Appendix 3: Data Analysis

CAST MODE DATA

Dumping data to a capture file provides useful archive data but the desired results are generally graphical, either plots of the acoustical profiles or the estimated size-abundances versus depth. The steps and the software we use to produce these depictions of the data are described below. These are offered as a guide to data processing only; you are encouraged to use methods and programs with which you are more comfortable if you so desire or to modify these programs for your special needs.

We generally use file names and extensions that help us to remember when the data were obtained. A typical data dump could be saved to a text file with a name like:

T970413.log

where the T signifies TAPS data, 970413 represents April 13, 1997, and the .log extension tells us that this is raw archive data. Unless you have a ***lot*** of casts in one day, you can append new data dumps to previous ones from the same day. If you do have lots of data, file names such as T970413C.log are useful in sorting the sequence of log files.

PLOTTING CAST DATA

At a convenient time after the .log files have been created, we run the program **plot6.m** in Matlab (5.0 or newer) This program reads the .log files you have created, figures out how many casts there are, and plots them. It also produces output data files for input to the inversion program, **inv6.m**

Plot6.m offers some useful options:

- You can elect to use the calibration constants as installed in TAPS or enter a set of calibration 'deltas'. If you are operating TAPS with zeroed calibration constants, you will need to enter values here (or modify the m-file to incorporate these constants).

- You can elect to ignore changes in absorption with actual water temperature or let the program compute a correction to the calibration constants due to temperature changes. You will have to enter the temperature and salinity at which the nominal absorption was computed (on the calibration sheet in Appendix 1: Calibrations) and enter a mean salinity estimate if you choose to correct for absorption.

- You may choose which cast to plot. When more than one cast is contained in a log file, you are free to examine them in any order.

- You can select the depth bin size, view the results, and change your mind as often as necessary.

- You can elect to bin downcast data only, upcast data only, or all cast data. Generally, downcast data is least perturbed by the presence of TAPS, any mounting hardware, and the cable.

- You can elect to use a noise-subtraction algorithm to enhance the data at low scattering levels. This algorithm uses the minimum scattering level in any cast as an estimate of the receiver noise floor and subtracts this value (in intensity units) from the remaining data to improve the signal-to-noise ratio. This is a simple, yet often effective, means of pulling 'signals' out of the noise in situations where the overall scattering levels are low.

When the data have been binned and corrections applied, you have the option of saving the results to a .mat file. This file can be input to the inversion program at some later time. You will <u>not</u> be able to invert these data unless you save them to a file.

Plot6 does not plot external sensor data but you could easily modify it to do so. The sensor frequency fields are read and annotated in the code. Help on converting data from some instruments is available. As an example, the file **plot6f.m** converts instrument field #1 from frequency to fluorescence according to the factory calibrations for our WetLabs WetStar fluorometer. This field is plotted in between the temperature and Sv plots so that correlations may be observed.

Sample plot outputs from the program are shown in Fig 1. The lower panel displays temperature versus depth on the left side and Sv versus depth (at the six frequencies) on the right side. This format allows you to inspect the behavior of acoustical backscattering with respect to the principal physical forcing function in the ocean, temperature.

The upper panel of Fig. 1 shows the frequency spectrum of the binned data. This sort of display is sometimes useful in discriminating between regions of different scatterer types. Overplots of data with similar scattering behavior versus frequency tend to produce a visual image of the mean scattering behavior. If two or more distinct types of scatterer are present, it is usually easy to discern the differences in the ‘mean’ spectra.

Plot6 considers a new cast to be flagged by a significant change in the time interval between samples. A significant time interval is defined as five times the average interval between data sets The program considers the end of a downcast to be at the maximum depth reached in the cast.

Binning consists of sorting the raw acoustic data by depth, determining which bin the depth corresponds to, and adding the acoustic intensities (not dB) to an accumulator using the depth bin index. The outputs after binning are divided by the number of data summed in each bin—this will usually differ from bin to bin—and converted back to dB. The user inputs the size of the bin in meters and the program calculates the maximum depth for you.

Note that bins are labeled by the maximum depth of the bin. For example, if the bin size is 0.5 m, the first bin holds data from 0m to 0.5 m, the second holds data from just greater than 0.5 m to 1 m, etc.

The binned acoustic data produced by **plot6.m** should be saved to a file with a descriptive name. We typically use the generic name of the .log file, TYYMMDD, with a letter suffix to indicate the cast number. No extension is needed as this file is saved in the Matlab internal format Thus, the third cast from the .log file used as an example above would be saved as

T970413c.mat.

The extension signifies binned acoustic data (to us, anyway) and the file name links the data to the archive (.log) file should we have any questions later on.

CREATING A MODEL

The inversion process is described in Appendix 2: Basic Acoustics and the references contained therein. The program **inv6.m** conducts the inversion process on data files produced by **plot6.m** using a scattering model or models supplied by you. The scattering models can be simple, such as the truncated fluid sphere model we employ for small crustaceans such as copepods, or complex, such as the finite bent cylinder models of Stanton or the arbitrary cross-section models of McGehee. More than one model can be used at one time. If no model file is available, however, **inv6.m** will signal an error and not attempt to invert the data file.

MakeTS.m is the m-file that creates files embodying the truncated fluid sphere model using parameters that you select. The emphasis on the user is necessary because successful inversion requires some sensible interpretation and model selection on the part of the user.

