NAME

mbnavadjust - Package that solves for optimal navigation by matching bathymetry of overlapping swaths.

VERSION

Version 5.0

SYNOPSIS

mbnavadjust [**-V -H**]

DESCRIPTION

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) are frequently not navigated with accuracy equivalent to the lateral resolution of swath bathymetry obtained with high frequency sonars operated close to the seafloor. This is particularly true for systems that depend on ultra-short baseline navigation (USBL) or on inertial navigation (as opposed to long baseline (LBL) navigation obtained using transponder networks). 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 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 Simrad 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 throughout a survey, it is possible to solve, or invert, for a navigation model which is consistent with the original navigational constraints, satisfies the requirement of reasonableness, and also 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 when all vertical offsets have been corrected prior to importation into **MBnavadjust** (e.g. tidal corrections have been made). Otherwise, the offset analysis and inversions are three dimensional.

Previous attempts to address this problem (e.g. Nishimura et al. [1988]) have focused on automating the identification of matching features and their navigational offsets. The automated approach is problematic because seafloor features are complex, and the cross correlation function of matching features often exhibits multiple local minima. Our approach is to automate the identification of matching features and the ultimate solution for an optimal indentification while depending on interactive determination of the navigational offsets. We have not found an automatic algorithm that can reliably replace the human eye and brain in correctly lining up real seafloor features.

MBnavadjust operates within the context of the **mbprocess** parallel processing scheme, but it should used on data for which the initial navigation processing (e.g. **mbnavedit** is complete. When swath data files are imported into **mbnavadjust**, the data is taken from processed files generated by **mbprocess** whenever possible (when those files exist). If no processed file exists, the raw data is used. When the processing with **mbnavadjust** is completed and an optimal navigation solution is achieved, new navigation files are generated (".na#" files, see below for full description) which supercede the ".nve" files generated by **mbnavedit**. Rerunning **mbprocess** will merge the new navigation into a new set of processed files. 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 complicating his or her efforts.

The inversion for an optimal navigation which fits the offsets identified at matching features must also be reasonable. This is achieved by minimizing perturbations to speed and acceleration in the inversion. This approach of penalizing first and second derivatives within a linear inverse problem is common within the geophysical inverse theory literature (e.g. Parker [1994]).

Users are advised that **mbnavadjust** is complicated in both conception and implementation. We strongly recommend that users read all of the documentation provided below prior to attempting a first use of this software. The meaning and use of the individual widgets and windows employed in the graphical interface is provided in the **INTERACTIVE CONTROLS** section. The **USING MBNAVADJUST** section provides a more coherent discussion of how to use **mbnavadjust**, how this program interacts with the **mbprocess** parallel processing scheme, and the underlying concepts and algorithms.

AUTHORSHIP

David W. Caress (caress@mbari.org)
Monterey Bay Aquarium Research Institute
Dale N. Chayes (dale@ldeo.columbia.edu)
Lamont-Doherty Earth Observatory

OPTIONS

–H This "help" flag cause the program to print out a description of its operation and then exit immediately.

–V Normally, **mbnavadjust** outputs nothing to the stderr stream. If the **-V** flag is given, then **mbnavadjust** works in a "verbose" mode and outputs the program version being used, all error status messages, and a large amount of other information.

INTERACTIVE CONTROLS

MBnavadjust Window

This is the primary window of **mbnavadjust**.

File This button activates a pulldown menu with six items: **New**, **Open**, **Close**, **Import Swath Data**, and **Quit**.

File->New

This menu item brings up a popup window which allows the user to specify a new **mbnavadjust** project. As part of initializing a new project, **mbnavadjust** creates a file with a ".nvh" suffix and a directory with a ".dir" suffix.

File->Open

This menu item brings up a popup window which allows the user to open an existing **mbnavadjust** project.

File->Close

This menu item causes **mbnavadjust** to close the current project.

File->Import Swath Data

This menu item brings up a popup window which allows the user to specify a swath sonar data file or a list of swath files to be imported into the current **mbnavadjust** project. This program uses the **MBIO** library and will read any swath sonar formats supported by **MBIO**. A list of the swath sonar data formats currently supported by **MBIO** and their identifier values is given in the **MBIO** manual page. If the swath sonar data file is named using the **MB-System** suffix convention (format 11 files end with ".mb11", format 41 files end with ".mb41", etc.), then the program will automatically use the appropriate format id; otherwise the format must be set by the user in the format text box.

File->Quit
This button causes **mbnavadjust** to quit. If a project has been created or opened, that project will be closed before the program quits.

Option This button brings up a pulldown menu with one item: **Controls**.

Option->Controls
This menu item brings up a dialog window with several slider widgets that set parameters controlling how swath data are imported and displayed. These widgets are discussed in the **Controls** section below.
.TP **View** This button brings up a pulldown menu with fifteen items in three groups. This menu controls the contents of the list widget in the lower right **MBnavadjust Window**. If an **mbnavadjust** project has been created or opened, the **Data Table** list widget will display a table of items controlled by these menu items, and the corresponding menu item will be disabled. Selection of another item items will change what is displayed in the **Data Table** list widget accordingly.

The first group includes the **Show Surveys**, **Show Data Files**, **Show Data File Sections**, **Show All Crossings**, **Show All Crossings**, **Show >25% Crossings**, **Show >50% Crossings**, **Show True Crossings**, and **Show Ties** items.

View->Show Surveys
This menu item causes the **Data Table** list widget to display a list of the surveys in the current **mbnavadjust** project. Within **MBnavadjust**, a survey is a group of swath files that are contiguous in time.

View->Show Data Files
This menu item causes the **Data Table** list widget to display a list of the swath data files in the current **mbnavadjust** project.

View->Show Data File Sections
This menu item causes the **Data Table** list widget to display a list of the swath data sections in the current **mbnavadjust** project.

View->Show All Crossings
This menu item causes the **Data Table** list widget to display a list of all the swath crossings in the current **mbnavadjust** project. When this option is set, selecting a list item causes **mbnavadjust** to load and display the selected selected crossing in the "Nav Err" window.

View->Show >25% Crossings
This menu item causes the **Data Table** list widget to display a list of all the swath crossings in the current **mbnavadjust** project for which the two sections overlap by more than 25%. When this option is set, selecting a list item causes **mbnavadjust** to load and display the selected selected crossing in the "Nav Err" window.

View->Show >50% Crossings
This menu item causes the **Data Table** list widget to display a list of all the swath crossings in the current **mbnavadjust** project for which the two sections overlap by more than 50%.. When this option is set, selecting a list item causes **mbnavadjust** to load and display the selected selected crossing in the "Nav Err" window.

