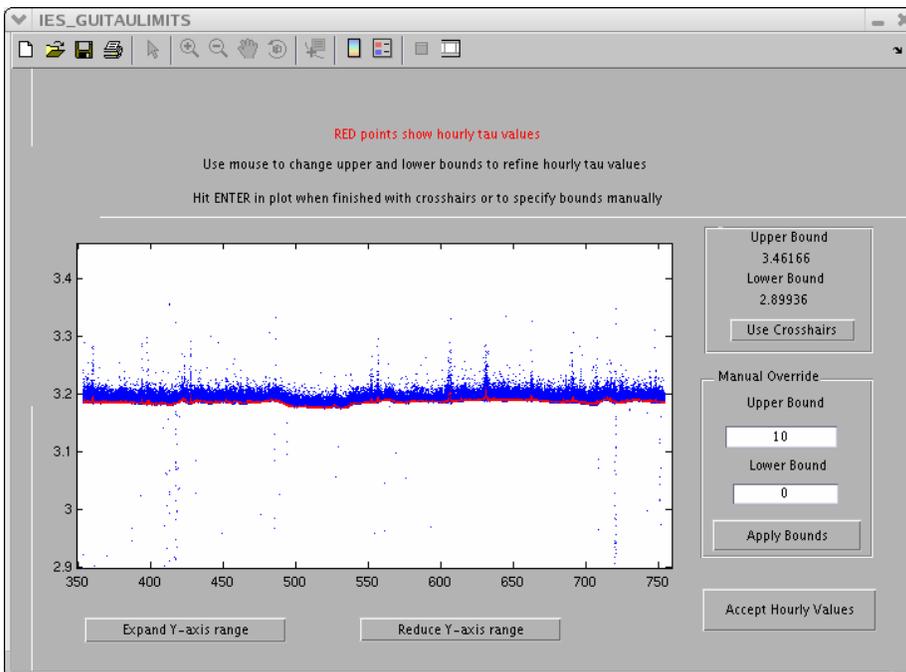


Figure 18 Example IES_GUITAULIMITS window

In Figure 18 the blue dots are the raw tau measurements and the red line shows the resulting hourly tau values obtained with the upper and lower tau bounds shown in the upper right box of the GUI (Upper Bound 3.46166, Lower Bound 2.89936). The goal is for the red curve (the calculated hourly tau values) to track the lower edge of the dense cloud of blue dots. For this example, the tau bounds are quite broad which biases the hourly taus to longer values, away from the lower edge of the cloud. To improve the hourly values, you need to adjust the upper and lower bounds to restrict the range. You can hit Enter immediately to end crosshairs, if you want to specify the bounds manually. Otherwise, position the crosshairs with the mouse above and below the blue cloud to select new tau bounds. New hourly values are calculated using the refined bounds and are replotted as shown in Figure 19. **NOTE:** even though you may reduce the range of the bounds to calculate the hourly (red) values, the smallest y-axis range generated by the program is 0.1 s. This insures that the range is sufficient to span the important signals for most deployments. For reference, as an approximation, a 1000 m change in thermocline depth would produce about a 0.05 second change in tau.

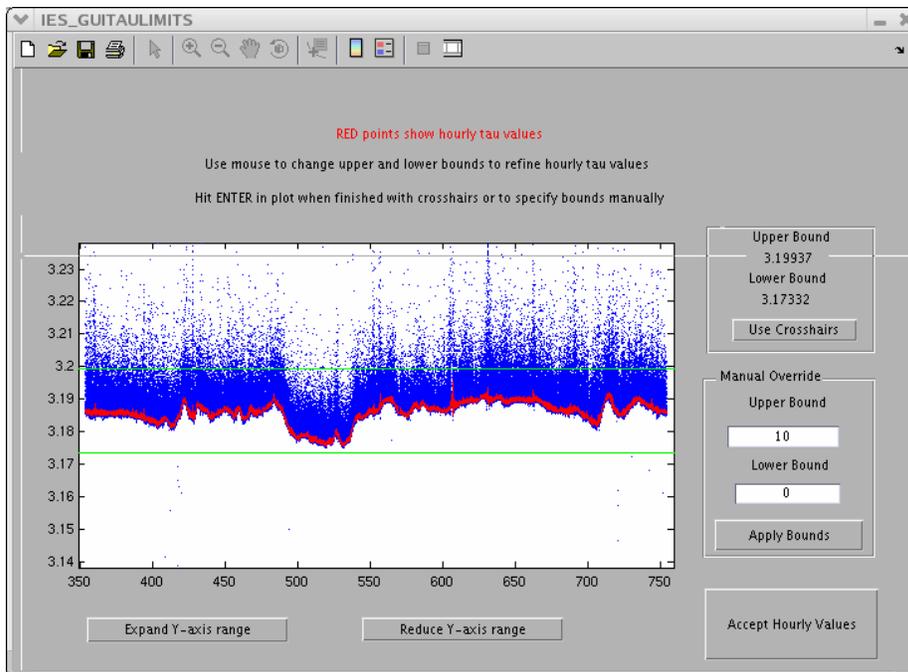


Figure 19 Example IES_GUITAULIMITS window with refined tau bounds

You can expand or reduce (Figure 20) the y-axis range by selecting the buttons at the bottom of the GUI window (**Expand Y-axis range**, **Reduce Y-axis range**).

However, you must terminate the crosshairs to adjust the y-axis range. Reducing will show less than the 0.1 s default y-axis range. You can repeatedly use the crosshairs and mouse to refine the upper and lower tau bounds (green lines). Each time the hourly taus are recalculated and replotted. Typically when good bounds are set, the lower bound is below the blue cloud since the leading edge is usually sharp/clean but the upper bound cuts through the cloud.

To terminate the crosshairs when you are satisfied with the hourly tau values, press **Enter** on the keyboard while the cursor is in the plot window. If you are finished refining the tau bounds, press the button **Accept Hourly Values** located in the lower right corner of the GUI window to end the processing and exit the window.

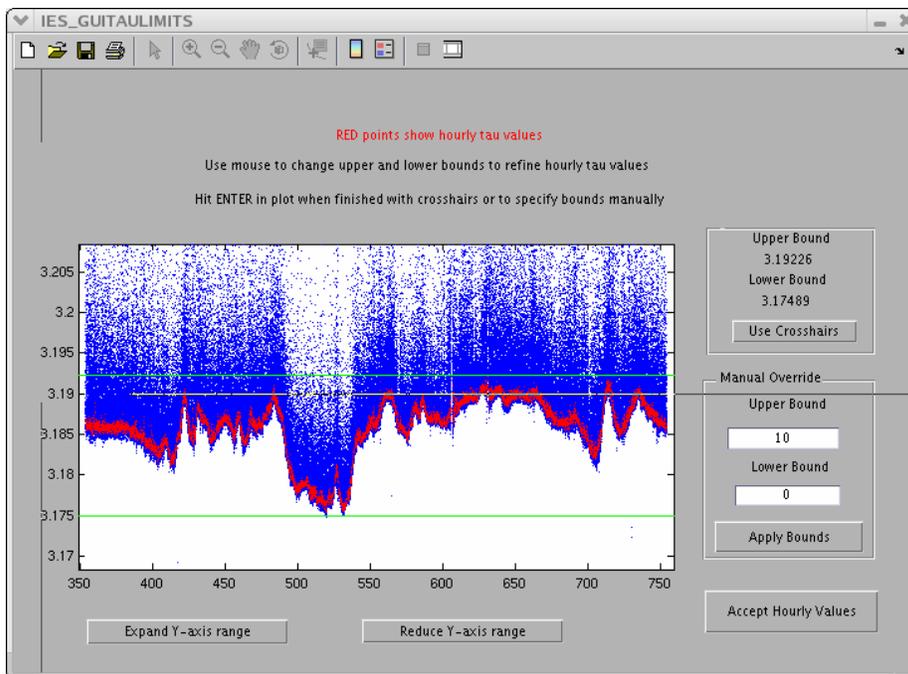


Figure 20 Example IES_GUITAULIMITS window with reduced Y-axis range

Otherwise you can refine the tau bounds further using crosshairs again or the Manual Override box on the right side of the GUI. If you choose the manual override option, enter the upper and lower bound values and press **Apply Bounds**. When satisfied with the hourly valued produced with manually selected tau bounds, press the button

Accept Hourly Values. NOTE: The manual override option is useful if you later need to repeat step 1, but were satisfied with the tau bounds recorded in the processing log.

Histograms are used to determine the acceptable ranges of good pressure and temperature measurements. The next figure window to open shows two histograms of pressure (Figure 21). Working with the histogram in the lower plot, set limits for the minimum and maximum pressures. **NOTE:** there are instructions in the MATLAB window as well as in red in the Figure window. Following the directions in the Figure window, click in the lower plot to select left and right limits of the histogram. Figure 1 is then replaced by the full pressure time series with the selected limits (Figure 22).

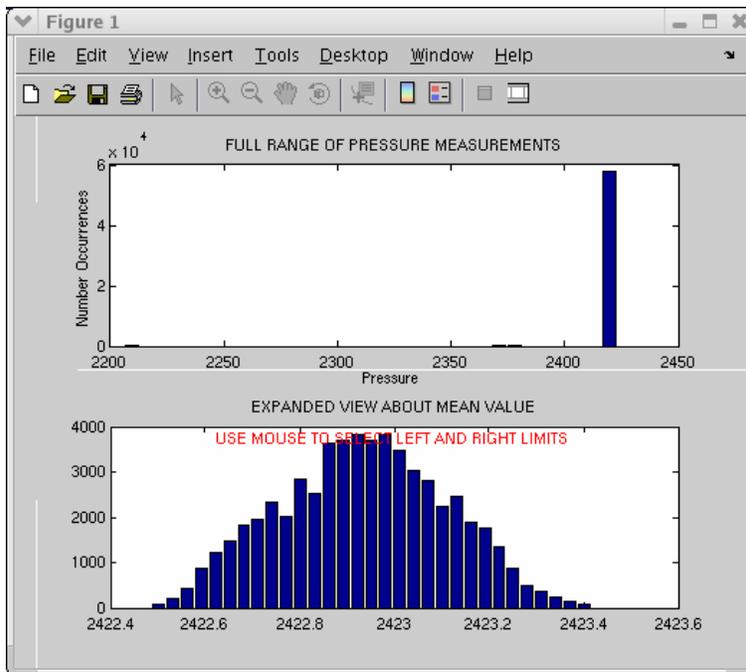


Figure 21 Figure window generated by Step 1 with full range of pressure measurements (top) and expanded view about mean value (bottom)

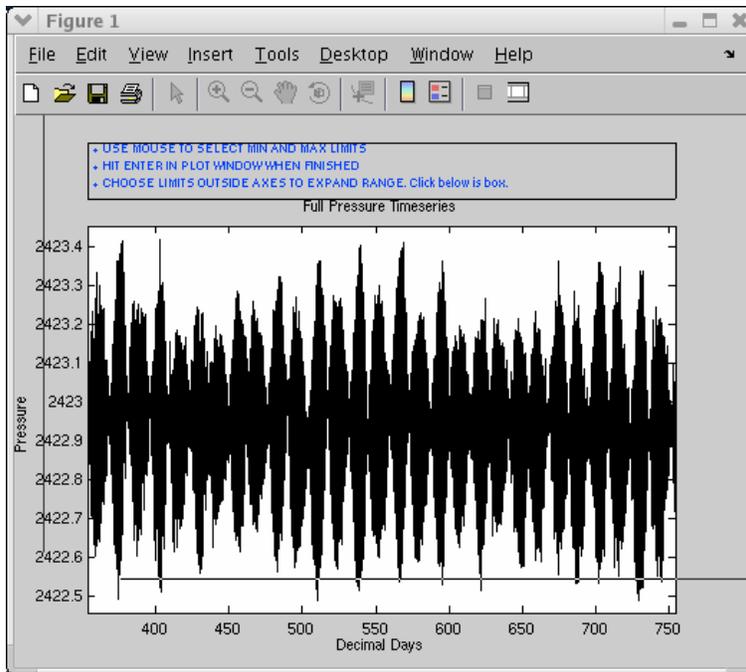


Figure 22 Figure window generated by Step 1 showing full pressure time series after initial windowing

Again directions in the Figure window guide the user. The pressure and temperature time series are typically very clean and may only need bounds changed once, if at all. To change the bounds click minimum and maximum limits. **NOTE:** you can press **Enter** at any point; you don't have to select new bounds. Repeat as many times as necessary to bound the pressure measurements. If you make a mistake and need to expand the axes range of the plot, click somewhere outside the plot and instruction boxes. When you are satisfied with the range of pressure, while the Figure window is active, press **Enter**. Repeat these steps with temperature (see Figure 23 and Figure 24).

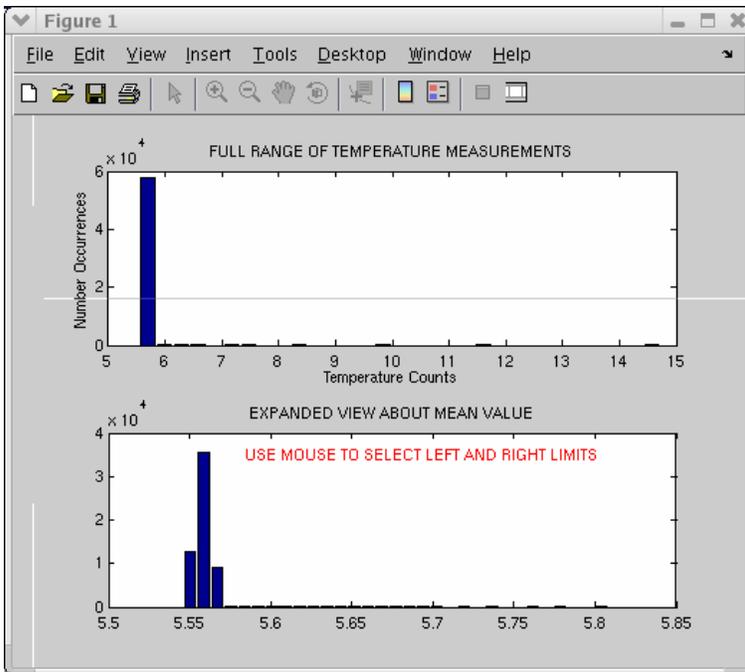


Figure 23 Figure window generated by Step 1 with full range of temperature measurements (top) and expanded view about mean value (bottom)

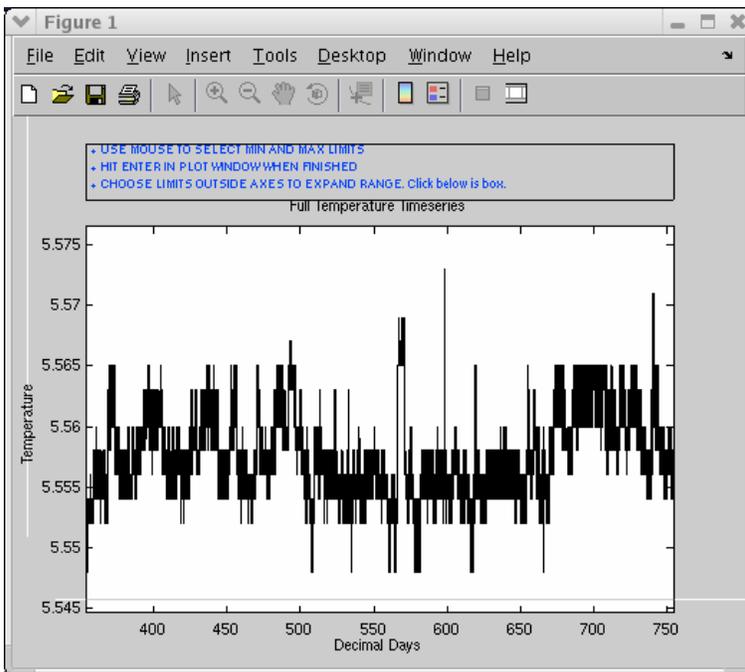


Figure 24 Figure window generated by Step1 showing full temperature time series after initial windowing

After the temperature bounds selection is completed, the next figure is a time series of hourly temperature from the beginning of the record (Figure 25). This plot allows the user to identify when the internal temperature of the IES reached equilibrium with the surroundings. Pressure and temperature data prior to the equilibrium time will be discarded. The internal temperature of the IES is measured by a Paroscientific sensor located within the IES glass sphere (*i.e.*, NOT in direct contact with seawater) and is needed for the conversion of pressure frequency to dbars. When the figure opens, a four-day long segment of temperature is plotted with values up to 12 hours after launch circled in red. These points will be eliminated unless the user selects a different value. Follow the directions in the Figure window. Position the crosshairs and then click the mouse on the last point to omit (Figure 26). Then hit **Enter**. **NOTE**: additional temperature measurements are obtained for CPIES by a temperature sensor on the DCS, which is in direct contact with seawater and therefore reaches equilibrium more quickly than the Paroscientific temperature sensor in the IES.