Appendix 3: Data Analysis

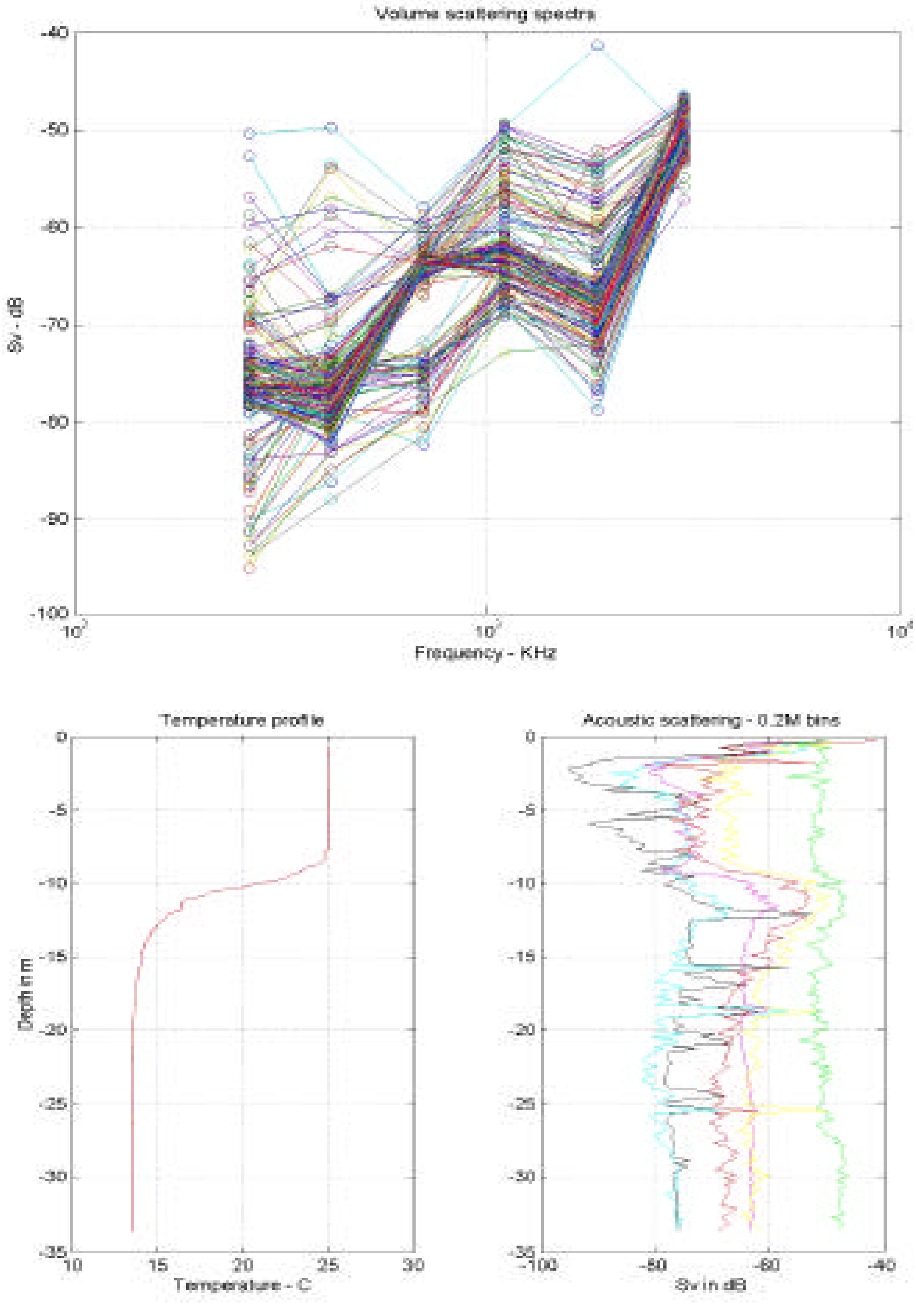

Figure 1. Sample output from **plot6**. Data from Lake San Vicente, CA. Upper panel: volume scattering spectrum. Lower panel: temperature vrs depth (left) and Sv vrs depth (right).

Creation of a model file from **makeTS** requires selection of the following:

- range of equivalent spherical radius (ESR) values
- linear or logarithmic spacing
- number of ESR values

The program begins by generating a file of ESR values. You will be prompted to enter the minimum and maximum values of ESR (in mm). The program then asks you to enter either 'log' or 'lin' to signal the type of spacing of the ESR values. Linear spacing ('lin') makes the intervals between ESR values equal. Logarithmic spacing ('log') makes the ratios of successive ESR values equal.

The last parameter to be entered is the 'underdetermination factor'. This number is multiplied by the number of frequencies (6) to determine the number of size classes in the model. As explained in Appendix 2: Basic Acoustics, the inversion algorithm will produce estimates of size-abundance even when the problem is underdetermined; that is, when the number of size classes is larger than the number of input data (frequencies). We have found that setting up the inverse problem as a highly under-determined one has significant advantages in improving the ability to select dominant size classes—at the cost of increased computation time. You get to make this tradeoff by selecting the under-determination factor.

Please also keep in mind that you won't get something for nothing; adding more size classes will not result in more information. You can expect at most to resolve five size classes (and the total biovolume, BV) with six frequencies; data in more than five size classes represents 'leakage' of information and should be treated with due caution.

Choosing a value of 5 will cause **makeTS** to setup 30 size classes (5 times the number of frequencies, 6). This is a reasonable number of size classes in many cases, although we have used under-determination factors of as much as 20 when trying to separate closely-spaced size classes.

MakeTS creates several matrices for use in the inversion. The ESR vector is computed as described above. Using this vector and the TAPS frequencies, the program computes the matrix of target strength values predicted by the model at the various frequency-size combinations. The inversion program displays both the measured Sv data and the Sv predicted from the size-abundance estimates; the latter require a target strength matrix computed sufficient frequencies that the plot is readable. We have selected 50 frequencies for this matrix (see Fig. 2).

The final user input to **makeTS** is a file name for saving these data. You are responsible for keeping notes on which model file was built with which parameters. We tend to use non-informative names like model1, model2, etc. for these files. Notes are clearly useful for keeping track.

INVERTING CAST DATA

The inversion process is performed by the program **inv6.m**. This program begins by searching the workspace for signs that a model file has already been loaded. If found, the program asks if you want to use it. If you don't answer 'y', you will be prompted to find the model file. A similar dialog lets you select the data file to invert.

You can elect to suppress one frequency, should you suspect that one channel may be misbehaving or just to assess the importance of particular channels to the results. It sometimes happens that better fits can be obtained using just 5 of the 6 frequencies. This is an area for experimentation. If you want to use all frequencies, enter a '0'.

Next, the program will let you enter a value for the Levenberg-Marquardt factor, **sfactr**. You can keep the current value by pressing ENTER or type a new value. Generally, values between 1e-3 and 1e-6 work best for us; you should experiment to find values that give the best fits for your data. Using too small a value can cause the program to crash with an “iteration limit exceeded” type of message. But such is life. If this happens, just run **inv6** again with a larger value for **sfactr**. See Appendix 1: Basic Acoustics for selection criteria for **sfactr**.