View->Show True Crossings
This menu item causes the **Data Table** list widget to display a list of all the swath crossings in the current **mbnavadjust** project for which the navigation tracks cross. When this option is set, selecting a list item causes **mbnavadjust** to load and display the selected selected crossing in the "Nav Err" window.

View->Show Ties
This menu item causes the **Data Table** list widget to display a list of the tie points in the current **mbnavadjust** project.

The second group includes the **Show All Surveys**, **Show Only Selected Survey**, **Show Only Selected File**, and **Show Only Selected Section** items. One of these options is always active, and modifies what is displayed in the **Data Table** list.

View->Show All Surveys
This menu item allows all of the possible items (file, sections, crossings, ties) to be displayed in the **Data Table** list.

View->Show Only Selected Surveys
This menu item limits the items (file, sections, crossings, ties) displayed in the **Data Table** list to those associated with the most recently selected survey (e.g. a survey selected by clicking in the **Data Table** list while the <View->Show Surveys> option is active).

View->Show Only Selected File
This menu item limits the items (file, sections, crossings, ties) displayed in the **Data Table** list to those associated with the most recently selected file (e.g. a file selected by clicking in the **Data Table** list while the <View->Show Files> option is active).

View->Show Only Selected Section
This menu item limits the items (file, sections, crossings, ties) displayed in the **Data Table** list to those associated with the most recently selected section (e.g. a section selected by clicking in the **Data Table** list while the <View->Show Sections> option is active).

The third group includes two options:

View->Visualize Survey
This option is not active, and is for future use.

View->Show Modelplot
This option causes an addition "Model Plot" window to be shown that displays the navigation adjustment model as three time series: longitude on top, latitude in the middle, and vertical on the bottom.

Action This button brings up a pulldown menu with nine items: **Set File Poor Navigation**, **Set File Good Navigation**, **Set File Fixed Navigation**, **Auto-Pick Offsets**, **Check for New Crossings**, **Analyze Crossings**, **Zero All Z Offsets**, **Invert Navigation**, and **Apply Adjusted Navigation**.

Action->Set File Poor Navigation
This menu item causes **mbnavadjust** to treat the navigation of a selected swath data file as poor. This option is only activated when the **Data Table** list widget shows a list of the imported swath files and one file has been selected by clicking in that list. In the case where no files are fixed, the final adjusted navigation model is centered on the average navigation offsets required for good files. The offsets tieing to files with poor navigation are ignored in calculating the averages.

Action->Set File Good Navigation
This menu item causes **mbnavadjust** to treat the navigation of a selected swath data file as good, which is in fact the default state for swath files. This option is only activated when the **Data Table** list widget shows a list of the imported swath files and one file has been selected by clicking in that list. In the case where no files are fixed, the final adjusted navigation model is centered on the average navigation offsets required for good files. The offsets tieing to files with good navigation are used in calculating the averages.

Action->Set File Fixed Navigation
This menu item causes **mbnavadjust** to treat the navigation of a selected swath data file as fixed, which means that all navigation points for this file will have zero offsets in the final adjusted navigation. Effectively, all of the offsets between fixed and unfixed files are taken up in the adjustments of the unfixed files. This option is only activated when the **Data Table** list widget shows a list of the imported swath files and one file has been selected by clicking in that list.

Action->Auto-Pick Offsets
This menu item causes **mbnavadjust** to automatically pick offsets in all of the unset crossings currently displayed in the **Data Table** list. If <View->Show True Crossings> has been selected, then only true crossings are displayed and only the unset true crossings will be automatically picked. In general, unsupervised offset picks are frequently erroneous, and use of this experimental option is not encouraged.

Action->Check For New Crossings
This menu item causes **mbnavadjust** to search for new crossings using the adjusted navigation from the current inversion. Often, new places where swaths overlap and cross emerge as the navigation is adjusted. The percentage of overlap is also recalculated for all of the existing crossings. If new true crossings are found, the user will be unable to perform a new inversion until all of the true crossings have been analyzed.

Action->Analyze Crossings
This menu item causes **mbnavadjust** to load and display the first crossing in the "Nav Err" window.

Action->Zero All Z Offsets
This menu item causes **mbnavadjust** to load and display the first crossing in the "Nav Err" window.

Action->Invert Navigation
This menu item causes **mbnavadjust** to invert for an optimal navigation solution that fits the offsets defined at the navigation tie points while minimizing speed and acceleration. This option is only activated when all of the true crossings in the current **mbnavadjust** project have been analyzed interactively.

Action->Apply Adjusted Navigation
This menu item causes **mbnavadjust** to export the adjusted navigation model from the current inversion to the swath files in the project.This option is only activated when the inversion is current, meaning that no new data have been imported and no changes to offset picks have been made since the last inversion.

About This button brings up a pulldown menu with one item: **About**.

About->About
This menu item brings up a popup window that displays the program's name, authors, and version. Clicking the **Dismiss** button will close the window.

Data Table
This list widget displays one of five tables of information regarding the current **mbnavadjust** project. These tables are a list of the imported swath surveys, the data files, the shorter data sections comprising each data file, a list of the identified crossing swaths, and a list of the navigation tie points defined interactively by the user. The **View** pulldown menu allows the user to set which table is displayed.

The survey list will look something like:

```
00 53 2009/08/03 08:18:49.484999 2009/08/03 22:52:59.375000
01 51 2009/08/04 09:03:11.938999 2009/08/04 23:02:03.470999
02 01 1998/05/13 01:33:36.791000 1998/05/13 02:42:11.703999
```

Here the first column is the survey counter, the second is the number of swath files included in each survey, and the following information consists of the start and end times of the data in each survey shown in YYYY/MM/DD HH:MM:SS.SSSSSS format.

The file list will look something like:

```
0000:00 gd   11  0.0  0.0 ../../20090803/20090803_081706.mb88
0001:00 gd   11  0.0  0.0 ../../20090803/20090803_083332.mb88
0002:00 gd   11  0.0  0.0 ../../20090803/20090803_085004.mb88
```

```
0003:00 gd   11  0.0  0.0 ../../20090803/20090803_090636.mb88
0004:00 gd   10  0.0  0.0 ../../20090803/20090803_092307.mb88
             ..........................................
0053:01 gd   12  0.0  0.0 ../../20090804/20090804_090127.mb88
0054:01 gd   10  0.0  0.0 ../../20090804/20090804_092036.mb88
0055:01 gd   11  0.0  0.0 ../../20090804/20090804_093707.mb88
0056:01 gd   10  0.0  0.0 ../../20090804/20090804_095339.mb88
0057:01 gd   11  0.0  0.0 ../../20090804/20090804_101010.mb88
             ..........................................
0104:02 fx    8  0.0  0.0 ../../MBARI/1998em300/mbari_1998_630_msn.mb57
```

Here the first column is the file counter and survey counter separated by a colon. The second column indicates the file navigations state; "gd" indicates good navigation, "pr" indicates poor navigation, and "fx" indicates fixed navigation. The third column shows the number of sections extracted from this file. The fourth and fifth columns show any heading or roll bias offsets in degrees applied to the swath data for that file. The sixth column gives the name of the swath data file imported into **mbnavadjust**. Note that the name shown here is that of the "raw" swath file. The data imported by **mbnavadjust** is, if possible, extracted from a "processed" swath file generated by **mbprocess** rather than the associated "raw" file.

