A Manual for MLOC,
A Program for Calibrated Multiple Event Relocation

This document reproduces the content, as of July 22, 2020, of the on-line documentation for the multiple-event relocation code MLOC, which is hosted at:

https://www.seismo.com/mloc/

Hypertext links in this document will go to the website. The website will always be the most up-to-date source of documentation for **mloc**.

The conversion from web pages to a single document presents some significant problems in terms of organization and formatting. I have endeavored to honor the original web content and organization to a high degree. The reader will recognize cases where this has led to repetition or peculiar phraseology and construction.

Eric Bergman
Global Seismological Services
1900 19th St., Golden, Colorado 80401
(720) 400-7835
bergman@seismo.com
https://www.seismo.com/

September 21, 2020

Table of Contents

MLOC, A Program for Calibrated Multiple Event Earthquake Relocation

This section of the GSS website deals with the installation and use of a program called **mloc** (sometimes **MLOC**) which implements the Hypocentroidal Decomposition (HD) method of multiple event earthquake relocation that was introduced in Jordan and Sverdrup (1981). The program is most commonly employed in the analysis of arrival time data from natural earthquakes but has also been used successfully on induced earthquakes and man-made explosions, including nuclear tests. It has been extensively developed for research on what are referred to as calibrated earthquake locations, which was not contemplated in the original publication. Calibrated locations arise from the use of procedures to minimize the biasing effect of unknown Earth structure and to more accurately estimate the uncertainties of the arrival time data, which leads to more reliable identification and rejection of outlier readings and more accurate estimates of uncertainties for derived hypocentral parameters.

Although the main purpose of **mloc** is to determine calibrated locations of a cluster of seismic events it can be used for traditional uncalibrated locations of single events or clusters based on minimizing the travel-time residuals of regional and/or teleseismic phases.

mloc is free. The entire source code for **mloc** is downloadable from this website, along with source code for several useful utilities, documentation, necessary data files, and some sample datasets. The navigation menu on the right sidebar of each page in the **mloc** section will help you to find what you want. The current release can be found here.

The code is written completely in Fortran; it uses some features from Fortran95 but nothing more recent than that. The program can be compiled and built with any one of several Fortran compilers running under macOS (formerly known as OS X) or most Unix-like operating systems, including Linux. **mloc** runs in a Terminal window.

mloc uses the Generic Mapping Tools (GMT) to create graphic output, so a recent version (preferably GMT6, but GMT5 works) is required. An installation of Google Earth is highly recommended as well, as one of the output files is a KML file of the relocated earthquakes. Finally, an installation of the tau-p software (Buland and Chapman, 1983) that implements the ak135 global travel-time model (Engdahl et al, 1998) may be required if the binary data files distributed with **mloc** do not work in your system.

Why Would I Want to Use mloc?

There are several other readily-available programs for doing multiple event relocation. Double Difference (DD) (Waldhauser and Ellsworth, 2000) has been widely used for 20 years. DD was originally specialized to exploit arrival time differences determined from waveform cross-correlation to obtain very high resolution hypocenters in the local distance range but has since been generalized to employ traditional arrival time readings, and for the full range of epicentral distances. BayesLoc is a recent entry into the field with an unusual and powerful set of features emphasizing a Bayesian approach to estimation of nearly every parameter in the relocation

problem. Older programs such as Jim Dewey's JHD and Gary Pavlis' PMEL are still available and they are quite suitable for certain types of relocation studies. Bill Rodi's GMEL program implements a powerful (but computationally intense) grid-search approach to the problem.

In comparison to these programs **mloc's** strength is in its specialization for pursuing calibrated hypocenters. The hypocentroidal decomposition algorithm on which it is based is particularly well-suited to the problem because it naturally and rigorously separates the relocation problem into two parts, the relative locations of events in a cluster and the absolute location of the cluster, allowing the user to control the datasets and weighting schemes most suitable for each part of the analysis.

Beyond that, no other relocation code (to my knowledge) provides the capability to determine empirical reading errors from the data itself and use those values for identifying outliers and weighting data for the inversion. **mloc** is particularly flexible in working with arrival time datasets compiled from different sources. Although it was developed primarily to work with traditional arrival time data, it can incorporate differential time data from cross-correlation analyses, as well as differential phase data (S-P and relative depth phases, e.g., pP-P). Based on relative depth phases **mloc** implements a novel and powerful method of analyzing teleseismic depth phases for constraint of focal depth.