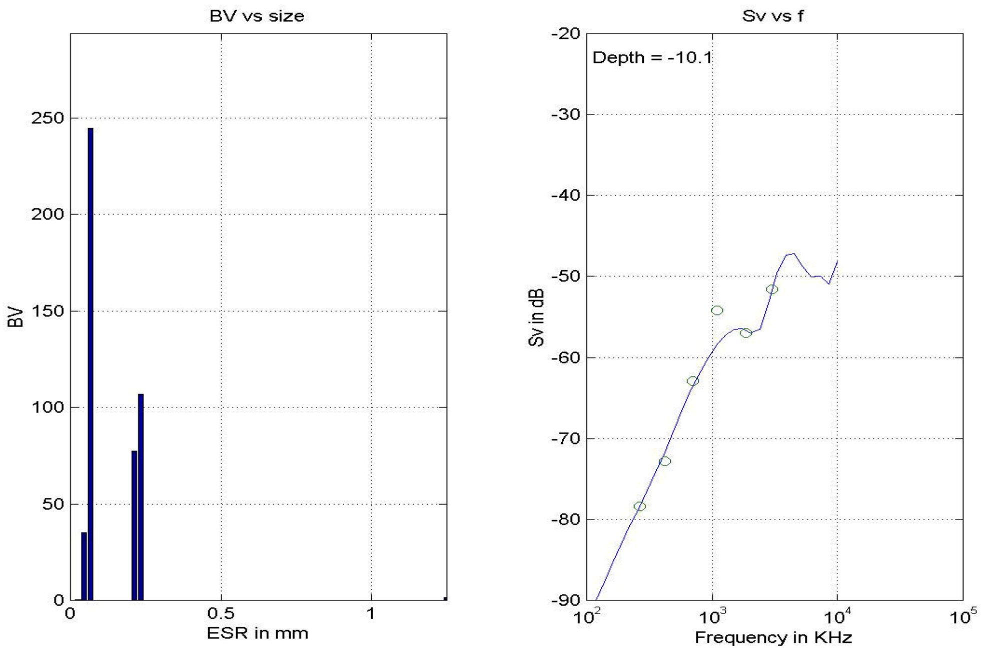

Figure 2. Typical intermediate data plot from **inv6**. Data from Lake San Vicente, CA.

The acoustic data at each depth bin are inverted sequentially and the results displayed graphically. This process plot (Fig. 2) consists of two panels: the left panel shows estimated size-abundance as biovolume (in mm^3/m^3) versus ESR (in mm) as a bar chart. These data are the outputs from inversion. On the right panel, the measured Sv data are plotted as circles versus the acoustic frequency. The size-abundance vector is used with the 50-frequency target strength matrix to generate a curve of Sv vrs frequency that is plotted as a line on the same axes. This permits inspection of the agreement between the measured data and the predicted acoustic response from the scatterer size-abundances estimated by the inversion process.

When all data have been processed, two further plots are generated. The first of these plots displays total biomass versus depth on the left panel and the norm of the residual error, rnorm, on the right panel (Fig. 3). The residual error plot allows you to gauge the quality of the fit in the various regions of biovolume. Normally, the value of rnorm will be unity or less. Values much larger than one suggest a poor fit (possibly measurement problems, a poor choice of size classes, or an incorrect model). Significantly smaller values of rnorm, while dramatic, do not necessarily mean significantly better fits than values just below 1.

The last plot (Fig. 4) presents the likely most useful suite of TAPS results on one page. Temperature is plotted versus depth on the left panel. Total biovolume is plotted versus depth on the center panel (in inv6f.m, fluorescence is plotted next). Biovolume is plotted on the right panel, this time as a color contour plot with depth and ESR as axes. This combination of data permits easy comparison of the structure of biovolume to the thermal structure, both in terms of total biovolume and as a function of size

When the current cast has been inverted and the data plotted, you are given the opportunity to save the results to disk. Enter a suitable file name if you wish to save the data or ENTER if not. This returns control to the top of the loop so you can invert another cast.

The ouput data file is a mat-file, readable only in Matlab. It contains the basic data necessary to recreate the plots, including the ESR vector, **a**; the set of TAPS frequencies, **fxv**; the binned temperatures, **t**; the set of depth bins, **adepth**; the input matrix of Sv's, **bx**; and the output matrix of BV vrs depth and ESR, **biov**. The latter matrix is organized as NDEPTH rows by NSIZE columns and contains the biovolume (BV in mm^3/m^3) for each ESR and depth.