The file section list will look something like:

```
00:0000:00 2009/08/03 08:17:07.546998 2009/08/03 08:18:49.484999
00:0000:01 2009/08/03 08:18:49.984999 2009/08/03 08:20:26.952999
00:0000:02 2009/08/03 08:20:27.452999 2009/08/03 08:22:05.890999
00:0000:03 2009/08/03 08:22:06.390999 2009/08/03 08:23:43.344001
00:0000:04 2009/08/03 08:23:43.844001 2009/08/03 08:25:19.796999
00:0000:05 2009/08/03 08:25:20.296999 2009/08/03 08:26:57.265997
00:0000:06 2009/08/03 08:26:57.765997 2009/08/03 08:28:35.219001
00:0000:07 2009/08/03 08:28:35.719001 2009/08/03 08:30:16.155999
00:0000:08 2009/08/03 08:30:16.655999 2009/08/03 08:31:57.594001
00:0000:09 2009/08/03 08:31:58.094001 2009/08/03 08:33:36.546999
00:0000:10 2009/08/03 08:33:37.046999 2009/08/03 08:33:37.546999
```

Here the first column shows the section id with the survey counter, the file counter, and the section counter separated by colons. The following information consists of the start and end times of each section shown in YYYY/MM/DD HH:MM:SS.SSSSSS format.

The crossing list will look something like:

```
-X    0 000:009 001:000  21  0
-     1 001:009 002:000  10  0
-     2 002:009 003:000  10  0
U     3 003:009 004:000   6  0
-     4 005:009 006:000  13  0
U     5 007:009 008:000   3  0
U     6 009:008 009:010   2  0
*     7 009:008 010:000  16  1
*X    8 009:009 010:000  41  1
-     9 009:008 010:001   1  0
```

Here the first column indicates the processing status for the crossing. The first character is the status flag. If the status flag is "U", then no decision has been made about skipping or tieing this crossing. New crossings always show a "U" flag prior to being inspected by a user. If the first character is "-", then the crossing has been skipped, and if the first character is "*", then at least one tie point has been set. The second column is the crossing counter. The third and fourth columns identify the swath data sections that overlap in this crossing. Each are identified by their file id and section id separated by ":". The fifth column indicates the percentage of overlap of the two sections in this crossing. The larger the degree of overlap, the more likely that diagnostic matching topographic features exist that can be used to determine the navigation offsets required for this crossing. The sixth, and last column gives the number of tie points that have been defined for each crossing. New crossings always begin with 0 tie points prior to being inspected by a user.

The tie point list will look something like:

```
 7 0 009:008:07 010:000:04 00:00  1.02 -2.87 0.00 |  9.61  8.49 1.40 | 0.041 0.107 0.027
 8 0 009:009:02 010:000:00 00:00  0.90 -4.47 0.00 |  7.37  6.30 2.29 | 0.040 0.184 0.033
12 0 009:008:04 010:002:04 00:00  4.11 -8.24 0.00 |  9.28  5.99 1.80 | 0.037 0.021 0.025
14 0 010:000:02 010:002:06 00:00  2.12 -4.38 0.00 |  7.50  6.70 1.10 | 0.007 0.065 0.005
16 0 009:007:04 010:003:04 00:00  5.90 -5.90 0.00 |  8.66  6.84 1.70 | 0.019 0.085 0.027
19 0 009:006:04 010:004:04 00:00  8.32 -6.83 0.00 |  8.78  8.55 6.65 | 0.026 0.066 0.024
21 0 009:005:05 010:005:03 00:00  8.96 -8.75 0.00 | 12.38  8.74 5.16 | 0.008 0.045 0.008
23 0 009:004:04 010:006:04 00:00 12.23 -5.98 0.00 |  7.03  4.49 1.40 | 0.019 0.034 0.010
26 0 009:003:04 010:007:04 00:00 17.42 -4.36 0.00 | 13.92 10.11 1.80 | 0.025 0.037 0.028
```

Here the first column indicates the crossing which contains the tie point, and the second column shows which tie point (of those defined for that crossing) is displayed in a particular line. The third and fourth columns identify the navigation control points of the tie point. The navigation control points are specified by file, section, and nav point numbers separated by ":". The fifth through seventh columns are the longitude, latitude and vertical offsets (in meters) set interactively by the user. These represent the distance the second navigation control point must be moved relative to the first in order to make the bathymetry in the two swaths match. The tenth through twelth columns show the magnitude of the three axes of the uncertainly ellipsoid associated with each tie. The uncertainly ellipsoid is estimated as a 3x3 tensor and used to weight the tie offsets in the navigation adjustment inversion. Here the major and second axes are always close to horizontal, and the minor is axis is always close to vertical. The last three columns are nonzero only after an inversion for an optimal navigation solution has been performed. These represent the residual, or difference, between the offset calculated for this tie point in the inversion and that set by the user (displayed in the fifth through seventh columns).

If the file list is displayed and one file is selected by clicking in the list, then the user can fix or unfix the navigation of that file using the <Action->Fix File> or <Action->Unfix File> pulldown menu items. If either the crossing list or the tie point list are displayed, selecting one crossing or one tie point by clicking in the list widget will cause the specified crossing to be loaded and displayed in the **Nav Err** window.

Controls Window

This dialog window is brought up by clicking the **Option->Controls** menu item in the **MBnavadjust Window**. This window includes two slider widgets setting data importation parameters, four slider widgets setting data display parameters, a button to apply any changes, and a button to dismiss the window.

Max Section Length (km)

When **mbnavadjust** imports swath data, it breaks each data file up into a set of sections, or pieces. The geographical coverages of all sections are then compared to determine where swaths overlap or cross. This control sets the section track length. The ideal section length varies with the sonar altitude and ping rate. If the section length is too short, there will be too many small sections and an excessive number of crossings to analyze. If the section length is too large, individual sections will be slow to load and display.

Max # Soundings in Section

This slider widget sets an additional or alternative control on swath data section size: the maximum number of soundings (not pings) that may be included in a given section. For some datasets, it may be more convenient to use the number of soundings rather than the track length as the criterea for breaking files up into sections.