Finally, **mloc** is designed to give the user a great deal of control over how a relocation analysis is done and to provide access to extensive information to judge the effects of different choices in approach, making it an excellent teaching tool for the subtleties of earthquake location, especially for users who already have some experience in this kind of analysis.

Disclaimer

mloc was not developed to be a black box tool that can be widely distributed to students, or even to post-graduate researchers with a casual knowledge of earthquake location and an interest in getting some quick results. Some aspects of its use are based on considerations that are rarely if ever found in other earthquake location codes. The analysis for calibrated locations typically requires many runs with carefully-considered editing of input files in between. Different datasets often require different approaches. **mloc** is adaptable to a wide variety of cases and as a result has tools that are only applicable in specific circumstances; they may conflict with other tools in other circumstances. In other words, successful use of **mloc** will require very careful review of the documentation, and ideally some guidance by someone who is familiar with it. If you are serious about using it for research you may want to consider attending a training course.

Development History

Development of **mloc** began in 1989 when I was a young post-doctoral researcher at the Massachusetts Institute of Technology. Tom Jordan, who had recently arrived at MIT as a faculty member, suggested looking into the HD algorithm as a research tool. Tom did not consider the computer code used for the 1981 paper to be a suitable basis for development, so **mloc** was built on the code base of an advanced single-event location program called LOC that had been written

in the early 1980s by Ken Creager. In its first incarnation **mloc** was, like other multiple event relocation codes, oriented to obtaining improved relative locations of seismic events, but not calibrated locations.

The next phase of development of **mloc** began in the late 1990s, when the application of **mloc** to the challenge of obtaining calibrated locations began in close collaboration with Bob Engdahl. At the time there was strong interest in (and funding for) research on "ground truth" seismic locations in support of the Comprehensive Test Ban Treaty. During this period, many code segments from Engdahl's single event location code (the basis for the EHB Catalog) migrated to **mloc** and many detailed comparisons of intermediate results between the two codes were made to ensure correct processing. The code that calculates travel times through custom crustal velocity models in **mloc** was borrowed from Johannes Schweitzer's HYPOSAT program during this phase.

Since about 2013 development of **mloc** has branched away from the EHB code, especially with the adoption of a new standard data format (MNF) that better supports the procedures used for calibrated relocations. Another major area of development has been the refinement of methodologies to more reliably constrain focal depths.

Development of **mloc** has been supported over the years by a number of research contracts and grants provided by several agencies of the U.S. Government, including the U.S. Geological Survey, the Department of Energy and the Air Force Research Laboratory. Much of the development was done in close collaboration with Bob Engdahl when we were both working in the Center for Imaging the Earth's Interior at the University of Colorado, under the watchful eye of Mike Ritzwoller. A partial list of contracts and grants that supported the development of **mloc** can be found here.

Acknowledgements

Many persons have contributed to the development of **mloc**. Tom Jordan deserves primary credit for setting me off on this particular research path and providing an algorithmic framework that proved to have enormous potential, but Bob Engdahl comes a close second for years of almost daily discussion on the realities and subtleties of earthquake location that don't appear in any textbook. István Bondár came up with the clever idea (Reciprocal Cluster Analysis, Bondar et al., 2008) that was the inspiration for the direct calibration method that is the standard approach for calibrated relocation with **mloc** today. Steve Myers, Reza Ghods, and Ezgi Karasözen all contributed to the development or refinement of important features of **mloc**. Bill Rodi has graciously shared his deep understanding of applied statistics and earthquake location on a number of occasions. Harley Benz has been a strong supporter of **mloc's** development with the vision that this kind of analysis can find its way into routine processing at the NEIC. It is unlikely that I would have bothered to spend so much of my professional life developing **mloc** if the International Seismological Centre did not exist. The ISC Bulletin is a gold mine of data for **mloc** and has nearly always been the main source of arrival time data for the published studies based on **mloc**.