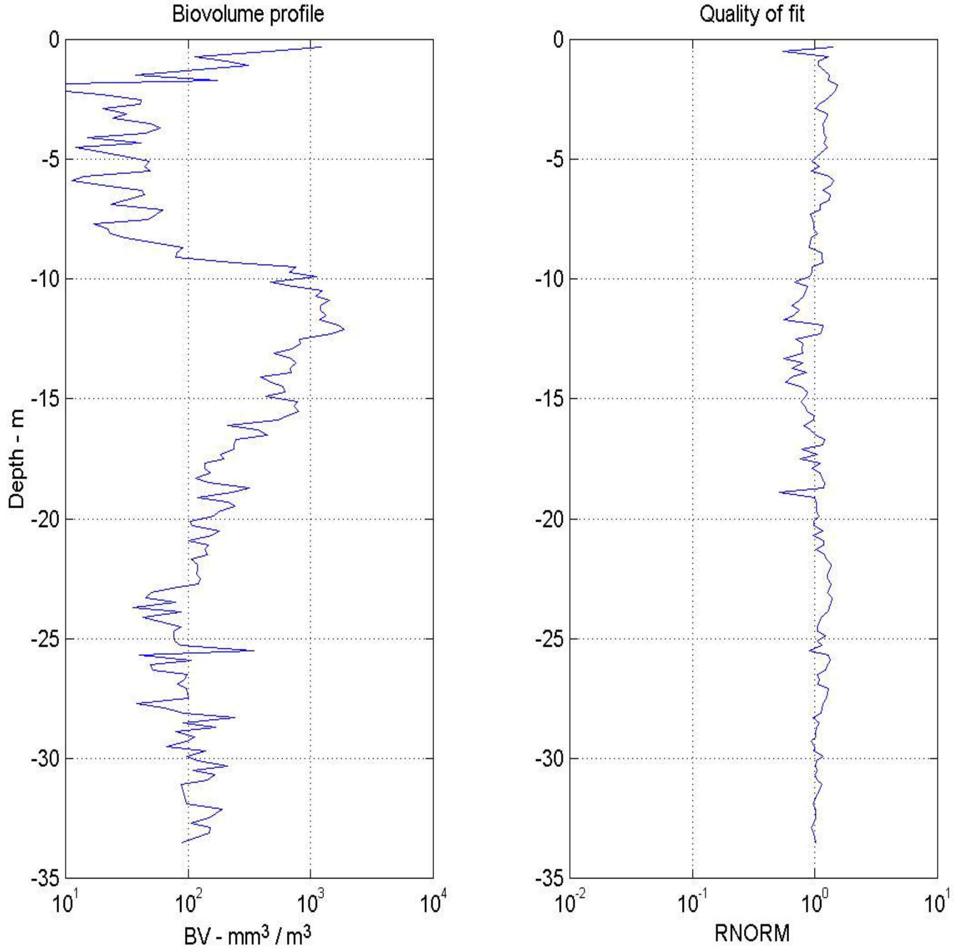

Figure 3. Typical output of **inv6** showing total BV and RNORM vrs depth. Data from Lake San Vicente, CA.

Appendix 3: Data Analysis

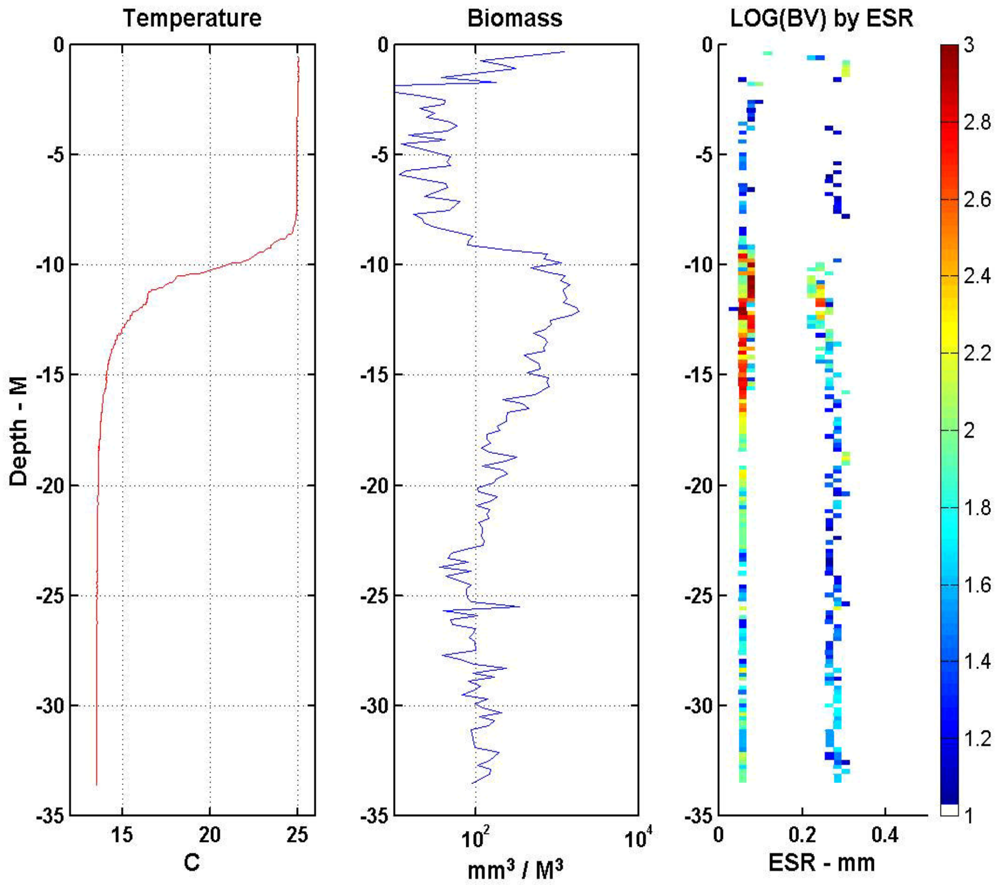

Figure 4. Typical output of **inv6** showing temperature profile, total BV by depth, and BV by depth and size class. Data from Lake San Vicente, CA. Zooplankters include a freshwater copepod and cladocerans. There is a clear peak in abundance just at the thermocline in both of the dominant size classes but some of each size is present almost throughout the water column.

PROGRAMS

The m-files needed to plot TAPS data are:

plot6.m	the main program
rdtaps6.m	a function to read files
absorb.m	a function to calculate absorption

The m-files needed to create model files are:

makeTS.m	the main program
fsa.m	truncated fluid sphere model
spbess.m	spherical Bessel function
neumann.m	neumann function

The m-files needed to invert the cast data are:

inv6.m	the main program
nnlsinv.m	the inversion function

These files are supplied on the TAPS CD in the folder **CAST**.

SUMMARY

In brief, these are the steps you would normally follow to turn .log files into plots of estimated biovolume versus depth and size:

Appendix 3: Data Analysis

- Run **plot6**. Break the data up into casts, correcting cal constants as required, and bin into suitable depth intervals. Save each cast to disk with a unique file name.

- Run **makeTS**, if necessary, to create a scattering model for the data set.

- Run **inv6**. Enter the model file name and cast file names as prompted.

INSTRUMENT DATA

A Matlab file is provided to convert INSTRUMENT mode data from ASCII-HEX to engineering data: **rdtapsinst.m**. This function is similar to the one used to read normal CAST mode data and is used in a program modified from **plot6.m**, called **plot6i.m**.

It will be necessary for you to modify this script prior to your first use, however. There is a line at the top of the function that sets the calibration constants to be used in scaling the volume scattering strength data. This line must be edited to contain the CAST mode calibration values for your TAPS.

Operation of **plot6i.m** is essentially identical to that described under CAST mode. Inversions should be done using **inv6.m** if the principal variable is depth (see below for data taken at constant depth).

TIME DATA

It will happen occasionally that CAST mode data will be taken at a fixed depth and it is desired to examine the data as a function of time instead of depth. A set of m-files are provided for processing this sort of data (**plottime.m**, **invtime.m**, and **replotinvtime.m**). These programs were modified from the equivalent CAST mode programs and operate, in general, just like these programs.

The major difference between **plot6.m** and **plottime.m** (other than the obvious one) is in the binning process. In **plot6.m** you bin data by depth; the number of data sets in any bin depends upon the speed of descent and is not necessarily constant. In **plottime.m** you bin N adjacent data sets, where N is a fixed number.

