

MB-System Training Tutorial

Authored by the MB-System Development Team

Christian Ferreira ([MARUM/University of Bremen](#))

Jenny Paduan ([MBARI](#))

David Caress ([MBARI](#))

With additional help from Evgenia Bazhenova ([MARUM](#))
in the frame of the Nippon Foundation-GEBCO Alumni project

Additional credits

This document is partially based on the “MB-System Training Tutorial”, originally created by:

MB-System Dev Team (2012 - 2024)
(Christian Ferreira ([MARUM/University of Bremen](#))
Krystle Anderson ([MBARI](#))
David Caress ([MBARI](#))
Dale Chayes ([CCOM/University of New Hampshire](#))

This document also includes excerpts from the “MB-System Cookbook”, originally created by:

Val Schmidt ([CCOM-UNH](#))
David Caress ([MBARI](#))
Dale Chayes ([LDEO/ CCOM-UNH](#))

Changelog:

Version	Date	Changes	Responsible
0.5	01/2025	First draft	CSF
0.8	03/2025	Second draft	CSF
0.9	05/2025	Final draft	CSF
1.0	07/2025	Initial release	CSF, DC, JP, EB

Download:

This tutorial is available from the GitHub repository at the link or QR code below:

<https://github.com/dwcaress/MB-System-Tutorial>



Feedback:

If you have feedback regarding this tutorial, please send it to cferreira@marum.de. Or post a comment in the “Issues” section on GitHub.

General notes:

Besides the sample data used in this tutorial (link in Chapter 1.2.1), there are also supplementary script files in the GitHub repository. One script covers all the commands from Ch. 1, which describes how to create a bathymetric map from raw data. A few others cover a real-world scenario on how to use MB-System in production mode. While this tutorial covers superficially nearly all the commands and concepts used in the scripts, this scenario goes far beyond the scope of this tutorial and is shared with two purposes: a) incentivizing users who want to learn more about MB-System; and b) build over time a collection of scripts to deal with all formats supported by MB-System. A quick explanation about how to use them is given in Chapter 5.3., item # 8.

Disclaimer:

This document is distributed under the GNU Free Documentation License version 1.3. For further details, please visit the following website: <http://www.gnu.org/copyleft/fdl.html>

Index

Chapter 1 - Basic MB-System

1. MB-System from raw data to map – page 5

1.1. Introduction – page 5

1.2. Using MB-System – page 5

1.2.1. Basic introduction and data download – page 6

1.2.2. Creating the first datalist (mbm_makedatalist) – page 7

1.2.3. Translation from the vendor's "raw" format to the "preprocessed" format (mbpreprocess) – page 8

1.3. Creating processed files (mbprocess) – page 9

1.4. Gridding the data (mbgrid) – page 9

1.5. Viewing your grid (mbgrdviz) – page 10

1.6. Plotting your grid (mbm_grdplot) – page 11

Chapter 2 - Data corrections

2. Corrections – page 13

2.1. Amplitude/Sidescan correction (mbbackangle) – page 13

2.2. Sound velocity correction (mbvelocitytool) – page 13

2.3. Tide corrections (mbotps) – page 15

2.4. Lever arms offsets and biases – page 15

Chapter 3 - Data cleaning

3. Cleaning data – page 16

3.1. Manual editing – page 16

3.1.1. 2D editor for multibeam data (mbedit) – page 16

3.1.2. 3D editor for multibeam data (mbeditviz) – page 17

3.1.3. Navigation editing (mbnavedit) – page 22

3.2. Editing automatically – page 23

3.2.1. Multibeam data (mbclean) – page 23

3.2.2. Sidescan data (mbfilter) – page 23

Chapter 4 - Metadata

4. Extracting basic information from your data (mbinfo) – page 23

4.1. Inserting metadata into every single file (mbset) – page 24

Chapter 5 - Miscellaneous and advanced topics

5.1. Multiple surveys in a single datalist – page 27

5.2. MB-System best practices – page 28

5.3. More advanced MB-System – page 29

5.4. Common MB-System workflow for Kongsberg data – page 30

Chapter 6 - AUV data processing

6.1. Main differences between the workflow of ship and AUV data – page 32

6.1.1. Preprocessing - page 32

6.2. Using mbnavadjust – page 32

6.3. Common MB-System workflow for AUV data – page 33

Chapter 7 - Dealing with other common formats in preprocessing

- 7.1. Kongsberg ALL/KMALL – page 35
- 7.2. Teledyne Reson S7K – page 35
- 7.3. Other formats – page 35

Appendices

- A. History – page 37
- B. MB-System structure and auxiliary files – page 37
 - B.1 The main structure of MB-System – page 37
 - B.2. Vendor format, editing format and processed files – page 37
 - B.2.1. Understanding the raw/processing format and the auxiliary files – page 38
- C. MB-System installation step-by-step on Mac OSX 15.5 (Sequoia) using Macports – page 40
- D. MB-System/OTPS installation on a sandbox (home folder) using Ubuntu 24.04 – page 42
- E. Extra steps after installing MB-System on Linux or MacOSX - page 43

Chapter 1 - Basic MB-System

1. MB-System from raw data to map

1.1. Introduction

MB-System (<https://www.mbari.org/technology/mb-system/>) is an open-source software package designed to process and display seafloor mapping data. This tutorial focuses on processing swath bathymetry and acoustic backscatter data obtained from multibeam, interferometry, and sidescan sonars. By swath data, we mean data that are organized as soundings and pixels associated with individual sonar transmissions, or pings, and that are located in the world using the position (or navigation) and orientation (or heading and attitude) of the sonar at the time of each ping. MB-System supports a number of different swath data formats, including the most common current open formats and many legacy formats associated with sonars that are no longer sold or operated:

http://www3.mbari.org/products/mbsystem/html/mbsystem_formats.html

This software is freely distributed in the form of source code that can be compiled and installed on Linux, BSD, and MacOs computers. To obtain and install the current MB-System distribution, follow the instructions in the link:
<https://www.mbari.org/technology/mb-system/installation/>

MB-System consists primarily of command-line programs, complemented by a few graphical tools for editing, analysis, and visualization. A comprehensive list of available MB-System commands and tools and their corresponding manual pages is at this link:

http://www3.mbari.org/products/mbsystem/html/mbsystem_man_list.html

Alternatively, these same manual pages are installed with the software and become accessible through your terminal window by executing the command “man” followed by the MB-System command of interest.

Commands commencing with the prefix “mb” are compiled programs written in C or C++, whereas those commencing with “mbm_” are scripts written in Perl or Python3 that are termed “macros” in MB-System documentation. These macros combine MB-System programs with system commands and programs from other packages to efficiently facilitate common tasks. The most commonly used macros combine MB-System and GMT (Generic Mapping Tools - <https://www.generic-mapping-tools.org/>) programs to easily generate maps from swath files or, more commonly, gridded bathymetry and mosaicked backscatter.

Help can also be found in the MB-System user discussion list archive or by posting questions to the user community, the address for which is at:

<https://www.mbari.org/technology/mb-system/discussion/>

For information regarding MB-System’s historical background, refer to Appendix A.

1.2. Using MB-System

The purpose of this tutorial is to provide a foundation for understanding MB-System, gradually building upon the basics. It is assumed that the reader has no prior experience with MB-System. The primary objective of this chapter is to transform raw data into a structured representation. Subsequent chapters will cover more advanced post-processing steps, organized by topic, and a complete description of the MB-System processing structure and data format handling is presented in Appendices A and B. However, for the purpose of this introduction, let us begin with a fundamental understanding of MB-System.

To demonstrate the functionality of MB-System, we will introduce it using a dataset acquired with a Kongsberg EM122 (a 1° by 2° degrees system for transmitting and receiving signals, respectively, and comprising 432 beams). The data identification (format) is specific to the third-generation Kongsberg systems (i.e., EM122, EM302, EM710, EM2040, and M3). For vendor format (files logged with the .ALL extension in their file names), the MB-System data ID is 58 (with extensions .mb58 or .all), while the processing files will have an ID of 59 (with an extension of .mb59).

For this tutorial, we utilized a computer running macOS 15.x (Sequoia) as evident from the screenshots. Rest assured, Linux users need not be concerned, as we have meticulously avoided any Mac-specific issues to ensure the universality of this tutorial. Furthermore, the graphical tools provided by MB-System maintain their identical appearance and functionality across both operating systems.

A number of the tools require a three-button mouse or a mouse that can at least emulate such functionality. Mac users with Apple mice 2-button mice might try a third-party software solution that facilitates the easy assignment of a third button.

1.2.1. Basic introduction and data download

We commence with the fundamental steps. Firstly, we download a data set and construct a file that lists all our multibeam data files. Subsequently, we employ the gridding tool to grid the data, followed by the plotting of the gridded data to generate a map. No post-processing has been undertaken at this stage. This methodology provides a gradual introduction to the typical workflow of MB-System, which predominantly relies on command-line operations and some graphical interface tools for processing the data.

For this tutorial, we will utilize a data set packaged as “PatchSurveyEM122.zip”. You may download it from Zenodo at the following link:

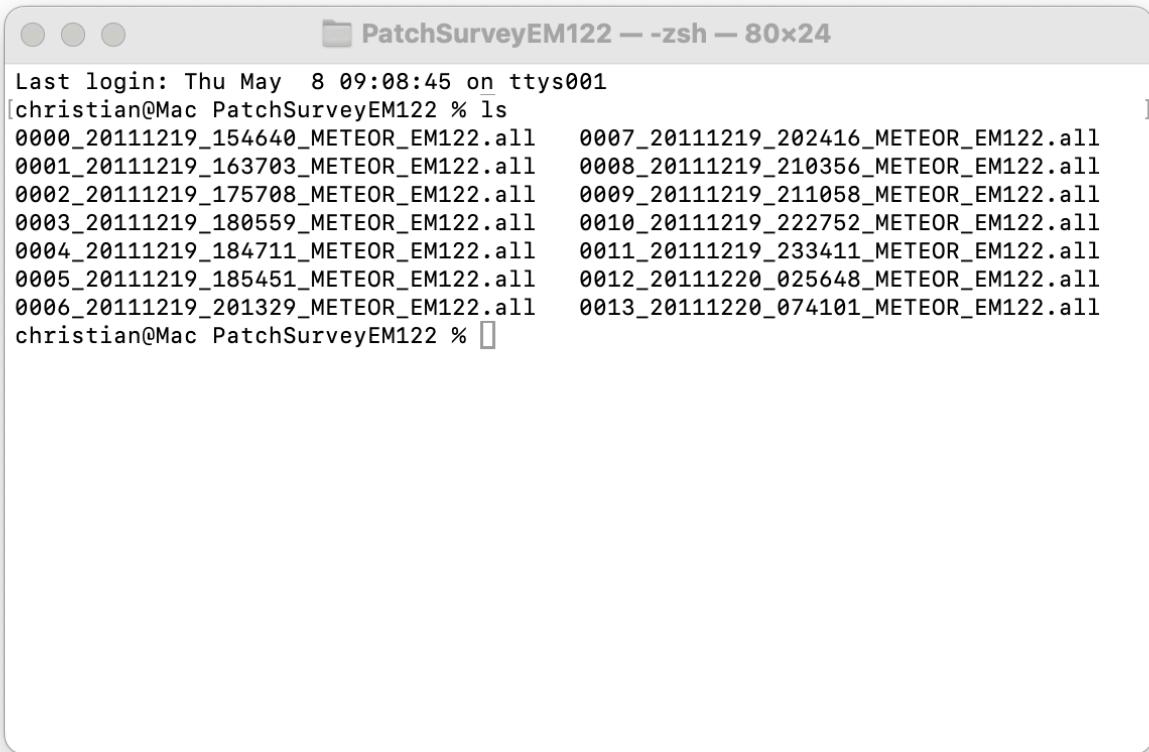
<https://doi.org/10.5281/zenodo.15689284>

After downloading and unzipping the file, navigate to the data directory. Open a terminal within the folder and execute the following command:

```
% ls
```

Note: The “%” symbol is **never** part of any command displayed during this tutorial. It merely indicates that the user is utilizing a terminal for inputting commands.

This command will list all the files in the current directory. The output should be presented in a tabular format:



A screenshot of a Mac OS X terminal window titled "PatchSurveyEM122 — zsh — 80x24". The window shows the output of the "ls" command, listing 14 files with ".all" extensions, each corresponding to a specific date and time. The files are arranged in two columns. The first column contains file names like "0000_20111219_154640_METEOR_EM122.all" through "0006_20111219_201329_METEOR_EM122.all". The second column contains file names like "0007_20111219_202416_METEOR_EM122.all" through "0013_20111220_074101_METEOR_EM122.all". The terminal prompt "christian@Mac PatchSurveyEM122 %" is visible at the bottom.

```
Last login: Thu May  8 09:08:45 on ttys001
[christian@Mac PatchSurveyEM122 % ls
0000_20111219_154640_METEOR_EM122.all  0007_20111219_202416_METEOR_EM122.all
0001_20111219_163703_METEOR_EM122.all  0008_20111219_210356_METEOR_EM122.all
0002_20111219_175708_METEOR_EM122.all  0009_20111219_211058_METEOR_EM122.all
0003_20111219_180559_METEOR_EM122.all  0010_20111219_222752_METEOR_EM122.all
0004_20111219_184711_METEOR_EM122.all  0011_20111219_233411_METEOR_EM122.all
0005_20111219_185451_METEOR_EM122.all  0012_20111220_025648_METEOR_EM122.all
0006_20111219_201329_METEOR_EM122.all  0013_20111220_074101_METEOR_EM122.all
christian@Mac PatchSurveyEM122 %
```

FIGURE1- TERMINAL OUTPUT GENERATED BY EXECUTING THE COMMAND “LS”.