Decimation

When swath bathymetry sections are displayed in the **Nav Err** window, the primary display consists of bathymetric contours. The time required for calculating these contours is linear with the number of soundings. This slider allows users to decimate the pings used for the contour displays. If the decimation is set to 1, all soundings are used. If the decimation is set to 2, the soundings from every second ping are used for contouring.

Contour Interval (m)

This slider widget sets the depth interval in meters for the bathymetric contours.

Color Interval (m)

This slider widget sets the the depth interval in meters at which the color of the bathymetric contours changes. The contours are generated in four colors: black, red, green, and blue. The color interval should be a multiple of the contour interval.

Tick Interval (m)

This slider widget sets the depth interval in meters at which contours have downhill facing tickmarks. The tick interval should be a multiple of the contour interval.

Inversion Smoothing

This slider widget sets the importance of smoothing in the adjusted navigation model. Larger values yield a smoother model, smaller values a rougher model. The default is 4.00, and possible ranges are 0.10 to 10.0, where the smoothing penalty weight applied to the first and second derivatives of the adjusted navigation perturbation is ten raised to this power.

Apply This button applies any changed values in the slider widgets of the **Controls Window**. If the **Controls Window** is closed without the **Apply** button being clicked, all changes to the values controlled by the sliders are lost.

Dismiss

This button closes the **Controls Window**.

Nav Err Window

This window allows users to interactively inspect crossing swaths, define navigation tie points where bathymetric features can be matched, and set the navigation offsets at those tie points. This window can be brought up in a number of ways. If one pulls down the menu option <Action->Analyze Crossings>, then the "Nav Error" window will come up with the first crossing loaded. Alternatively, if the <View->Show All Crossings> menu item has been selected so that a table of crossings is displayed, clicking once on any of the crossing items in this tabel will bring up the "Nav Error" window with that crossing loaded. Similarly, if tie points have been established and the tie point table displayed by selecting the <View->Show Ties> menu item, then clicking on any of the tie items this table will bring up the "Nav Error" window and load the crossing that includes the selected tie point. If the "Nav Error" window is already displayed, clicking on crossing or tie items in the display tables will load the selected crossing in place of whatever crossing was previously shown.

The Nav Err window includes a number of button widgets and three display canvases. The larger display to the right shows bathymetric contour maps of the overlapping swaths overlain by navigation tracks and any tie points that have been defined for the current crossing. The smaller canvas on the middle left shows the RMS bathymetry misfit between the two swaths as a function of lateral (x and y) offset using the current vertical (z)offset. The smallest canvas on the lower left shows the RMS Bathymetry misfit between the two swaths as a function of vertical offset using the current lateral offset.

Contour Display

This canvas displays color bathymetric contours from both overlapping swaths in the current swath crossing. Initially, the display is scaled so that the entire area covered by the current crossing is visible. The bathymetric contours of the second swath can be moved relative to those of the first by dragging with the left mouse button in the canvas. The current navigational offset changes as the contours shift. The location of the current lateral offset also changes simultaneously on the misfit plot. The user can display a subregion of the current map by dragging a rectangle in the canvas with the middle mouse button. This zoom operation can be repeated as many times as desired. The right mouse button is used to set the location of the current navigational tie point. When the right mouse button is clicked on a point in the contour map, the closest navigation control points from each swath to that position are selected for the current tie point. In practice, the user should right-click on the bathymetric feature being matched to set the tie point correctly.

Lateral Misfit Display

This canvas shows a color two dimensional plot of the RMS bathymetric misfit between the two overlapping swaths. The misfit is shown as a function of relative lateral (x and y) offsets between the two swaths using the current vertical (z) offset. Put another way, the misfit plot shows how good, or bad, the misfit becomes as one moves one swath relative to another. The lowest misfit values are shown in red; higher misfits are shown in blues to purples. The location of the minimum misfit is marked by a black "X", and the location of the current navigation offset is shown by a small red square with a black outline. If an inversion has been performed, a small black "+" will mark the location of the offset calculated by the inversion. If a user left-clicks in the Lateral Misfit Display, the location of the cursor is taken as the new navigation offset and the red square moves to that location. Simultaneously, the bathymetric contours in the Contour Display shift to reflect the new offset.

Vertical Misfit Display

This canvas shows a color profile plot of the RMS bathymetric misfit between the two overlapping swaths. The misfit is shown as a function of relative vertical (z) offset between the two swaths using the current lateral (x andy) offsets. Put another way, the misfit plot shows how good, or bad, the misfit becomes as one moves one swath vertically relative to another. The lowest misfit values are shown in red and as high points in the profile; higher misfits are shown in blues to purples and as low points. The location of the minimum misfit is marked by a black "X", and the location of the current vertical offset is shown by a solid line. If an inversion has been performed, a small black "+" will mark the location of the offset calculated by the inversion. If a user left-clicks in the Vertical Misfit Display, the location of the cursor is taken as the new vertical offset and the solid line moves to that location. Simultaneously, the bathymetric contours in the Contour Display shift to reflect the new offset.

Settings

This button brings up a pulldown menu with two items: **Biases** and **ContoursB**

Settings->Biases

This menu item brings up a dialog window with several slider widgets that set roll and heading bias values. These widgets are discussed in the **Biases** section below.

Settings->Contours

This menu item brings up a dialog window with several slider widgets that set parameters controlling how swath data are imported and displayed. These widgets are discussed in the **Controls** section above.

Previous

Clicking this button causes the Nav Err window to load and display the previous crossing.

Next Clicking this button causes the Nav Err window to load and display the next crossing.

Next Unset

Clicking this button causes the Nav Err window to load and display the next crossing which has no tie points set and has not been explicitly skipped.

Skip Crossing

This button indicates that no good tie points can be defined for the current crossing. The program **mbnavadjust** will not allow an inversion to be calculated until all true crossings either have tie points defined or have been explicitly skipped.

Full Size
Clicking this button causes the contour display to resize so that it shows the entire area covered by the swath sections in the current crossing.

Unset Crossing
This button returns the current crossing to the original unset state, deleting any ties that have been set.

Reset Crossing
This button is active only if a tie has been set for the current crossing. If a tie exists, and the user has changed the current offset from that of the current tie, then this button returns the working offset values to that of the existing tie.

Set Offset
This very important button causes the current navigational offset to be set for the current tie point and crossing. This button is the only way that the navigational offsets are actually applied to the **mbnavadjust** project. If the user closes the Nav Err window, changes the loaded crossing, or changes the active tie point without clicking <Set Offset>, whatever navigational offset was viewed will be lost. Tie points are displayed in yellow once they are set, and in red when either the navigational offset or the navigation points have been changed but not yet set.

Zero Z Offset
This button sets the current vertical offset to zero meters.