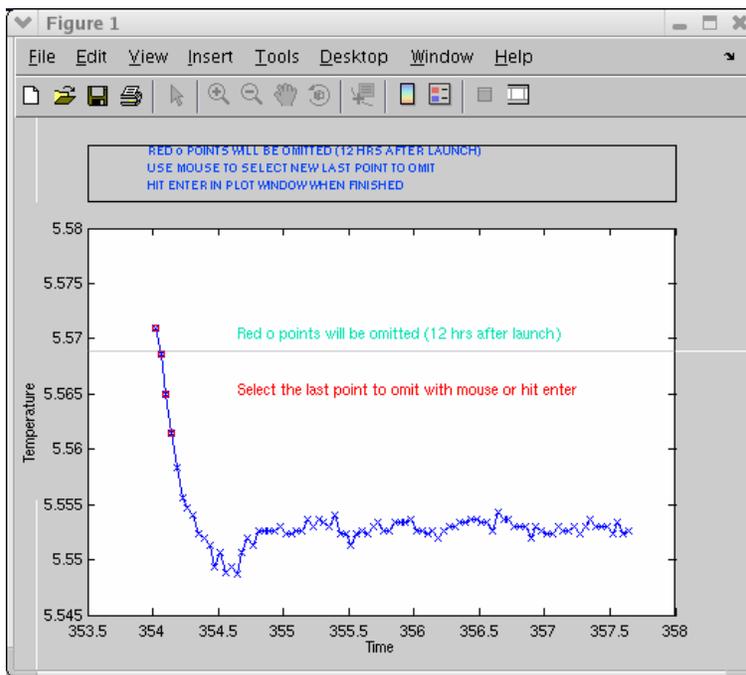


Figure 25 Figure window generated by Step 1 showing beginning of temperature time series with points to omit (12 hours after launch) shown in red

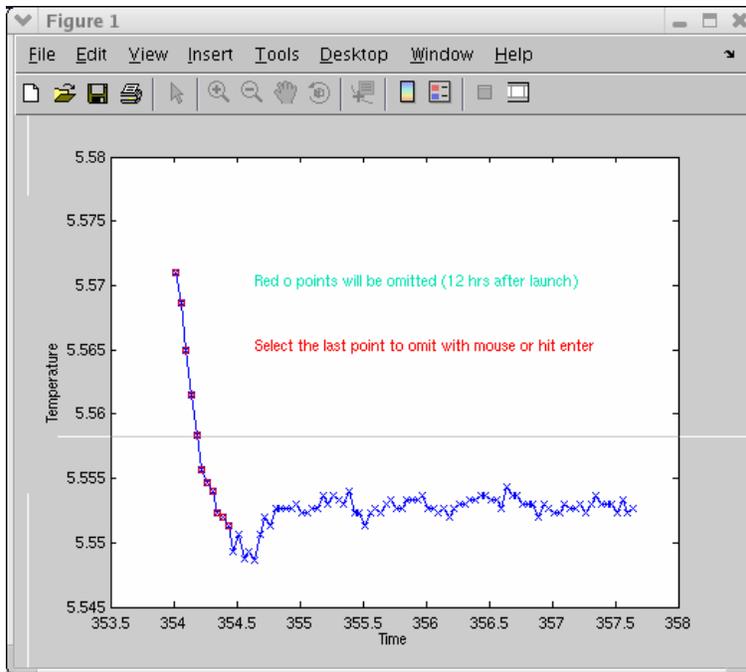


Figure 26 Figure window generated by Step 1 showing beginning of temperature time series with user selected points to omit shown in red

The IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. At the completion of this step, a Site1.mat file is created with the structure array contents shown in Figure 27. The output file and structure array names are derived from the site designator set in the IES_GUIDRIVER window (see Figure 15). The Site1.comment should read 'Created after windowing'. The log file (PIES076_1_processing.log) should have entries similar to Figure 28, listing launch and release times, time base information, number of gaps filled, the method for tau processing, and based on the interactive windowing, the bounds for tau, pressure and temperature and when temperature equilibrium was achieved. If a time offset had been used in processing, a line would appear in the log preceding the tau processing method stating "Time offset of XX minutes was included". **NOTE:** If a parameter could not be processed, for example, missing or bad tau, pressure, temperature, or velocity it will be noted in the log. Instrument 076_1 is a PIES, therefore the log (Figure 28) states that "Velocity data will not be processed".

```

// mbs
>> load Site1
>> Site1

Site1 =

    tau: [9607x1 double]
   taudd: [9607x1 double]
    tmp: [9588x1 double]
   tmpdd: [9588x1 double]
    prs: [9588x1 double]
   prsdd: [9588x1 double]
   refyr: 2004
  comment: {'Created after windowing'}

```

Figure 27 Example Site1.mat file contents after Step 1, windowing and hourly values

```

*****

Processing of 076_1 Site Site1
*****

Site ID is Site1
*****
STEP 1 processing run on 10-Apr-2006 12:23:53

Velocity data will not be processed

On bottom records are determined from launch and release times
Launch time: 2004,12,19,15,32,0 ==> decimal day = 3.536472e+02
Release time: 2006,1,24,1,23,0 ==> decimal day = 7.540576e+02

**** RELEASE information differs by 6.8667 minutes ****
User specified 754.0576 versus IES.LOG 754.0624

Time base:
Number of records added to fill time gaps in Travel Time: 2
Total Clock Drift: -0.000162 days (-0.233333 minutes)
Launch occurred 17 hours after instrument was turned on
Clock adjustment of -0.116667 minutes was applied
Number of records added to fill time gaps in Pressure and Temperature:
2

Tau processed using the QRT method
Tau bounds: 3.17517 to 3.19562

Pressure bounds: 2422.468182 to 2423.437879

Temperature bounds: 5.547049 to 5.571271

Temperature equilibrium reached about 19 hours after launch
*****
**** End of Processing ****

```

Figure 28 Example processing log after Step 1, windowing and hourly values

3.5.2 Step 2: Despiking

Step 2 is done to identify data spikes in the time series of travel time, temperature and pressure and uses either a slope or Tukey method to replace them with interpolated values. The slope method identifies outliers as data points that change more rapidly from the preceding value by more than the specified rate (tolerance). The Tukey method runs a median filter on the time series and identifies outliers as values that differ from the filtered record by the specified tolerance. Travel times can only be despiked using the slope method. Either method can be used to despike pressure and temperature.

To begin, enter the IES serial number and file index (e.g. **076_1**) and press the **Initialize** button in the upper middle of the IES_GUIDRIVER window. Change the upper right pull down menu to **Step 2: Begin at Despiking**. If you are running the steps consecutively, rather than all in one run, the second pull down menu automatically changes to **Stop before Initial Detiding**. If you have restarted IES_GUIDRIVER you will need to change the second step from **End of Processing** to **Stop before Initial Detiding**. Note, the values for some of the parameters are replaced with blanks in the GUI. However, the parameters in the processing program retain the values that were set in Step 1. If you wish to change a value, you can enter the new one at this time. Note that the tau processing method updates to “Do not reprocess” which refers to the processing that occurred in Step 1. Press the button **Begin Processing** in the lower left (Figure 29).

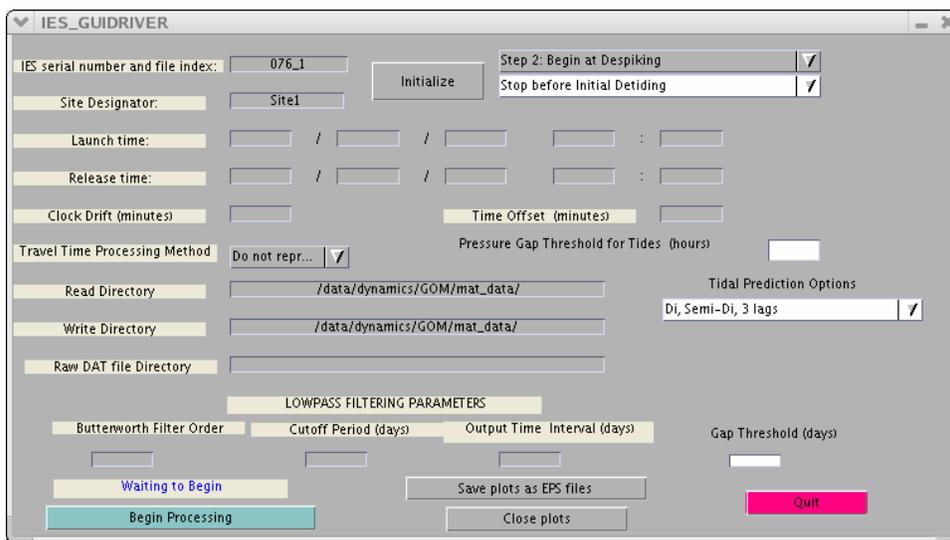


Figure 29 Example IES_GUIDRIVER window after initialization for processing Step 2 only

A new GUI window appears with a toolbar (Figure 30) showing the tau time series before despiking in the upper panel. On the right side of the GUI you can select the averaging length (default, 5), the tolerance (default, 0.0008 seconds) and the despiking method (travel times can only be despiked with the slope method). Using the GUI defaults, press the button **Run Despike**.

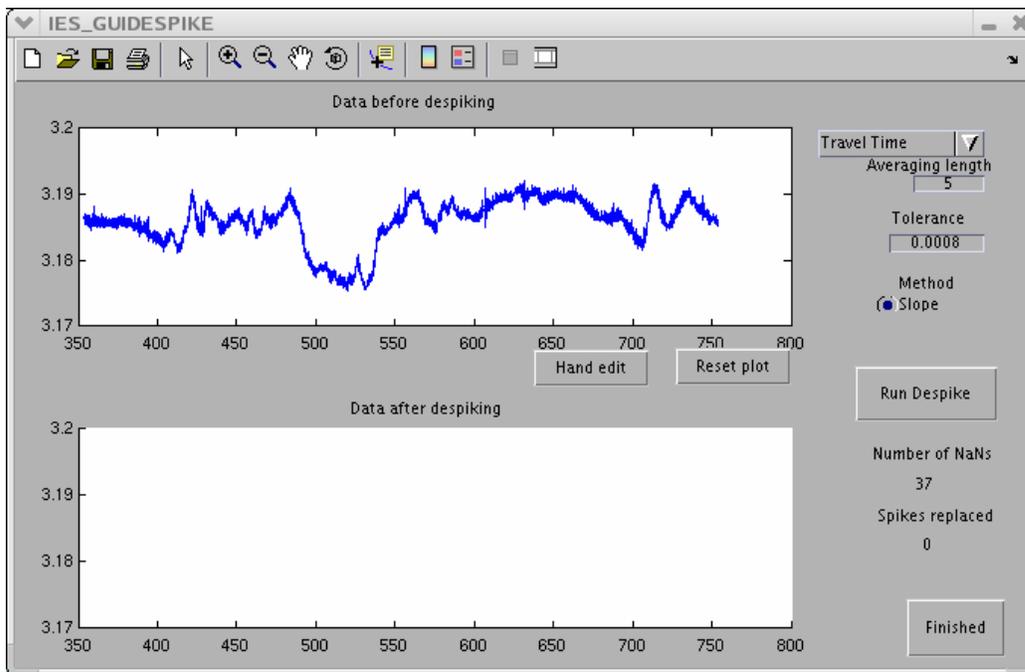


Figure 30 Example IES_GUIDESPIKE window showing data before despiking (top)

For 076_1, the default despiking replaces 5 data spikes and the new despiked time series appears in the lower plot of the GUI (Figure 31). If you are not satisfied with the despiked tau time series you can change the averaging length or tolerance. Decreasing the tolerance removes more data spikes from the time series.

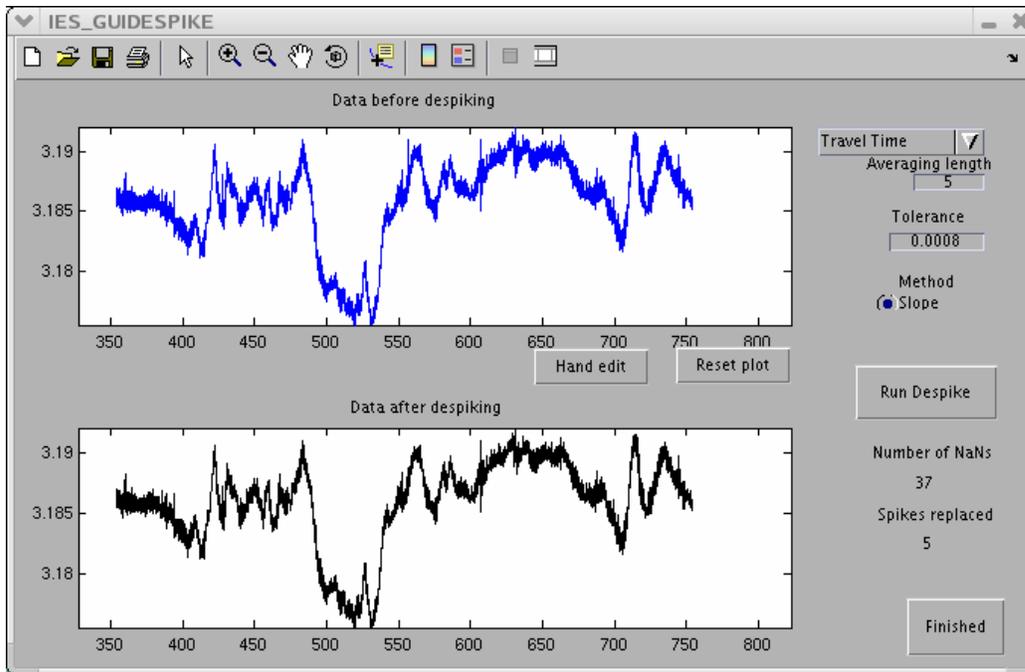


Figure 31 Example IES_GUIDESPIKE window showing data before (top) and after despiking (bottom)

If you wish to manually edit the time series, it is best to first zoom in using the MATLAB toolbar to show the area to be edited in more detail. Press the **Hand Edit** button between the two plots. The upper panel plot now shows the data before hand editing (Figure 32). A dialog box appears, directing the user to “Click on the good end points spanning the bad values to replace in the upper graph.” Click **OK**. **NOTE:** this dialog box only appears the first time you select the **Hand edit** button. Click on the left end point first, and then the right end point. The two points you selected with the mouse will then be circled in red in the upper panel. The new interpolated values will appear between the two circled red points in the lower plot (Figure 33). A dialog box will ask if you want to keep this interpolation. Click **Yes** if satisfied. Click **No** to reject the interpolation and restore to the values shown in the upper plot. Click **Replace with NaNs** if you want the interpolated values changed to NaNs.

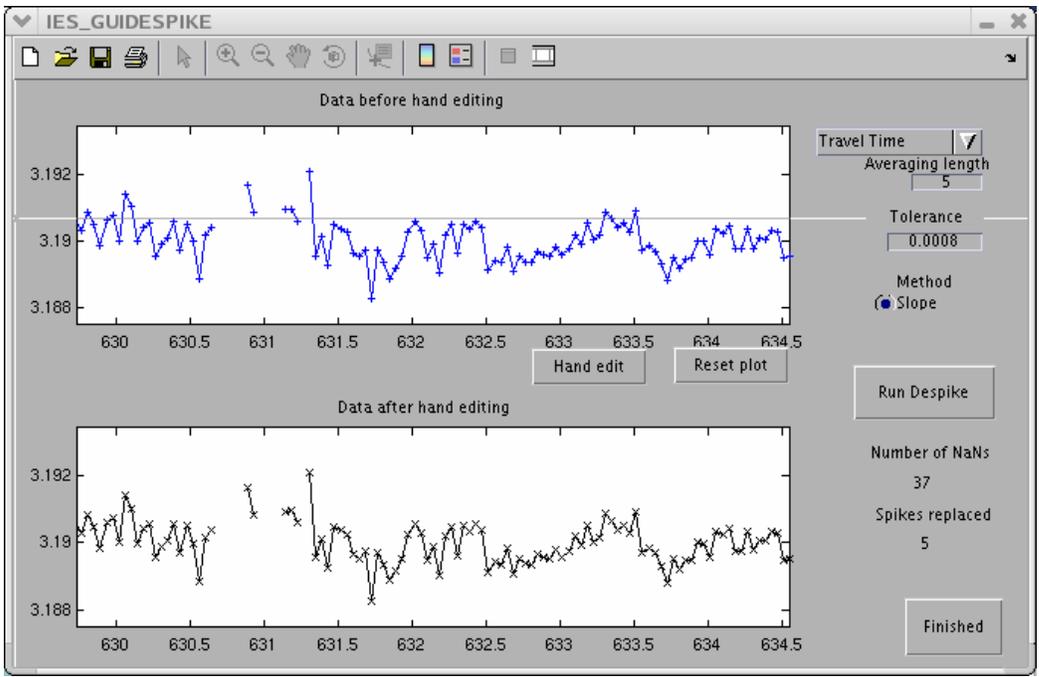


Figure 32 Example IES_GUIDESPIKE window showing expanded view for hand editing

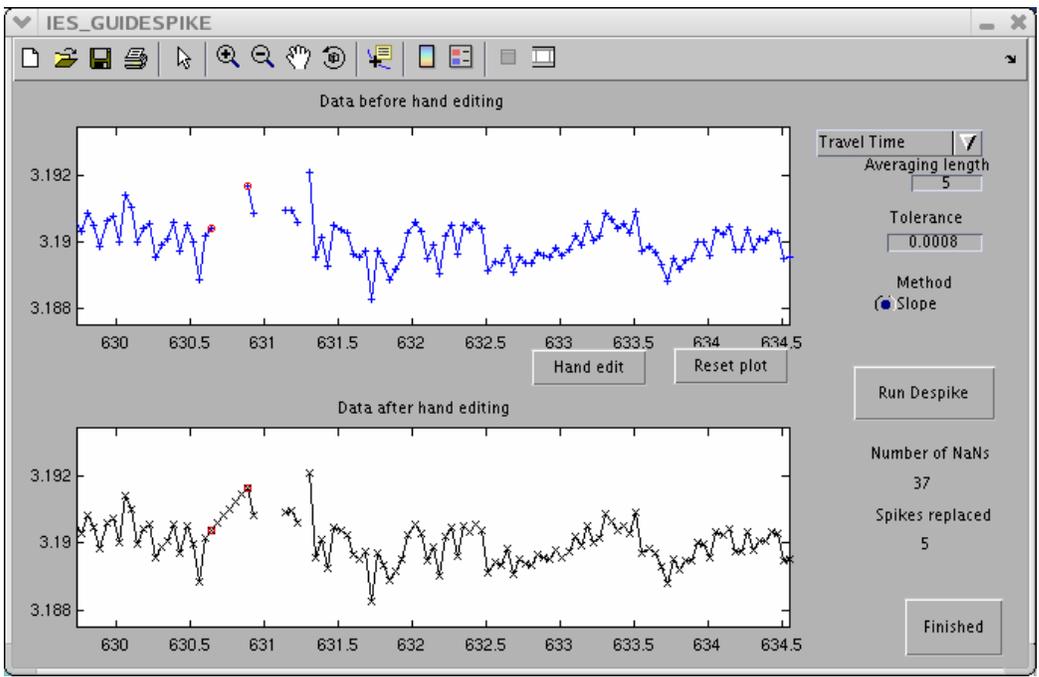


Figure 33 Example IES_GUIDESPIKE window showing data before (top) and after hand editing (bottom)

If you wish to despiking the pressure and or temperature time series **DO NOT** click **Finished**. Rather, go up to the top right pull down menu and select pressure or temperature. The default averaging lengths and tolerances are 5 and 0.5 for pressure and 5 and 0.1 for temperature. Repeat the steps as for tau. Click **Finished** when satisfied with *all three* time series. The IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. The Site1.mat file variables (tau, tmp and prs) are updated with the despiked data and the contents of the Site1 structure should now display the comment 'Created after despiking' (Figure 34). A summary of the step 2 processing should now be appended to the log file listing the number of NaNs, the tolerance, the averaging length and the numbers of records changed by travel time, pressure and temperature despiking (Figure 35).

```
>> load Site1
>> Site1

Site1 =
    tau: [9607x1 double]
   taudd: [9607x1 double]
    tmp: [9588x1 double]
   tmpdd: [9588x1 double]
    prs: [9588x1 double]
   prsdd: [9588x1 double]
   refyr: 2004
  comment: {'Created after despiking'}

>>
```

Figure 34 Example Site1.mat file contents after Step 2, despiking

```
*****
STEP 2 DESPIKE processing run on 24-Apr-2006 11:30:20

Travel Time Despiking:
NaNs (not replaced by despike = 33
tolerance = 0.00040
averaging length = 5
Total records changed: 116

Pressure Despiking:
NaNs (not replaced by despike = 2
tolerance = 0.50000
averaging length = 5
Total records changed: 3

Temperature Despiking:
NaNs (not replaced by despike = 5
tolerance = 0.02500
averaging length = 5
Total records changed: 23
*****
**** End of Processing ****
```

Figure 35 Example processing log after Step 2, despiking

3.5.3 Step 3: Initial Detiding

This step removes the tidal contribution from the pressure record (despiked but still containing drift) using a FORTRAN program called RESPO.for (Response Analysis of Tides), based on the work of Munk and Cartwright (1966). Response analysis constructs and applies a predictive filter which represents the ocean's response to gravitational forcing. Unlike the related harmonic analysis, the response analysis assumes nothing about which frequencies are present, because the input function is derived directly from Newtonian-Keplerian orbital motions; the input function contains all the variations of the astronomic forcing regardless of size. The oceanic response is considered distinctly from the astronomic forcing. The method also has a more physical basis than harmonic analysis since it treats the ocean as a dynamical system. Details of the response method are given in Appendix C.