1.2.2. Creating the first datalist (`mbm_makedatalist`)

MB-System programs each act on a catalog of data files, henceforth referred to as a “datalist”. First we will make a datalist of the “logged” files that we just downloaded, “datalist_raw.mb-1”. Then, in a preprocessing step we will prepare the logged files for further processing, which converts them to what will be called “preprocessed” files, and their datalist is “datalist.mb-1”. Later, all programs will act on either the preprocessed files or, to utilize the processing we will apply to them, on “processed” files, which will be accessed with “datalistp.mb-1”. Datalists can be lists of other datalists, which becomes a useful shortcut to working with multiple surveys of diverse data (see Chapter 5). For all MB-System programs, use the option “-I” or “--input=” to input the desired datalist. The expected format ID for a datalist is mb-1, so adding “-F -1” to your commands is not necessary, however, if you wish to open a single data file in a program, you will need to specify the format ID of the file, so for example, by typing:

```
% mbedit -F 59 -I 0012_20111220_025648_METEOR_EM122.mb59
```

The first step is to create the datalist of the logged files. The macro command “`mbm_makedatalist`” will enumerate all the files with the extension “.all”, generate a secondary column with this sonar’s MB-System format ID number (which for these files is 58), and output these to a file named “datalist_raw.mb-1”. The command line should be structured as follows:

```
% mbm_makedatalist -S.all -Odatalist_raw.mb-1
```

The resulting datalist_raw.mb-1 file should appear as follows when opened in a text editor:

```

1 0000_20111219_154640_METEOR_EM122.all 58
2 0001_20111219_163703_METEOR_EM122.all 58
3 0002_20111219_175708_METEOR_EM122.all 58
4 0003_20111219_180559_METEOR_EM122.all 58
5 0004_20111219_184711_METEOR_EM122.all 58
6 0005_20111219_185451_METEOR_EM122.all 58
7 0006_20111219_201329_METEOR_EM122.all 58
8 0007_20111219_202416_METEOR_EM122.all 58
9 0008_20111219_210356_METEOR_EM122.all 58
10 0009_20111219_211058_METEOR_EM122.all 58
11 0010_20111219_222752_METEOR_EM122.all 58
12 0011_20111219_233411_METEOR_EM122.all 58
13 0012_20111220_025648_METEOR_EM122.all 58
14 0013_20111220_074101_METEOR_EM122.all 58
15

```

Lines: 14 Characters: 574 Location: 574 bytes | Unicode (UTF-8) | LF

FIGURE 2 - CONTENT WHEN WE OPEN THE DATALIST OF LOGGED FILES WITH A TEXT EDITOR.

1.2.3. Translation from the vendor's “raw” format to the “preprocessed” format (`mbpreprocess`)

Most modern data formats necessitate a data translation from the vendor's format as it was logged, which we will refer to here as the “raw” data, to an enhanced format that can be utilized by MB-System, which we will refer to as the “preprocessed” data. It is to the preprocessed data that edits and corrections will be applied to generate “processed” data, and the raw files will no longer be used. This step is not essential for simple plotting purposes but is mandatory for processing tasks that will be discussed in the subsequent chapters, because the original formats typically lack the capability to accommodate user-generated edits, additional metadata or comments, or might contain data from multiple sensors logged in asynchronous order.

We will transform the logged EM122 data from its raw format (ID 58) to the processing-compatible, preprocessed format (ID 59) with the “`mbpreprocess`” command. It will read data from this third-generation Kongsberg multibeam sonar and interpolate the asynchronous navigation, heading, and attitude (motion) data and embed it correctly with the multibeam data. The new files it writes will have the names of the original files but with the extension of “.mb59”. The command for this task is:

```
% mbpreprocess --input=datalist_raw.mb-1 --verbose
```

Under most circumstances, the positional offsets of the sensor array on a ship will have been carefully measured and regularly recalibrated and the angular biases should be accommodated by motion sensors, and their values should be already be incorporated by the sonar into the data stream. A tool to evaluate the angular biases resides in the program “mbeditviz” and will be introduced later in this tutorial. The values being used can be output as a “platform” file

(extension “.plf”), which can be edited using a text editor if they need to be adjusted. The following commands will create the platform file (which the user can edit manually) and apply it using mbpreprocess.

```
% mbmakeplatform --swath=datalist_raw.mb-1 --verbose --output=Meteor_EM122.plf  
% mbpreprocess --input=datalist_raw.mb-1 --verbose --platform-file=Meteor_EM122.plf
```

Upon completion of this task, we must create two new datalists. One (datalist.mb-1), will contain a list of our newly created, preprocessed format 59 files, while the other (datalistp.mb-1) will be used in commands to call all those files but utilize processed versions of the files, if they exist.

```
% mbm_makedatalist -S.mb59 -P -V  
% mbdatalist -I datalist.mb-1 -Z -V
```

The command “mbdatalist” given alone will output all of the files called by the datalist.mb-1, with their paths as found on your system, format IDs, and priorities (discussed in Chapter 5).

The option “-Z” generates the datalistp.mb-1 that you will specify as the input file in commands to use processed data. That file has the form:

```
$PROCESSED  
datalist.mb-1 -1
```

The option “-I” by default reads datalist.mb-1 for the input file so might be omitted from commands. However, it could instead specify a custom datalist (see Chapter 5). Then the option “-Z” will append “p.mb-1” to that custom datalist name and put that name in the file rather than “datalist.mb-1” as shown here.

1.3. Creating “processed” files (mbprocess)

During the preprocessing step, some “auxiliary” (also known as “ancillary”) files will have been generated, which are files that accompany the data and are used by MB-System for various purposes. Many of the subsequent processing steps will create their own auxiliary files or require a common file to be associated with all of the data files. The parameter files (with the extension “.par”) control what files are associated with the data files. These are text files that can be modified individually in a text editor or as a group with the “mbset” command; type “man mbset” for details.

After any processing steps have been undertaken, the modifications stored within auxiliary files need to be applied to the preprocessed data to generate or update the processed files. Processed files will have “p.mb59” in their filenames. To process the files, execute the “mbprocess” command:

```
% mbprocess -I datalist.mb-1
```

You can add the option -C to specify the number of CPUs mbprocess can utilize. It is recommended to refrain from using a number exceeding the actual number of cores available on your machine. Even for machines with octa-core (8 CPUs) like the Intel Core i7, informal tests have demonstrated that, despite utilizing a very fast Solid State Drive (SSD), data read/write speeds may be slower with 8 cores compared to 4. For Hard Disk Drives (HDD), the data output is significantly slower, and a number between 2 and 4 is typically the maximum recommendation.

1.4. Gridding the data (mbgrid)

Once mbprocess has completed, it is time to make a map. First, we make a grid. To interpolate the bathymetric data, we execute the command “mbgrid” using the processed datalist. By default this will output a COARDS compliant netCDF grid with the extension “.grd” in Geographic coordinates (WGS84 horizontal datum), which can be further imported into a GIS software (e.g. QGIS).

```
% mbgrid -A2 -N -I datalistp.mb-1 -F5 -E75/75m! -OMeteor_M86-1C_75m_bathy_F5
```

Where:

- A2: Sets the grid as “topography” mode. Depth values are interpreted as positive upwards. Alternatively, one can use “-A3” to grid amplitude data or “-A4” to grid sidescan data.
- N: Causes grid cells with no data and no interpolation to be assigned a value of “NaN” (Not-a-Number) instead of the default value of 99999.9.
- I: Specifies the input datalist; note that we instructed mbgrid to use datalistp.mb-1.
- O: Assigns a name to the output grid. It is advisable to choose a meaningful, descriptive name to avoid confusion and save time in identifying the grid. In this example, we use “Meteor_M86-1C_75m_bathy_F5” to indicate that a 75m grid has been constructed.
- F: Selects the interpolation method. Please refer to the man page for further details.
- E: Sets the X-Y resolution for our grid. In this case, we use 75 meters. If this option is not set, the program will calculate a resolution. By default the coordinate system is WGS-1984 geographic, so except at the equator, an unprojected grid's cells will not be square. If you want square cells, which some GIS programs require, use “-E75/0!”.

Other useful options:

- R <west/east/south/north>: With this option, you can set geographical limits for your map region when you want to plot only a subset of the entire survey area. The values must be in decimal degrees. If this option is not set, by default the entire area covered by the datalist will be gridded. Note that too large an area at too high a resolution may cause a segmentation fault.
- S speed: Sets the minimum speed (in km/hr) allowed in the input data. Can be useful if the sonar was left on while the ship stopped for other operations.
- C clip: can help interpolate between grid cells not filled by swath data
- J projection: specifies a desired projection for the grid. A complete list is in the man page for “mbgrid”.
- V: verbose mode: causes the program to reveal its progress, which includes listing the data files being read.

We can also grid the amplitude (-A3) and sidescan (-A4) data from our files:

```
% mbgrid -A3 -N -I datalistp.mb-1 -F5 -E75/75m! -OMeteor_M86-1C_75m_amplitude_F5
```

```
% mbgrid -A4 -N -I datalistp.mb-1 -F5 -E75/75m! -OMeteor_M86-1C_75m_sidescan_F5
```

For further details, refer to the man page for mbgrid.

Information about your grid such as lat/long bounds, projection, and cell size can be retrieved by typing:

```
% gmt grdinfo Meteor_M86-1C_75m_sidescan_F5.grd
```

1.5. Viewing your grid (mbgrdviz)

A grid can be viewed and explored in both 2D and 3D with the MB-System program “mbgrdviz”:

```
% mbgrdviz -I Meteor_M86-1C_75m_bathy_F5.grd
```

The grid will appear at its full extent, in 2D, symbolized with the Haxby color palette and shaded by slope by default. Click on “View” in the upper left to change these view settings or enable others. The default “Pan & Zoom” mouse behavior allows you to pan with the middle mouse button and zoom with the right mouse button. Picking a point or drawing a line with the left mouse button displays location and distance information in the left panel and in your terminal window, where it can be grabbed if you want to paste the values elsewhere. Draw a line across a feature on the seafloor and visualize it in cross-section by selecting “View --> Profile”.

Expand the window to show a list of other mouse behaviors along the left, and the corresponding mouse button actions will display in an upper left window as you select behaviors. Change the mouse to “Pick Area”, and coordinates of the corner of the box you draw with the left or middle buttons will appear in the left panel and in your terminal window.

Click on “Action --> Navigation” and choose `datalistp.mb-1` to display the ship tracks on the map. Change the mouse to “Pick Nav File”, select a stretch of the nav track, and observe that by choosing “Action” in the upper left you can open the selected swath data files in some of the tools that will be introduced later in this tutorial.

1.6. Plotting your grid (`mbm_grdplot`)

Once our grid is created, we can now make a plot of it. For this example, use the “`mbm_grdplot`” command:

```
% mbm_grdplot -G2 -I Meteor_M86-1C_75m_bathy_F5.grd
```

Upon completion, execute the “.cmd” file that was created to generate the final PostScript file:

```
% bash Meteor_M86-1C_75m_bathy_F5.grd.cmd
```

Note that this map was created from data that have not yet been fully processed. After some corrections and editing, as discussed in Chapter 2, a significantly more accurate map will be generated.

Note: Please be aware that plots are generated in PostScript format. Although this format is still supported by Linux, Mac users are no longer able to open these files using Preview. The following options are available: set “gv” as the default PostScript viewer (using the command “`mbdefaults -Dgv`”), or convert these files to the PDF format using the command “`psconvert`” or to JPG format (at 300 dpi) with this command:

```
% gmt psconvert -Tj -E300 -A *.ps
```

For more sophisticated plots, we recommend incorporating the following options (details are in the man page for “`mbm_grdplot`”):

-R *west/east/south/north*: As above, to further restrict the plotted area. Suggestion: try to increase or decrease the area from one of our previous plots using this.

-G *color_mode*: For synthetic illumination (also known as sun illumination or hillshade), use -G2. For shading by slope magnitude, use -G5.

-MGQ *dpi*: Set the resolution in dots per inch (dpi). The default is 100, but for publications, 300 is typically recommended. Higher dpi values result in larger PostScript plot files.

-MNA: Enumerate each multibeam file name adjacent to the navigation track. Subsequently, append a “p” to plot the names perpendicular to the navigation track.

-O *name*: Specify the name for the output script that generates the map. (Don't forget to run this script!)

-T: Plot the coastline (if the survey is conducted at a scale that allows the coast to be visible within the map area).

-MTG: Coastline plotting options. Append a shade, color, pattern, or c for clipping. Specify the shade (0-255) or color (red/green/blue).

-U: Select the desired orientation: portrait (1) or landscape (2).

-W *color_style/palette*: The color style “1” (default) represents a continuous gradient of colors, while “2” denotes a set of distinct color intervals. Palette “1” (default) comprises Haxby colors, palette “2” comprises high-intensity colors, palette “3” comprises low-intensity colors, palette “4” comprises grayscale, and palette “5” comprises uniform grayscale. For instance, the command “-W1/2” transforms the color to high intensity, and “-W1/4” renders it in grayscale.

-Z *min/max*: Assign minimum and maximum values to the bathymetry data to influence the color palette. Please note that since the data represents topography, the values are negative and the deeper value will be assigned first, followed by the shallower value. Add a “-Y” to enforce your min/max range, as by default it will choose a nice, rounded interval between colors, which is likely to extend the range.

At this point, you are welcome to experiment with “`mbm_grdplot`” by incorporating or discarding various options until you achieve the desired outcome. Additionally, please refer to the man page for further exploration of available options.

It is important to note that the “.cmd” files created by these commands are simply text files that combine MB-System and GMT tools. Therefore, by learning GMT commands, you can unlock a wide range of plotting possibilities and extend your MB-System plots to achieve anything GMT can do.

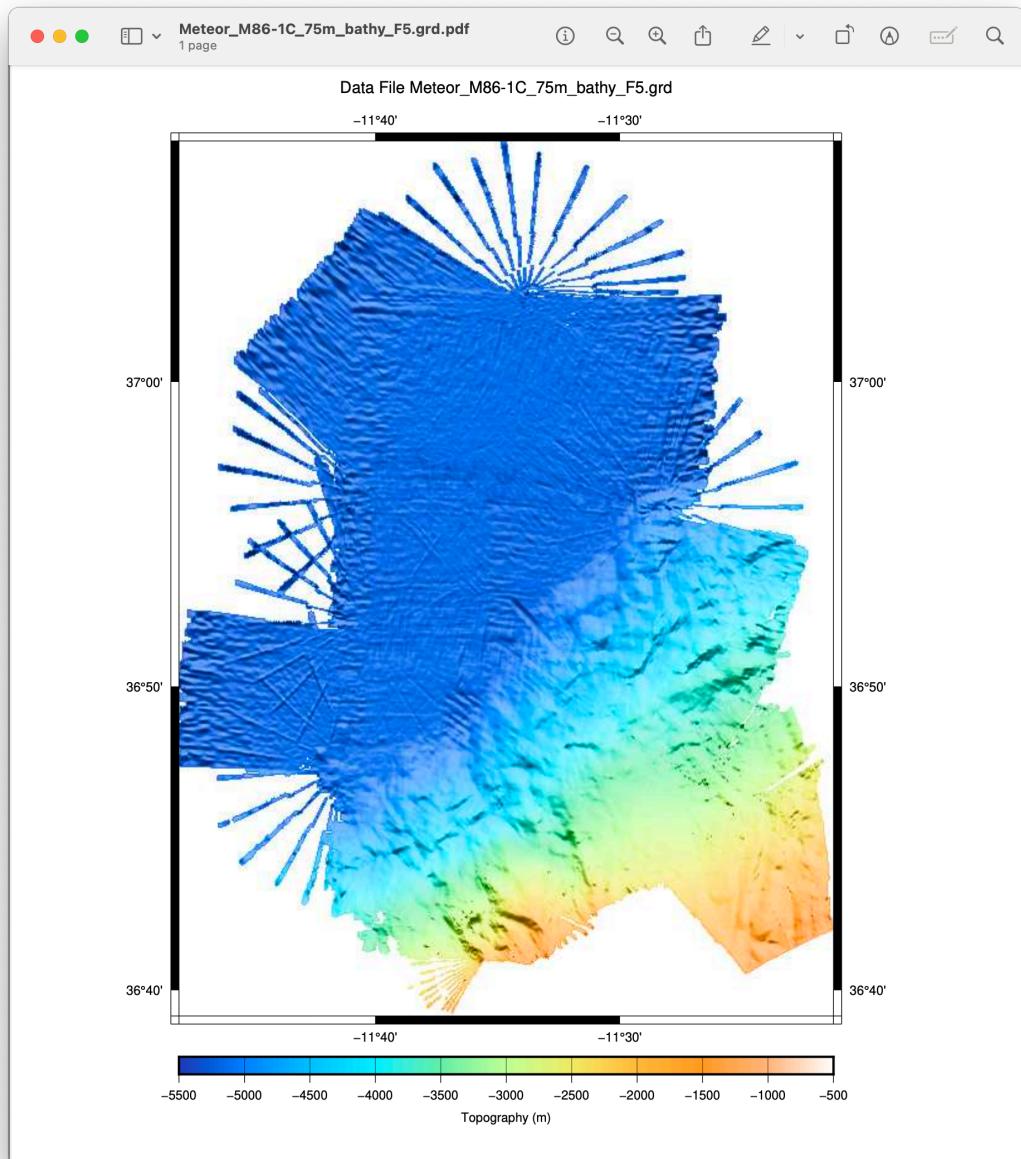


FIGURE 3: GRID PLOT GENERATED USING MBM_GRDPLOT.