Invtime.m outputs data versus time rather than depth but is otherwise identical to **inv6.m**.

Appendix 3: Data Analysis

SOUNDER MODE DATA

SOUNDER mode data are quite different from CAST mode data. They contain some of the same information, but with a unique format. Moreover, the data require different processing from CAST mode data. The programs to read, convert, and display SOUNDER data are described in this section.

Because of the several modes of SOUNDER operation and data collection, the format of a data file can vary drastically. The data can be in ASCII text or binary form, with or without a data header, and with or without ancilliary information such as a STATUS screen. The quantity of data can vary as well, from a handful of files taken in internal SOUNDER mode to weeks of data taken in external SOUNDER mode. We have developed separate programs to deal with these different types of data: **makesndr.m** and **makermt.m**.

Makesndr.m is the program you would use to process data from internal SOUNDER mode or single external SOUNDER data files. This program can read ASCII or BINARY data files and convert them into volume scattering strength matrices that are suitable for inversion. Because the SOUNDER mode of TAPS operation emphasizes speed of data download, the raw acoustic accumulators are not converted into volume scattering strengths internally; this task is left to the conversion/analysis program.

NOTE: before you can use **Makesndr** to convert any of your data, you will have to edit the m-file (program) to include the calibration constants for your TAPS. The values to be used are the SOUNDER mode calibration constants shown on the cal sheet for your unit in Appendix 1: Calibration. Edit the line that appears as

cals=[0 0 0 0 0 0];

by entering the six values of SOUNDER calibrations constants given on the cal sheet in place of the six zeroes. Note that these will be positive numbers. The order is as shown on the calibration data sheet, from lowest frequency to highest. Anytime your TAPS is re-calibrated, the latest calibration constants should be entered in this program.

Conversion begins by entering the data file name. We customarily use filenames similar to those of CAST mode TAPS; e.g,

S990413b.log

would be the second SOUNDER mode data file obtained on the 13th of April, 1999. The program verifies that this file exists and then asks for the type of file—ASCII or binary.

SOUNDER mode permits data transfer as either ASCII text data or a binary data. **Makesndr** can read either type of data but you must tell it which type of data is in the file named previously. You did write this information down, didn't you?

While **Makesndr** can discriminate between text lines and data lines in an ASCII file, it cannot do so in a binary file. If you save SOUNDER data in binary mode, you must make sure that the data files do not contain extraneous characters (such as the commands used to RUN and DUMP data). They should contain only data.

Because of the stripped-down nature of the SOUNDER mode data storage, it is also necessary for you to enter the values for number of range bins and number of pings per average. You did write these numbers down also, didn't you?

Next, **makesndr** asks for the "deployment mode." Since TAPS can be oriented in any direction imaginable, it was difficult to incorporate all these choices for data display in one program. **Makesndr** makes distinctions between three cases: (d)own-looking, (u)p-looking, and (s)ide-looking. If TAPS was oriented looking down, and you answer ‘d’ to the request for deployment mode, then plots of Sv will be presented in depth format with the minimum depth (normally zero) at the top. Increasing depths will be displayed as negative numbers. If TAPS was oriented looking up or sideways, and you answer ‘u’ or ‘s’, then plots of Sv will be presented with minimum range at the bottom of the plot and maximum range at the top.

In the particular case of up-looking data, you have the opportunity to let the software detect the surface for you and clear all range bins above this point. **Makesndr** uses data from the 1850 kHz channel to locate the surface. It might be wise to verify that the surface can be seen before selecting this mode. When it is feasible, however, this mode produces quite pleasing displays of sub-surface scattering with tides clearly evident in the plots (Fig. 5).

If you select automatic surface detection, you will also be offered the choice to ‘surface-lock’ the displays. This option uses the surface detected above to re-arrange the data matrix, putting the surface echo at zero range and data below the surface at increasing depths. The resulting plot is identical to that for down-looking mode, except that tidal changes show as variations in the maximum depth of the data on the plot. All bins outside the region between TAPS and the surface are cleared (set to NaN—not a number) and do not display on the plot.

After the plots are presented, you are given the opportunity to replot the data using different depth ranges. The full range of depths available is displayed and you are given the opportunity to modify either the min or max or both values. The plots will then be re-drawn using the new depth axes. This process continues until you answer ‘n’ to the question, ‘Enter new depth ranges (y/n?)’.

Finally, you are asked to enter a file name for the binned Sv data. If you enter a name, the program will save the data necessary for performing the inverse estimates on the data. These data include: the decimal date, **zdate**; the depth bins, **dep**; the TAPS depths (one per profile), **d**; the TAPS temperatures (one per profile), **t**; six matrices containing the volume scattering strength profiles for the six channels, **sv1** to **sv6**; and a vector, **iv**, containing the data indices corresponding to the depth range you selected for the display plots. This latter vector can be used to select interesting portions of the data set for inversion.

Analysis of data obtained using **EZSOUND** proceeds much like was described just above. The major difference arises from the unique format of the data files stored by the datalogger program.

Files stored by **EZSOUND** contain up to one hour’s worth of profiles in binary format. The program supplied for SOUNDER mode data processing, **makesndr**, can read these files, one at a time, and the outputs can be inverted and plotted as described above.

An alternate version of **makesndr**, called **makermt**, is provided to simplify the initial processing of TAPS data obtained in REMOTE mode. This program operates very much like it’s SOUNDER counterpart. The major difference is in the file selection dialog.

Before running **makermt**, be sure to edit the calibration values as described above. Use the same values you would use for SOUNDER mode, from the calibration table in Appendix 1: Calibrations.

Makermt begins by asking you to select a data file (this setups up the path to the data, which could be on a floppy disk, CD, or on another computer accessed by a LAN). It then changes the active Matlab directory to this path.

Appendix 3: Data Analysis

Files are selected by entering the dates and times of interest. Recall that file names from **EZSOUND** are of the form SMMDDHH where S stands for SOUNDER, MM is the month, DD is the day, and HH is the hour at which the file was opened.

You are first asked to enter the month of interest. Then you are asked to enter the starting day and hour, followed by the ending day and hour. For example, if you wished to process all data from the 19th and 20th of May, you would enter 5 for the month, 19 for the starting day, 0 for the starting hour, 20 for the ending day, and 23 for the ending hour. (Recall the hour marks when the file begins, not when it ends.)