A pull down menu provides three tidal prediction options in the IES_GUIDRIVER window:

Di, Semi-Di, 3 lags (default)

Di, Semi-Di, 5 lags

Di, Semi-Di, Mf, Mssa, 5 lags

where Di refers to diurnal, Semi-Di to semi-diurnal, Mf to lunisolar fortnightly, and Mssa to solar semi-annual.

The tidal prediction option to use depends on the length of the pressure time series. Table 3 provides recommended guidelines for choosing the tidal prediction option from the pull down menu.

Record length	Tidal Prediction Option
<1 year	Di, Semi-Di, 3 lags
> 1 year and < 2 years	Di, Semi-Di, 5 lags
< 2 years *	Di, Semi-Di, Mf, Mssa, 5 lags

Table 3 Record Length for Tidal Prediction Options

*Di, Semi-Di, Mf, Mssa, 5 lags may be used for records as short as 1.5 years, but it is crucial that the instrumental drift be accurately estimated and removed. A warning statement will be written to the processing log if the record length is shorter than 1.5 years and the user opted to remove the fortnightly and semi-annual tides.

Currently, these options calculate the total gravitational potential for the lunar and solar tidal constituents. In the future, an option may be added to calculate the sun's radiational potential. Listings of the files associated with the three tidal prediction options are included in Appendix C-2 as well as a description of the file contents.

Warning: The third processing option (Di, Semi-Di, Mf, Mssa) calculates the fortnightly and longer period tides. The predicted fortnightly and longer period tides output by RESPO have been compared with results from harmonic analysis and are computed correctly. However, the phases and amplitudes returned by RESPO for these constituents are incorrect at this time. There appears to be a bug in the RESPO subroutine which returns the phases and amplitudes for the fortnightly and longer period tides. Work is ongoing to resolve this problem for the next version of the IES processing package.

To begin detiding, enter the IES serial number and file index (e.g. **076_1**) and initialize the IES_GUIDRIVER window. Change the upper right pull down menu to **Step 3: Begin at Initial Detiding**. If you are running the steps consecutively, rather than all in one run, the second pull down menu automatically changes to **Stop before Dedrifting**. If you have restarted IES_GUIDRIVER you will need to change the second step from **End of processing** to **Stop before Dedrifting**. The values for some of the parameters are replaced with blanks in the GUI. However, the values set previously are retained. **NOTE:** The value for the pressure gap threshold for tides (maximum allowable data gap size that

can be linearly interpolated before performing the tidal analysis) was likely set in Step 1 (see middle of the IES_GUIDRIVER window in Figure 15). The default value is 24 hours. Press the button **Begin Processing** in the lower left of the IES_GUIDRIVER window (Figure 36).

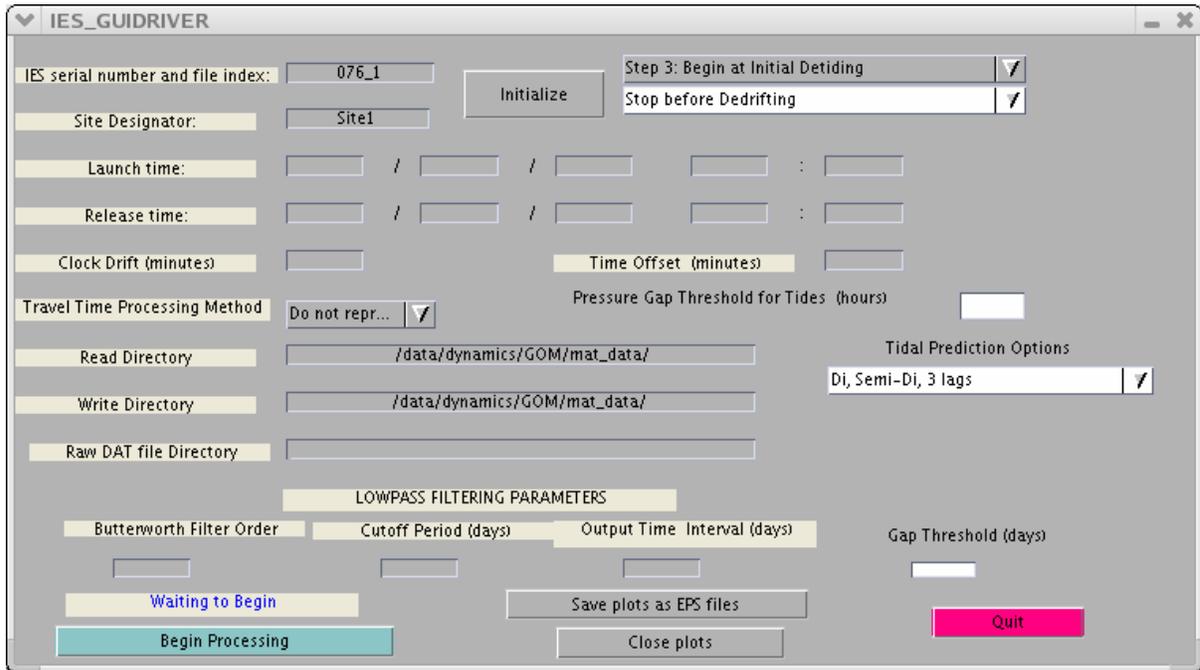


Figure 36 Example IES_GUIDRIVER window after initialization for processing Step 3, initial detiding only

There is no further user input for this step. The RESPO.for program creates a detailed log file with the filename derived from the site designator ending with .tidelog, (e.g., Site1.tidelog) which is written to the wrkspc subdirectory.

If your data contains gaps larger than the threshold (default = 24 hours), this step performs the tidal analysis on the longest segment of the time series. Then the calculated weights are passed to RESPO.for and the tide is predicted for the full record. A second log file is created with “_fullrecord.tidelog” as part of the name.

The IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. The Site1.mat file (Figure 37) now contains the detided pressure data (Site1.prs), the time series of the predicted tide (Site1.tide), and the tidal amplitudes (Site1.amplitude) and phases (Site1.phase) extracted from the separate log file created by

the RESPO program (Site1.tidelog). Figure 38 lists amplitudes (in dbar) and phases (in degrees) calculated for the Site 1 example. The Site1 structure comment will consist of two lines containing 'Created after initial detiding' and 'prs=prs-tide'. You may want to write down the values of Site1.amplitude and Site1.phase for later comparison with the values calculated after step 5 (final detiding). The default case (Di, Semi-Di, 3 lags) provides the amplitudes and phases for the tidal constituents listed in Table 4.

```

>>
>> Site1
Site1 =
    tau: [9607x1 double]
    taudd: [9607x1 double]
    tmp: [9588x1 double]
    tmpdd: [9588x1 double]
    prs: [9588x1 double]
    prsdd: [9588x1 double]
    refyr: 2004
    comment: {2x1 cell}
            tide: [9588x1 double]
    amplitude: [1x1 struct]
    phase: [1x1 struct]

>> Site1.comment
ans =
    'Created after initial detiding'
    'prs = prs - tide'

```

Figure 37 Example Site1.mat file contents after Step 3, detiding

```

>>
>> Site1.phase
ans =
    O1: 11.7340
    K1: 18.6000
    Q1: 1.6270
    P1: 19.3580
    M2: 102.9810
    K2: 103.3770
    N2: 115.5460
    S2: 102.6960

>> Site1.amplitude
ans =
    O1: 0.1419
    K1: 0.1424
    Q1: 0.0325
    P1: 0.0482
    M2: 0.0237
    K2: 0.0060
    N2: 0.0046
    S2: 0.0208

```

Figure 38 Example Site1.mat tidal phases and amplitudes

Name	Symbol	Approximate Period (Solar Hours)
Semi-diurnal:		
Principal lunar	M2	12.4
Principal solar	S2	12.0
Larger lunar elliptic	N2	12.7
Luni-solar semi-diurnal	K2	11.97
Diurnal:		
Luni-solar diurnal	K1	23.9
Principal lunar diurnal	O1	25.8
Principal solar diurnal	P1	24.1
Larger lunar elliptic	Q1	26.9

Table 4 Tidal Constituents Provided by Default Processing

The processing log file (Figure 39) will list the gap threshold, the tidal constituents removed (default: diurnal and semi-diurnal), and the maximum admittance control value (default: 3 lags). Only pressure data are detided.

STEP 3 INITIAL DETIDE processing run on 28-Apr-2006 11:53:28

Data gaps less than 24 hours will be interpolated before detiding
 Diurnal and semidiurnal tides removed
 Maximum Lags for Admittance: 3

Figure 39 Example processing log after Step 3, initial detiding

3.5.4 Step 4: Dedrifting

This step fits and removes drift from the pressure record. Drifts are typically associated with variations in the properties of the pressure sensor crystal over long time-scales or slight imperfections in the IES master clock. The pressure records can either be dedrifted by a linear-exponential function or by double exponential functions. The linear exponential function method was recommended by Watts and Kontoyiannis (1990) who have examined sensor drift and performance where the rate of drift decays with time and is best approximated by a function of the form

$$Drift = Ae^{Bt} + Ct + D$$

The double exponential function has the form

$$Drift = Ae^{Bt} + Ce^{Dt} + E$$

The processing code calculates the initial estimates for the coefficients A, B, C, D and E using one of two MATLAB m-files. If the MATLAB optimization toolbox is available, `fmincon.m` is used to calculate these parameters. If the optimization toolbox is not available, the program `fminsearchbnd.m` is used instead and a message appears in the GUI window.

Real ocean signals may appear like instrumental drift. For preliminary drift removal, choose the exponential-linear equation and allow MATLAB to select the drift coefficients. Subsequently, comparisons with coincident current measurements should be made to determine if the preliminary drift curve contains any ocean signal. If so, it will be necessary to dedrift the pressure record again.

To begin, enter the IES serial number and file index (e.g. 076_1) and press the **Initialize** button in the upper middle of the IES_GUIDRIVER window. Change the upper right pull down menu to **Step 4: Begin at Dedrifting**. If you are running the steps consecutively, the second pull down menu automatically changes to **Stop before Final Detiding**. If you have restarted IES_GUIDRIVER you will need to change the second step from **End of Processing** to **Stop before Final Detiding**. Press the button **Begin Processing** in the lower left of the IES_GUIDRIVER window (Figure 40).

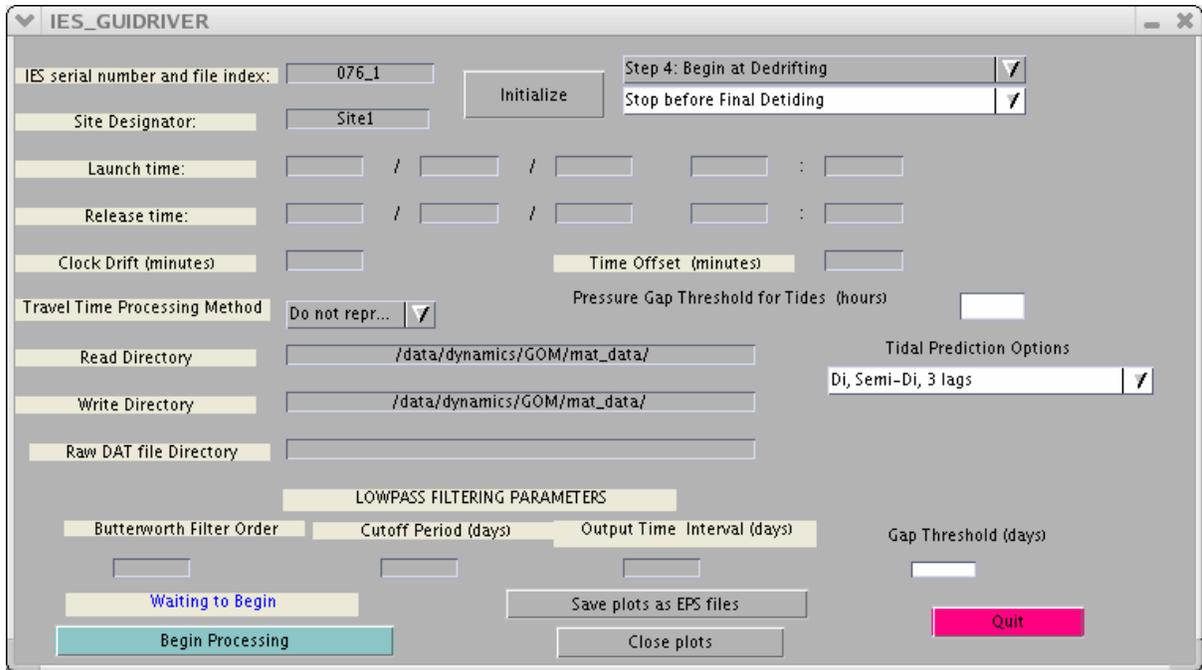


Figure 40 Example IES_GUIDRIVER window after initialization for processing Step 4, dedrifting only

The IES_GUIDEDRIFT2EXP window appears (Figure 41) showing the data prior to fitting drift in the left plot. Below the plot, the default method of dedrifting is selected (linear-exponential) with initial guesses for the coefficients A, B, C and D.

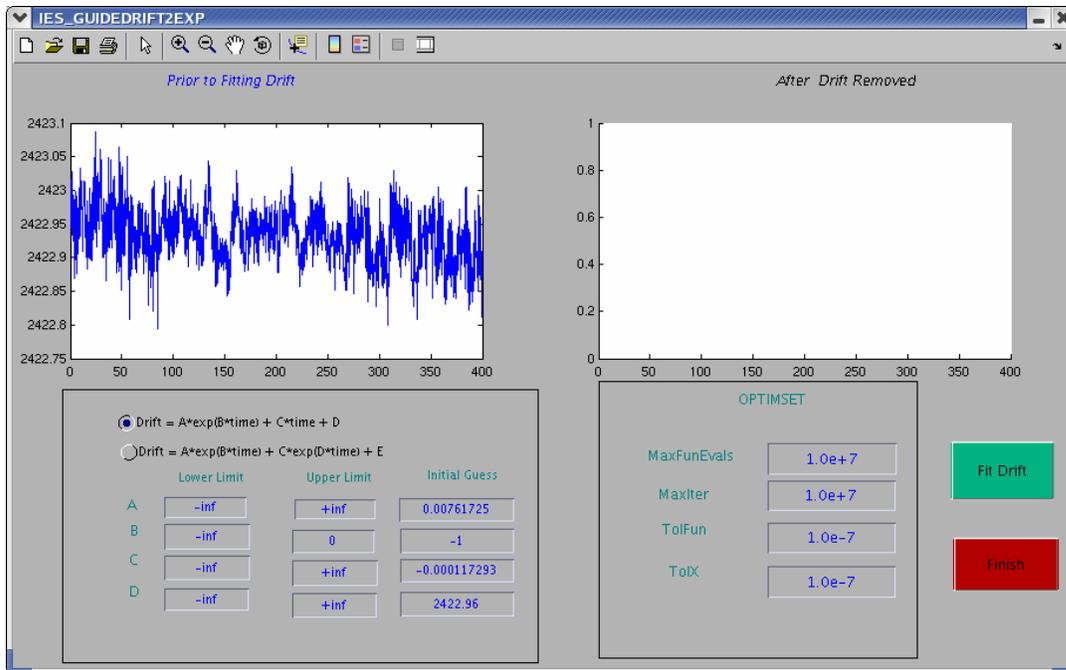


Figure 41 Example IES_GUIDEDRIFT2EXP window prior to fitting drift

The user can modify the OPTIMSET parameters on the right in Figure 41 and/or specify constraints on the coefficients A-D in the GUI. The OPTIMSET parameters are defined in MATLAB (type *help optimset* for more information) as follows:

MaxFunEvals – Maximum number of function evaluations allowed (positive integer)

MaxIter – Maximum number of iterations allowed (positive integer)

TolFun – Termination tolerance on the function value (positive scalar)

TolX – Termination tolerance on X (positive scalar)

Press the **Fit Drift** button. The drift curve (in red) is superimposed on the data in the “Prior to Fitting Drift” plot and the dedrifted data (data minus drift) is plotted in the “After Drift Removed” plot (Figure 42).