Chapter 2 - Data Corrections

2. Corrections

Even the most sophisticated multibeam system is not entirely immune to errors. The environment in which we are surveying can sometimes be highly dynamic (waves, water masses, complex topography, etc.). Therefore, applying corrections is always necessary, even if they are minimal. In this Chapter, we will provide an overview of the most common corrections for multibeam data.

2.1. Amplitude/Sidescan correction (mbbackangle)

Certain sonar brands typically do not apply corrections to the backscatter/sidescan records. Consequently, it is imperative for the user to apply corrections prior to interpreting the characteristics of the surface sediments and morphology.

To accomplish this with MB-System, the command “mbbackangle” can be utilized. A typical example of its usage is as follows:

```
% mbbackangle -A1 -A2 -Q -T Meteor_M86-1C_75m_bathy_F5.grd -V
```

Where:

-A: The algorithm is designed to apply a grazing angle correction to amplitude (-A1) and/or sidescan (-A2) records. The user can specify a reference beam, with a default of a 30-degree (off nadir) angle. Data is recalculated in groups of 2 degrees each.

-Q: This calculation determines the slope across the track within a single ping and applies it to the calculations from option “-A”.

-T grid: If a grid with the morphology from the area is provided (typically processed multibeam data from the same survey), the program calculates slopes along the track and applies them to the calculations from option “-A”.

-V: This is the verbose mode.

Note: MB-System typically calculates the correction based on option “-A” for groups of 50 pings. However, the most effective approach is to apply the overview values calculated for the entire survey. This can be achieved by executing the following two commands to modify the .par files to apply the files generated by mbbackangle:

```
% mbset -PAMPCORRFILe:datalist.mb-1_tot.agr  
% mbset -PSSCORRFILe:datalist.mb-1_tot.sga
```

Then run the command “mbprocess” to merge the preprocessed files with the modified auxiliary files and generate new processed files.

2.2. Sound velocity correction (mbvelocitytool)

Multibeam sonars calculate bathymetry by raytracing through a model of the speed of sound of the ocean. The sound velocity profile (SVP) that is used can dramatically impact the calculation. Typically, a profile should be acquired and applied at the start of the survey. However, if you observe that the outer beams of the multibeam data appear to be curved upward or downward, it is advisable to take a new SVP. Unfortunately, due to time constraints, cruises may only be able to take a new SVP every few hours or daily.

We will construct an example file, “my_SVP.svp”. This file is a straightforward ASCII file with two columns: depth and sound velocity. It can be space or tab-separated. Files generated using SIS (Seafloor Information System) typically work, but users should be mindful of the ASCII encoding, as UNIX systems and MS Windows employ distinct systems for Carriage Return (also known as New Line). To address this issue, users can open the SVP ASCII file in any text editor on a UNIX system, copy and paste the content to a new file, and then save it again.

For this exercise, extract the SVP data from the file “0000” located within our dataset and write the values to a new file, “my_SVP.svp”. (You can refer to it as an example to construct future SVP files using data from XBT or CTD casts). The command is:

```
% mbsvplist -F58 -I 0000_20111219_154640_METEOR_EM122.all > my_SVP.svp
```

Then turn around and specify that this SVP file should be called in all of the .par files for your datalist by executing the command:

```
% mbset -I datalist.mb-1 -PSVPFILE:my_SVP.svp
```

You can utilize the tool, “mbvelocitytool”, to interactively model the impact of a new SVP over the data without actually implementing it. To access this tool, type the command:

```
% mbvelocitytool
```

This command launches a window where you can select “File --> Open Swath Sonar Data”. Choose the first .mb59 file (e.g., 0000...) and press “OK”. You will then see the following window:

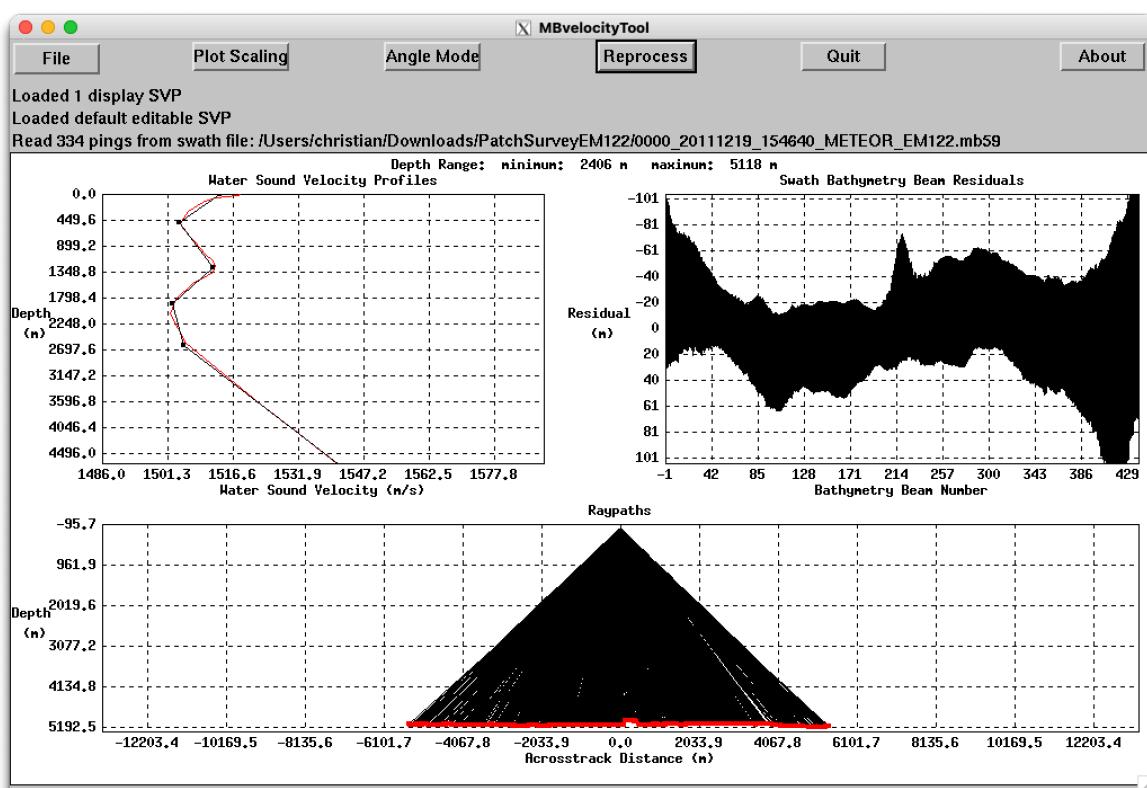


FIGURE 4 – THE USER INTERFACE OF THE MBVELOCITY TOOL ENABLES THE VISUALIZATION OF ONE MULTIBEAM FILE’S RESIDUALS AND RAY PATHS. THE RED LINE REPRESENTS THE PROFILE QUERIED USING THE COMMAND “MBLEVITUS”, WHICH IS AN ANNUAL AVERAGE SVP FOR THE SPECIFIED AREA.

Regrettably, the sample dataset was not acquired over a flat bottom, resulting in the “Swath Bathymetry Beam Residuals” appearing distorted. Therefore, please disregard this aspect and treat the dataset as presented. Subsequently, open your SVP file (accessible through “File --> Open display/editable profile”). If you are able to view the file, it indicates that MB-System has successfully interpreted it.

For further details, consult the man page or access our YouTube video, which explains the fundamental functionalities.

<http://www.youtube.com/user/MBSYSTEM1993>

Notes:

- a) The distinction between “display” and “editable” lies in the fact that “display” is not applied to the data when you select “Reprocess,” whereas the second option is.
- b) Although mbvelocitytool is designed to operate on a single file, once a suitable SVP has been identified for your data, you can apply the same file to an entire dataset or survey using the mbset command with a datalist (above).
- c) Use the “Quit” button rather than closing the windows, as this will prevent the loss of any work done on the respective files.

2.3. Tide corrections (mbotps)

When processing multibeam data from surface ships at depths exceeding 2000 meters, the influence of tides becomes virtually imperceptible due to inherent uncertainties in the measurements. Conversely, for ship operations in shallower depths, the effect of tides can contaminate depth measurements, resulting in artificially shallower or deeper areas depending on the time of the survey in the tide cycle. Note that for datasets originating from AUVs or ROVs, this consideration remains paramount regardless of the depth.

Correcting for tides with MB-System is relatively straightforward if OTPS is installed and `mbotps` is compiled during the installation process. See Appendix E for instructions on how to download and install OTPS for all operating systems. Afterward, type the following:

```
% mbotps -F-1 -I datalist.mb-1 -D60 -M
```

The interval is set by the “-D” flag, which specifies the duration in seconds. Adjust the value accordingly.

Note that, alternatively, tides can be corrected if you have direct measurements from a bottom-mounted pressure sensor.

2.4. Lever arms offsets and biases

Lever arms are normally read from raw files using the “mbm_makeplatform” command (see Ch. 1.2.3). However, new lever arms and biases can be applied. They are extensively explained in the man page with the “`mbset`” command. A concise method to obtain an overview of potential modifications is by accessing a “.par” (parameter file) from any edited file using MB-System tools. The procedure to apply changes to a specific file is to edit the “.par” file directly, while for groups of files, use “`mbset`” with changes to lever and other parameters as demonstrated in Chapter 2.2 for the SVP correction.

Note: For AUVs or formats that require preprocessing (.ALL, KMALL and S7K), one can apply these in the platform file and improve on them with the bias optimization tools in `mbeditviz`.

Chapter 3 - Data Cleaning

3. Cleaning data

Data cleaning is necessary unless you possess the perfect echo sounder and vessel.

Important: in “mbedit”, “mbeditviz”, and other programs (with the exception of “mbfilter”), work on the preprocessed versions of files (e.g., import the `datalist.mb-1`), and choose to apply existing saved edits to the data. Editing the “`p.mb59`” files (or importing `datalistp.mb-1`), has the undesirable result of creating cascades of “`pp.mb59`” files.

By “cleaning” or “editing” data, we are actually simply flagging soundings so that they will not be utilized by commands that work on processed data (the `datalistp.mb-1`). Data are not being deleted.

Many of these tools require a 3-button mouse. Also, always close graphical interface windows with the “Quit” or “Dismiss” buttons so that files locking the data against conflicting processes acting on them will be gracefully closed. Stray lock files can be deleted by typing:

```
% mbdatalist -I datalist.mb-1 --unlock
```

Or navigate to each survey directory and type:

```
% rm *.lck
```

3.1. Manual editing

No data can be completely cleaned using automatic filters (author’s opinion - please feel free to disagree and provide evidence to support your position). Therefore, it is essential to master both of the “mbedit” and “mbeditviz” interactive editors from MB-System.

3.1.1. 2D Editor for multibeam data (mbedit)

To initiate the 2D data editor and load the initial file from our `datalist`, execute the following command at the terminal:

```
% mbedit -I datalist.mb-1
```

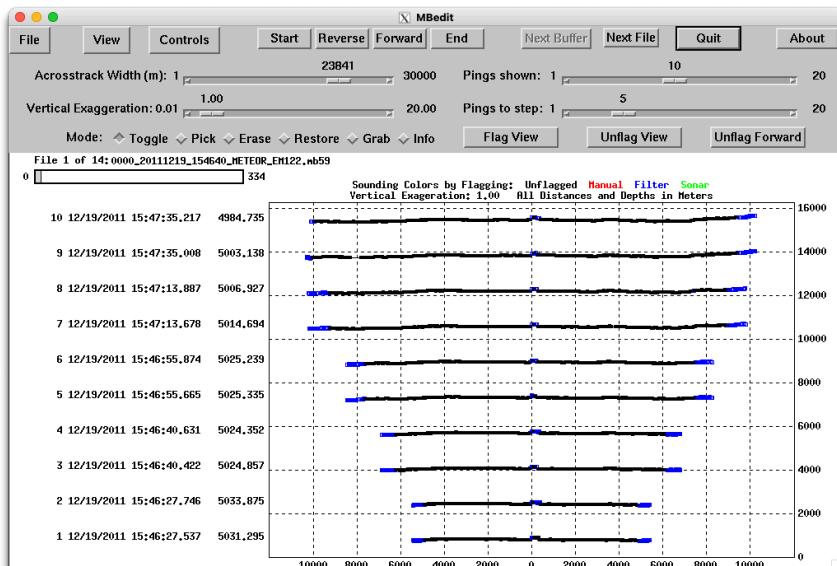


FIGURE 5: A TYPICAL MBEDIT SESSION DISPLAYING SEVERAL PINGS.

You can either open a single file or a datalist; opening a datalist enables you to work through each sequential individual file by pressing the “Next File” button. Be sure to select preprocessed, not processed files. When opening a file that was previously edited, choose to accept previous edits (unless you want to start over). Whenever a file is opened by “mbedit,” this will generate an auxiliary “.esf” file (for “edit save file”), whether or not you actually make any edits.

Soundings can be flagged with the “Erase” and “Grab” modes and unflagged with “Restore” (there is no Undo). This flags - doesn't delete - the soundings to be excluded from use by gridding and other programs acting on processed data. With settings in the View menu, soundings can be viewed in “Waterfall”, “Alongtrack”, and “Acrosstrack” views, and along the left side, time stamps and plots of heading, roll, sonar depth, and several other parameters can display those data for each of the pings.