The program will calculate how many files (hours) this includes and display that value. You are asked to verify that everything is correct (enter ‘y’ if it is).

Next, you are asked if you want to surface-lock the data. This is the only option of this type available; answer ‘y’ if this option is desired.

You can enter a plot title, if desired. This is a text string that appears on the contour plots of Sv produced by **makermt**. (See Fig. 5 for an example.)

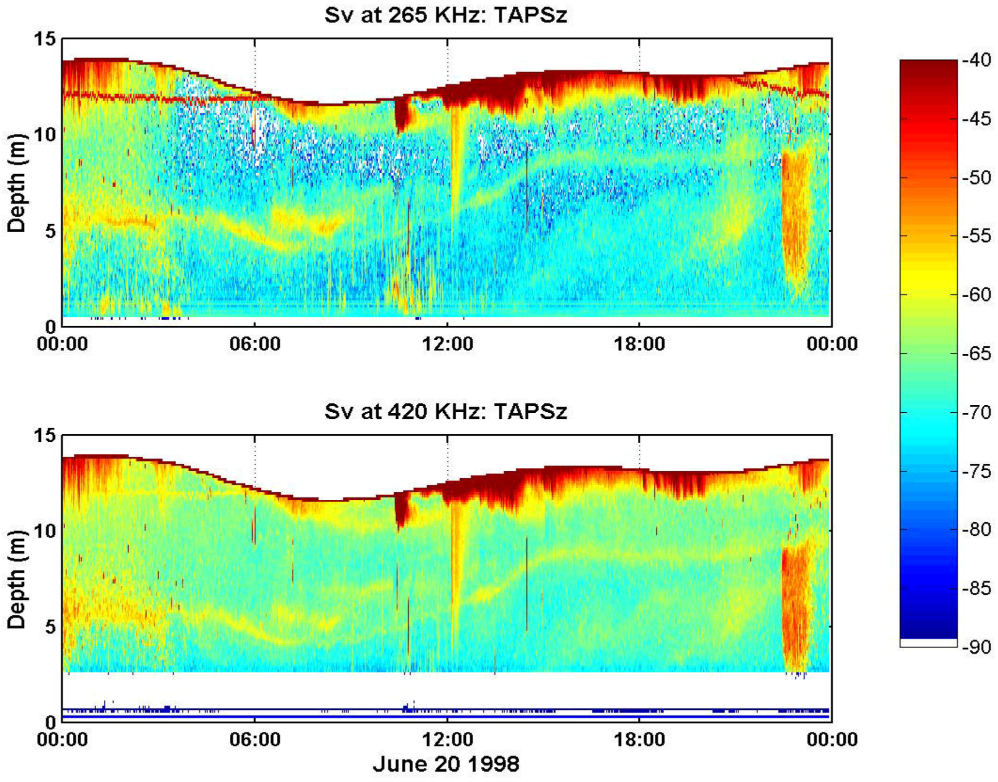

Figure 5. Example of processed Sv data from **makermt.m**. Data taken in East Sound, WA from TAPS-6 moored at approximately 12 m below the surface, looking upwards. Profiles were taken at 1 minute intervals. The thin layer at 5-9 m was the feature of principal interest in this study. Note the occurrence of individual fish and fish schools at night and the strong surface scattering in the afternoon when the wind speed increased.

You must also enter certain pertinent parameters of the data set, including the number of range bins, number of pings, and the file prefix. Normally, this latter will be an ‘S’ but it is possible to operate an array of TAPS remotely, when the data from each unit might be identified by a unique prefix like X, Y, and Z. In fact, we have done just that. Contact us for details if you are interested in such a setup.

The ouput of **makermt** is a single, possibly large, mat-file. This file is in the proper format, however, for use by **invsndr**. Thus, subsequent processing of REMOTE mode data can be accomplished by the programs already described in the SOUNDER section of this appendix.

Figures 5-8 display data obtained using the REMOTE SOUNDER MODE of operation for TAPS.

INVERTING SOUNDER DATA

It is assumed that you have created a suitable scattering model using **makets** as described in the CAST mode section above.

The inversion process is performed by the program **invsndr.m**. This program operates very much like **inv6** does. It checks first for the presence of a model on the workspace and requests you select a model file if one is not found. You are then prompted to find the data file.

The first difference is that you are asked for an output file name before any processing has been done. This is because the file structure for SOUNDER data differs from that for CAST data. If you think about it for a moment, you will see that each SOUNDER profile is basically identical to a complete CAST profile. To keep file sizes within reason, we have elected to save each SOUNDER profile as a separate file. In order to keep files associated properly, the file names are composed of two parts: a generic part and a sequential number. The file name you are asked to enter is the generic part, the program will take care of adding the appropriate number to form the complete file name.

Generic output file names should be short—as Matlab doesn’t really like long file names and **invsndr** adds 3 characters to the end of your input—and descriptive, so you can remember what the files represent. They can contain numbers, though that may end up confusing you a bit.

Every time you process SOUNDER data using **invsndr**, you will obtain a minimum of two mat-files. If your generic name were, say, ‘SPB’ (to remind you this was the San Pedro Basin data set), the first file would be named SPB000.mat and the second file SPB001.mat. The first file holds several vectors that make it possible to plot the data, which is contained in the second file.

SPB000.mat contains the ESR vector, **a**; the bin depth vector, **dep**; the TAPS temperature and depth vectors, **t** and **d**; a decimal date vector, **zdate**; and the vector of depth indices, **iv**. The temperature, depth, and decimal date vectors are as long as the number of profiles in the input data—if you have 5 profiles in the input data file, you will have 5 temperatures and 5 dates.

SPB001.mat contains a single matrix, **outdat**. This matrix consists of **ndepth** rows (**ndepth** is the number of depth bins). Each row is **nsizes** + 2 elements long (**nsizes** is the number of ESR’s in the model). The first **nsizes** columns contain biovolume (BV in mm^3/m^3) for each of the ESR classes. The next column contains the total BV, summed over all ESR’s. The last column contains the value of **rnorm**, a measure of the goodness of fit of the inverse solution.

The input data will normally consist of several separate SOUNDER profiles and the same number of data files will be generated by **invsndr**. If the input data for San Pedro Basin contained 45 profiles, then **invsndr** would generate 46 output files: SPB000, SPB001, SPB002, ..., SPB0045.