Add Tie
This button adds a new tie point for the current crossing and makes this new tie point active. The tie point initially adopts the current navigational offset.

Zero Offset
This button sets the current navigational offset to zero meters for longitude, latitude, and vertical.

Select Tie
If more than one navigation tie point has been defined for a crossing, then only one tie point can be active at any time. The navigation control points of the active tie points will be displayed as red boxes, and those of any inactive tie points will be shown as smaller red boxes. This button changes the active tie point. Repeatedly clicking this button will cycle through all of the tie points in the current crossing. Note that each tie point has its own offset defined, so in general the positions of the bathymetric contours will change as the active tie point changes.

Delete Tie
This button deletes the current active tie point.

Dismiss
This button closes the Nav Err window.

Minimum Misfit
This button shifts the current offset to the value identified as the minimum misfit in the current misfit display. The minimum misfit is shown as the large black X on the color lateral misfit plot and on the vertical misfit profile.

Minimum XY Misfit
This button shifts the current lateral offset to the values identified as the minimum lateral misfit in the current misfit display. The vertical offset is not changed. The minimum lateral misfit is shown as the small black X on the lateral misfit plot. This button is generally used when the data are already corrected for all vertical offsets (e.g. tides), and so all z offsets can be confined to zero values.

Center on Zero Offset
This button causes the misfit plot to be regenerated using a zero navigational offset as the plot origin.

Center on Current Offset

This button causes the misfit plot to be regenerated using the current navigational offset as the plot origin.

Biases Window

This dialog window is brought up by clicking the **Settings->Biases** menu item in the **Nav Err Window**. The controls allow users to set roll and heading bias (or offset) values to be a applied to some or all of the swath bathymetry in the **mbnavadjust** project. Under most circumstances, roll bias and pitch bias problems should be resolved before swath data are imported into **mbnavadjust**. However, these settings make it possible to deal with bias issues if they are discovered during the navigation adjustment process. This window includes two toggle buttons to set the bias mode, four slider widgets to set the bias parameters, a button to apply the bias settings to all swath files, a button to apply the bias settings to the current swath file, and a button to dismiss the window.

Same Biases (km)

This toggle button sets the bias mode so that the heading and roll bias is identical for both swath data sections in the current crossing. The <Different Biases> toggle is deselected when this toggle is selected.

Different Biases (km)

This toggle button sets the bias mode so that the heading and roll bias can be different between the two swath data sections in the current crossing. The <Same Biases> toggle is deselected when this toggle is selected.

Section 1 Biases (deg) -> Heading

This slider widget sets the heading bias for swath section 1. The

Section 1 Biases (deg) -> Roll

This slider widget sets the roll bias for swath section 1.

Section 2 Biases (deg) -> Heading

This slider widget sets the heading bias for swath section 2.

Section 2 Biases (deg) -> Roll

This slider widget sets the roll bias for swath section 2.

Apply to All Files

This button sets the current heading and roll bias values as the bias values for all swath files in the **mbnavadjust** project. This button is only enabled when the <Same Biases> toggle is selected.

Apply This button sets the current heading and roll bias values as the bias values for the swath sections in the current crossing. The bathymetry contours and misfit plot are recalculated and redisplayed using the new bias values.

Dismiss

This button closes the **Controls Window**.

USING MBNAVADJUST 1: IMPORTING DATA

Users begin using **mbnavadjust** by starting up the program on the command line:

mbnavadjust

In order to get started, one must either open an existing **mbnavadjust** project or initialize a new one. Both options are available under the <File> menu with the <File->New> and <File->Open> menu buttons. When one pulls down the <File->New> option, a dialog opens that allows navigation of the filesystem and specification of a project name. Whatever project location and name is chosen, a directory "{project_path}/{project_name}.dir" and a file "{project_path}/{project_name}.nvh" will be created. Once a project exists and has been opened, swath data can be imported and then analyzed. To open an existing **mbnavadjust** project, click the <File->Open> option and select the corresponding "*.nvh" file.

The main **mbnavadjust** window displays basic information in a set of labels in the upper left, including the open project name, the number of files imported, the number of crossings found, the number of crossings analyzed, the number of navigation tie points set, and whether an up-to-date inversion for optimal navigation has been performed. A scrollable text window in the lower left displays messages regarding all actions performed by the program during the current session. Another scrollable window on the right displays one of three tables of information according a user selection under the <View> menu. The three choices are a table of the imported swath files, a table of the swath crossings, and a table of the interactively defined navigation tie points. If no swath data has yet been imported, then the all of the tables will be empty. Once some data files are imported, the swath file table will have entries and some number of crossings will be defined (assuming that swaths do overlap and cross), but no tie points will yet be defined.

In order to import swath data into an **mbnavadjust** project, pull down the <File->Import Swath Data> menu item. A file selection dialog will appear. Swath data can be imported in single files or through datalists (see the **MB-System** manual page for a description of recursive datalists). As with other **MB-System** programs, the format id will be automatically determined if possible for each file selected. If a filename does not follow the the **MB-System** naming convention, the user may need to manually set the format id in the appropriately labeled dialog text widget.

Each file that is imported is broken into a number of sections. The maximum size of the sections in line length or in number of soundings is set in a dialog opened by clicking on <Option->Controls>. Navigation control points are selected at regular intervals within each section. The control point distance interval is approximately one tenth of the specified segment length, so there are up to 11 control points for each section. The data for each section is written as a format 71 bathymetry-only file in the "*.dir" directory. As the files are imported, the areal extent of each section is compared to the other sections. Any pair of sections that overlap is added to a list of crossings to be investigated graphically. This list can be displayed by pulling down the <View->Show Crossings> menu item.

As the user later works through the crossings, he or she will will define tie points whenever the overlapping bathymetry allows the navigational offset to be determined reliably in three dimensions. Each tie point will reference two of the navigation control points, one from each swath in the crossing. Ultimately, some crossings will allow one, or in some case multiple tie points to be defined. Other crossings will still have no tie points, either because the swaths don't really overlap or because there aren't any distinctive features to match. When all of the crossings have been inspected, then the user can invoke inversion for an optimal navigation solution. In cases where the data are known to be already corrected for vertical offsets, such as tides, then the user can uniformly set the vertical (z) components of offsets to zero.

USING MBNAVADJUST 2: INSPECTING CROSSINGS

Bringing Up The Nav Error Window

The heart of **mbnavadjust** is the interactive inspection of the swath crossings. One can bring up the crossing inspection window (entitled "Nav Error") in a number of ways. If one pulls down the menu option <Action->Analyze Crossings>, then the "Nav Error" window will come up with the first crossing loaded. Alternatively, if the <View->Show Crossings> menu item has been selected so that a table of crossings is displayed, clicking once on any of the crossing items in this tabel will bring up the "Nav Error" window with that crossing loaded. Similarly, if tie points have been established and the tie point table displayed by selecting the <View->Show Ties> menu item, then clicking on any of the tie items this table will bring up the "Nav Error" window and load the crossing that includes the selected tie point. If the "Nav Error" window is already displayed, clicking on crossing or tie items in the display tables will load the selected crossing in place of whatever crossing was previously shown.