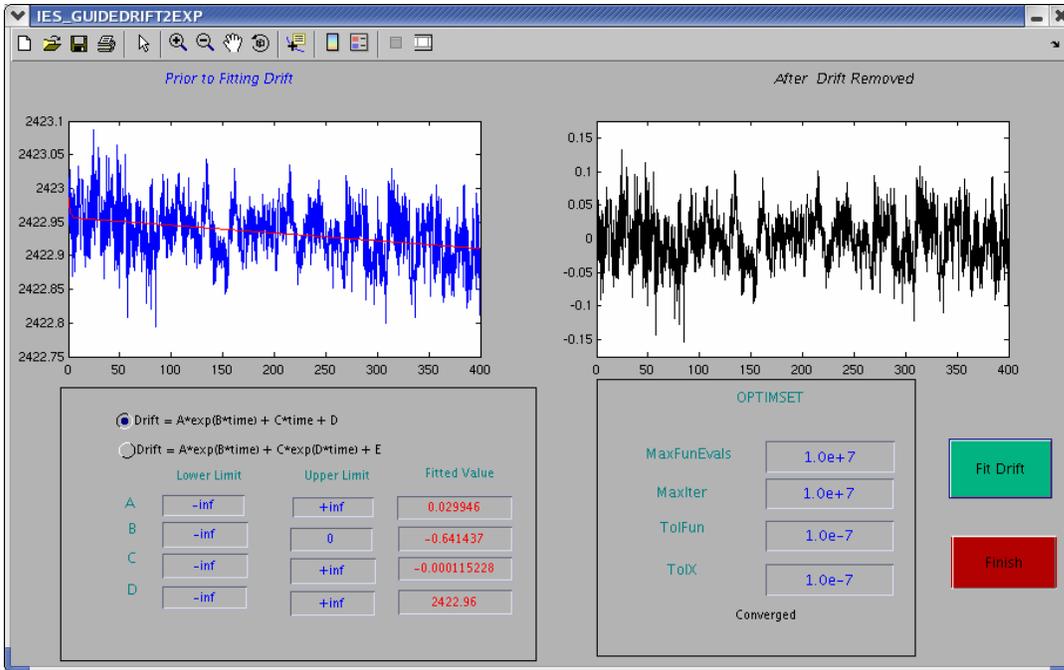


Figure 42 Example IES_GUIDEDRIFT2EXP window showing data prior to fitting drift (left) and after drift removed (right)

Note, the Initial Guesses (Figure 41) have been replaced with Fitted Values (Figure 42) and at the bottom of the OPTIMSET box “Converged” should appear. In general, the OPTIMSET parameters should be appropriate. However, if the GUI seems to hanging, decreasing MaxIter and increasing TolX (in the OPTIMSET box) may help. Use the toolbar to zoom in on the beginning of the pressure record in the left panel to examine the fit (Figure 43). You can constrain the values of the A, B, C and D coefficients to fall within certain limits. For each coefficient you can specify upper and lower limits. For example, if you believe that a linear fit better describes the entire record, change the upper and lower limits of A or B to zero in the IES_GUIDEDRIFT2EXP window and press **Fit Drift** again. The Fitted Values should then be zero.

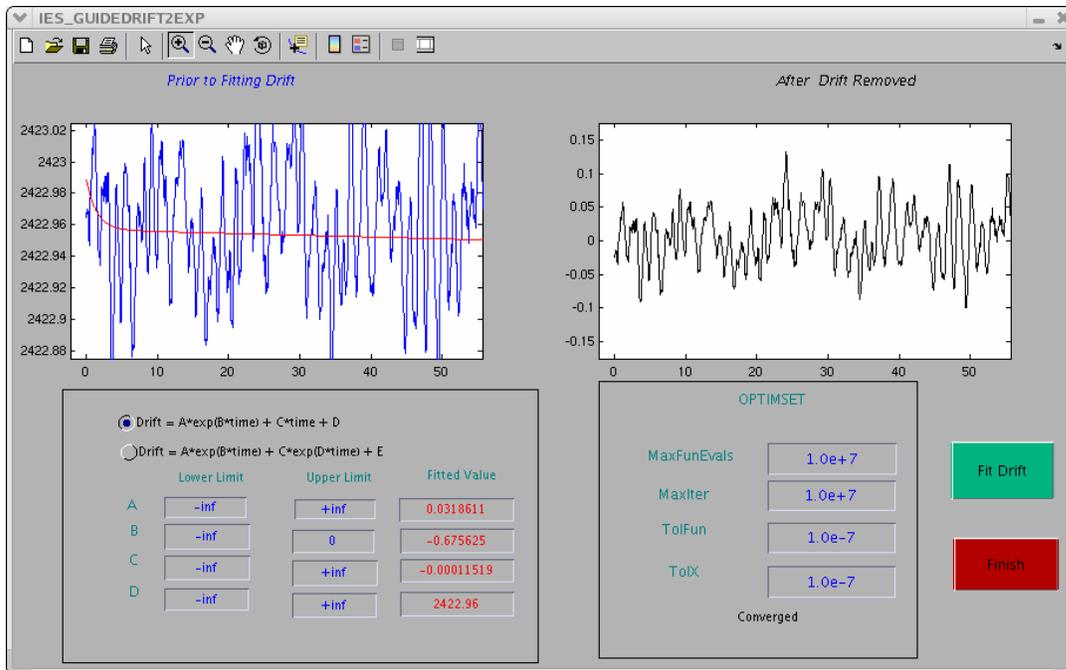


Figure 43 Example IES_GUIDEDRIFT2EXP window showing expanded view of data prior to fitting drift (left) and after drift removed (right)

Press the **Finish** button. The IES_GUIDEDRIFT2EXP window will close and the IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. The Site1.mat file (Figure 44) now contains the dedrifted pressure data (Site1.prs), the time series of the drift (Site1.drift) and the drift coefficients (Site1.driftcoef). The Site1 structure comment will read 'Created after dedrifting', 'prs=prs-tide-drift' (Figure 44) and the values of the drift coefficients are appended to the log file (Figure 45).

```

Site1 =
    tau: [9607x1 double]
    taudd: [9607x1 double]
    tmp: [9588x1 double]
    tmpdd: [9588x1 double]
    prs: [9588x1 double]
    prsdd: [9588x1 double]
    refyr: 2004
    comment: {2x1 cell}
        tide: [9588x1 double]
    amplitude: [1x1 struct]
    phase: [1x1 struct]
    drift: [9588x1 double]
    prsave: 2.4229e+03
    driftcoef: [1x1 struct]

>> Site1.comment
ans =
    'Created after dedrifting'
    'prs = prs - tide - drift'

```

Figure 44 Example Site1.mat file contents after Step 4, dedrifting

STEP 4 DEDRIFT processing run on 10-May-2006 12:47:38

```

Drift equation: A*exp(B*time) + C*time + D
Drift coefficients A = 3.186259e-02
Drift coefficients B = -6.753087e-01
Drift coefficients C = -1.151889e-04
Drift coefficients D = 2.422957e+03
*****
**** End of Processing ****

```

Figure 45 Example processing log after Step 4, dedrifting

3.5.5 Step 5: Final Detiding

After the pressure record has been dedrifted, it is customary to detide the record again to improve the tide prediction. This step adds the tides removed by initial detiding (Section 3.5.3, Step 3) to the dedrifted pressure record and then recalculates the tides.

To begin, enter the IES serial number and file index (e.g. 076_1) and initialize the GUI window. Change the upper right pull down menu to **Step 5: Begin at Final Detiding**. If you are running the steps consecutively, the second pull down menu automatically changes to **Stop before Velocity Processing**. If you have restarted IES_GUIDRIVER you will need to change the second step from **End of Processing** to **Stop before Velocity Processing**. Press the button **Begin Processing** in the lower left (Figure 46). There is no further user input for this step.

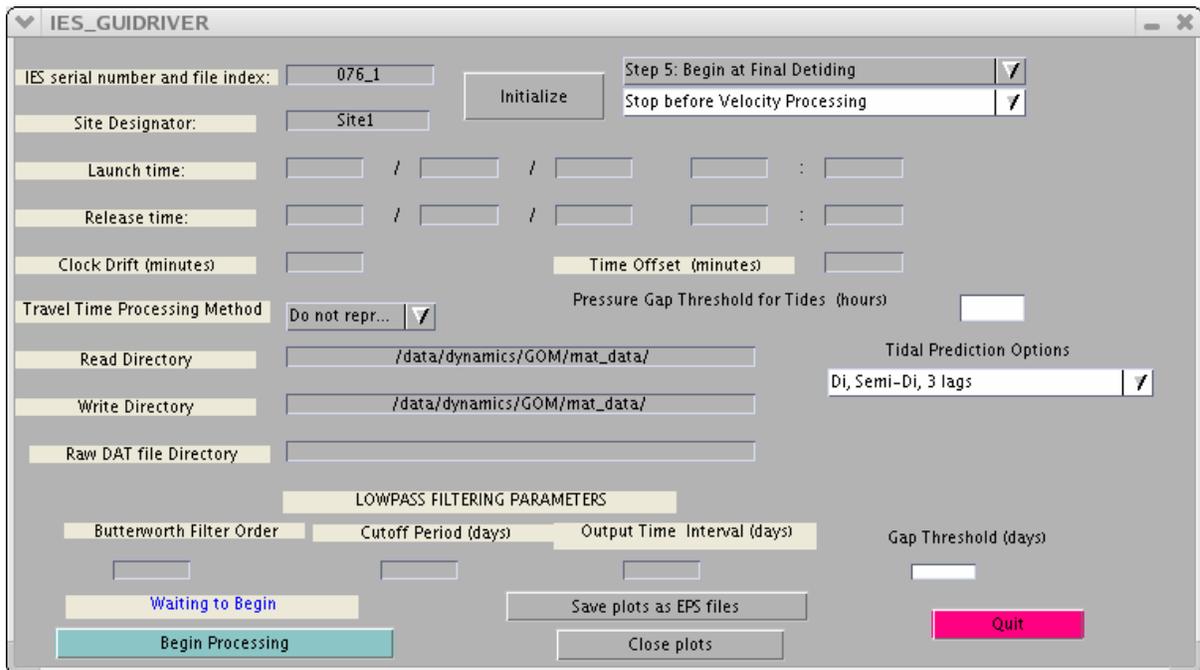


Figure 46 Example IES_GUIDRIVER window after initialization for processing Step 5, final detiding only

The IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. The Site1.mat file (Figure 47) now contains the dedrifted pressure data after final detiding (Site1.prs). The Site1 structure comment will read 'Created after final dedrifting', 'prs=prs-tide-drift' (Figure 47). Site1.tide now contains the recalculated time series of the tide. The amplitudes (Site1.amplitude) and phases

(Site1.phase) of the tidal constituents have been overwritten with the recalculated values. The amplitudes and phases after final detiding will likely show only slight differences from those calculated in Step 3 (initial detiding).

If your data contains gaps larger than the threshold (default = 24 hours), processing will again be done on the longest segment as was done in Step 3, Initial Detiding. Once the weights are calculated for the longest segment they are passed to RESPO.for and the tide is predicted for the full record. A second log file is created with “_fullrecord.tidelog” as part of the name whenever there are gaps greater than the threshold.

```
>>
>> load Site1
>> Site1

Site1 =

    tau: [9607x1 double]
   taudd: [9607x1 double]
    tmp: [9588x1 double]
   tmpdd: [9588x1 double]
    prs: [9588x1 double]
   prsdd: [9588x1 double]
   refyr: 2004
  comment: {2x1 cell}
    tide: [9588x1 double]
  amplitude: [1x1 struct]
    phase: [1x1 struct]
    drift: [9588x1 double]
   prsave: 2.4229e+03
  driftcoef: [1x1 struct]

>>
>> Site1.comment

ans =

'Created after final detiding'
'prs = prs - tide - drift'
```

Figure 47 Example Site1.mat file contents after Step 5, final detiding

The log file will have appended step 5 processing information (Figure 48) and state that the tidal coefficient amplitudes and phases were recalculated after drift removal.

```
*****
STEP 5 FINAL DETIDE processing run on 10-May-2006 13:25:46

Data gaps less than 24 hours will be interpolated before detiding
Diurnal and semidiurnal tides removed
Maximum Lags for Admittance: 3
Tidal coefficient amplitudes and phases were recalculated after drift
removal
*****
**** End of Processing ****
```

Figure 48 Example processing log after Step 5, final detiding

3.5.6 Step 6: Velocity Processing

Velocity processing calculates hourly averages of the velocity components if the instrument is a CPIES and “Currents” were selected at the DAT2MAT stage of processing (see Figure 5). The example instrument used in this manual, P076_1, is a PIES rather than a CPIES. Attempting velocity processing on a PIES will not result in an error, but data are not modified and no information is appended to the processing log. **NOTE**, the log created after Step 1 for 076_1 showed that velocity data would not be processed (refer to Figure 28 Section 3.5.1). To illustrate velocity processing a different instrument (a CPIES, 102_1) will be used.

To begin, enter the IES serial number and file index and press the **Initialize** button in the GUI window (Figure 49). Change the upper right pull down menu to **Step 6: Begin at Velocity Processing**. If you are running the steps consecutively, the second pull down menu automatically changes to **Stop before Lowpass Filtering**. If you have restarted IES_GUIDRIVER you will need to change the second step from **End of Processing** to **Stop before Lowpass Filtering**. Press the button **Begin Processing** in the lower left (Figure 49).

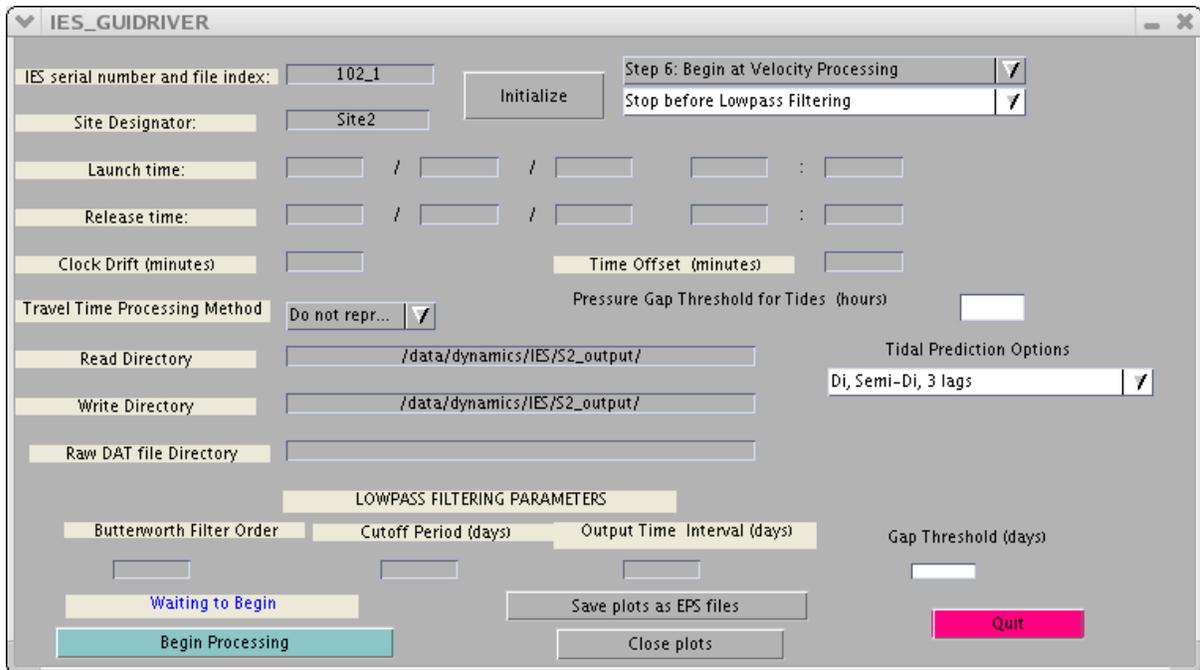


Figure 49 Example IES_GUIDRIVER window after initialization for processing Step 6, Velocity Processing Only

This step starts with determining when the current meter temperature has reached equilibrium with the surroundings (similar to Step 1 for the IES Paroscientific temperature sensor, Figure 25 in Section 3.5.1). Equilibration occurs more rapidly than for the Paroscientific sensor because the current meter temperature sensor is in direct contact with the water. Follow the directions in the Figure window (see Figure 50) to select the last temperature record to omit prior to equilibrium with surroundings and hit **Enter** in the plot window.