If you had selected a limited set of ranges (depths) to plot in **makesndr**, **invsndr** will assume that you only want to invert those data. However, you will be asked if you want instead to invert all the data. Answer ‘y’ if you do.

You will be next be asked to choose whether to delete selected frequencies and to select the range at which to begin deleting each frequency. This option lets you, for example, delete the 3 Mhz data past 4m range if the display from **makesndr** or **makermt** shows that there are no echoes above the noise past this range. You may delete more than one frequency; enter 0 when you are done.

You may keep the default Levenberg-Marquardt factor or enter a new value, as described under CAST mode data processing and in Appendix 2: Basic Acoustics.

Displays for **invsndr** are similar to those described in CAST mode inversion during the inversion process. Data from each range bin is inverted separately; when a profile has been completed, a plot of BV vrs depth and RNORM versus depth is displayed for a short time to allow inspection of the results for the entire profile.

DISPLAY INVERSION RESULTS

The possibilities for displaying SOUNDER data are nearly endless. The number of programs supplied to display these data is not, however. In fact, there is only one: **plotbvs4.m**.

This program reads the files generated by **invsndr**, sums up the BV into four categories (small, medium, large and really large), and plots these as color contours on a depth by time axis (Fig. 6-7). A summary figure, showing mean ESR in the upper frame and total BV in the lower frame is also generated (Fig. 8). What could be simpler?

You may not find the simplistic division of size classes in this program to be very satisfactory. We tend to set the size class divisions to enhance a biological interpretation of the data whenever possible—sometimes we are even interested in a single size class— you might wish to do the same. This will require that you edit the m-file, possibly often. Find the section of code entitled “Select size categories ...”. The lines of code following that heading compute indices that split the ESR vector into four segments. These indices (**n1** to **n3**) are used in subsequent code to sum the biovolumes into the three sub-categories:

SMALL:	1-n1
MEDIUM:	(n1+1):n2
LARGE:	(n2+1):n3
WHALES:	(n3+1):na

Total biovolume is computed as the sum over all sizes.

To run the program, you need only enter the generic part of the data file names. Three figures, each with two plots, will be displayed (see Fig. 6-8). Biovolume in the four size classes

Appendix 3: Data Analysis

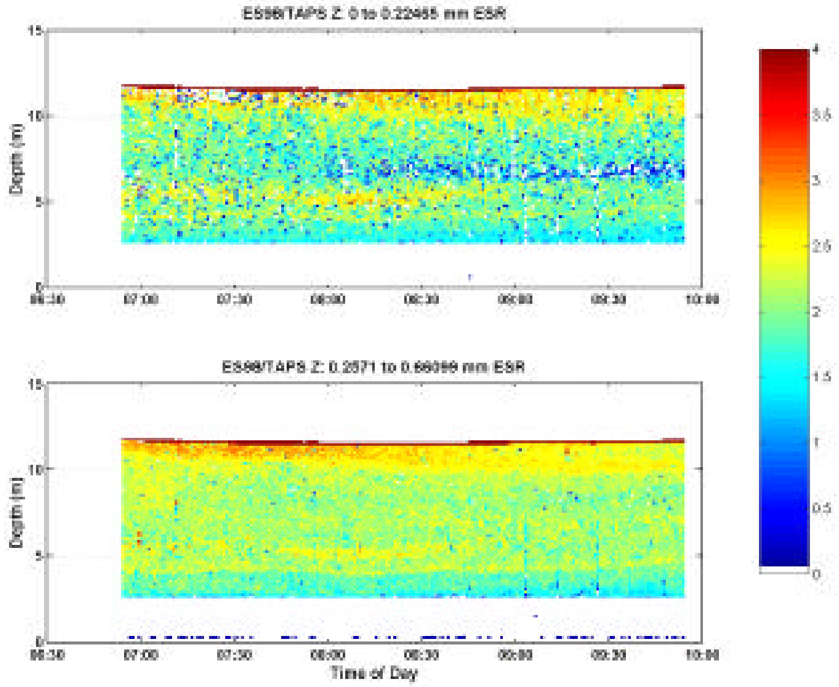

Figure 6. Typical output from **plotbvs4**. Data from East Sound, WA. Log BV in smallest two size classes versus depth and time. TAPS moored about 12m below surface.

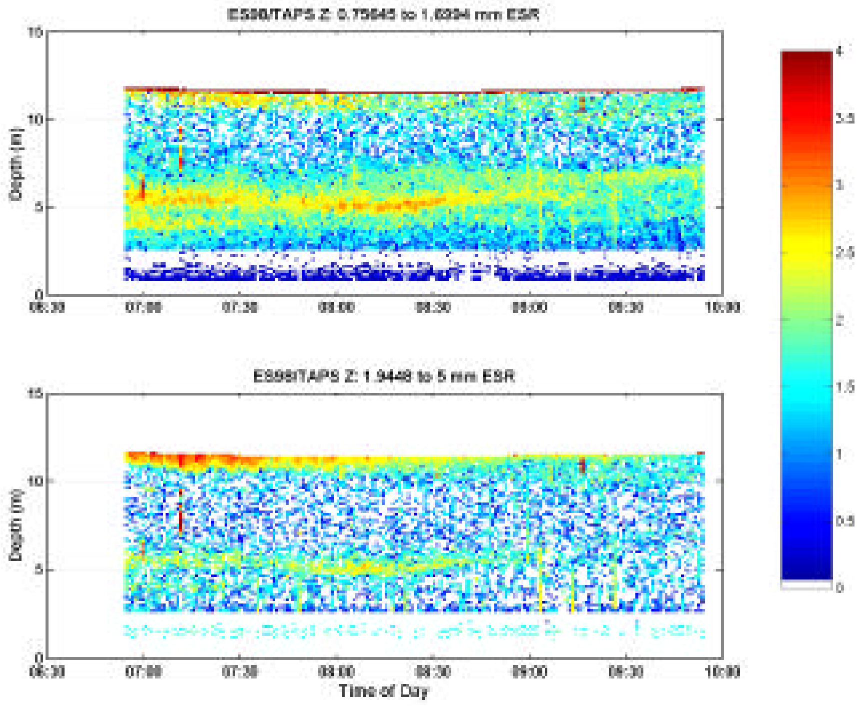

Figure 7. Typical output from **plotbvs4**. Data from East Sound, WA. Log BV in largest two size classes versus depth and time. TAPS moored about 12m below the surface.