We recommend referring to the man page for the usage of mbedit. We have created a video that demonstrates its key features:

<http://www.youtube.com/user/MBSysytem1993>

Notes:

- a) mbedit offers a set of keyboard shortcuts that significantly enhances the editing process. The comprehensive list of shortcuts is detailed in the official documentation.
- b) To close any MB-System graphical windows, you must utilize the “Quit” or “Dismiss” button.
- c) Run “mbprocess” on the datalist to apply the edits.

3.1.2. 3D Editor for multibeam data (mbeditviz)

The 3D editor is the most efficient method for achieving a thoroughly cleaned map. Utilizing this editor, you can load a single file or a group of files, visualize the soundings as a 3D cloud of points and interactively flag data as unsatisfactory. Interactions with other sonars, random noise, side lobe interference, and behaviors characteristic of your sonar will become apparent when viewed in the context of an obvious surface of the seafloor. Your assessment of what constitutes good or bad soundings will improve as you gain experience with your data.

To initiate the 3D data editor and load our datalist of preprocessed data, execute the following command:

```
% mbeditviz -I datalist.mb-1
```

Or if just one file is needed, type:

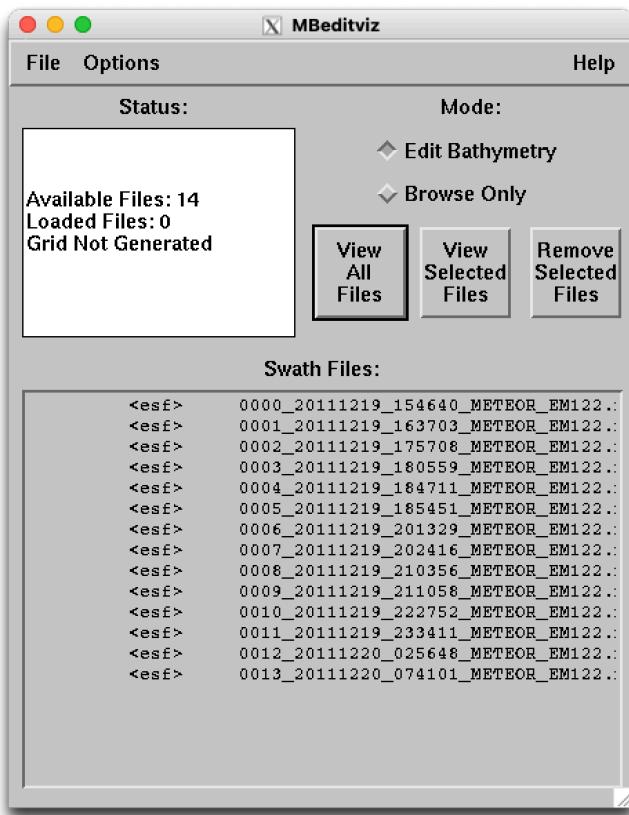
```
%mbeditviz -F59 -I 0000_20111219_154640_METEOR_EM122.mb59
```

Another way to initiate the editor is from “mbgrdviz” (Chapter 1.5): open your grid, select “Action --> Open Navigation”, choose datalistp.mb-1, change your mouse behavior to “Pick Nav File”, use the middle button to select a file, then select “Action --> Open Selected Nav in MBeditviz”.

These methods open the window in Figure 6, presenting the user with three distinct options.

- 1) “View All Files”
- 2) “View Selected Files” (use the shift or control keys to select lines)
- 3) “Remove Selected Files”

Note: An auxiliary file containing the extension “.esf” was created when we opened data files using the “mbedit” editor, even if no edits were made. Making modifications to the data within “mbeditviz” will modify existing .esf files and make new ones if none existed yet.



*FIGURE 6: INITIAL WINDOW OF MBEDITVIZ,
DISPLAYING A LOADED DATALIST.*

When you select files to view, the software will read the auxiliary “.inf” files and determine an appropriate resolution based on the depth and footprint cell size of the shallowest point in the dataset (Figure 7).

The dialog in Figure 7 allows you to choose a custom resolution; it is best to use the same resolution you are targeting for your final grid. The interpolation method used by “mbeditviz” is equivalent to “-F5” for “mbgrid” so the resulting surface will look great (or the best it can). However, because this is a very robust model the process can take some time at high resolution with a large dataset or might even crash. Therefore, if you would like to use a higher resolution consider opening only a few files at a time.

In this instance, we will utilize a 75-meter resolution for our EM122 dataset. Once you have modified the dialog to read 75, click “Apply” to initiate the interpolation process.

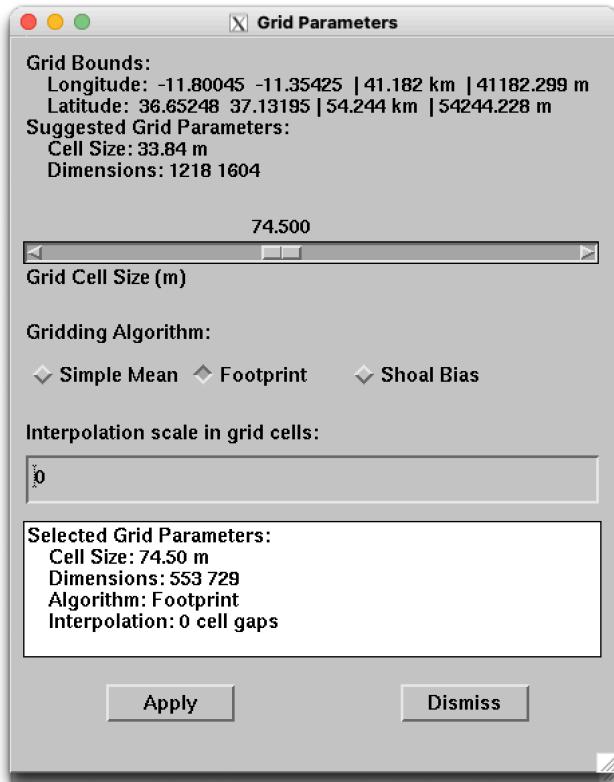


FIGURE 7: SELECTING THE RESOLUTION FOR THE DISPLAY GRID.

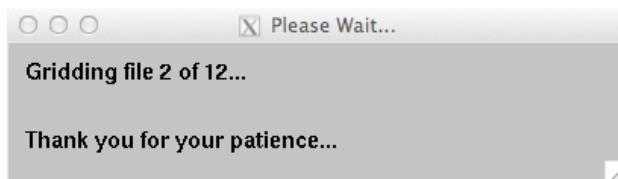


FIGURE 8: GRIDDING THE FILES

Upon completion, the primary window from "mbeditviz" will be opened.

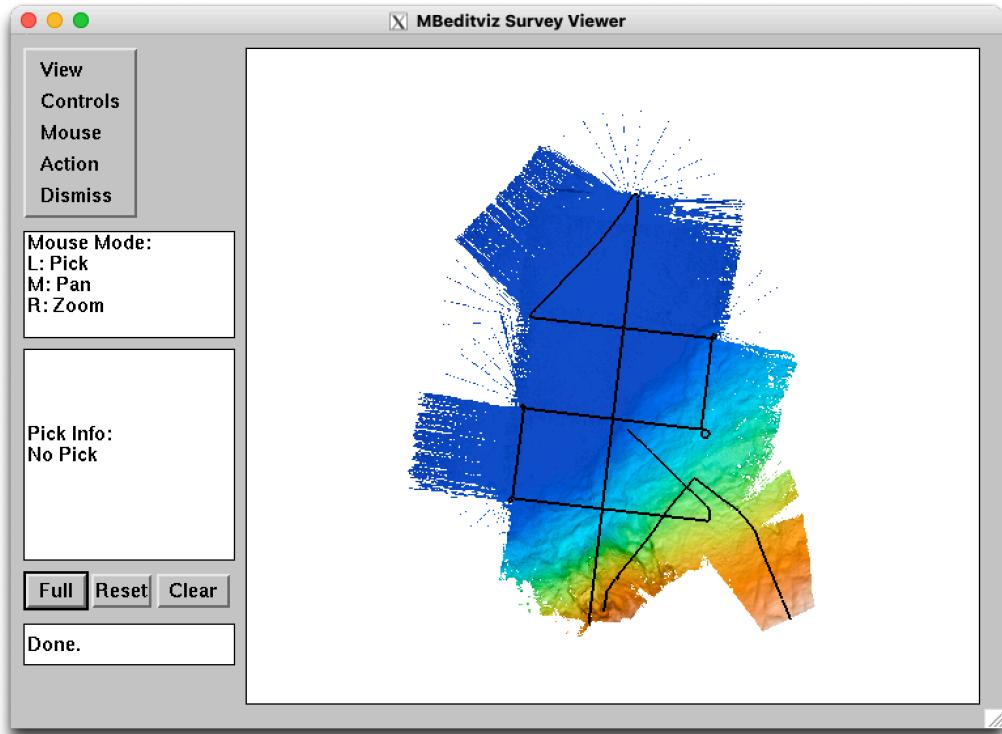


FIGURE 9: THE SURVEY VIEWER WINDOW DISPLAYING OUR MULTIBEAM DATASET AND NAVIGATION DATA.

Enlarge the window to reveal additional buttons in the left side of the window.

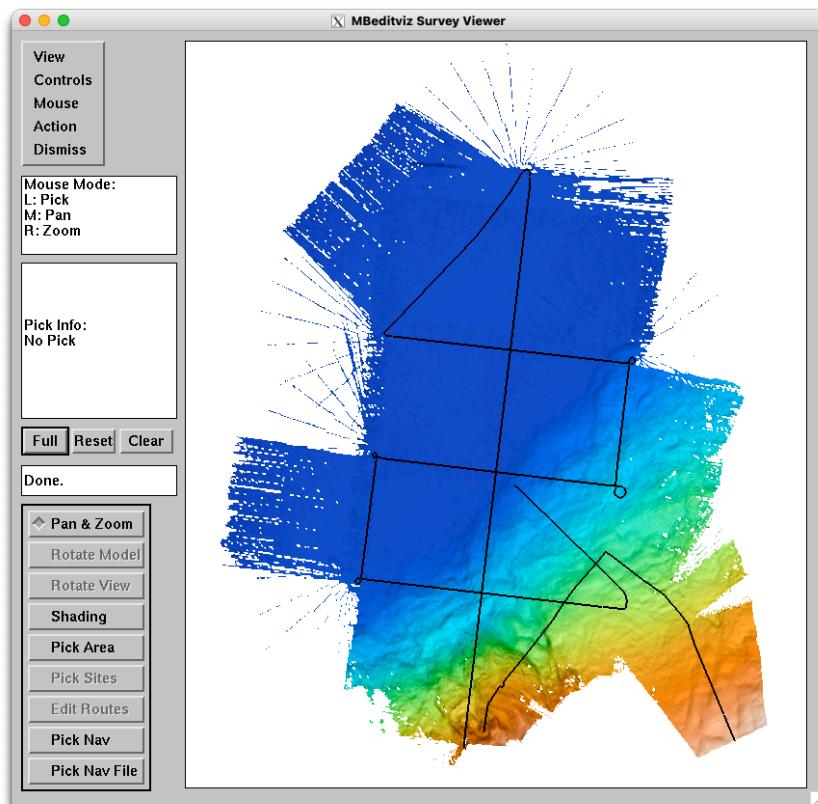


FIGURE 10: WHEN THE WINDOW IS ENLARGED, ALL BUTTONS WILL BE VISIBLE.

****Note:**** Upon selecting a button on the left, a “Mouse Mode” above indicates the functions of the left, middle, and right mouse buttons. For instance, in Figure 10, when the “Pan & Zoom” button is selected, the “Mouse Mode” above reveals that the left button selects an area of interest, the middle button pans, and the right button zooms in and out.

To initiate the data cleaning process, select an area of interest. Begin by clicking the “Pick Area” button located on the left side of the screen. Use either the left or middle mouse button to draw a selection, which will be displayed within a yellow box. This action will open a 3D soundings visualization window, as depicted in Figure 11.

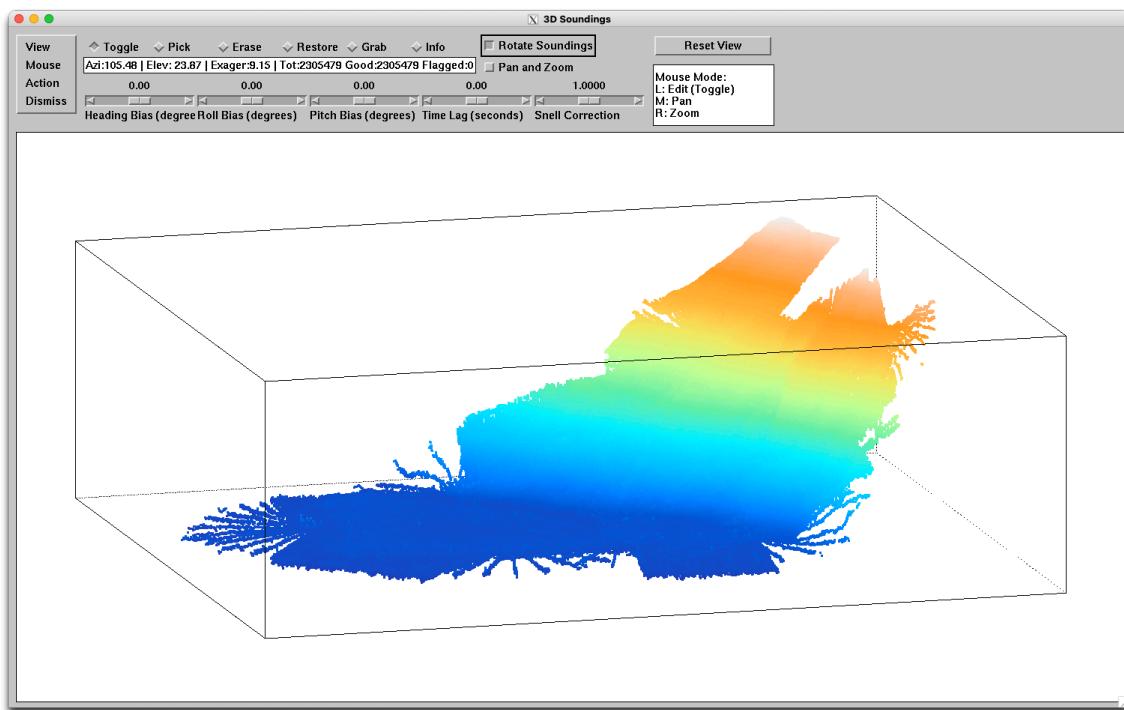


FIGURE 11: THE 3D SOUNDINGS WINDOW. SOUNDINGS COLORED BY TOPOGRAPHY. THE DEFAULT IS BY FLAG STATE (FLAGGED - RED, UNFLAG - BLACK, OR FLAGGED BY SONAR - GREEN).