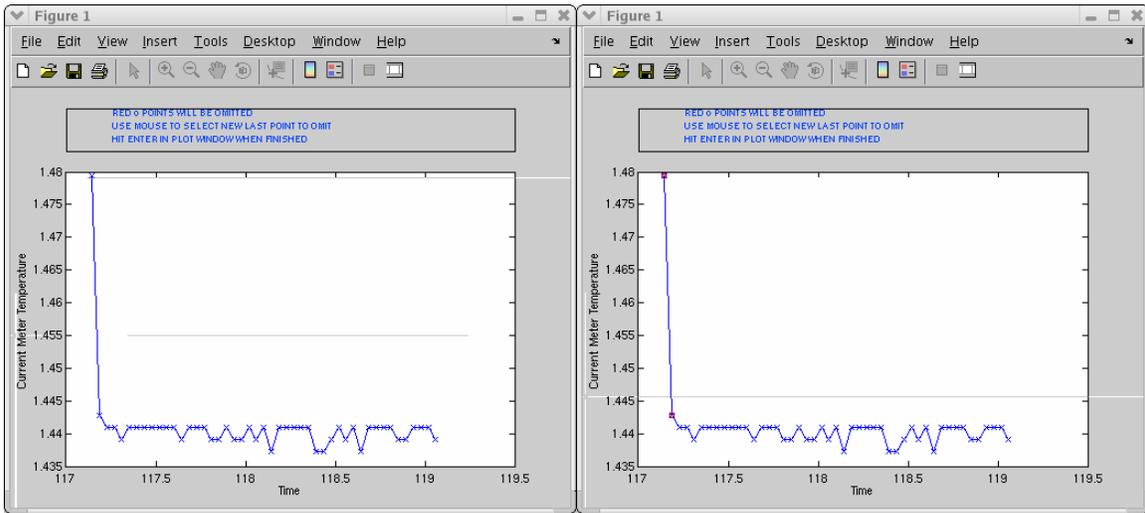


Figure 50 Figure windows generated by velocity processing showing beginning of DCS temperature time series with points to omit shown in red

Next the despiking GUI window appears (Figure 51). The variables u , v , and current meter temperature can be despiked using either the slope or Tukey methods (similar to step 2 for despiking temperature and pressure). The default averaging lengths and tolerances for u and v are 5 and 5. For IES 102_1, no spikes are replaced with these defaults for either u or v (Figure 52 and Figure 53). However, more experience with current meter processing is needed before the defaults will be changed.

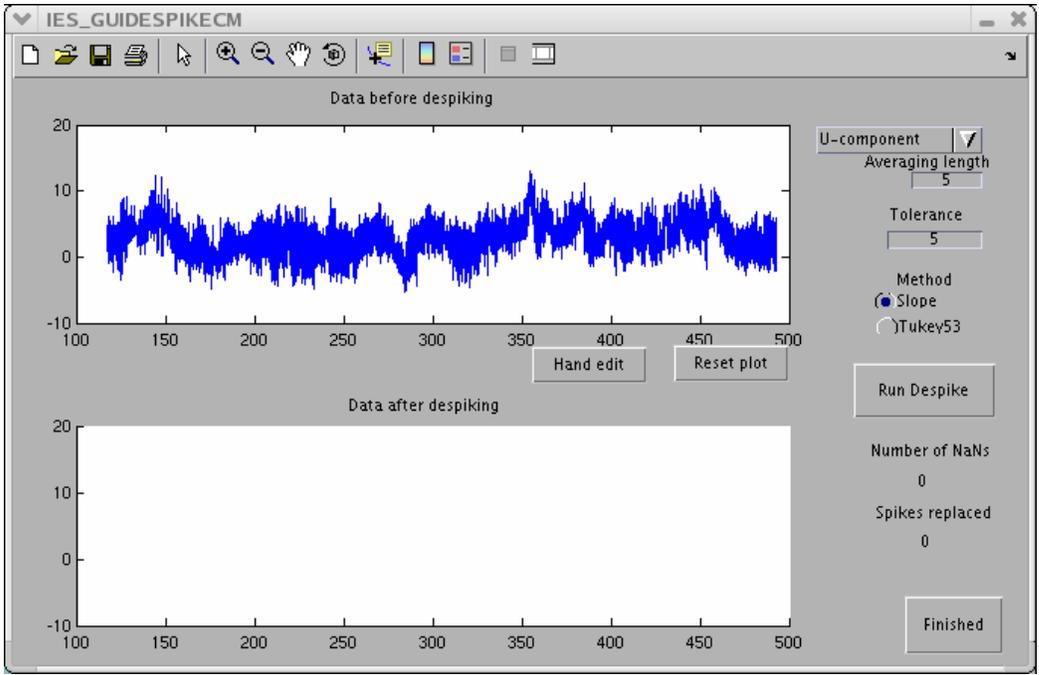


Figure 51 Example of IES_GUIDESPIKE CM window showing data before despiking (top)

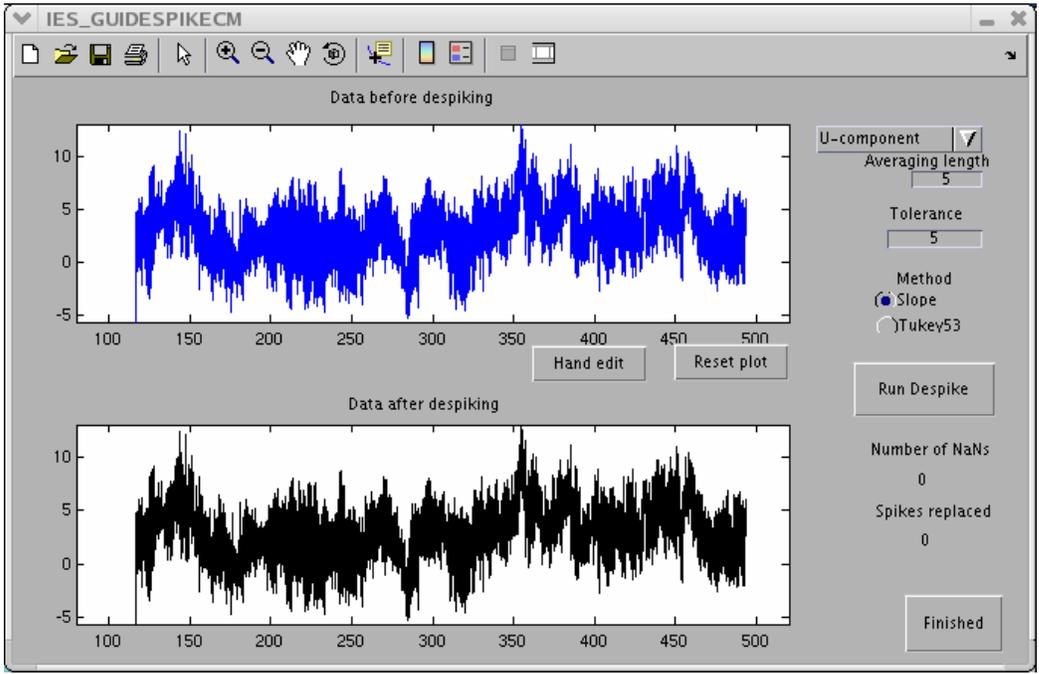


Figure 52 Example IES_GUIDESPIKE CM window showing U component data before (top) and after despiking (bottom)

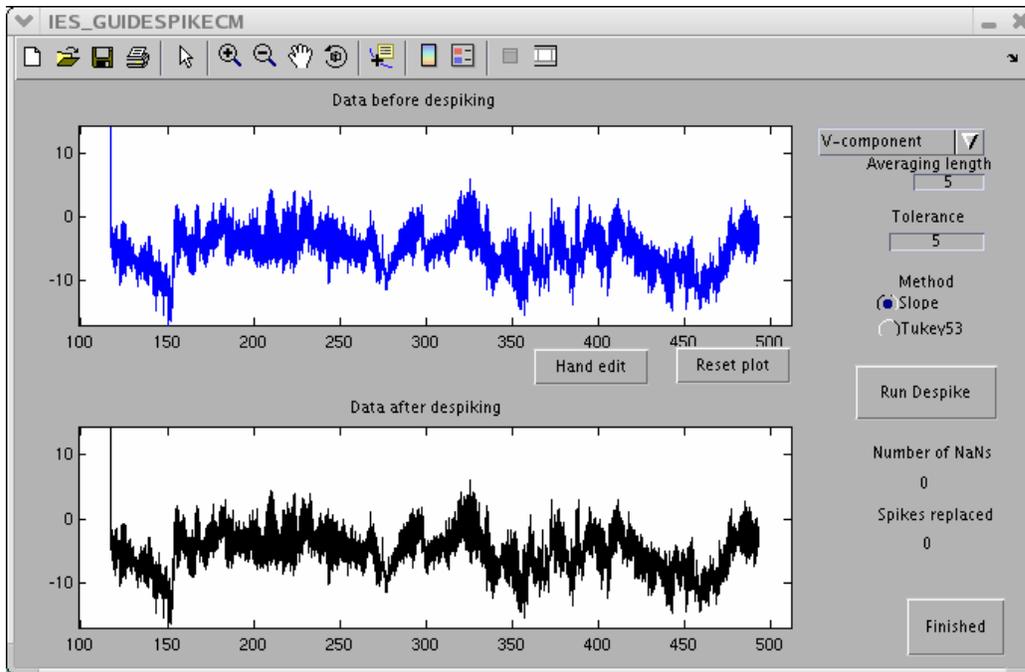


Figure 53 Example IES_GUIDESPIKE window showing V component data before (top) and after despiking (bottom)

Close examination of Figure 53 shows some spikes that should be hand edited (similar to hand editing in Step 2). It is not uncommon to need to edit a few points at the beginning and end of the record. Hand editing of the v component at the beginning of the time series shortens the v time series (Figure 54) and subsequently shortens the u time series as well, since u and v share the same time base. However, only the beginning and end of the u and v records are changed together. A spike in the middle of a u record must be changed separately from a v spike.

DCS temperature may have a different time base than u and v and is also treated separately. Default averaging length and tolerance for current meter temperature despiking are currently set at 5 and 0.1 deg (Figure 55). These values are likely too large, but again more experience is needed to adopt new defaults. For comparison a tolerance of .001 deg is shown in Figure 56.

Click **Finished** in the IES_GUIDESPIKECM window when satisfied with *all three* time series. The IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. The Site2.mat file will now have added current meter variables (u, v, cmtmp, see Figure 57). The Site2 comment should include 'Created after Velocity Processing'. A summary of the step 6 processing should now be appended

to the log file including when the current meter temperature reached equilibrium, the tolerances and averaging lengths used in despiking, the number of spikes replaced and if the time series were truncated (Figure 58).

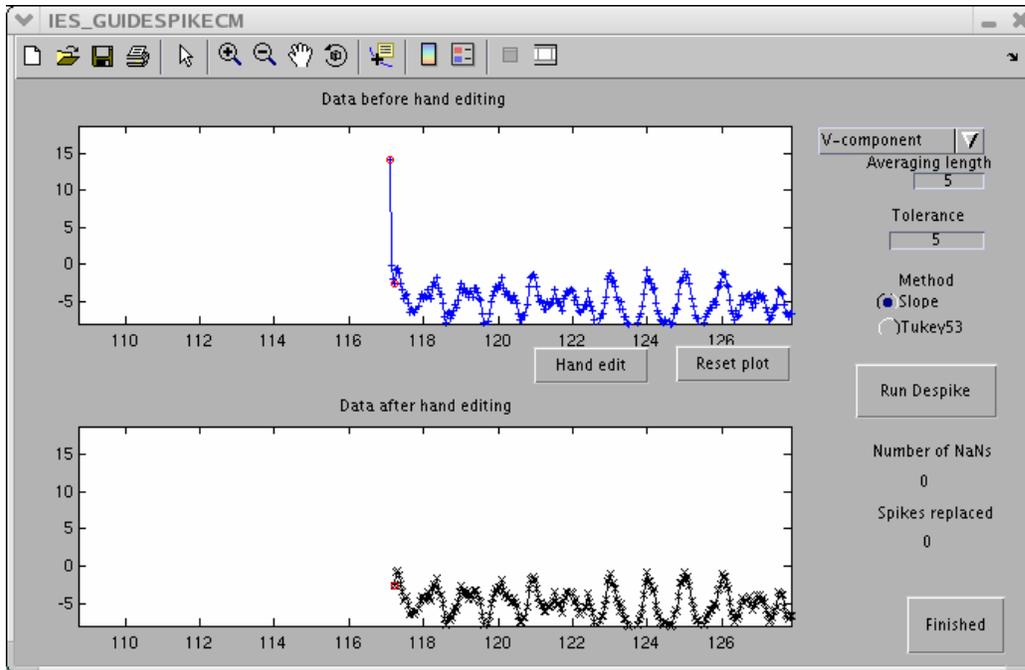


Figure 54 Example of IES_GUIDESPIKE CM window showing expanded view for V component hand editing

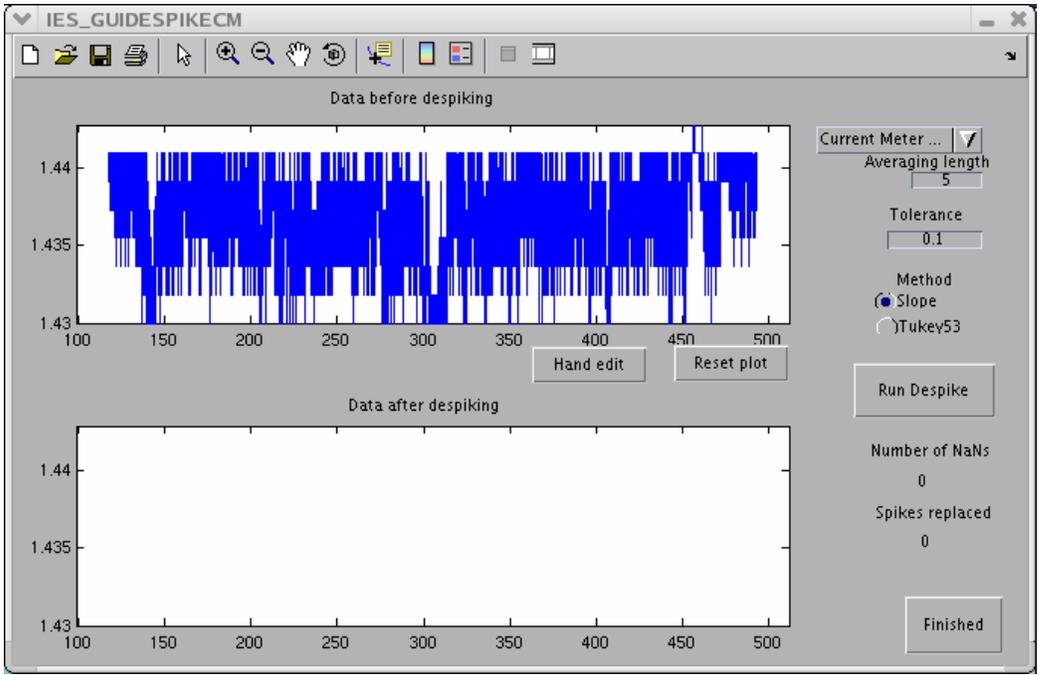


Figure 55 Example of IES_GUIDESPIKE CM window showing DCS temperature data before despiking (top)

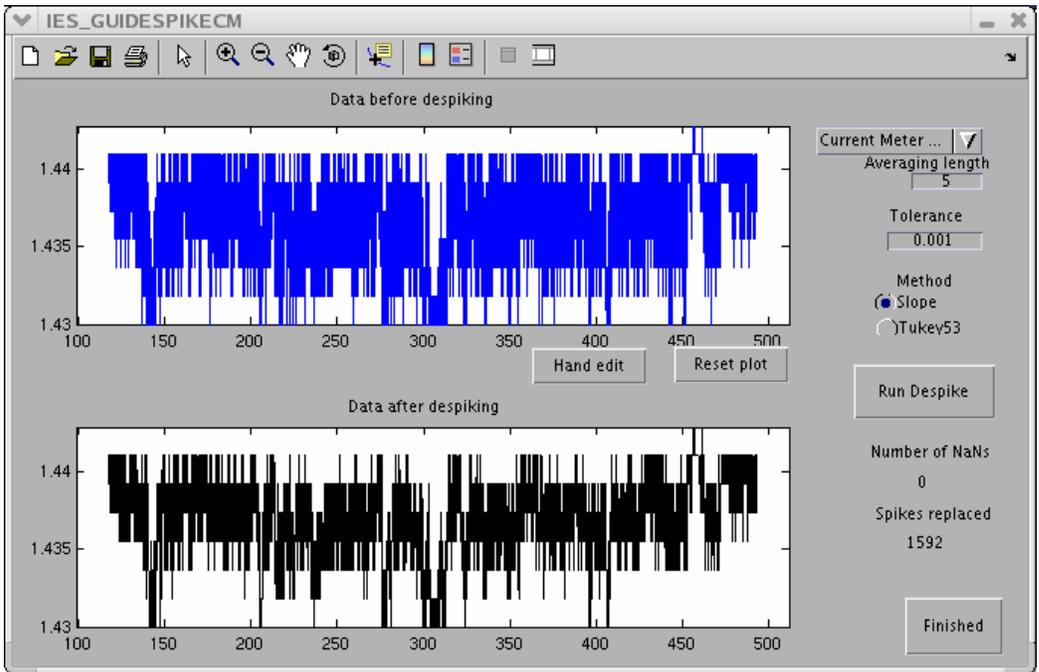


Figure 56 Example IES_GUIDESPIKE CM window showing DCS temperature data before (top) and after despiking (bottom)

```

>> load Site2
>> Site2

Site2 =

    tau: [10039x1 double]
    taudd: [10039x1 double]
    tmp: [10020x1 double]
    tmpdd: [10020x1 double]
    prs: [10020x1 double]
    prsdd: [10020x1 double]
    refyr: 2004
    comment: (2x1 cell)
    tide: [10020x1 double]
    amplitude: [1x1 struct]
    phase: [1x1 struct]
    drift: [10020x1 double]
    prsave: 5.7312e+03
    driftcoef: [1x1 struct]
    uvdd: [9024x1 double]
    u: [9024x1 double]
    v: [9024x1 double]
    cmtmp: [9024x1 double]
    cmtmpdd: [9024x1 double]

>> Site2.comment

ans =

    'Created after velocity processing'
    'prs = prs - tide - drift'

>>

```

Figure 57 Example of Site2.mat file contents after step 6, velocity processing

```

*****

STEP 6 CURRENT processing run on 17-May-2006 14:35:17

Current meter temperature equilibrium reached about 3 hours after
launch

U-component Despiking:
NaNs (not replaced by despike = 0
tolerance = 5.00000
averaging length = 5
Total records changed: 0

V-component Despiking:
NaNs (not replaced by despike = 0
tolerance = 5.00000
averaging length = 5
Total records changed: 4
V-component record shortened to remove leading or trailing NaNs
U- and V-component record lengths forced to agree

Current Meter Temperature Despiking:
NaNs (not replaced by despike = 0
tolerance = 0.00100
averaging length = 5
Total records changed: 1592
*****
**** End of Processing ****
*****

```

Figure 58 Example processing log after Step 6, velocity processing

3.5.7 Step 7: Lowpass filtering

Residual pressure, temperature and travel time are filtered (using a Butterworth filter from MATLAB's Signal Processing Toolbox) and subsampled. The Butterworth filter design is known for its characteristic sharp monotonic transition between flat pass and stop bands with a minimum of coefficients. The Butterworth is also well known and used regularly in the oceanographic field. The filtering is done twice, once forward and once in the reverse direction. Three parameters control lowpass filtering:

Butterworth Filter Order	default = 4
Cutoff Period (days)	default = 3 days
Output Time Interval (days)	default = 0.5 days

Transients at the records' ends are reduced by removing a linear ramp generated from the first and last points of the series before filtering. The same linear ramp is added after filtering. To avoid contamination by startup transients, data at each end of the filtered series are discarded. The amount of data discarded is one third of the cutoff period. After filtering, the time series is interpolated onto the output time interval.