Hypocentroidal Decomposition

mloc is based on the Hypocentroidal Decomposition (HD) method for multiple event relocation introduced by Jordan and Sverdrup (1981). The basic algorithm is completely described in that reference. A PDF of the paper is included in the mloc distribution in the /documents directory. The essence of the HD algorithm is the use of orthogonal projection operators to separate the relocation problem into two parts:

- The **cluster vectors**, which describe the relative locations in space and time of each event in the cluster. They are defined in kilometers and seconds, relative to the current position of the hypocentroid.
- The **hypocentroid**, which is defined as the centroid of the current locations of the cluster events. It is defined in geographic coordinates and Coordinated Universal Time (UTC).

Both methods of location calibration implemented in **mloc** depend fundamentally on this decomposition of the relocation problem. Similar approaches could conceivably be implemented in other multiple event relocation algorithms but it seems likely to be considerably more difficult than it is with hypocentroidal decomposition.

The cluster vectors are defined only in relation to the hypocentroid. The hypocentroid can be thought of as a virtual event with geographic coordinates and origin time in UTC. The orthogonal projection operators act on the data set of arrival times to produce a data set that includes only data that actually bears on the relative location of cluster events, i.e., multiple reports of a given seismic phase at the same station for two or more events in the cluster.

The hypocentroid is located very much as an earthquake would be, except that the data are drawn from all the cluster events. Thus it is typical for the hypocentroid to be determined by many thousands of readings. Nevertheless, the hypocentroid is subject to unknown bias because the theoretical travel times (typically ak135) do not fully account for the three-dimensional velocity structure of the Earth. Geographic locations for the cluster events are found by adding the cluster vectors to the hypocentroid.

The HD method works iteratively. At each iteration, two inversions are performed, first for the cluster vectors relative to the current hypocentroid, then for an improved hypocentroid. The cluster vectors are added to the new hypocentroid to obtain updated absolute coordinates for each event. The convergence criteria are based on the change in relative location of each event (0.5 km) and the change in the hypocentroid (0.005°). The convergence limits for origin time and depth, for cluster vectors and hypocentroid, are 0.1 s and 0.5 km, respectively. Convergence is normally reached in 2 or 3 iterations.

The data sets used for the two problems need not be (and usually are not) the same. Because the inverse problem for changes in cluster vectors is based solely on arrival time differences, baseline errors in the theoretical travel times drop out and it is desirable to use all available phases at all distances outside the immediate source region. For the hypocentroid, baseline errors in theoretical travel times are more important and one may wish to limit the data set to a phase

set, e.g., teleseismic P arrivals in the range 30-90°, to achieve a more stable (but uncalibrated) result. The choice of data set for determining the hypocentroid has great importance in the direct calibration method.

Similarly, weighting schemes are different for the two inversions (cluster vectors and hypocentroid) that comprise HD, reflecting the different natures of the two problems. Empirical reading errors for each station-phase pair are used in weighting data for estimating both the hypocentroid and cluster vectors, but the uncertainty of the theoretical travel times, which are estimated empirically for each phase from the residuals of previous runs, is relevant only to the hypocentroid.

The HD algorithm as developed by Jordan and Sverdrup (1981) is used only to obtain improved relative locations for the cluster events, with a geographic location for the cluster as a whole (the hypocentroid) that is biased to an unknown degree by unmodeled Earth structure that has been convolved with the (typically) unbalanced geographic distribution of reporting seismic stations. The capabilities of **mloc** go far beyond this, including especially the tools implemented for location calibration which attempts to minimize this bias.

Distribution

This page contains the latest distribution of the complete **mloc** package. The archive unpacks into a directory structure containing all necessary executables, source codes, compiler directions, and data files. The distribution is served in a zipped archive. The */mloc_distribution/* directory structure contained in the archive may be installed anywhere in the user's file system, but it is *strongly* recommended that none of the directory or file names be changed until and unless the user has a thorough understanding of the **mloc** ecosystem.

Current Version

The current version of **mloc** is v10.5.1 with a release date of August 6, 2020. The file size is 27.4 MB.

Download

Change Log

The full version history of development of **mloc** is contained in the text file *mloc_version_history.txt* in the directory *mloc_directory/mloc_src/*. Recent entries are reproduced here, in reverse chronological order, along with any notes about changes to other elements of the distribution.

2020/8/6: Some new output in ~.depth_phases to help analysis.