Enable the “Erase” or “Grab” functions, find some stray soundings and click on or draw a box around them, respectively, with the left mouse button. This flags the soundings to be excluded from use by gridding and other programs acting on processed data. Rotate the view with the middle mouse button to look for more stray soundings. Increase the vertical exaggeration with the right mouse button, turn off “View --> Scale with Flagged Soundings”, and draw a smaller box to get in closer to see more detail. There is no “Undo”, but you can enable the “Restore” function and unflag soundings. Soundings that you flag will appear as red dots, and soundings identified by the sonar as poor quality will appear as green dots (when “View --> Show Flagged Soundings” and “View --> Color by Flag State” are selected). The grid you see in the Survey Viewer window updates on the fly but not perfectly, so to fully update the grid select “Action --> Update Bathymetry Grid”.

To help diagnose where there are problems with the data, in the 3D Soundings window you can change the colors of pings in areas or along nav tracks by clicking on “Action --> Color Unflagged Soundings Blue” (or another color). You can also view the soundings by topography or amplitude. Another way to see where problems are is by changing the view of the grid in the Survey Viewer to “View --> Standard Deviation”. Select the “Info” function to view in the editing window and also in your terminal window the file name, ping number, time, and other metadata for each sounding you select.

Refer to the man page for mbeditviz for details of its operation. We have also created a video that demonstrates additional features:

<http://www.youtube.com/user/MBSYSTEM1993>

Angular offsets (roll, pitch, heading biases, time lag) can be evaluated by viewing an area in the 3D soundings window and using the sliders along the top or with “Action --> Optimize Bias Values”, and applied as described in Chapter 2.4.

Notes:

- a) This tool necessitates a three-button mouse or a mouse that can at least emulate such functionality.
- b) mbeditviz occasionally experiences crashes. Upon such an event, the lock files for any open files will still be present. To resume editing, the lock file must be removed. The most straightforward method for eliminating all lock files is to execute the following command in a terminal window:

```
% mbdatalist --unlock
```

- c) To ensure your work editing the data is not lost and lock files are removed gracefully, close all graphical windows by utilizing the “Quit” or “Dismiss” button.
- d) Run “mbprocess” on the datalist to apply the edits in the newly modified .esf files to update the processed (“p.mb59”) files.

3.1.3. Navigation editing (mbnavedit)

In general, GPS data is consistent and reliable. However, no ship's GPS reception is 100% reliable all the time, and the GPS may transmit erroneous data to the multibeam system. ROV dive tracks might also show deviations due to spurious USBL hits. To clean these problems, you must edit the navigation data. While “mbnavedit” is designed to operate with single files, which you can do by typing:

```
%mbnavedit -F59 -I 0000_20111219_154640_METEOR_EM122.mb59
```

it is possible to navigate to the next file in a datalist by pressing the “Next File” button if you use the following command:

```
% mbnavedit -I datalist.mb-1
```

We recommend referring to the man page for mbnavedit. We have also prepared a video that effectively demonstrates additional features:

<http://www.youtube.com/user/MBSYSTEM1993>

Notes:

- a) Within the “mbnavedit” program, you can access the attitude data. While other software may permit editing of the attitude data, the MB-System philosophy advocates against such modifications. Instead, it is recommended to ensure the proper installation and calibration of the motion sensor. Attempting compensatory measures for malfunctioning motion sensors is not advisable. In such cases, the only viable option is to replace the attitude data with data from a secondary motion sensor (if your ship or AUV is equipped with two motion sensors). For individuals without a second attitude stream, a workaround involves extracting the erroneous attitude data using “mbnavlist”, smoothing it using a third party tool (R, Matlab, Python, etc), and then reapplying it using “mbset” and “mbprocess”.
- b) You can enlarge vertically (only) the “mbnavedit” window to view more information.
- c) Run “mbprocess” on the datalist to apply the edits.
- d) The program “mbnavadjust” will be introduced in the Chapter about working with data from AUVs. With it you adjust the navigation model, for example, to correct for errors that accumulate while a vehicle is submerged and out of GPS satellite contact.

3.2. Editing automatically

Some individuals simply lack the time to manually clean data. For those, MB-System offers two specialized filtering tools: one for multibeam data and another for sidescan data. It is important to note that filters can occasionally delete valuable data. Additionally, filter settings that are effective for one dataset may not necessarily be suitable for another. Consequently, determining the optimal filter settings can sometimes be a time-consuming process.

3.2.1. Multibeam data (mbclean)

The automatic filter for multibeam data within MB-System is named “mbclean”. This tool offers a comprehensive set of options, as detailed in the “man” page. In practice, the mbclean tool is typically employed to remove significant outliers (excluding those within the specified range defined by the option “-Z”) and eliminate certain outer beams. For instance, in this scenario, 20 beams are removed from each side, resulting in a total of 40 beams being eliminated:

```
% mbclean -X20 -D-5500/-1300 -V
```

3.2.2. Sidescan data (mbfilter)

The tool “mbfilter” enables the filtering of sidescan data. The man page provides detailed instructions on its usage, while an article published during the GLORIA project elucidates the filters implemented within the tool. For instance:

```
% mbfilter -Idatalistp.mb-1 -S2/5/3 -V
```

Reference: Sauter, D., and L. Parson, Spatial filtering for speckle reduction, contrast enhancement, and texture analysis of GLORIA images, IEEE J. Ocean. Eng., 19, 563-576, 1994.

Note that the command “mbfilter” with the desired settings should be applied to the processed data (datalistp.mb-1, rather than to datalist.mb-1). Due to the complexity involved, a separate document will be created in the future to discuss sidescan in more detail.

Chapter 4 - Metadata

4.1. Extracting basic information from your data (mbinfo)

Any surveyor requires fundamental information about multibeam data, whether obtained from a single file or a survey. The command for this purpose is “mbinfo”, and a concise data summary (Figure 12) can be readily generated for our survey using this command.

```
% mbinfo -I datalist.mb-1
```

```

Data Totals:
Number of Records: 5734
Bathymetry Data (432 beams):
  Number of Beams: 2477088
  Number of Good Beams: 2305479 93.07%
  Number of Zero Beams: 8424 0.34%
  Number of Flagged Beams: 163185 6.59%
Amplitude Data (432 beams):
  Number of Beams: 2477088
  Number of Good Beams: 2305479 93.07%
  Number of Zero Beams: 8424 0.34%
  Number of Flagged Beams: 163185 6.59%
Sidescan Data (2048 pixels):
  Number of Pixels: 11743232
  Number of Good Pixels: 8874987 75.58%
  Number of Zero Pixels: 0 0.00%
  Number of Flagged Pixels: 2868245 24.42%
Navigation Totals:
Total Time: 16.0633 hours
Total Track Length: 177.1359 km
Average Speed: 11.0273 km/hr ( 5.9607 knots)

Start of Data:
Time: 12 19 2011 15:46:27.537000 JD353 (2011-12-19T15:46:27.537000)
Lon: -11.561664100 Lat: 36.847807750 Depth: 4931.7554 meters
Speed: 17.6040 km/hr ( 9.5157 knots) Heading: 129.2600 degrees
Sonar Depth: 5.9950 m Sonar Altitude: 5025.2998 m

End of Data:
Time: 12 20 2011 07:50:15.517000 JD354 (2011-12-20T07:50:15.517000)
Lon: -11.382564000 Lat: 36.688677300 Depth: 1473.9969 meters
Speed: 13.1400 km/hr ( 7.1027 knots) Heading: 152.9100 degrees
Sonar Depth: 6.0650 m Sonar Altitude: 1468.7335 m

Limits:
Minimum Longitude: -11.800451506 Maximum Longitude: -11.354249929
Minimum Latitude: 36.652482624 Maximum Latitude: 37.131946147
Minimum Sonar Depth: 4.2323 Maximum Sonar Depth: 7.4170
Minimum Altitude: 1468.7335 Maximum Altitude: 5138.4932
Minimum Depth: 1350.4994 Maximum Depth: 5552.4209
Minimum Amplitude: -59.6600 Maximum Amplitude: 0.4000
Minimum Sidescan: -97.8000 Maximum Sidescan: 3.0500
christian@Mac PatchSurveyEM122 %

```

FIGURE 12: SUMMARY OF OUR RAW TEST DATASET, SHOWING ONLY THE CONCLUSION OF THE EXTRACTION PROCESS.

The command “mbinfo” offers a range of options, including:

- N: the capability to display notifications from the data
- C: outputs all metadata, which is particularly useful for processed files generated by MB-System, as it contains all corrections performed or applied.
- X: By default, “mbinfo” reports information in a text format designed for easy reading. However, the “-X” option enables the selection of an alternative output format, such as JSON or XML.

4.2. Inserting metadata into every single file (mbset)

Another method to record the collection date and location of your dataset is by inserting metadata into the files using the command “mbset”. An example of this command is provided below:

```
% mbset -PMETAVESSEL:Meteor -PMETAINSTITUTION:University_of_Hamburg \
-PMETAPLATFORM:M -PMETASONAR:EM122 -PMETASONARVERSION:SIS_3.7 \
-PMETACRUISEID:M86-1C -PMETACRUISENAME:M86-1C_EM122_EM710_calibration \
-PMETAPI:Christian_Ferreira \
-PMETAPIINSTITUTION:MARUM -PMETACLIENT:MARUM -PMETASVCORRECTED:1 \
-PMETATIDECRECTED:1 -PMETABATHEDITMANUAL:1 -PMETABATHEDITAUTO:0 \
-PMETAROLLBIAS:0.0 -PMETAPITCHBIAS:0 -PMETAHEADINGBIAS:0 -PMETADRAFT:0.00
```

When a single file is queried using the “mbinfo” command, the output appears as follows:

```
% mbinfo -C -F59 -I 0000_20111219_154640_METEOR_EM122p.mb59
```

christian@Mac PatchSurveyEM122 % mbinfo -C -F59 -I 0000_20111219_154640_METEOR_EM122p.mb59

Swath Data File: 0000_20111219_154640_METEOR_EM122p.mb59
MBIO Data Format ID: 59
Format name: MBF_EM710MBA
Informal Description: Kongsberg current multibeam processing format
Attributes: Kongsberg EM122, EM302, EM710,
bathymetry, amplitude, and sidescan,
up to 400 beams, variable pixels, binary, MBARI.

Metadata:

Vessel: Meteor
Institution: University_of_Hamburg
Platform: M
Sonar: EM122
Sonar Version: SIS_3.7
Cruise ID: M86-1C
Cruise Name: M86-1C_EM122_EM710_calibration
PI: Christian_Ferreira
PI Institution: MARUM
Client: MARUM
Corrected Depths: YES
Tide Corrected: YES
Depths Manually Edited: YES
Depths Auto-Edited: NO
Roll Bias: 0.000000 degrees
Pitch Bias: 0.000000 degrees
Heading Bias: 0.000000 degrees
Draft: 0.000000 m

Comments in file 0000_20111219_154640_METEOR_EM122p.mb59:

Swath data modified by program mbprocess
MB-system Version 5.8.2beta21
Run by user <christian> on cpu <Mac> at <Thu May 8 11:07:26 2025>
Depths and crosstrack distances recalculated from travel times
by raytracing through a water velocity profile specified
by the user. The depths have been saved in units of
corrected meters (the depth values obtained by
raytracing are not adjusted further).

Control Parameters:

MBIO data format: 59
Input file:/Users/christian/Downloads/PatchSurveyEM122/0000_20111219_154640_METEOR_EM122.mb59
Output file:/Users/christian/Downloads/PatchSurveyEM122/0000_20111219_154640_METEOR_EM122p.mb59
Angle mode: angles corrected using Snell's Law
Default SSV: 1518.100000
SSV initial pass: on
SVP file:/Users/christian/Downloads/PatchSurveyEM122/my_SVP.svp
Input water sound velocity profile:
depth (m) velocity (m/s)
0.00 1518.10
2.60 1518.10
16.60 1518.20
[...Whole SVP suppressed to avoid several additional pages...]

4701.30 1539.80
4711.50 1539.90
4722.80 1540.10
5108.50 1547.20
6000.00 1558.20
8500.00 1602.80
12000.00 1669.00

Output bathymetry reference: CORRECTED
Roll bias: OFF
Pitch bias: OFF
Draft: not modified
Heave: not modified
Travel time: not modified
Lever calculation off.
Tide calculation off.

Merged navigation file:/Users/christian/Downloads/PatchSurveyEM122/ 0000_20111219_154640_METEOR_EM122(mb59.nve
Merged navigation format: 9
Heading merge: ON
Speed merge: ON

Draft merge: ON
Attitude merge: ON
Navigation algorithm: linear interpolation
Navigation time shift: 0.000000
Navigation positions not shifted.
Merge adjusted navigation: OFF
Attitude merging: OFF.
Sensordepth merging: OFF.
Heading modify: OFF
Amplitude Corrections:
Amplitude vs grazing angle corrections applied to amplitudes.
Amplitude correction file: /Users/christian/Downloads/PatchSurveyEM122/datalist.mb-1_tot.agm
Amplitude correction by subtraction (dB scale)
AVGA tables allowed to be asymmetric
Reference grazing angle: 30.000000 deg
Amplitude correction uses topography grid
Topography grid file: /Users/christian/Downloads/PatchSurveyEM122/Meteor_M86-1C_75m_bathy_F5.grd m
Amplitude correction uses seafloor slope
Sidescan Corrections:
Amplitude vs grazing angle corrections applied to sidescan.
Sidescan correction file: /Users/christian/Downloads/PatchSurveyEM122/datalist.mb-1_tot.sgm
Sidescan correction by subtraction (dB scale)
AVGA tables allowed to be asymmetric
Reference grazing angle: 30.000000 deg
Sidescan correction uses topography grid
Topography grid file: /Users/christian/Downloads/PatchSurveyEM122/Meteor_M86-1C_75m_bathy_F5.grd m
Sidescan correction uses seafloor slope
Sidescan Recalculation:
Sidescan not recalculated.
Data cutting disabled.
Bathy edit file:/Users/christian/Downloads/PatchSurveyEM122/0000_20111219_154640_METEOR_EM122. mb59.esf

Data Totals:

Number of Records: 334

Bathymetry Data (432 beams):

Number of Beams: 144288

Number of Good Beams: 119638 82.92%

Number of Zero Beams: 571 0.40%

Number of Flagged Beams: 24079 16.69%

Amplitude Data (432 beams):

Number of Beams: 144288

Number of Good Beams: 119638 82.92%

Number of Zero Beams: 571 0.40%

Number of Flagged Beams: 24079 16.69%

Sidescan Data (2048 pixels):