To begin, enter the IES serial number and file index (e.g. 076_1) and press the **Initialize** button in the GUI window. Change the upper right pull down menu to **Step 7: Begin at Lowpass Filtering**. If you are running the steps consecutively, the second pull down menu automatically changes to **Stop before Plotting**. If you have restarted IES_GUIDRIVER you will need to change the second step from **End of Processing** to **Stop before Plotting** (Figure 59).

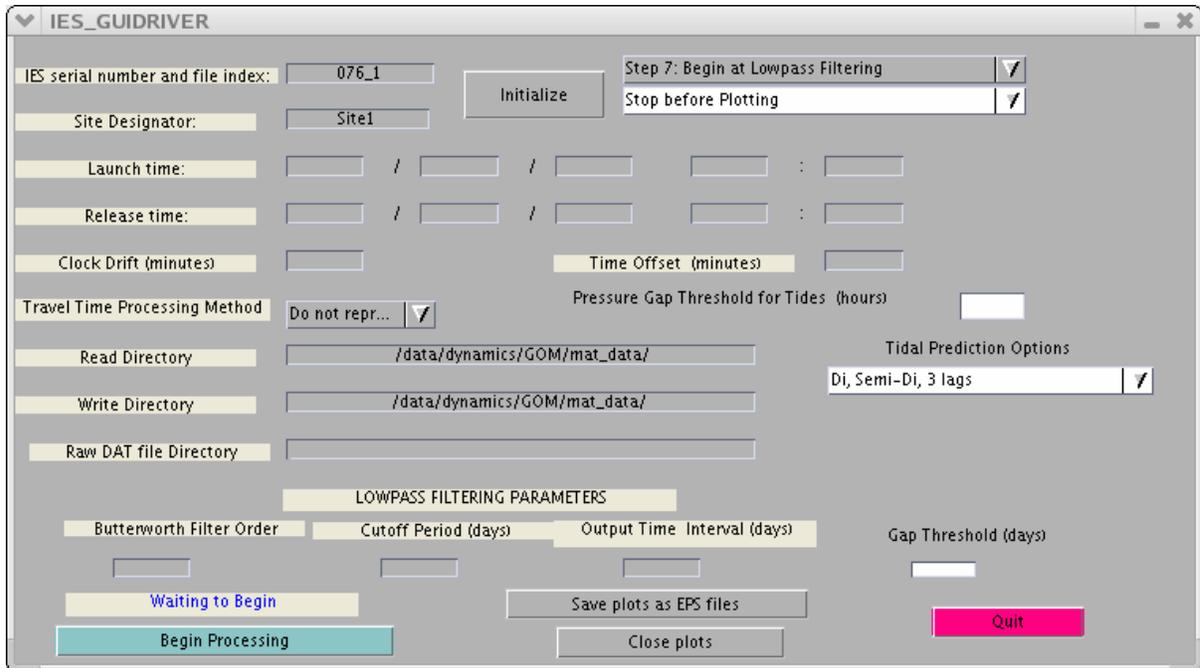


Figure 59 Example IES_GUIDRIVER window after initialization for processing Step 7, lowpass filtering only

Parameters used by this step are those listed in the GUI as LOWPASS FILTERING PARAMETERS: Butterworth Filter Order, Cutoff Period, and Output Time Interval. Gap Threshold, located in the lower right of the GUI window is also used. Gaps less than this threshold will be interpolated before filtering (default = 2 days). Note the values for these parameters are blank in the GUI (Figure 59). However, in our example, LOWPASS FILTERING PARAMETERS were specified during initialization (Figure 16) as 4, 3, and 0.5 (from the defaults) and the Gap Threshold was specified as 1 (from the InitGuiDriver.dat file where lpgap=1 under the array variables, see Figure 14) and have been retained for this step. You do not need to re-enter these parameters unless you want to change the values that were entered during initialization.

Press the button **Begin Processing** in the lower left of the IES_GUIDRIVER window (Figure 14). **NOTE:** If no values for the LOWPASS FILTERING PARAMETERS were ever specified a message will appear in the MATLAB window “No lowpass filtering will be done”.

The IES_GUIDRIVER and MATLAB windows will show **Finished** and **End of Processing** respectively. The Site1lp.mat file (Figure 60) is created and contains the

lowpass filtered, subsampled travel time, temperature and pressure (for a PIES). For a CPIES the lp.mat file will contain the additional variables u, v, and cmtmp (DCS temperature, Figure 61). The log file will have appended the step 7 processing information listing the Butterworth Filter Order, the Cutoff Period, the Output Time Interval and the Gap Threshold for filtering (Figure 62).

```
>>
>> load Site1lp.mat
>> Site1lp

Site1lp =

    tau: [1x796 double]
   taudd: [1x796 double]
    tmp: [1x795 double]
   tmpdd: [1x795 double]
    prs: [1x795 double]
   prsdd: [1x795 double]
   refyr: 2004
```

Figure 60 Example Site1lp.mat file contents for a PIES after Step 7, lowpass filtering

```
Site2lp =

    tau: [1x832 double]
   taudd: [1x832 double]
    tmp: [1x831 double]
   tmpdd: [1x831 double]
    prs: [1x831 double]
   prsdd: [1x831 double]
   refyr: 2004
     u: [1x748 double]
     v: [1x748 double]
   uvdd: [1x748 double]
   cmtmp: [1x748 double]
  cmtmpdd: [1x748 double]

>>
```

Figure 61 Example Site2lp.mat file for a CPIES after Step 7, lowpass filtering

```
*****
STEP 7 LOW-PASS FILTER processing run on 12-May-2006 11:25:13

Butterworth lowpass filter: Order = 4, Cutoff period = 3 days
Filtered record has sampling interval of 0.5 days
Data gaps less than 1 days will be interpolated before filtering
*****
**** End of Processing ****
*****
```

Figure 62 Example processing log after Step 7, lowpass filtering

3.5.8 Step 8: Plotting

To begin, enter the IES serial number and file index (e.g. 076_1) and press the **Initialize** button in the GUI window. Change the upper right pull down menu to **Step 8: Begin at Plotting**. The second pull down menu automatically changes to **End of processing**. Press the button **Begin Processing** in the lower left (Figure 63).

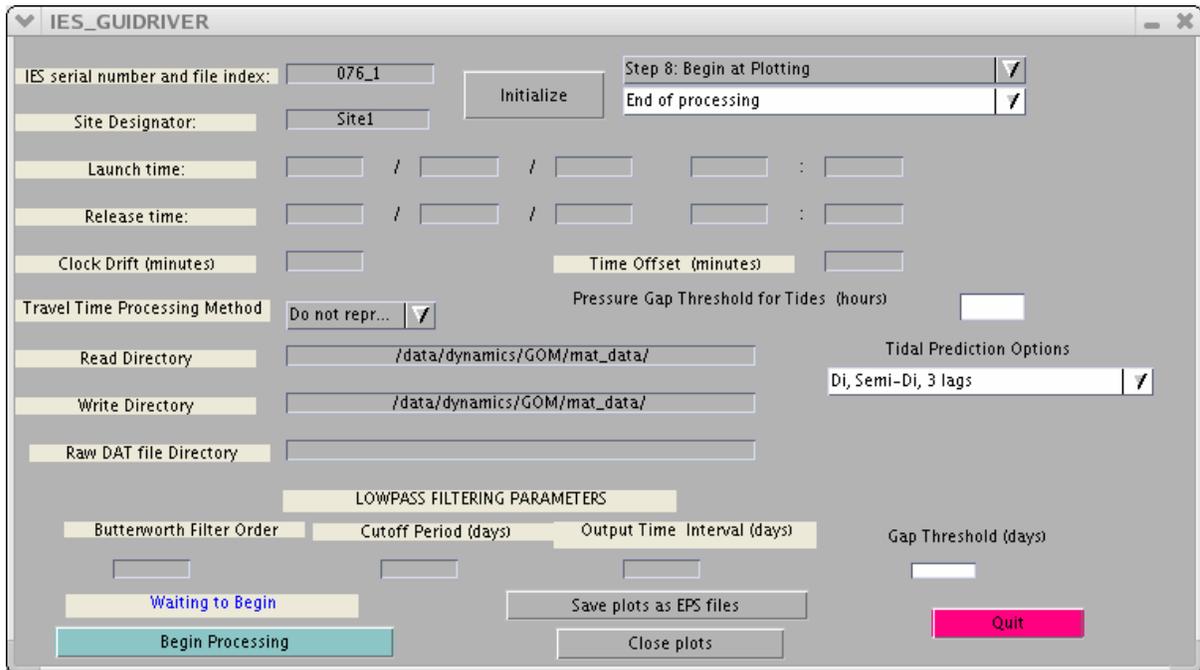


Figure 63 Example IES_GUIDRIVER window after initialization for processing Step 8, plotting only

Three Figure windows will be generated. The Figure 1 window for a PIES (Figure 64) contains the hourly tau, pressure and temperature time series from the final Site1.mat file. If the instrument is a CPIES, the Figure 1 window (Figure 65) contains the hourly tau, pressure, temperature, u and v and DCS temperature time series from the final Site1.mat file. The Figure 2 window (Figure 66) is the same for both PIES and CPIES and provides a chronology of the pressure data processing, showing from top to bottom, the despiked pressure, the tides, the detided pressure with drift, and the dedrifted pressure time series. The Figure 3 window for a PIES (Figure 67) contains the lowpass filtered, subsampled tau, pressure and temperature time series from the Site1lp.mat file. For a CPIES, u, v and DCS temperature are added to the Figure 3 window (Figure 68).

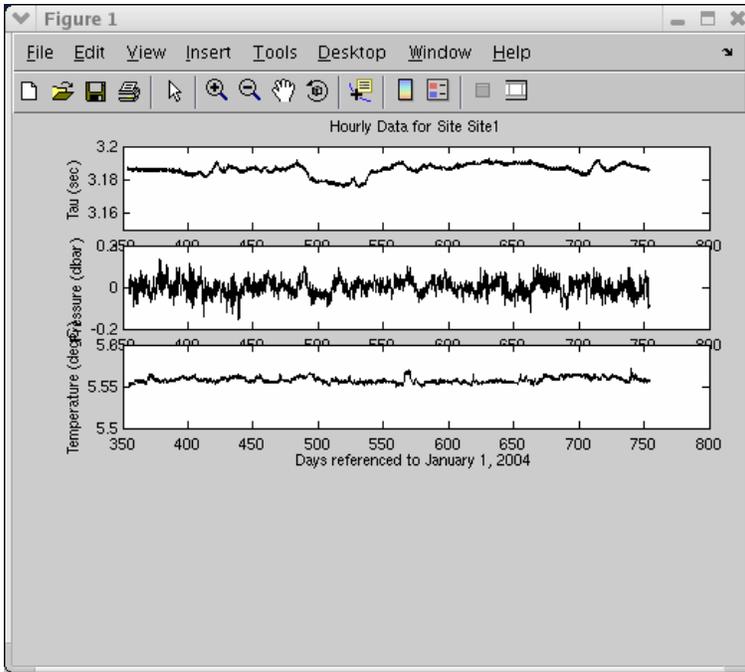


Figure 64 Example Figure 1 window showing the hourly tau, pressure and temperature time series for a PIES

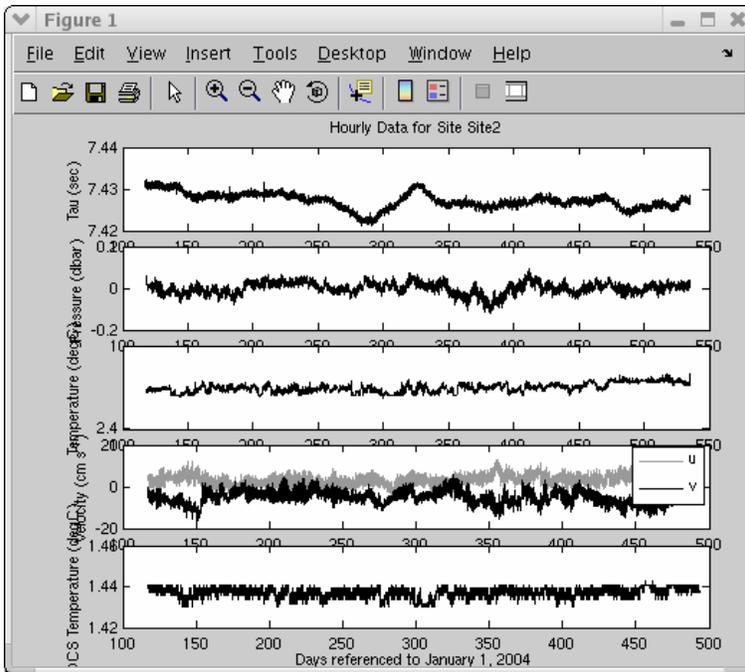


Figure 65 Same as Figure 64 with addition of u, v and DCS temperature for a CPIES

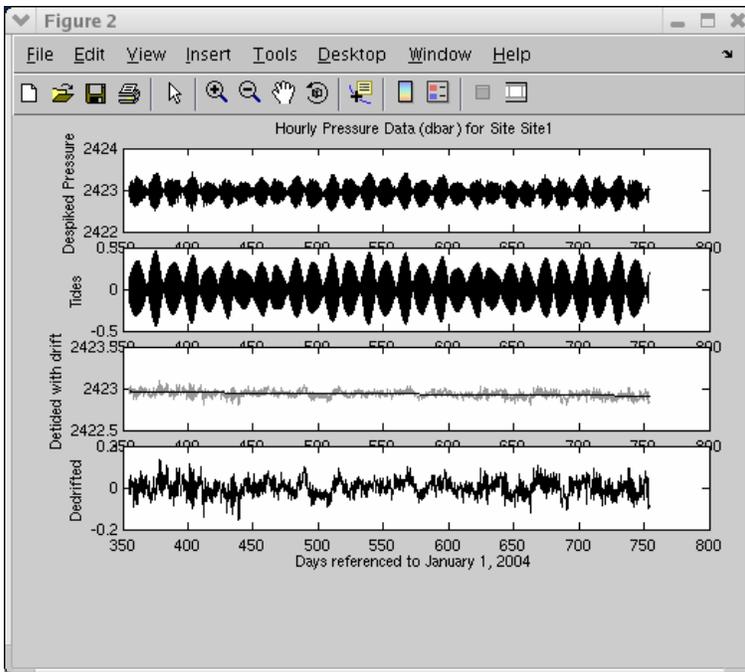


Figure 66 Example Figure 2 window showing the despiked pressure, tides, detided pressure with drift and dedrified pressure time series

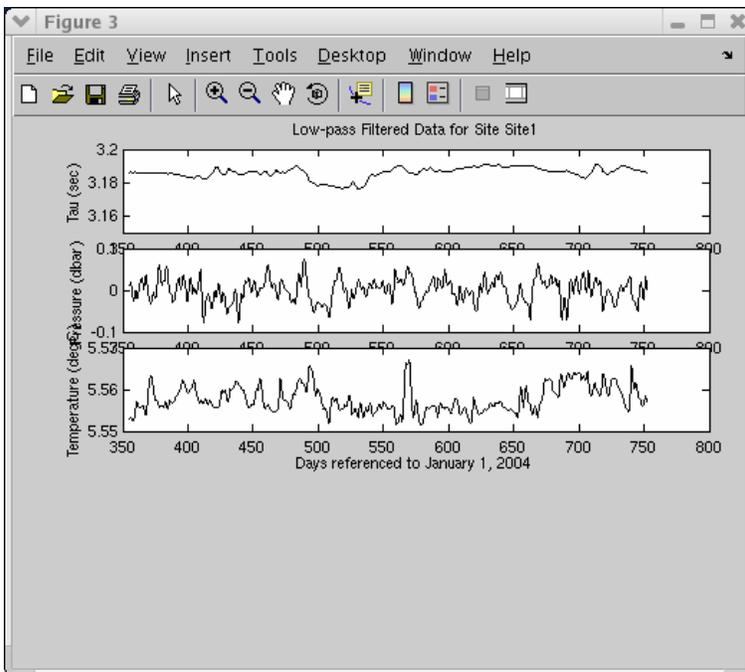


Figure 67 Example Figure 3 window showing the lowpass filtered, subsampled tau, pressure and temperature time series for a PIES

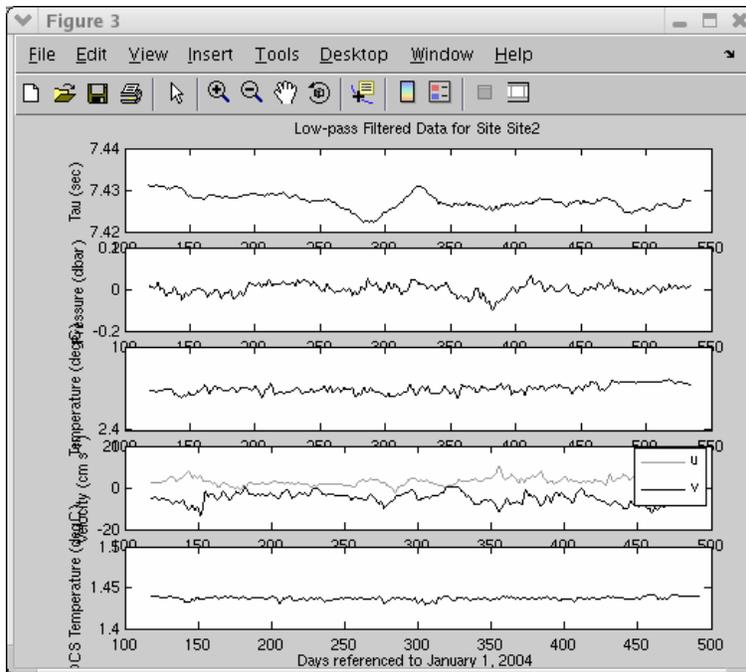


Figure 68 Same as Figure 67 with addition of u, v and DCS temperature for a CPIES

It is recommended that you save (and later print) the three figures by pressing the button **Save plots as EPS files** in the center bottom of the GUI window (Figure 63).