Once the "Nav Err" window is displayed, the user can also move through the crossings by clicking on the <Previous>, <Next>, and <Next Unset> buttons. The <Previous> and <Next> buttons will load the previous or next, respectively, crossings in the crossing list. As discussed below, each crossing must ultimately be "resolved" by either having one or more navigation offsets set at particular "tie points", or by being "skipped" because no matchable seafloor features are found. The <Next Unset> button will load the next crossing that has not been resolved. To close the "Nav Err" window, click the <Dismiss> button.

Interactively Matching Seafloor Features

The "Nav Error" window is complicated in appearance, and regrettably complicated in function also. The purpose is to allow the user to determine if any seafloor features can be confidently matched in the overlapping swaths. If so, one or more tie points can be defined. In order to ease the identification of matching features, two simultaneous displays are provided. The larger plot on the right consists of overlain bathymetric contours derived from each of the two swaths. The smaller canvas on the left shows a color two dimensional plot of the RMS misfit between the two swaths. The misfit is shown as a function of relative lateral offsets between the two swaths. Put another way, the misfit plot shows how good, or bad, the misfit becomes as one moves one swath relative to another. The lowest misfit values are shown in red; higher misfits are shown in blues to purples. The location of the global (three dimensional) minimum misfit is marked by the large black "X", the location of the minimum misfit using the current vertical offset value is marked by a small black "x", and the location of the current navigation offset is shown by a small red square with a black outline.

The interactive aspect of the "Nav Error" window works simultaneously in both displays. If the user holds down the left mouse button in the contour plot and moves the cursor, the bathymetry contours from one swath will move along with the cursor. In this way, the user can move one of the swaths around relative to the other until the contours line up and features match. As the contours move, the red square showing the current offset location also moves on the misfit plot. The user can thus visually relate the contour matching to the misfit function. The combination of these two displays greatly improves a user's ability to reliably determine navigational offsets (and to conclude where navigational offsets cannot be determined).

The "Nav Err" window includes two buttons that are particularly useful during efforts to match seafloor features. The <Minimum Misfit> button below the misfit display will cause **mbnavadjust** to set the current navigational offset to that associated with the smallest misfit for the current misfit display. This button is often used first to get close to the right offsets. The <Zero Offset> button above the contour display will return both displays to a state of zero navigational offset.

The relationship between the contoured bathymetry and the misfit plot is usually quite clear. If a strongly matching seafloor feature exists, then a distinct minimum will show up in the misfit plot. If the navigation is good and the feature already matches, then the misfit minimum will be located at the center of the plot, corresponding to an offset that is zero distance in both the east-west and north-south directions. If the navigation is bad, then the misfit minimum will be offset from the origin, and the offset vector will correspond to how far and what direction one must move one swath so that the features in both swaths match. In cases where there is no distinctive seafloor feature to match, the misfit plot will not display a strong minimum and it will be impossible to determine the relative navigational offset. Alternatively, the existence of multiple similar features can produce multiple local minima in the misfit map. In this case, the ambiguity between multiple possible solutions prevents the determination of the navigational offset. We have found that combining both contour and misfit displays allows interactive, visually based decision making that is more generally reliable than any automated scheme we can devise.

Navigational offsets can only be used when they are associated with specific points on the overlapping swath navigation. These points are called "tie points". All crossings will begin with no tie points, and users can generate one or more tie points for any crossing as required. The creation and manipulation of tie points is discussed in detail in a later section.

It is also important to understand that any apparent navigation offset observed in the contour and misfit plots is relative. It may turn out that both swaths are poorly navigated and have to be moved, or that all of the offset can be applied to one swath or the other. The set of decisions about how to distribute the relative navigation offsets among the affected swath files intrinsically involves speed and acceleration changes. Fortunately, we are able to formulate the automated inversion process discussed below to obtain an optimal navigation solution.

Display Controls

The user controls the appearance of the bathymetry contour plot. The contours are generated at regular intervals in depth, and also change color and are annotated with downhill facing tickmarks at regular intervals. A controls dialog brought up by clicking on the <Settings->Contours> button allows the user to set the contour, color change, and tickmark intervals. This same dialog also sets a decimation parameter that causes the contours to be calculated from fewer soundings (the data are decimated by ignoring pings). The application of decimation may speed up the crossing loadings, but is not generally recommended unless the bathymetry is strongly oversampled.

Users may also use a "zoom" feature to focus on small areas in the contour plot. The center mouse button is used to drag a box over a region of interest in the contour plot. When the center button is released, both the contour and misfit plots are regenerated to show the smaller area. Users may zoom as many times in succession as desired. One cannot undo the individual zoom events, but clicking the <Full Size> button in the "Nav Err" window will cause the plots to show the original, full area covered by the two swaths in the current crossing.

The misfit plot represents lateral offsets scaled according to the current contour plot display. Specifically, the width and height of the misfit plot correspond to one half the width and height of the bathymetry contour plot. When the bathymetry plot area changes due to a zoom event, the misfit is recalculated and redisplayed centered around the current offset. The color map used for the misfit display is automatically scaled according to the minimum and maximum misfit values.

Setting Crossing Ties and Offsets

In order to actually set the relative navigational offset between two particular points on overlapping or crossing swaths, the user must first create a tie point. This is accomplished by clicking on the <Add Tie> button in the "Nav Err" window. Once a tie point exists, it will be shown on the contour plot as two yellow or red-filled, black outlined squares connected by a thin black line. Each of the squares is located along one of the swath navigation tracks, and represents one of the navigation control points defined during data importation. There can be multiple tie points for each crossing, and each one is created by clicking on the <Add Tie> button.

Only one tie point will be active at any time. The active tie point is displayed with larger boxes (the inactive tie points are only 1/4 the size. If only one tie point has been defined, it will always be active. If more than one tie point exists, clicking on the <Select Tie> button in the "Nav Err" window will change the active tie point to the next in the list for the displayed crossing (the tie points are selected in the order in which they were created). If the user wishes to delete a tie point, then click <Select Tie> until the undesirable tie point is active, and then click <Delete Tie>. The active tie point will be displayed in red fill if either the associated navigation control points or the navigational offset have been changed since it was created or last set. If the active tie point is up-to-date, then it will be displayed in yellow fill. Each new tie point is initially displayed in yellow. All inactive tie points will be shown in yellow fill.