Number of Pixels: 684032

Number of Good Pixels: 504003 73.68%

Number of Zero Pixels: 0 0.00%

Number of Flagged Pixels: 180029 26.32%

Navigation Totals:

Total Time: 0.8327 hours

Total Track Length: 13.5764 km

Average Speed: 16.3038 km/hr (8.8129 knots)

Start of Data:

Time: 12 19 2011 15:46:27.537000 JD353 (2011-12-19T15:46:27.537000)

Lon: -11.561664100 Lat: 36.847807750 Depth: 5027.6503 meters

Speed: 17.6040 km/hr (9.5157 knots) Heading: 129.2600 degrees

Sonar Depth: 5.9950 m Sonar Altitude: 5021.6553 m

End of Data:

Time: 12 19 2011 16:36:25.308000 JD353 (2011-12-19T16:36:25.308000)

Lon: -11.489473400 Lat: 36.771997350 Depth: 3243.1497 meters

Speed: 12.7800 km/hr (6.9081 knots) Heading: 274.7900 degrees

Sonar Depth: 6.4350 m Sonar Altitude: 3230.6101 m

Limits:

Minimum Longitude: -11.635634729 Maximum Longitude: -11.391462541

Minimum Latitude: 36.724204953 Maximum Latitude: 36.918012338

Minimum Sonar Depth: 4.9650 Maximum Sonar Depth: 6.9150

Minimum Altitude: 3171.4417 Maximum Altitude: 5021.6553

Minimum Depth: 2442.5078 Maximum Depth: 5124.6543
 Minimum Amplitude: -58.9000 Maximum Amplitude: -9.2000
 Minimum Sidescan: -94.4300 Maximum Sidescan: 0.4400

Chapter 5. Miscellaneous and advanced topics

5.1. Multiple surveys in a single datalist

In certain work environments, your team may gather data from multiple hydrographic databases (e.g., [NOAA-NCEI](#), [BSH](#), etc.). Alternatively, the data may be divided into multiple folders, resulting in numerous distinct datalists. To consolidate all the relevant data, you simply create a master datalist that references the server paths for all the cruises. For illustration purposes, we present a datalist that originates from a compilation of 10 cruises in Svalbard, and Figure 13 is a plot of the grid made from it.

```
2011_JCR253/EM122/datalist.mb-1 -1 10
2012_HE387/EM710/datalist.mb-1 -1 100
2012_MSM21-4/EM120/datalist.mb-1 -1 1
2013_MSM30/EM120/datalist.mb-1 -1 1
2013_MSM31/EM120/datalist.mb-1 -1 1
2015_HE449/EM710/datalist.mb-1 -1 100
2015_HE450/EM710/datalist.mb-1 -1 100
2016_MSM57/EM122/datalist.mb-1 -1 10
2017_MSM67/EM122/datalist.mb-1 -1 10
2017_MSM68/EM122/datalist.mb-1 -1 10
```

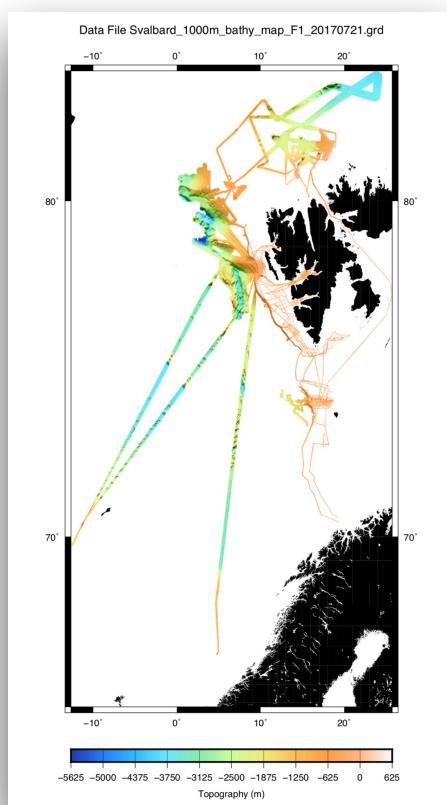


FIGURE 13: SVALBARD COMPILATION - 10 CRUISES.

This datalist has been constructed to reside in the directory above these data sets. You can name it `datalist_Svalbard.mb-1` (so the corresponding “`datalistp.mb-1`” made with the command “`mbdatalist -l datalist_Svalbard.mb-1`” will be “`datalist_Svalbardp.mb-1`”) and you can call it from within any of these survey directories by addressing it (following “`-l`” or “`--input=`”) as “`../../datalist_Svalbardp.mb-1`”.

Have you observed an additional column in this datalist? With MB-System, you can attribute data quality factors for each cruise. These priority rankings are determined by the user. The larger the number, the higher the priority that the datalist will have when the grid is calculated. Please note that for further information, refer to item “6” at Chapter 5.3.

****Note:**** There is no predetermined minimum or maximum value for the classification. The user determines the appropriate ranking criteria. Therefore, it is crucial to carefully consider the factors that should be used to categorize cruises.

Avoid calling the same datalist multiple times or through multiple levels of the same dataset. This practice may inadvertently create an infinite loop. Our recommendation is to organize your data storage into subfolders by categories as follows:

- a) Upper-level: Region (e.g., ATLANTIC or NORTH_ATLANTIC);
- b) Second-level: Year, ship name, and/or cruise number within that area (e.g., 2018_METEOR_M149);
- c) Sounder model: You may have multiple sounder models with different qualities and/or resolutions (e.g., EM122 or etc.);
- d) Lowest-level: Survey per area or day (e.g., AREA1 or DAY01).

5.2 MB-System best practices

These are the lessons acquired over several years through the use of MB-System.

- Combine your commands into a processing script that you can execute and go do something else.
- The number of times you type the option “-V” determines the verbosity of the program’s output. Omitting it will result in few messages, while using it 1 time will output essential messages. Typing it 2 or more times enables debugging modes for each tool within MB-System, with a maximum of 5 times.
- Refrain from utilizing or processing your data on volumes using the Windows file system (e.g., FAT32 or NTFS). These systems occasionally encounter issues related to permissions and file locking. Any file system from Linux or macOS is significantly superior.
- NEVER use folders or names that contain spaces, as this practice will invariably lead to issues. Underscores are acceptable (and personally preferable).
- MB-System has a large user base. Kindly subscribe to the User’s Discussion List (mailing list) and post your inquiries. Direct communication with Dave Caress, Dale Chayes, or myself should be avoided. Links to subscribe and read the archives are here: <https://www.mbari.org/technology/mb-system/discussion/>
- Before posting questions, please Read The “Friendly” Manual (RTFM). Your efforts in this regard are greatly appreciated.
- Maintain an open mind and explore beyond the commands you have learned. While we are focusing here on fundamental or essential commands, it is beneficial to venture into other options.
- Learn also GMT (Generic Mapping Tools, <https://www.generic-mapping-tools.org/>). This software and MB-System interact harmoniously, and GMT can significantly expand your toolset when processing final grids. In fact, all MB-System scripts can be enhanced by understanding GMT commands and syntax. Documentation for GMT is at:
<https://docs.generic-mapping-tools.org/6.5/gmt.html>

- When you are unfamiliar with a data format, you can compare the file extensions with the corresponding list at:
http://www3.mbari.org/products/mbsystem/html/mbsystem_formats.html

If your data are not well documented, consider developing and implementing a comprehensive metadata system that effectively describes the files. This will not only enhance the accessibility of the files for future generations but also ensure their long-term preservation and usability.

- The mouse resolution significantly impacts your cleaning performance when utilizing tools such as “mbeditviz”. Typically, a laser mouse is preferred over a regular optical mouse, and a wired over a wireless mouse, due to substantially higher accuracy and responsiveness.

5.3. More advanced MB-System

These are more advanced terminal commands that may assist you or save you time:

- 1) Construct a recursive datalist. This program must be executed from the root directory and will recursively search through all subdirectories to locate all datalists and write them to a file:

```
% find . -name 'datalist.mb-1' > datalist.mb-1
```

Please note that the file created must still be edited to remove the “./” from the beginning of each path. The (normally the last one) line with “datalist.mb-1” must also be removed, as should any datalist that refers to other datalists, otherwise this datalist will become an infinite loop (since it references itself).

- 2) Re-export compressed grid (see also Appendix E, item “d”) disabling the compression:

```
% gmt grdconvert compressed_grid.grd uncompressed_grid.grd --IO_NC4_CHUNK_SIZE=classic
```

- 3) Recursively delete lock files. When mbeditviz or other tools crash, they may leave behind “.lck” files. If a recursive datalist was being used, these lock files will be left in multiple levels. To fix that this command deletes all lock files from the main folder and its subfolders.

```
% mbdatalist --unlock
```

- 4) Utilize the command history feature of your terminal to retrieve a previous command or search for a specific keyword:

```
% history | grep -i "command or keyword"
```

- 5) Smooth Grids. If the gridding resolution is set too high or the data are not clean of artifacts that will be visible on the map, smoothing can be applied to try improve the map. To smooth a grid, use GMT’s command grdfilter.

```
% gmt grdfilter my_10m_bathyGrid_.grd -Gmy_10m_bathGrid_smoothed50m.grd -D4 -Fm0.05 -V
```

Option “-F” sets the resolution, expressed in kilometers. Refer to the “grdfilter” man page for further details.

- 6) The two recommended approaches for weighting datalists (as presented in Chapter 5) are:

One approach is to assign weights to each cruise or survey in the top level’s datalist, where you likely have multiple cruises (or surveys from the same cruise). Assign each cruise or survey a weight, and the value will determine the extent to which that data will influence the interpolation. Assign a weight of 1 (or less) to a poor cruise or survey, and a weight of 100 (or bigger) to a good one. When grid cells of both a good and a bad cruise and/or survey overlap, the data from the bad cruise or survey will vanish from our grids.

The second option is to assign a weight to each file listed in a datalist. This allows you to adjust the weighting for each individual file. However, re-weighting the files using this approach is significantly more time-consuming.

Please note that the weighting mechanism does not function effectively in multiple levels. It is advisable to set the values only once, as “mbgrid” will default to using the value from the lowest level and closest to the files. Please refer to the example below for clarification.

If, in the master datalist, a value is assigned to a cruise as,

```
cruise/datalist.mb-1 -1 100
```

but within a cruise's datalist, we have a weight value for each survey, these will override the weight assigned to the cruise. Similarly, if I were to assign weight values to my surveys and then assign weights to each one of the files (which would then override the survey values), the same problem would occur.

Our recommendation is to assign weight to “units” where your files are located in a similar area, time period, or oceanographic condition (i.e., per cruise, per survey, per day, per area, etc.).

7) Adding a scale bar when using “mbm_grdplot”. A few examples of scale bars are:

Lower center-left, within map, lat 24°N, 5 km length:

```
-MGLfx5/1/24/5.0_l"km"
```

Below map, centered horizontally, lat 36.4°N, 25 m length:

```
-MGLx8/-2.5+f+c36.4+w25e+l"meters"
```

Half inch above bottom axis, 3" into plot from left, lat 46°N, length 50m

```
-MGLx3/0.5+f+c46+w50e+l"meters"
```

** Note **: don't forget to edit the latitude value to scale the bar to the Earth correctly! Refer to GMT documentation for the specifics.

8) Production mode scripts included in the documentation repository on GitHub

These scripts cover a real-world scenario where a more advanced user of MB-System uses it on a real-world production environment offshore or onshore. Here, most of the tasks are nearly fully automated. They are separated into 3 scripts/steps. The first script is for preprocessing and/or data preparation, and where we have a script tailored for each individual format. It's currently covering the .ALL, .KMALL, and .S7K formats with more to come in the near future. The second script is for post-processing the data, and where manual editing is called (either using mbedit or mbeditviz) and done, and where many corrections are applied automatically and final processed files are made. The last and third is just about gridding the data (using mbgrid and mbmosaic), plotting it, and creating additional data products (plots and geotiffs) from the grid files. The second and third scripts are universal since the data used by them has been either preprocessed or translated.

5.4. Common MB-System workflow for Kongsberg data

See figure 14. This workflow is also published in GitHub.

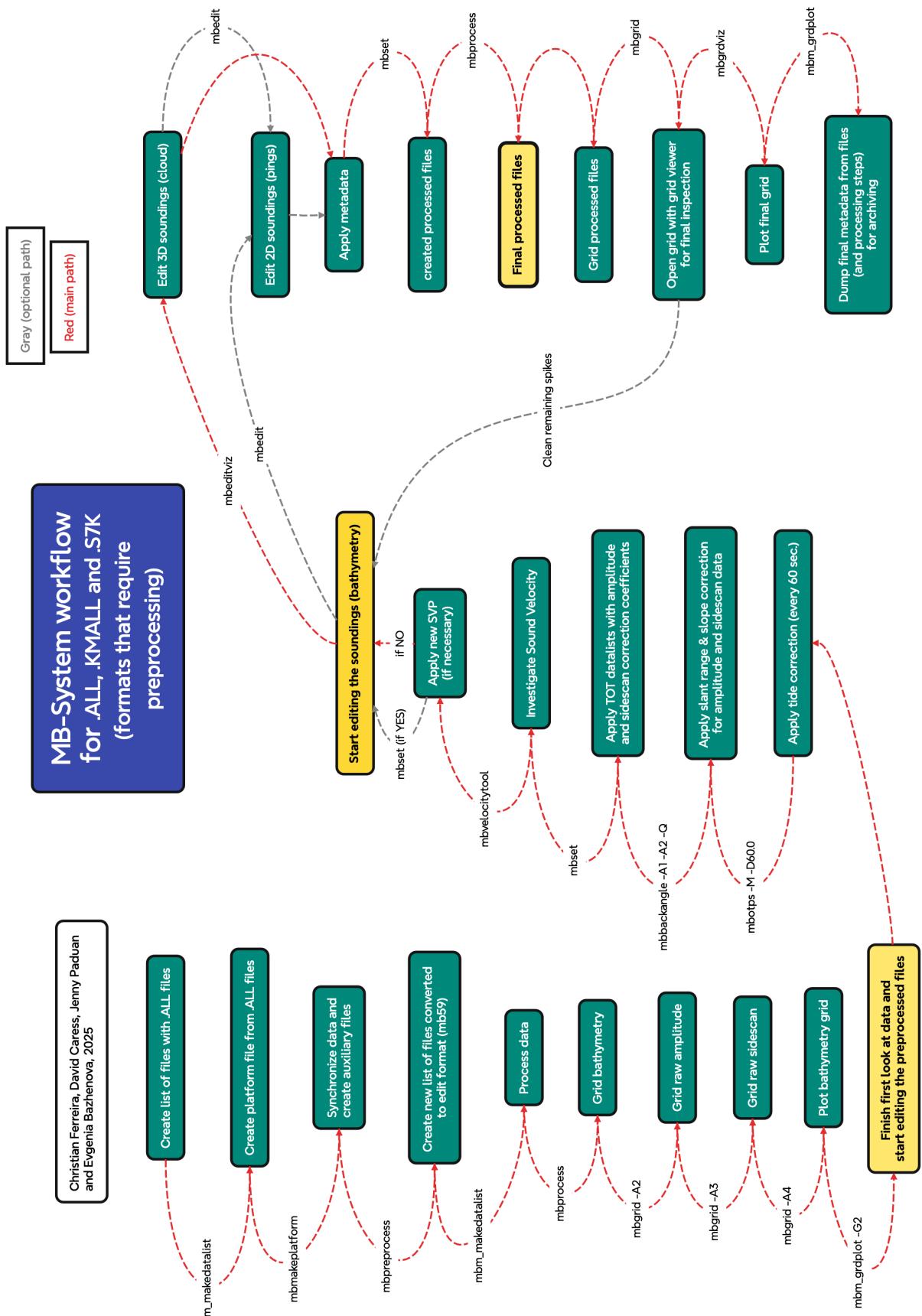


FIGURE 14 - COMMON MB-SYSTEM WORKFLOW FOR SHIP DATA.