Following our example, the saved EPS filenames will have the syntax

Site1_finalfigX.eps for 076_1. Press the red **Quit** button in the lower right of the GUI window or **X** to exit.

Chapter 4. IES_GUIDRIVER Special Cases

4.1 No Travel Time Data

If the travel time file (TXXX_#.mat) is missing, the remaining data can still be processed with IES_GUIDRIVER. During Step 1, pressure and temperature limit selection will proceed as described in Chapter 3 starting at Figure 21. Without travel time data, the program will read the instrument log file (IESXXX.LOG) to find the travel time measurement interval. Time bases for the other measurements will then be calculated using this travel time measurement interval. The user will be prompted for the travel time measurement interval if it could not be read from the instrument log file. Subsequent processing will proceed as described in Chapter 3.

References

Fields, E., K. Tracey and D.R. Watts, 1991. Inverted echo sounder data processing report. University of Rhode Island GSO Tech. Rept. No. 91-3, 150 pp.

Munk, W.H. and D.E. Cartwright. 1966. Tidal spectroscopy and prediction. *Phil. Trans. Roy. Soc. London*, 259, 533-581.

Watts, D.R. and H. Kontoyiannis. 1990. Deep-ocean bottom pressure measurements: Drift removal and performance. *J. Atmos. Ocean. Technol.*, 7, 296-306.

Acknowledgments

The MATLAB IES processing codes described in this manual have evolved from earlier versions created by Douglas Mitchell and Charles James. Mark Wimbush wrote the Respo.for program used for tidal analysis and helped implement it for our application. We gratefully acknowledge these programming efforts.

This research program has been sponsored by the National Science Foundation under grants numbers OCE00-95572 and OCE02-21008, the Office of Naval Research under contract N00014-02-1-0271, the US Department of the Interior, Minerals Management Service and Science Applications International Corporation under contracts 4400061089 and 4400099204 and by the US Department of the Interior, Minerals Management Service and Evans Hamilton, Inc. under contract 01440 .

Appendix A Example plots generated by IES_GUIDAT2MAT for PIES

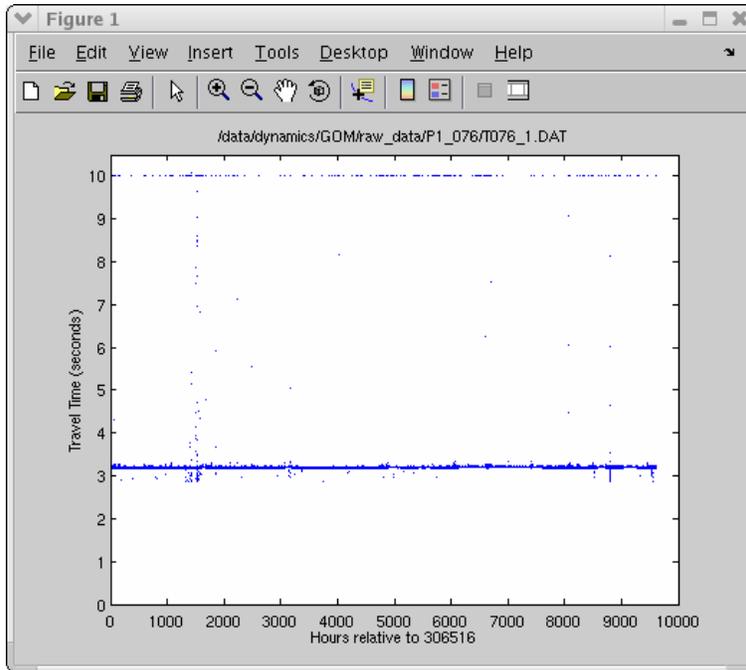


Figure 69 IES_GUIDAT2MAT default Figure 1, Travel Time (seconds)

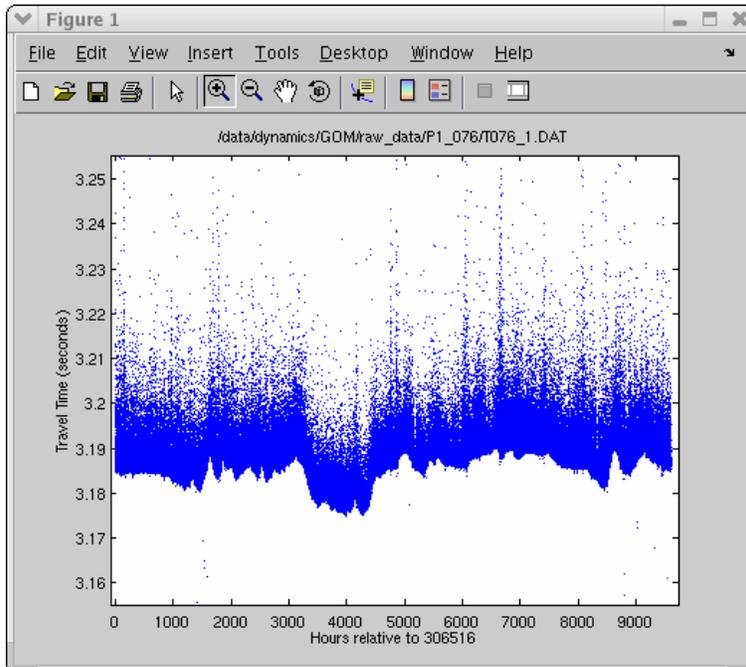


Figure 70 IES_GUIDAT2MAT Figure 1, Travel Time (seconds) with reduced y-axis limits

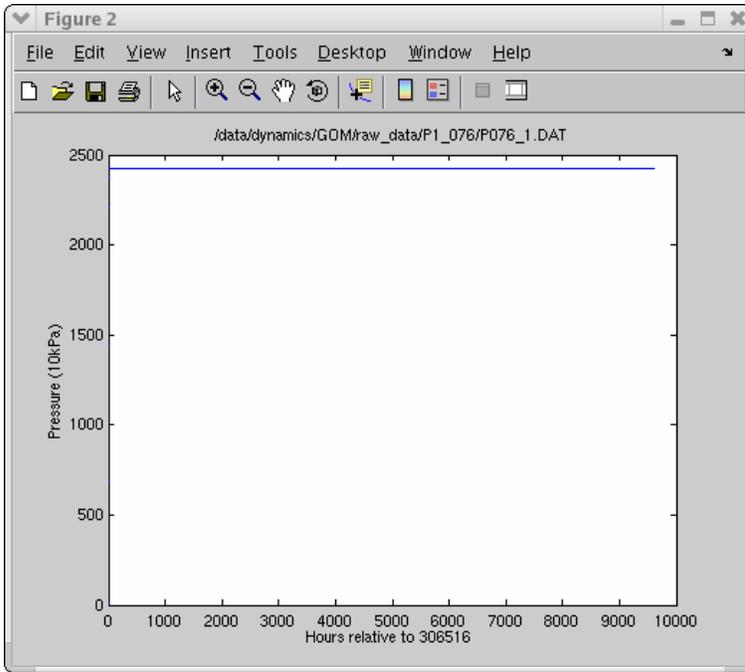


Figure 71 IES_GUIDAT2MAT default Figure 2, Pressure (10kPa)

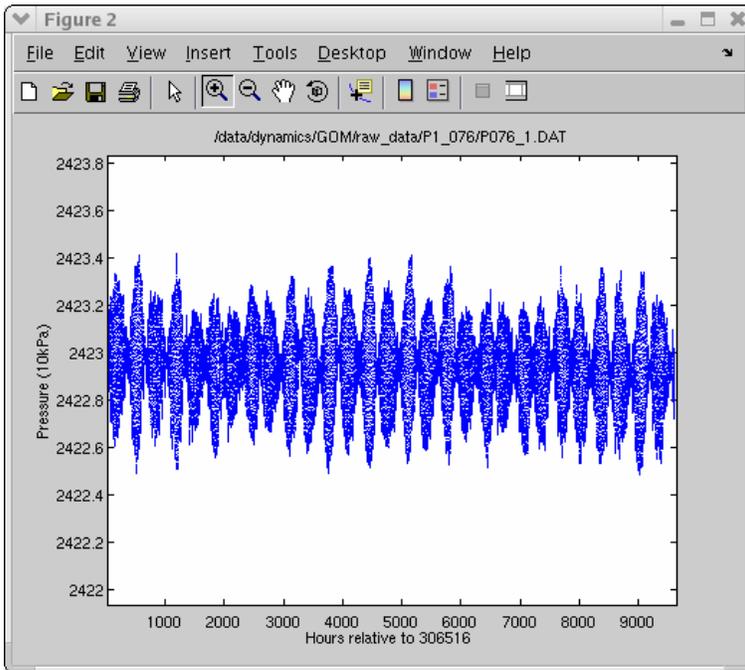


Figure 72 IES_GUIDAT2MAT Figure 2, Pressure (10kPa) with reduced y-axis limits

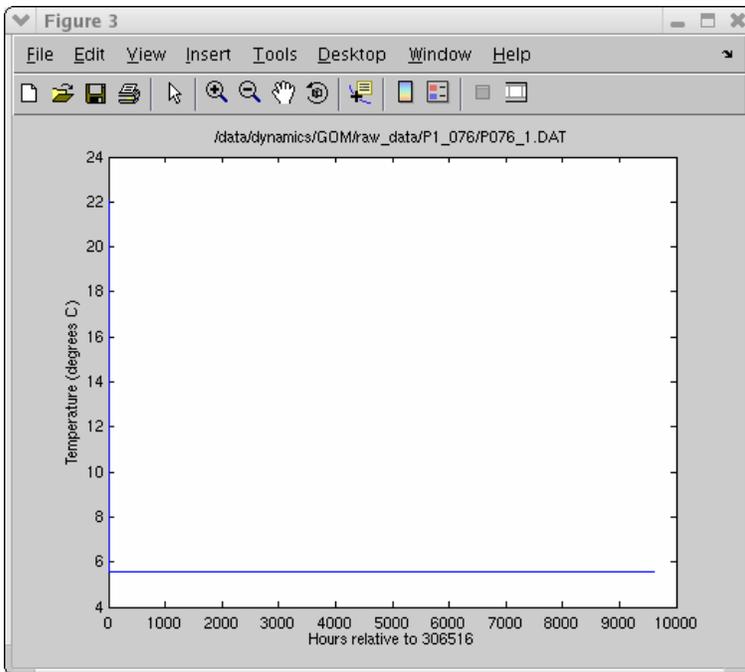


Figure 73 IES_GUIDAT2MAT default Figure 3, Temperature (degrees C)

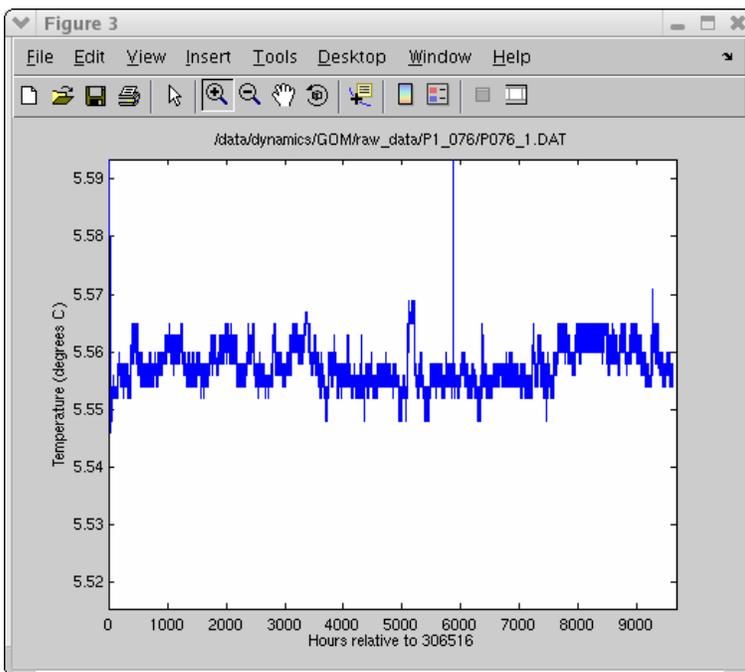


Figure 74 IES_GUIDAT2MAT Figure 3, Temperature (degrees C) with reduced y-axis limits

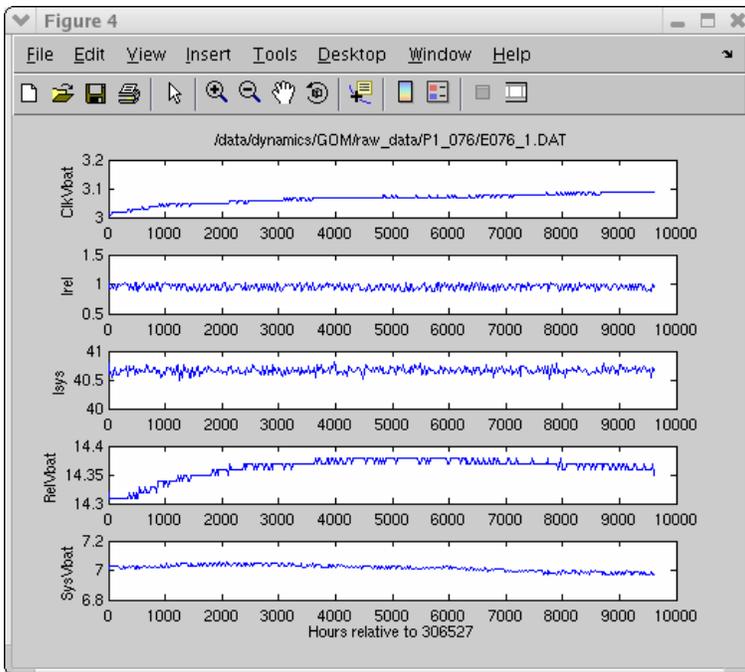


Figure 75 IES_GUIDAT2MAT default Figure 4, Engineering measurements

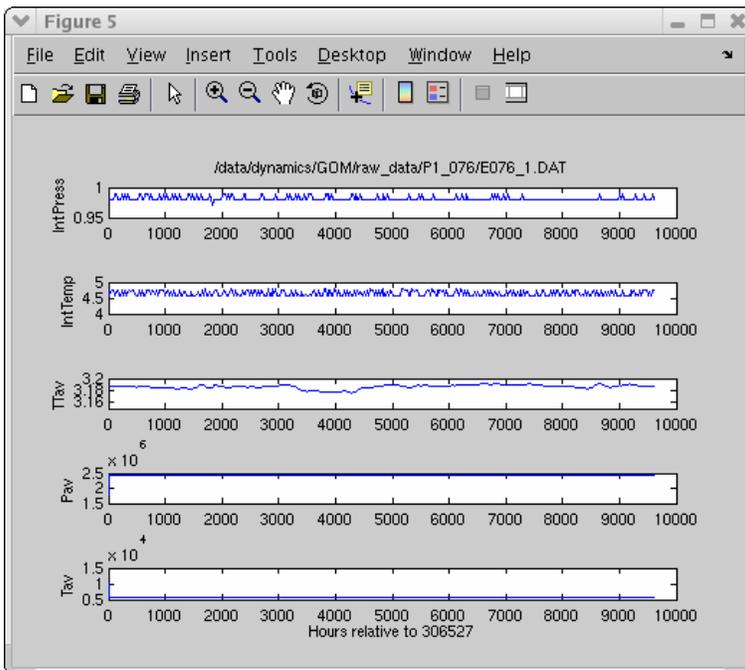


Figure 76 IES_GUIDAT2MAT default Figure 5, Engineering measurements

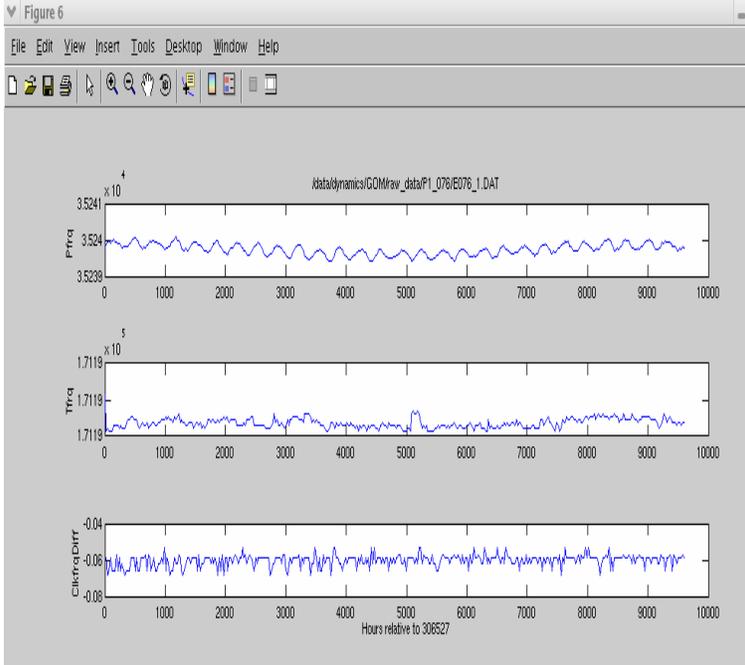


Figure 77 IES_GUIDAT2MAT default Figure 6, Engineering measurements

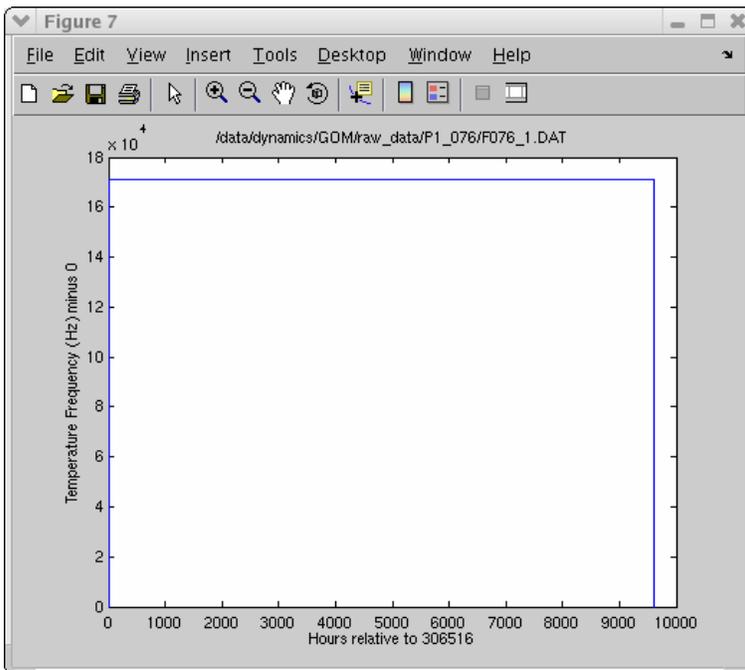


Figure 78 IES_GUIDAT2MAT default Figure 7, Temperature Frequency (Hz)