When viewing a crossing with one or more tie points, the offsets displayed are associated with the current tie point. As one moves the contours to match overlapping features, it is important to remember that the navigational offset will be applied to the navigation points indicated for the current tie. Thus, the feature being matched should derive from data (soundings) associated with those particular points on the shiptrack. This is accomplished by clicking on the bathymetric feature in the contour display with the right mouse button. The "right-click" causes **mbnavadjust** to find the soundings from each swath that are closest to the point clicked, and then to shift the current tie point to the navigation points on each swath that include these soundings (that is, the navigation points associated with the sonar pings that include the closest soundings).

In order to set, and save, a navigational offset that causes a bathymetric feature to be matched in overlapping swaths, the user must click the <Set Offset> button above the contour display. If a user changes the active tie point or loads a different crossing without clicking <Set Tie>, then no offset information will be saved. Conversely, for any crossing with one or more tie points, the <Reset Offset> button will reset the navigational offset to the last value set for the current crossing and tie point.

So, in practice, setting navigational offsets that will be used in obtaining an optimal navigation solution involves the following steps:

1. Identify a bathymetric feature with overlapping data so that it can be matched.
2. Create a tie point by clicking the <Add Tie> button.
3. Set the tie point location by right -clicking on the feature.
4. If necessary, zoom the display to focus on the bathymetric feature of interest by dragging a box with the middle mouse button.
5. Adjust the offsets so that the feature is matched in both swaths (operating in the contour display, the misfit display, or both).
6. Click the <Set Offset> button.

These steps should be followed for every feature that can be matched in overlapping swaths.

In some cases, the user will find it useful to create and set multiple tie points in a single crossing. Other crossings may not allow any features to be matched. Users should click the <Skip Crossing> button on crossings that do not allow one or more offsets to be determined. In fact, **mbnavadjust** will only allow the calculation of a navigation solution when all of the crossings have been acted on by either having tie points set or by having been explicitly skipped.

Users should feel free to iterate any way they like during crossing inspection. Crossings may be displayed as many times as desired, and ties can be created, deleted, and changed without restriction. Users may also quit **mbnavadjust** and then later reopen the same project without losing any information.

Setting Good, Poor and Fixed Navigation

The adjusted navigation model produced by **MBnavadjust** should be accurate to the bathymetric resolution in a relative sense, but fitting a set of relative offset ties does not provide constraints on the global location of the survey data. **MBnavadjust** provides two means to control the global location of the adjusted navigation. First, the global location of the model is essentially an average of the overall offsets associated with good navigation. Users may use the <Action->Set File Poor Navigation> menu item to set selected surveys or files to be ignored in setting the global model. Second, if certain data files are thought to have correct navigation, they can be fixed (e.g. to have zero adjustment) using the <Action->Set File Fixed Navigation> menu item. In this case, all of the non-fixed files are adjusted relative to the fixed files.

USING MBNAVADJUST 3: INVERTING FOR AN OPTIMAL NAVIGATION SOLUTION

Performing the Inversion

Once the user has interactively analyzed all of the crossings and closed the "Nav Err" window by clicking the <Dismiss> button, the <Action->Invert navigation> button becomes enabled. Clicking this button causes **mbnavadjust** to construct and solve an inversion for the optimal navigation.

The inversion solves for navigation adjustments at each navigation control point which satisfy the offsets at the tie points while minimizing speed and acceleration. The speed and acceleration penalty is set using a penalty weight value that may be varied. If the penalty weight is large, the navigation adjustments may be smooth and small but not fit the tie point offsets well. An infinite penalty weight produces uniformly zero adjustments. In contrast, a small penalty weight allows the tie point offsets to be fit as well as possible even if large speed and acceleration spikes are a consequence. Hoever, even with a zero penalty weight the inversion may not be able to exactly satisfy all of the tie point offsets. If some of the tie point offsets are conflicting (e.g. one tie point requires a navigation control point to move to the west while another tie point requires a move to the east), then the offsets cannot all be simultaneously fit exactly.

The inversion is actually performed many times with different penalty weights, and the "best" solution is selected and applied to the data. The details of how the "best" solution is identified are given in the section "Details of the Inversion" section below. A log of the inversion parameters, the results from each of the inversion iterations, and detailed results from the final inversion are output to the Message text window. The program then outputs an adjusted navigation file for each of the input swath files and updates (or creates) the parameter file for each swath file so that **mbprocess** will merge the adjusted navigation.

The output adjusted navigation files are named by adding a ".na#" suffix to the original swath data filename. If a swath file imported into **mbnavadjust** for the first time is named:

    mbari_1998_55.mb57

then the adjusted navigation resulting from that project will be named:

    mbari_1998_55.mb57.na0

If this file is imported into a second **mbnavadjust** project, the result from inversion in that project will be:

    mbari_1998_55.mb57.na1

In addition to generating the adjusted navigation, **mbnavadjust** also sets the NAVADJMODE and NAVADJFILE values in the **mbprocess** parameter file. In this case, the parameter file is named:

    mbari_1998_55.mb57.par

and the processed swath file generated by running **mbprocess** is:

    mbari_1998_55p.mb57

Refer to the **mbprocess** and **mbset** manual pages for details on the control and use of **mbprocess**.

Note that the relevant parameter file settings will reflect the most recent inversion in **mbnavadjust**. Users do need to be aware that the order of navigation processing is important because, when possible, **mbnavadjust** imports existing processed data files. The data within the **mbnavadjust** projects are not, however, updated when the source data are updated. Consequently, standard navigation processing should be completed and applied with **mbprocess** before a swath file is imported into an **mbnavadjust** project. If a swath file is to be used in multiple **mbnavadjust** projects, the first project should be finalized and the results applied before data are imported into the second.

Fine-Tuning the Inversion

Once an inversion has been performed, the user should inspect the fit for each of the tie points before accepting and applying the adjusted navigation. The relationship between the interactively defined navigation offsets and the offsets associated with the inversion can be investigated numerically in the tie points table or visually in the "Nav Err" window.

We suggest first examining the tie points table by pulling down the <View->Show Ties> menu item. This table shows, from left to right, the identity of each tie point, the longitude and latitude offsets defined by the user (in meters), and the longitude and latitude residuals, or differences between these offsets and those associated with the inversion (in meters). If any of these residuals are unexpectedly large, simply clicking on the table line showing the suspect navigation tie will bring up the "Nav Err" window and load the crossing including that navigation tie. Once an inversion is performed, the "Nav Err" crossing displays show the inverted offset as a small '+' symbol on the misfit plot. The user can then determine whether the previously set navigation offset is truly required by the data. On occasion, one discovers that the offset obtained in the inversion is as consistent with the bathymetry as the offset originally set by the user. Once the offset values have been adjusted as necessary, they can be reset by clicking on the <Set Offset> button (just as in the earlier interactive sessions).