Chapter 6 - AUV data processing

6.1. Main differences between the workflow of ship and AUV data

While the majority of MB-System users process data originating from surface vessels, there exists a substantial segment that works with submersible platforms such as autonomous underwater vehicles (AUVs) and remotely operated vehicles (ROVs). Both groups exhibit a notable degree of similarity within their workflows, to the extent that the majority of post-processing steps are indistinguishable. Nevertheless, there are certain key distinctions that warrant our attention, which we shall explore in detail.

6.1.1. Preprocessing

The preprocessing steps are largely identical, with only one notable modification: the depth sensor data. While the MBES measurements provide the distance between an AUV and the seabed, the depth sensor data is crucial for determining the distance from the surface. This data is of paramount importance, albeit it may occasionally contain inaccuracies or noise. To mitigate these issues, depth sensor data smoothing/filtering techniques are usually deployed.

6.1.2. Navigation adjustment with mbnavadjust

“MBnavadjust” is an interactive graphical program used to adjust swath data navigation by matching bathymetric features in overlapping and crossing swaths. The primary purpose of mbnavadjust is to eliminate relative navigational errors in swath data obtained from poorly navigated sonars. Submerged platforms such as towed vehicles, remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) frequently depend on ultra-short baseline (USBL) or inertial navigation systems (INS) for navigation, so are not navigated with accuracy equivalent to the lateral resolution of swath bathymetry obtained with high-frequency sonars operated close to the seafloor. Many old, pre-GPS surveys with hull-mounted sonars also suffer from poor navigation relative to the resolution of the swath bathymetry, and can thus be improved with the application of this tool. “MBnavadjust” also works well to co-register surveys at different scales. For instance, our initial use of this tool involved co-registering a deep-towed Reson 8101 multibeam survey of the Loihi Seamount summit (offshore Hawaii) with a hull-mounted Kongsberg EM300 multibeam survey.

When swath surveys are poorly navigated, features clearly imaged by the sonar may not match in overlapping and crossing swaths. However, by systematically determining the positional offsets between matching features (making “ties”) throughout a survey, it is possible to solve, or “invert”, for a navigation model that is consistent with the original navigational constraints, satisfies the requirement of reasonableness, and causes features in overlapping swaths to match within the sonar’s resolution. The analysis and adjustment can be limited to lateral (x and y or longitude and latitude) offsets, such as for when all vertical offsets have been corrected for a survey prior to import into “MBnavadjust” (e.g. tidal corrections have been made). Between surveys, the offsets likely need to be three-dimensional (x, y, and z).

“MBnavadjust” should be used after the initial navigation processing (e.g. “mbnavedit”) and cleaning (e.g., “mbeditviz”) is complete and “mbprocess” has been run. Import the preprocessed datalist into “mbnavadjust”, and the data are taken from processed files when those files exist (and are not locked), and if no processed file exists, the raw data file is used.

Begin by analyzing true crossings: “List --> Show True Crossings”, click on the first crossing. In the analysis window that opens, zoom in on the overlap area, select “Minimum XY Misfit”, then “Add Tie” when it looks like contours are matching. Misfit visualizations and offset statistics on the left of this window, and other tools accessible from the main project window, can guide your assessments (e.g., “View --> Model Plot” graphically shows the offsets in X, Y, and Z; or “List --> Show Crossing Ties Sorted” lists the tie statistics, where the final columns show how poorly the model fits the ties). Invert to update the working navigation model with your ties (“Action --> Invert Navigation”). View a grid within the project to see where your ties are being made and how well they are helping (invert, “Action --> Update Grids” and “View --> Visualize Survey”). Proceed to 50% and 25% crossings.

Often with AUV data it is necessary to shift the entire survey toward a better-navigated regional grid (e.g., of GPS navigated ship data): “File --> Import Reference Grid”, “List --> Data File Sections”, and in the analysis window, zoom in and select “Minimum Misfit” multiple times, and when they overlap closely, “Add Tie”, which makes a “global tie”. A thorough series of these will shift your survey to match the regional grid.

When the survey has been well tied, ties with impossibly large offsets have been deleted, and you are satisfied with navigation solution, invert once again and select “Action --> Apply Adjusted Navigation”. This applies the navigation model to the survey data (until now, changes have only been made within the mbnavadjust project). New navigation files are generated (“.na#” files) which supersede the “.nve” files generated by “mbnavedit”. Finally, run “mbprocess” to update the processed files with the new navigation model.

Once a swath file is processed with “mbnavadjust”, any further attempts to process the navigation with “mbnavedit” will only produce navigation files that are ignored by “mbprocess”. This behavior can be reversed manually using “mbset”, but then the user will unnecessarily complicate his or her efforts.

A companion program “mbnavadjustmerge” can merge “mbnavadjust” projects together or modify “mbnavadjust” project settings.

Users are advised that “mbnavadjust” is complicated in both conception and implementation. We strongly recommend that users read all of the documentation provided on the man page prior to attempting a first use of this software. A future tutorial specific to “mbnavadjust” is being prepared.

We have also prepared a video that effectively demonstrates visually these features:

<http://www.youtube.com/user/MBSYSTEM1993>

6.1.3. Common MB-System workflow for AUV data

See figure 15. This workflow is also published in GitHub.

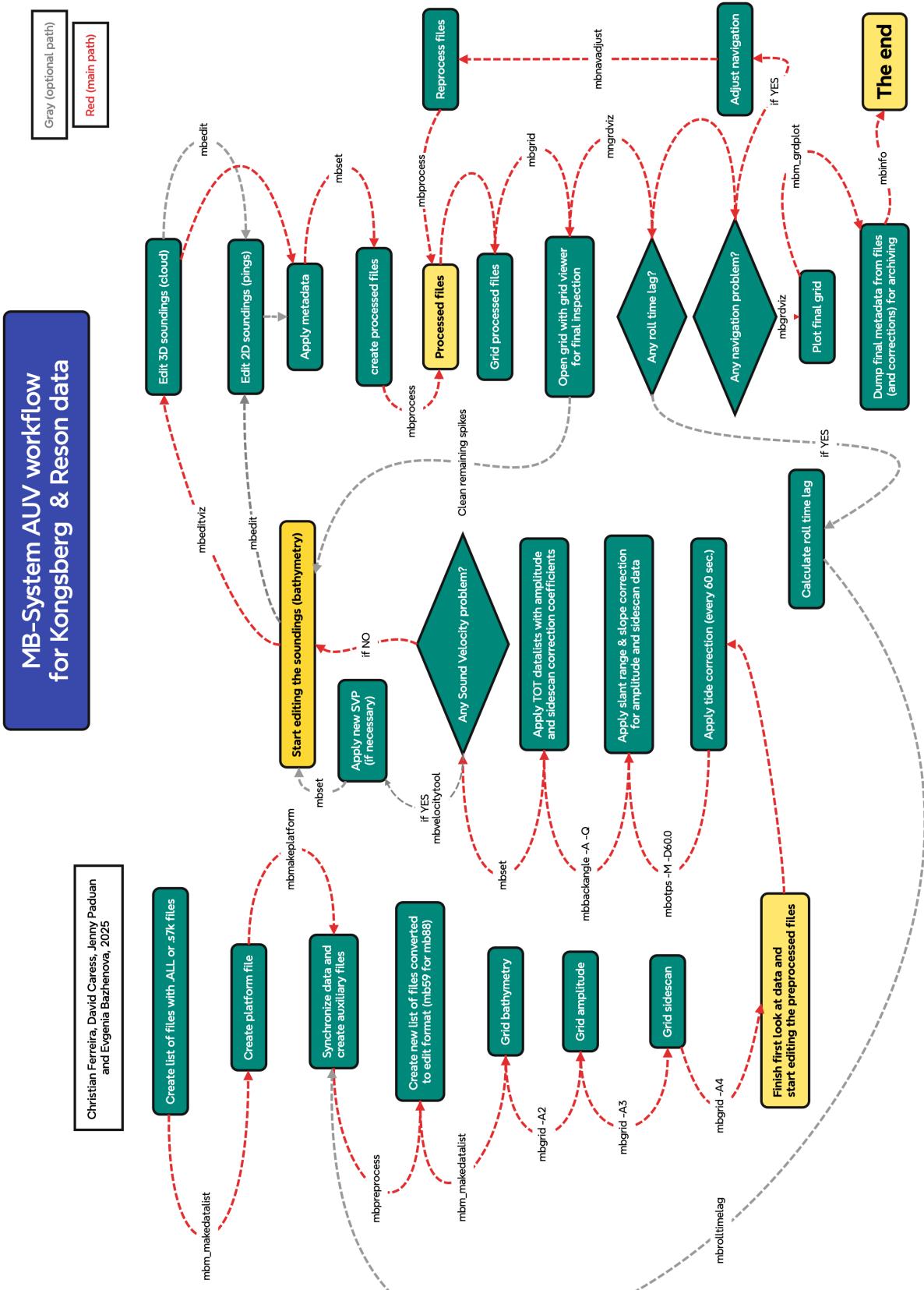


FIGURE 15 - COMMON MB-SYSTEM WORKFLOW FOR AUV/ROV DATA.

Chapter 7 - Dealing with other common formats in preprocessing

7.1. Kongsberg KMALL & ALL

The latest multibeam echosounders (EM124, EM712, EM304, etc.) from Kongsberg have now adopted the latest evolution of format ALL, which is now known as KMALL. The KMALL format enables the addition of new data fields to datagrams without compromising compatibility with existing decoders. Additionally, the design facilitates backwards compatibility when future versions of specific datagrams are released. Consequently, the datagrams in KMALL must be decoded in a specific manner to enable the aforementioned features.

In light of the commands presented in Chapter 1, the two primary distinctions between the ALL format and KMALL lie in the following commands:

```
% mbm_makedatalist -S.kmall -Odatalist_raw.mb-1
```

In this instance, one must to select the ID format for KMALL (261) instead of ALL (58).

```
% mbm_makedatalist -S.mb261 -P -V
```

The remainder of this tutorial is identical for the KMALL format.

7.2. Teledyne Reson S7K

The most recent multibeam echosounders from Teledyne have incorporated the latest iteration of the S7K series 3.x format, an advancement over the original S7K. This development was undertaken to support the integration of several multibeam echosounder devices that have recently been added to Teledyne's portfolio, including the Hydroweep 3rd generation (HS3) devices. It is noteworthy that other multibeam echosounder vendors, such as Norbit, have also adopted S7K as their default logging format.

In light of the commands presented in Chapter 1, the two primary distinctions between the ALL format and S7K lie in the following commands:

```
% mbm_makedatalist -S.s7k -Odatalist_raw.mb-1
```

In this instance, one must to select the ID format for S7K (89) instead of ALL (58).

```
% mbm_makedatalist -S.mb89 -P -V
```

The remainder of this tutorial is identical for the S7K format.

It is worth mentioning two points:

- a) Data from S7K series 2.x or older are ID 88 (.mb88), and S7K series 3.x are ID 89 (.mb89) when (pre)processed;
- b) S7K data exported by QINSY is already preprocessed and its logged data already synchronized. Users are advised to skip the steps where a dataset containing raw files is created and subsequently preprocessed. It is only necessary to create a dataset with the S7K files using ID 89 and create the auxiliary files as outlined in this tutorial.

7.3. Other formats

Many older data formats do not necessitate preprocessing, as they lacked support for concurrent asynchronous datagrams. Consequently, preprocessing is either unnecessary or even unfeasible. In many instances, the workflow for these formats involves simply creating a datalist containing these files, generating the auxiliary data files, and subsequently working with the data.

Nevertheless, there are certain instances where a data translation is necessitated from the original format to the MB-System generic format (ID 71, mb71). An example of that is format SURF (mb181), and the tool used for the translation is "mbcopy".

Data sets downloaded from the National Centers for Environmental Information (NCEI), formerly the National Geophysical Data Center(NGDC), are already converted to their equivalent MB-System data format.

<https://www.ncei.noaa.gov/maps/bathymetry/>

Appendices

Appendix A – MB-System’s History

MB-System was originally developed at the Lamont-Doherty Earth Observatory of Columbia University (L-DEO) and is now a collaborative effort between the Monterey Bay Aquarium Research Institute (MBARI), University of New Hampshire and MARUM at University of Bremen. The U.S. National Science Foundation (NSF) provided the initial support for MB-System development in 1993, and continued support through a series of grants extending through 2018. The Packard Foundation has provided significant support through MBARI since 1998. Additional support has derived from SeaBeam Instruments (1994-1997), NOAA (2002-2004), German National Research Data Infrastructure - NFDI (2022-2024) and others.

MB-System’s team has been since the origin maintained by Dave Caress (MBARI) and Dale Chayes (LDEO, now at CCOM-UNH). In the 2010’s the team was joined by Christian Ferreira (MARUM) and Jenny Paduan (MBARI), and more recently by Tom O'Reilly and Kent Headley (both from MBARI). In the last years it has received contributions made by Krystle Anderson (MBARI), Joaquim Luis (GMT Dev Team and University of Algarve), Val Schmidt (CCOM-UNH), Gordon Keith (CSIRO), Kurt Schwehr (Google), Bob Covill (Tekmap Consulting), Hamish Bowman (University of Otago), Andrew Bell (Hobu Inc.), Jonathan Beaudoin (CCOM-UNH), Ammar Aljuhne (University of Oldenburg), David Finlayson (USGS), Suzanne O'Hara (LDEO), Frank Delahoyde (Scripps Institution of Oceanography), Jens Renken (MARUM), Simon Dreutter (AWI), Monica Schwehr (MBARI), Vikram Unnithan (Constructor University), Johannes Schauer Marin Rodrigues, and others.