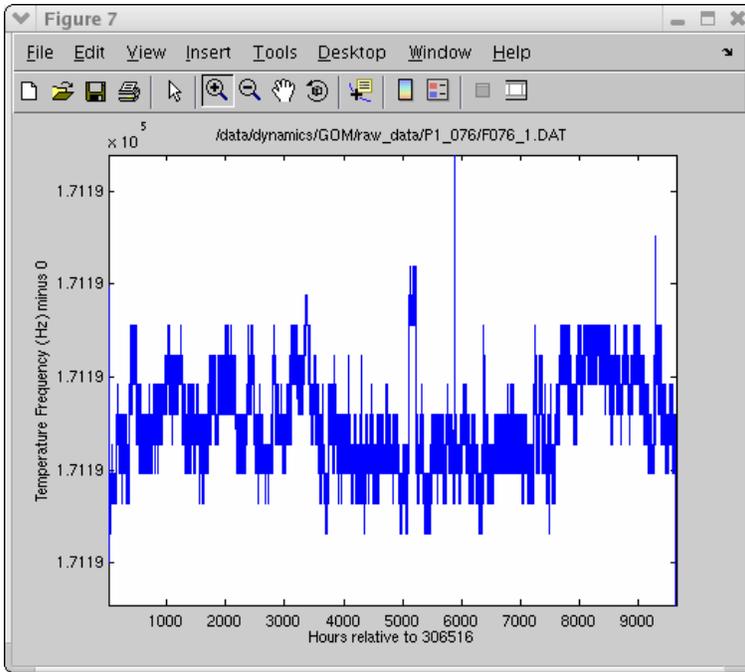


Figure 79 IES_GUIDAT2MAT Figure 7, Temperature Frequency (Hz) with reduced y-axis limits

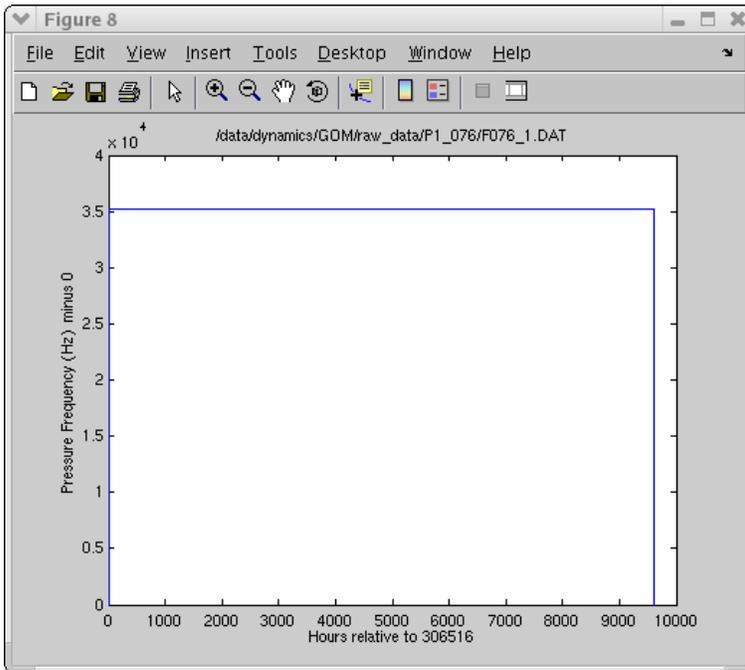


Figure 80 IES_GUIDAT2MAT default Figure 8, Pressure Frequency (Hz)

Appendix B Examples of additional plots generated by IES_GUIDAT2MAT for currents (CPIES only)

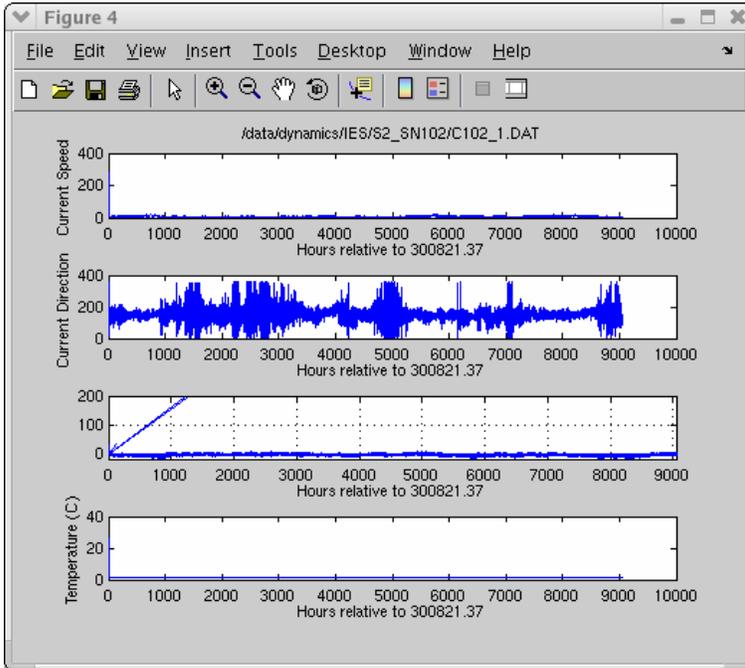


Figure 82 IES_GUIDAT2MAT default Figure 4 for CPIES

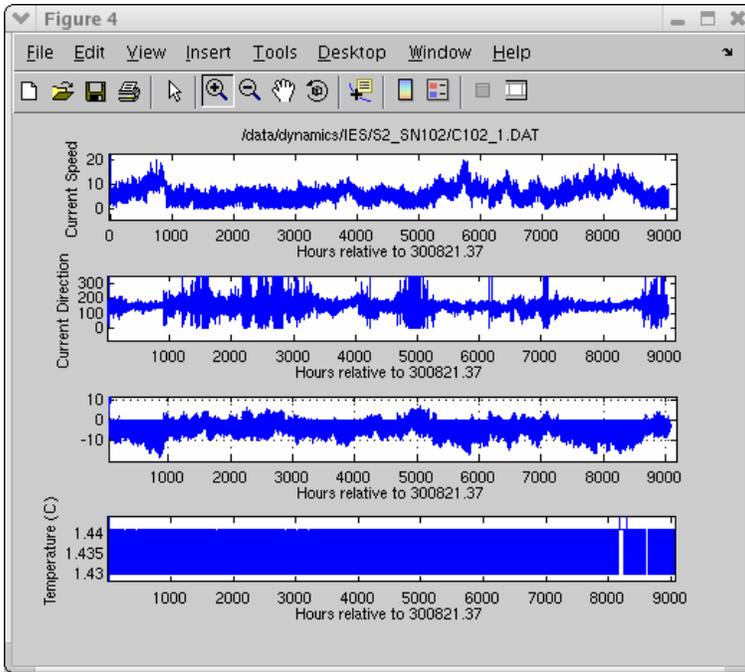


Figure 83 IES_GUIDAT2MAT Figure 4 with reduced y-axis limits for CIES

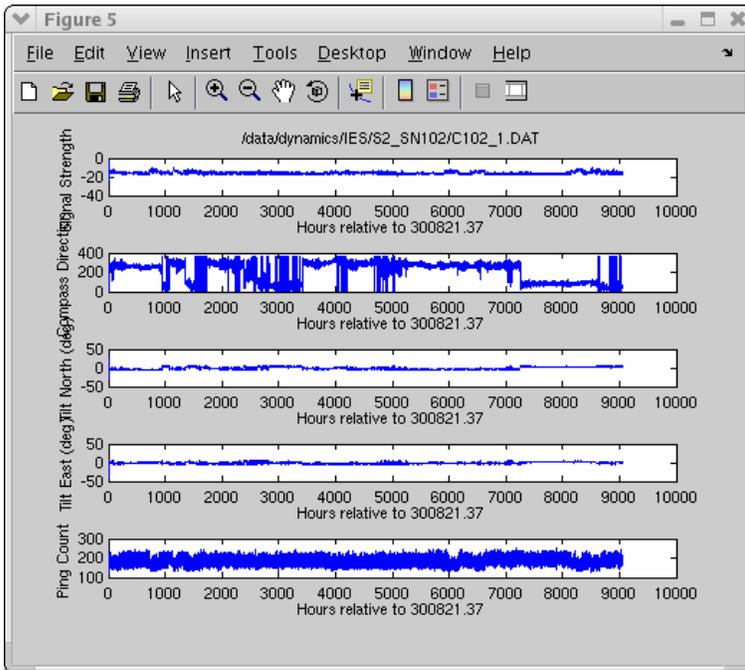


Figure 84 IES_GUIDAT2MAT default Figure 5 for CIES

Appendix C Response Analysis of Tides

C.1 Theory

A simple filter may be expressed as,

$$y(t) = \sum_k l(\tau_k) x(t - \tau_k) \quad (1)$$

where y is the predicted tide, x is the input function, and l is the response of the ocean to a unit of impulse of x at time zero.

The tidal prediction illustrated by Equation 1 depends only on the input's temporal variation at that particular location. As a refinement, the forcing at other locations may be included in the prediction:

$$y(t) = \sum_i \sum_k l_i(\tau_k) x_i(t - \tau_k) \quad (2)$$

where ' i ' represents forcing at neighboring locations that might influence sea level at the site of interest. Response analysis systematically includes spatial dependence by expanding x in surface spherical harmonics. The predicted tide is expressed as a filter acting on the complex-valued, time-varying amplitudes of the spherical harmonic functions representing the equilibrium-tidal potential.

$$y(t) = \text{Re} \sum_k \sum_{n=2}^{\infty} \sum_{m=0}^n w_n^m(\tau_k) C_n^m(t - \tau_k) \quad (3)$$

The indices n and m are the degree and order of the surface spherical harmonic functions. $C_n^m(t)$, which replaces x , is the set of time-varying amplitudes of the corresponding spherical functions. The w_n^m , which replace l , are the complex weights associated with $C_n^m(t)$. Using the data to be detided, $(\eta(t_j))$ the weights (w_n^m) are found by solving the over determined set of equations such that the difference between the data and the predicted tide $(\eta(t_j) - y(t_j))$ is minimized in a least squares-sense.

The equilibrium-tidal potential for a mass M , whose center of mass is at distance ρ from the point of observation, is:

$$\frac{V(t)}{g} = \frac{GM}{g\rho}$$

Where $V(t)$ is the gravitational potential due to mass M , G is the gravitational constant, and g is local gravity. Typically, ninety-nine percent of the gravitational tidal variance can be explained with the equilibrium-tidal potential (due to the masses of the moon and sun) represented by just the C_2^1 and C_2^2 $\{n=2, m=1, 2\}$ amplitude functions.

The spherical harmonics corresponding to C_2^1 and C_2^2 are illustrated as viewed down the axis of rotation in Figure 85. The plus sign represents bulging relative to the geoid, the minus sign flattening. From this illustration, it is apparent that C_2^1 and C_2^2 are associated with the diurnal and semi-diurnal species of the harmonic analysis. RESPO is set up to use these functions (C_2^1 and C_2^2) as input at time-lags of $\tau_k = k * 48$ hr, $k=-1,0,1$. Thus, with the default case of $n=2$, Equation 3 is then truncated to:

$$y(t) = \text{Re} \sum_{k=-1}^1 \sum_{m=1}^2 w_2^m(k * 48) C_2^m(t - k * 48) \quad (4)$$

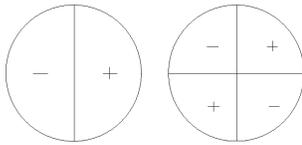
The 6 weights, $w_2^m(k * 48)$, $m=1,2$, $k=-1,0,1$ are found from the overdetermined set of equations.

$$\eta(t_j) = \text{Re}(C(t_j) * W)$$

Where $C(t_j) = [C_2^1(t_j + 48)C_2^1(t_j)C_2^1(t_j - 48)C_2^2(t_j + 48)C_2^2(t_j)C_2^2(t_j - 48)]$

And

$$W = \begin{pmatrix} w_2^1(48) \\ w_2^1(0) \\ w_2^1(-48) \\ w_2^2(48) \\ w_2^2(0) \\ w_2^2(-48) \end{pmatrix}$$



$$C_2^1 \quad C_2^2$$

Figure 85 Spherical harmonics corresponding to tidal amplitude functions

These weights are applied to C_2^m to generate the predicted tide. This is subtracted from the original data $\eta(t_j)$ to give the residual tide.

C.2 Tidal Prediction Options

Currently, RESPO is set up to calculate the total gravitational potential (LGAMMA=3) for the lunar and solar tidal constituents.

Di, Semi-Di, 3 lags (Diurnal, Semi-Diurnal, 3 lags): BASECTRL_DS3.txt

```
2,  
3, 1, 2,  
3, 2, 2,  
2, 1, 2,  
3, -48, 0, 48,  
2, 0, 2, 0, 3,  
1, 1.0, -1, 1, 1, 1, 0, -1, 1, -1, 1,  
'tidedata.txt',  
'(f12.6)',  
'tidedataout.txt',  
'(3f15.6)',
```

Di, Semi-Di, 5 lags (Diurnal, Semi-Diurnal, 5 lags): BASECTRL_DS5.txt

```
2,  
3, 1, 2,  
3, 2, 2,  
2, 1, 2,  
5, -96, -48, 0, 48, 96  
2, 0, 2, 0, 5,  
1, 1.0, -1, 1, 1, 1, 0, -1, 1, -1, 1,  
'tidedata.txt',  
'(f12.6)',  
'tidedataout.txt',  
'(3f15.6)',
```

Di, Semi-Di, Mf, Mssa, 5lags (Diurnal, Semi-Diurnal, Fortnightly, and Solar Semi-annual 5 lags): BASECTRL_DSL5.txt

```
3,  
3, 1, 2,  
3, 2, 2,  
3, 0, 2,  
3, 1, 2, 3,  
6, -96, -48, 0, 48, 96, 0,  
3, 0, 2, 3, 0, 5, 6,  
1, 1.0, -1, 1, 1, 1, 0, -1, 1, -1, 1,  
'tidedata.txt',  
'(f12.6)',  
'tidedataout.txt',  
'(3f15.6)',
```

```

NUMGMN
LGAMMA, MORDER, NDEGRE
LGAMMA, MORDER, NDEGRE
KP (JP(K), K=1,KP)
KH (HH(K), K=1,KH)
NPHGP1 (NP(K), K=1,nphgp1) (NH(K), K=1,NPHGP1)
IADJD, CALIB, IPRIND, IDWTS, IDADM, IDRSP, KOMPLX, IPRINP, IRESID,
IPRINR, IOUT

NUMGMN = Number of gamma, M, N triplets
LGAMMA = 1 for moon's gravitational potential
         = 2 for sun's gravitational potential
         = 3 for total gravitational potential
         = 4 for sun's radiational potential
MORDER = value of M in ith gamma, M, N triplet (order of spherical
harmonic)
NDEGRE = value of N in ith gamma, M, N triplet (degree of spherical
harmonic)
KP = Dimension of array JP
JP = Array of P Values (X component numbers) to be used in forming the
prediction weights
KH= Dimension of array HH
HH = Array of H values (X time leads) to be used in forming the
prediction values
NPHGP1 = Dimension of array NP and of array NH
        NP(I) NP(1) = 0
        NP(I .GT. 1) is the number of the last term in the
        (I-1)th group of array jp
        NH(I) NH(1) = 0
        NH(I .GT. 1) is the number of the last term in the
        (I-1)th group of array HH
IADJD (if EQ 1) subtract mean from series and multiply by calib
CALIB calibration
IPRIND (if LT 0) print only partial list of normalized input data
IDWTS (if LT 0) weights are not to be computed, they are given
IDADM (if EQ 1) compute admittance
IDRSP (if EQ 1) create predicted tide series
KOMPLX (if EQ 0) only real part of predicted series is to be retained
IPRINP (if LT 0) print only partial list of predicted tide
IRESID (if EQ 1) compute residual series
IPRINR (If LT 0) print only partial list of residual series
IOUT (if LT 0) do not write output time-series file

```

Figure 86 Description of contents of Tidal Prediction Option Files