Once all of the suspect navigation ties have been inspected, and perhaps corrected, another inversion can be generated using the revised set of offsets. In this way, users can iterate over cycles of inversion and inspection until a satisfactory (self-consistemt) solution is obtained. When the final inversion has been performed, the user can then run **mbprocess** on all of the affected swath data files to produce a set of processed files incorporating the optimally adjusted navigation. Once again, we emphasize that users should always finalize an **mbnavadjust** project before importing affected swath data into another **mbnavadjust** project.

Details of the Inversion

The inversion is constructed as a sparse overdetermined least squares matrix problem. Suppose we have N navigation control points in all of the swath files and have defined M tie points. The form of the problem is:

A X = D

Here **X** is the vector of unknowns, which happen to be the changes in the longitude and latitude values of all of the navigation control points. So, there are 2N unknowns. Note that we do not solve directly for longitude and latitude, but rather for the change, or perturbation, to the longitude and latitude values. The matrix **A** contains 2N columns corresponding to the unknowns and a row for each of the constraints we can apply to the navigation adjustment problem. The number of elements in the "data" vector **D** also corresponds to the number of constraints. We apply four sets of constraints in this inverse problem:

1) Fixed navigation points
2) Penalize speed (first derivative)
3) Penalize acceleration (second derivative)
4) Fit navigation offsets at tie points

The first kind of constraint is simply expressed as:

XLONj = 0

XLATj = 0

where XLONj is the longitude change and XLATj is the latitude change for the "j"th navigation control point.

The second contraint (speed) is also one of minimization:

```
-XLONj + XLONj+1
----------------  = 0
   -Tj + Tj+1

-XLATj + XLATj+1
----------------  = 0
   -Tj + Tj+1
```

Here XLONj+1 and XLATj+1 are the longitude and latitude changes for the "j+1"th navigational control point and Tj and Tj+1 are the times of the "j"th and "j+1"th navigational control points. The denominator in these expressions is thus the time difference between the two navigation points. The speed constraint can only be applied to navigation control points that are sequential, and is not applied across breaks in the swath data. Note that multiple swath files may be sequential without breaks, while time gaps or breaks can occur within a single swath file. The existence of gaps or breaks in the swath data is determined solely on the basis of time gaps as the data are imported.

The third contraint (acceleration) is also one of minimization:

```
XLONj - 2 * XLONj+1 + XLONj+2
----------------------------     = 0
   -Tj + Tj+2
```

$$\frac{XLAT_j - 2 * XLAT_{j+1} + XLAT_{j+2}}{-T_j + T_{j+2}} = 0$$

The second derivative calculation requires three sequential navigation control points: j, j+1, and j+2. Here XLONj+2 and XLATj+2 are the longitude and latitude changes for the "j+2"th navigational control point and Tj and Tj+2 are the times of the "j"th and "j+2"th navigational control points. The denominator in these expressions is thus the time difference between the "j"th and "j+1"th navigation points.

The final, and most important constraints are the relative navigation ofsets defined for each of the M tie points. Since each offset has a longitude and a latitude value, there are 2M rows in **A** and elements in **D** associated with the tie points. If the "i"th tie point specifies an offset DLONi and DLATi between the "j1"th and "j2"th navigation control points, then the constraint may be expressed as:

$$-XLON_{j1} + XLON_{j2} = DLON_i$$

$$-XLAT_{j1} + XLAT_{j2} = DLAT_i$$

The size of the matrix problem will vary with the number of navigation control points, tie points, fixed points, and time gaps. However, the addition of the speed and acceleration minimization constraints guarentees that the number of constraints will always be larger than the number of unknowns, and so we will always be solving an overdetermined least squares problem. Each of the above equations contribute one row to the matrix problem, and each of these rows has at most three nonzero elements in **A**. As a result, this matrix problem is also always extremely sparse. This condition allows us to use one of a class of approximate least squares solution algorithms that are efficient in solving sparse problems. The algorithm used for **mbnavadjust** inversions is that of Olsen [1987].

The importance of the speed and acceleration minimization constraints is varied by multiplying the associated matrix row elements by a penalty weight value. In practice, the inversion is constructed and solved with many different penalty weights, and the "best" inversion is selected and applied. Generally speaking, we seek the smoothest inversion that satisfactorily fits the tie point offsets. We set smoothness using the penalty weight value so that larger penalty weights correspond to smoother solutions. We measure the fit to the tie point offsets using the usual least squares calculation:

$$Fit^2 = Chi^2 = SUM((DLON_i - (XLON_{j2} - XLON_{j1}))^2 + (DLAT_i - (XLAT_{j2} - XLAT_{j1}))^2)$$

using the same notation as above. Note that the units of Chi are distance, and so are physically meaningful. A smaller Chi corresponds to a better fit to the tie point offsets. The Chi value will be smallest for a zero penalty weight, and increase as more smoothing is applied.

The initial solution is generated using a very small penalty weight to insure that the smoothing is negligible and that the tie point offsets are fit to the maximum degree possible. This first inversion is used to set a "reference" value of Chi. In some cases, the tie points offsets do not substantially conflict and it is possible for the inversion to fit the offsets nearly exactly so that Chi is quite small. If the initial Chi is greater than the target precision for the inversion (set from the Controls panel), **mbnavadjust** sets the reference fit to this initial value. Otherwise, the reference fit is set to the target precision. The default value is 0.1 meters, which is appropriate for high resolution, low-altitude surveys. Larger values will be more appropriate for large altitude (e.g. deep water hull-mounted) surveys. The program then recalculates solutions with different penalty weights until the resulting Chi is between 1.05 and 1.1 times the reference fit. This last solution is chosen as the best solution, reported as the navigation adjustment inversion solution, and applied to that swath data navigation. The justification for choosing the optimal solution in this fashion is that this approach insures that the smoothness constraint is sufficient large to be impacting the fit, but not so large that the fit has been substantially degraded.

REFERENCES

Nishimura, C. E., and D. W. Forsyth, Improvements in navigation using SeaBeam crossing errors, *Mar. Geophys. Res.*, **9**, 333-352, 1988.
Olson, A. H., A Chebyshev condition for accelerating convergence of iterative tomographic methods - Solving large least squares problems, Phys. Earth Planet. Inter., 47, 333-345, 1987.
Parker, R. L., **Geophysical Inverse Theory**, Princeton University Press, Princeton, NJ, 1994.

SEE ALSO

mbsystem(l), **mbio**(l), **mbprocess**(l), **mbnavedit**(l), **mbset**(l)

BUGS

It used to be too new to be bulletproof, now its too complicated to be bulletproof. Good luck.