Appendix B – MB-System structure and auxiliary files

B1. MB-System datalists

In contrast to most other packages, MB-System does not have a “project” file. Additionally, it does not necessitate loading or opening all data files prior to commencing work with them, or converting disparate sonar data formats to a single common format. MB-System employs a file named “datalist.mb-1” that stores all file names and their sonar data formats (here represented by numbers with two or three digits). All tools within the MB-System package utilize a file named like “datalist.mb-1” to read and process these files. An accompanying file named “datalistp.mb-1” instructs tools to use processed files if they exist.

The datalist is designed to have no limitations on the number of files it can contain. As discussed in Chapter 5.1, any number of custom datalists can be created. Additionally, files of various formats can be seamlessly mixed within a single list. Furthermore, a “master” datalist can be created to establish connections between multiple datalists stored within the same storage location or remotely accessible network drives. This capability enables the creation of a comprehensive link to a nearly infinite number of files logged by different sonars on many cruises over the years. These features empower users to compile files and cruises in a manner akin to a database, facilitating the management of substantial quantities of data through diverse methods such as filtering, extracting metadata, and gridding, among others.

B2. Vendor format, editing format and processed files

As introduced in Chapters 1.2.3 and 1.3, MB-System seeks to preserve the integrity of data files while operating on them. Before further processing, many formats necessitate a data translation from the “raw” vendor format to an enhanced state that prepares the vendor format for processing but otherwise preserves the original data. This conversion is necessary because, typically, the original format lacks the ability to accommodate user edits (flagged data), additional metadata/comments or contains data from multiple sonars that were logged in asynchronous order. The files in this editing-compatible format are considered “preprocessed” and are called by the input file “datalist.mb-1”. Editing and other corrections will not change these files, as user edits and filters will be stored in “auxiliary files”. These files will be merged with the preprocessed files by using the command “mbprocess”, resulting in “processed” files, which have file names with a “p” appended to the end and are called by the input file “datalistp.mb-1”.

B2.1. Understand the raw/processing format and the auxiliary files

The use of auxiliary files (also known as “ancillary files”) is consistent with MB-System’s philosophy of preserving the integrity of data files.

Auxiliary files introduced in this tutorial store outputs of corrections made to the data (from data cleaning, navigation correction, filters applied, etc.). When mbprocess is run, they are applied to the preprocessed files to output processed files. An advantage to this philosophy is that if edits or other corrections are deemed unsatisfactory, it is straightforward to revert them by simply deleting the auxiliary files that contained those edits.

Other auxiliary files are a subset of information extracted from the original files with mbpreprocess, mbdatalist, and other MB-System programs. Reading these files (such as by editing or gridding programs) is significantly faster than reading the full soundings data.

Here is a detailed examination of some of the auxiliary files and their respective functions. (Adapted from MB-System Cookbook, Schmidt et al. 2004). A partial list can also be found in the man page for mbprocess.

.inf - file statistics

These files contain metadata about the multibeam data, including geographic, temporal and depth bounds, ship track information, and numbers of flagged or zero beams. They are created automatically by mbpreprocess and mbprocess, or by mbinfo via mbdatalist.

.fbt - fast bathymetry

These files are bathymetry only format 71 files that are intended to provide faster access to swath bathymetry and amplitude data than in the original format. They are created automatically by mbpreprocess and mbprocess, or by mbcopy via mbdatalist.

.fnv - fast navigation

These files are simply ASCII navigation files that are intended to provide faster access to swath navigation than data in the original format. These are created for all swath data formats except single beam, navigation only, or XYZ files. These are created automatically by mbpreprocess and mbprocess, or by mblist via mbdatalist.

.par - parameter file

These files specify settings and parameters controlling how mbprocess generates a processed swath data file from a preprocessed swath data file. These are generated or updated by all of the data editing and analysis tools, including mbedit, mbnavedit, mbreditviz, mbvelocitytool, mbclean, mbbackangle, and mbnavadjust. They can also be directly altered by mbset or using a text editor.

.esf - bathymetry edit flags

These files contain the bathymetry edit flags output by mbredit, mbreditviz and/or mbclean. The edits are stored in binary format (and not human-readable).

.nve - edited navigation

These files contain the edited navigation output by mbnavedit.

.na0 - adjusted navigation

These files contain adjusted navigation output by mbnavadjust. When enabled, the mbnavadjust navigation files supersede .nve files output by mbnavedit.

.bas - asynchronous attitude (roll and pitch)

They contain all attitude data acquired by motion sensor and stored in the multibeam file. While the fast navigation contains only the attitude associated with each ping, here the attitude is complete and asynchronous (from the bathymetry data). The amount of data is set by the rate which the motion sensor sends attitude data to the acquisition software. These files are created by mbpreprocess, and if present can be used by mbeditviz to assess time latency between the sonar and the attitude, heading, and sensor depth sensors.

.bah - asynchronous heading

The same as above but for the heading data. These files are created by mbpreprocess.

.bas - asynchronous sensor depth

The same as above but for the sensor depth data. These files are created by mbpreprocess.

.baa - synchronous attitude (roll and pitch)

They contain the attitude data acquired by motion sensors and associated with each ping. Here the attitude is incomplete and synchronous (with the bathymetry data). The amount of data is correlated to the number of pings inside the multibeam file. These files are created by mbpreprocess, and can be used by mbeditviz to assess time latency between the sonar and the attitude sensors if full asynchronous data are not available.

.tde - tide files

These files are used for tidal correction. They can be generated using mbotps, or can be supplied from other sources by users, in which case the use of the tide files must be enabled using mbset.

.lck - lock file

Lock files allow multiple users to work on the same dataset simultaneously. With this feature the user who opens file(s) for editing automatically locks the file(s) to prevent that another user(s) edit the same file(s) at same time. Once the user properly closes the file (using the "Quit" or "Dismiss" buttons) it will unlock and will become available for another user(s) to open. Lock files will also prevent mbprocess from acting on a file that is actively being edited or analyzed. It is not recommended, but this feature can be deactivated using mbdefaults.

Appendix C – MB-System installation step-by-step on Mac OSX 16 (Sequoia) using Macports on a “sandbox” (user folder)

The following is modified from the instructions at:

<https://www.mbari.org/technology/mb-system/installation/>

```
#-----
# MacOs Example Using MacPorts to Install Prerequisite packages:
#-----
#
# The use of MacPorts to install the MB-System prerequisite packages is
# recommended on Apple Mac computers, particularly because this approach enables
# cleanly building with a complete X11 + Motif + OpenGL infrastructure separate
# from the libraries and header files associated with the XQuartz package.
# One usually still installs XQuartz and uses it as the X11 display server, but
# confining the MB-System compilation and linking to headers and libraries
# within the MacPorts structure avoids several issues.
#
# This example is relevant for MacOs 10.13 High Sierra to the current MacOs 15
# Sequoia on both Intel and ARM (Apple Silicon) architecture computers.
#
# Install Xcode and the Xcode command line tools, which includes the LLVM
# compiler suite.
    xcode-select --install
#
# Install the XQuartz X11 server from
# https://www.xquartz.org
# XQuartz 2.8.5 or later is required for all MB-System installations.
#
# Install MacPorts using the appropriate downloadable installer package from:
# https://www.macports.org/install.php
# Or use the automate installation procedure bellow:
# curl -L https://github.com/XQuartz/XQuartz/releases/download/XQuartz-2.8.5/XQuartz-2.8.5.pkg
# sudo installer -pkg "XQuartz-2.8.5.pkg" -target /
#
# After MacPorts installation, first make sure the default port packages are
# current by running selfupdate and then install the MB-System prerequisites.
# GGC14 is required for compiling the OTPS fortran code since it provides gfortran.
# However, no symlink to gfortran is created by GCC, and must be done manually.
# Also, no symlink to GMT exists and must be created too.
sudo port -v selfupdate
sudo port install gmt6 proj9 fftw-3 xorg mesa libGLU openmotif gcc14 gdal opencv4 gv sqlite3-tcl
sudo ln -s /opt/local/bin/gfortran-mp-14 /opt/local/bin/gfortran
sudo ln -s /opt/local/bin/gmt6 /opt/local/bin/gmt
#
# Also make sure that a current version of Python3 is available. First list the
# available Python3 versions, install the most recent, and then set port to link
# that version to python3:
port search python
sudo port install python313
sudo port select --set python3 python313
# Also install the most recent Python imaging library Pillow
sudo port install py313-Pillow
#
# Download the MB-System source package from the repository at GitHub:
```

```

# https://github.com/dwcaress/MB-System
# There are often beta releases that are more recent than the current stable
# release. For instance, to download 5.8.2.beta21 go to:
# https://github.com/dwcaress/MB-System/releases

#
# Make a folder in the user home folder called: sandbox
cd
mkdir sandbox
#
# Now upzip the tar.gz file in the folder and rename it to simply: mbsystem
# with that done update the path in line 75-76, 87 and 93 to match your username.
#
# Unpack the MB-System distribution tarball, and then cd into the top directory
# of the resulting structure. This will typically be named something like
# "MB-System-5.8.2". At that location, execute the configure script, named
# "configure", with the options necessary for your context. The XCode compiler
# tools do not look for header files or libraries in the locations used by
# MacPorts, and so it is necessary to specify these locations for several of
# the prerequisite packages.
cd /sandbox/mbsystem
#
# This command should successfully enable building the current MB-System
# (5.8.2 or later) on any Mac computer with the prerequisites installed through
# MacPorts. This has been tested with computers running Sequoia, Ventura,
# and Monterey.
#
# mkdir build
cd build
cmake .. -DCMAKE_INSTALL_PREFIX=/Users/christian/sandbox/mbsystem \
-DotpsDir=/Users/christian/sandbox/otps -DbuildTRN=OFF
#
# Once the makefiles have been generated by cmake, build and install
# MB-System using:
make
make check
make install
#
# Edit the paths to know where the binaries of MB-System are
sudo nano /etc/paths
# and add a line with:
/Users/christian/sandbox/mbsystem/bin
#
# In order for GMT to successfully execute the MB-System modules, the location of the shared
# libraries containing the dynamically loaded modules must be known to GMT. This can be
# accomplished by using the GMT module gmtset. This step must be done in your home directory.
cd
gmt gmtset GMT_CUSTOM_LIBS /Users/christian/sandbox/mbsystem/lib/mbsystem.so
#
# The parallel processing modules mbm_multiprocess, mbm_multidatalist, and mbm_multicopy
# require the perl module Parallel::ForkManager. This can be added to your local perl installation
# using the command line utility cpan:
sudo cpan LWP::Simple
sudo cpan Parallel::ForkManager
#
#-----

```

Appendix D – MB-System/OTPS installation on a sandbox (home folder) using Ubuntu 24.04

```
sudo apt-get install gmt libgmt6 libgmt-dev gmt-gshg gmt-common libx11-dev xorg-dev libmotif-dev libmotif-common libglu1-mesa libglu1-mesa-dev mesa-common-dev libsdl1.2-dev libsdl-image1.2-dev build-essential gfortran libfftw3-3 libfftw3-dev libproj-dev gdal-bin libgdal-dev cmake xfonts-base xfonts-100dpi xfonts-75dpi xfonts-cyrillic libopencv-dev
```

```
# download latest version from releases: https://github.com/dwcaress/MB-System/releases
```

```
# unpack and rename folder to "mbsystem"
```

```
# run "sudo nautilus" on a terminal, and copy folder "mbsystem" to "/usr/local", afterwards close this Nautilus window, and don't use it for anything else!
```

```
# now on a terminal run the following commands:
```

```
cd /usr/local/mbsystem/
```

```
sudo mkdir build
```

```
cd build
```

```
sudo cmake .. -DCMAKE_INSTALL_PREFIX=/usr/local/mbsystem -DotsDir=/usr/local/otps
```

```
sudo make
```

```
sudo make install
```

```
# run the command bellow and add this path bellow ":/usr/local/mbsystem/bin" at the end of the line)  
sudo nano /etc/environment
```

```
# press CONTROL-O to save, and after CONTROL-X to exit
```

```
# Now connect GMT to the MB-System modules, for that open a terminal and enter the following commands:
```

```
cd
```

```
gmt gmtset GMT_CUSTOM_LIBS /usr/local/mbsystem/lib/mbsystem.so
```

```
# Install the Perl modules for parallel processing
```

```
sudo cpan Parallel::ForkManager
```

```
# Then reboot and the installation is ready
```

Appendix E – Extra steps after installing MB-System on Linux or Mac OSX

These additional steps are necessary to fully operate MB-System with all add-ons.

a) Configuring Multi-CPU Macros:

Users will require the installation of a Perl module to utilize these macros. The installation process is straightforward. The library is known as Parallel-ForkManager. To install it simply run:

```
% sudo cpan Parallel::ForkManager
```

Now the multi CPU macros should work.

b) Open Terminal within the folder:

This previously necessitated third-party software, but more recently, all operating systems appear to have incorporated this feature. It is typically accessible by right-clicking on a folder name and selecting “Open Terminal in Folder” (or a similar option).

c) Enable the MB-System/GMT modules (e.g., mbswath, mbcontour, and mbm_grdtiff):

To create the MB-System/GMT modules, users must first compile them. Inside the MB-System home folder, there is a folder called /lib, and inside it should be a file named "mbsystem.so".

Next, we must enable these modules by linking the library inside the gmt.conf file. To do this, run the following command:

```
% cd  
% gmt gmtset GMT_CUSTOM_LIBS /usr/local/lib/mbsystem.so
```

Please note that this configuration must be run at the user’s home folder, which is why the first command is “cd” (change directory). This will create a file named gmt.conf that must remain in place.

d) Disable compressed netCDF grids in MB-System:

The NetCDF 4.0 standard (from 2008) added the possibility of grid file compression. The compression benefits the users with grids that are up 90% smaller, however some software doesn’t support this feature either because it’s old software or new software where developers still haven’t updated their NetCDF driver. To disable the grid compression and make grids compatible with this software one should run this command:

```
% gmt gmtset IO_NC4_CHUNK_SIZE classic
```

But I urge people to contact their software developers and ask for updating the NetCDF driver if possible.

PS: Please note that upgrading GMT via MacPorts or manually could eventually revert this setting back to the default.

e) Install OTPS

To install OTPS first head to their website and download OTPS.

<https://www.tpxo.net/home>

Next register there and download the models (currently Version 5) and place the files inside the subfolder “DATA/TPXO9_atlas_v5” where OTPS was installed. Now run “make” inside the OTPS folder to build the tool “predict_tide”. At last compile MB-System, and if you followed the sandbox instructions when installing MB-System change the default

location where MB-System looks for OTPS (at /usr/local) to your user folder. That can be done when calling “cmake” for compiling MB-System by using option “-DotpsDir=/Users/christian/sandbox/otps”. If have already installed MB-System, you can run “cmake” with this additional option, and run “make” and “make install” to update your installation. If you installed OTPS at the default local then MB-System will find and connect to it without any additional steps.