Appendix 3: Data Analysis

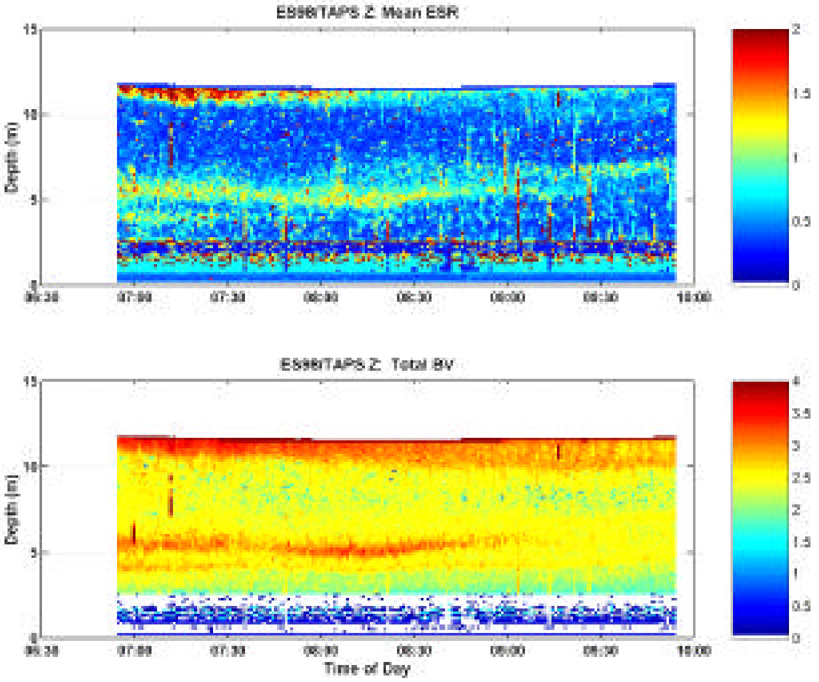

Figure 8. Typical output from **plotbvs4**. Data from East Sound, WA. Mean ESR (top) and total BV (bottom) versus depth and time. TAPS moored about 12m below surface.

(small, medium, large, and really large) are plotted versus depth (or range) and time. Color represents $\log_{10}$(biovolume in mm^3/m^3) and, for the first iteration, is autoscaled.

The third figure displays the mean ESR and total BV together. Mean ESR is computed as the 'center of mass' measure of ESR (sum of the products of ESR and BV(ESR) divided by the total BV) for each depth bin. Color represents ESR on this plot. Total BV is the sum of BV over all size classes. Color represents $\log_{10}$(biovolume in mm^3/m^3) as above.

After the plots are displayed, you are asked if you wish to change the plot limits. If you answer 'y', you will be asked to enter a new value for the low limit and for the high limit of BV, ESR, and depth. You should use the $\log_{10}$ values displayed on the colorbar (e.g., -2 for 0.01) for BV. When all data are entered, the plots are redrawn using these new limits for the color contours.

PROGRAMS

The m-files needed to plot SOUNDER data are:

makesndr.m	the main program, or
makermt.m	
rdbsndr.m	reads binary files
rdasndr.m	reads ASCII files
rdhsndr.m	reads hourly files.

The m-files needed to create model files are:

makeTS.m	the main program
fsa.m	truncated fluid sphere model
spbess.m	spherical Bessel function
neumann.m	neumann function.

Appendix 3: Data Analysis

The m-files needed to invert the SOUNDER data are:

invsndr.m	the main program
nnlsinv.m	the inversion function.

The m-files needed to plot the biovolume data are:

plotbvs4.m	the main program.

These files are supplied on the TAPS CD in the folder **SOUNDER MODE**.

SUMMARY

In brief, these are the steps you would normally follow to turn .log files into plots of estimated biovolume versus depth and size:

- Run **makesndr** or **makermt**. Save the data to disk with a unique file name.
- Run **makeTS**, if necessary, to create a scattering model for the data set.
- Run **invsndr**. Enter the model file name and data file names as prompted.
- Run **plotbvs4**. Plot the data in several size classes versus depth or range.

Appendix 3: Data Analysis

SAMPLE DATA

Sample data files have been provided on the TAPS CD in the folder, SAMPLES. Examples of CAST data, INSTRUMENT data, and SOUNDER data taken at sea and in our local lakes have been put here so that you might have some actual data on which to exercise the processing programs.

CAST data includes the following files:

```
T6EX1.txt is a profile taken off
   Bermuda.
T6EX2.txt is a profile taken in a
   Lake San Vicente, CA.
T6EX3.txt is a profile taken off
   Long Beach, CA.
T6EX4.txt is a profile taken off
   Long Beach, CA.
T6EX5.txt is a profile taken off
   Long Beach, CA.
CAST01_02.log is a pair of
   profiles from Lake Miramar,
   CA.
T010223.log is a profile from
   Lake Miramar with a fluor-
   ometer attached.
```

These profiles may be used to exercise the CAST mode processing programs.

One file of INSTRUMENT data, taken in a local lake, has been provided (in the CASTSAMPS directory). This file, INSTDAT.LOG, is not a particulary useful file except for the purposes of verifying how the program operates.

There are two sets of SOUNDER data in the SAMPLES folder. The file SOUNDER01.LOG is an ASCII-text file taken in internal SOUNDER mode in a local lake. TAPS was suspended from a dock with the transducers pointed down. For these data, you will need to run **makesndr.m**; enter 'asc' for the data type, 160 for # of range bins, 24 for # of pings, and 2000 for the year. The default cal constants in the program are suitable for this data set.

If you save the processed data from this sample set, you will have a useful-sized data set for exercising the inversion program, **invsndr.m**.

A set of data files taken with a predecessor of **EZSOUND** is provided to demonstrate **makermt.m**. These data are from the data set used as examples in this section of the manual.

Run the program and browse to the sample folder and click on any of the files named Zxxxxxx.dat. These data cover a time period from 1200 on 6/19 to 1200 on 6/21. You may enter any part or all of this time period in **makermt.m**. If you wish, you can save processed data from this program for exercising the inversion program, **invsndr.m**. Keep in mind that inversion is a cpu-intensive process; for purposes of demonstration it is better to deal with fairly small data sets.

For this data set, the number of bins is 117, the number of pings is 24, and the prefix for the data is "Z".