2020/8/3/: Fixed a bug in subroutine tt_rdp_summary (mlocout_gmt.f90) related to the recent rewrite of that routine. It was caused by zero-depth events, just a problem finding the plot boundaries. I also made some changes to the logging under command vlog to reduce the size of the output file and make it easier to find stuff that's likely to be of importance.

2020/7/30: I rewrote the code (in mloclib_tt.f90) related to doing analysis of relative depth phases to handle focal depths over the full range 0-760 km. I also rewrote the plotting codes that deal with focal depths to handle the full depth range.

2020/7/18: I upgraded my gfortran from 4.9.2 to 9.2.0 and was presented with quite a few warnings and a few errors. The most serious ones were ancient code structures in libtau.f90, such as "arithmetic if", "computed go to" and having a do loop end on an executable statement are now causing compiler errors. In the process of fixing those I went through the entire module and cleaned up the coding style a bit so it's more readable now. I tested to make sure the new version of libtau.f90 yields the same results in a cluster as before, using both the Absoft and gfortran compiler versions. After working so much on the tau-p code I added the command "tlog" to carry out extensive logging of the tau-p calculations; it would only be needed for testing and debugging the tau-p code (libtau.f90). I also removed the "libtau.inc" include file from the project. It was left over from an earlier attempt to simplify the tau-p code, and was not actually used anywhere. Using the flags in the standard makefile for mloc, gfortran v9.2.0 now issues no warnings or errors. In late testing I discovered a bug in the routine topo_corr in mloclib_tt.f90

that screwed up the topographic correction for events in the ocean. It would return zero correction for both crust and water layer.

2020/7/14: A new user reported compile errors under gfortran (v10) for two routines in mloclib.f90 that simply change the case of a string to lower case or upper case. The problem was related to the use of "pure function" call, which is not necessary. However, neither routine is actually called so I simply removed them. However this raised the issue of mloc not being entirely compatible with recent versions of gfortran. I've been using a much older version (4.9.2) from around 2013.

2020/6/30: Fixed a couple bugs related to the routine "check_station" in mloclib_stations.f90. The stations that moved without changing station code were not being handled correctly.

2020/6/27 (v10.5.1): I removed several commands that have become obsolete: cvou, fplt, mdou, mech, puke. I added a command "ngmt" to be able to turn off all plotting in a case where GMT is not available.

2020/6/24: I added some logic in mloc.f90 to handle the case of a missing mloc.conf file more gracefully, and also added a mechanism to try to ensure that new users edit the sample file before running mloc. There's a new keyword "SAMPLE" in the first line of the example file, with a reminder to edit the other keywords. mloc won't run unless that line is deleted.

2020/4/22: Getting a segmentation fault with a cluster that had a lot of events with stations right on top of them. I think it was the section in subroutine delaz (mloclib_geog.f90) that handled zero epicentral distance. I was setting delta=0. when it got to be less than about 50 m. Setting the distance to a very small number (but non-zero) fixed the problem.

2020/4/10: In the Iwaki cluster I ran into a bug in mlocout_ttsprd.f90 where the number of P readings exceeded the hard-wired limit of 30,000. This didn't break anything but produced a large number of warnings. I just set the limit of those arrays to be equal to the parameter that sets the limit for number of readings used (ntmax1).

2019/12/18: I changed the functioning of the RADF command, so that reading (and using) agency and deployment fields to resolve station code conflicts is restricted to specified station codes, not universal. In practice there are seldom more than a couple cases in any given cluster where agency and deployment codes are needed. Making sure that the entire dataset had correct specification of agency and deployment (to match whatever was in the station files) when you only need it for a couple readings from a single station was a nightmare. In fact, the code now uses the agency and deployment fields along with the station code for every comparison, but in most cases the fields are simply blank. RADF just specifies that those fields, both for station files and phase readings, will actually be read for certain station codes.

2019/12/5 (v10.5.0): Major changes in plotting, related to the adoption of GMT6, which has a nice new option for handling plotting of topography using the "@earth_relief_rru" option in grdcut. I was also running into trouble with the old .grd files when using GMTv6. Some of the GLOBE tiles were ill-constituted and are now breaking stricter requirements. Rather than try to

fix my old .grd files I decided to abandon GLOBE and GINA as options and use ETOPO1 (the "01m" option from the GMT server) as the sole option for dem1. The new system also supports high-rez DEMs from SRTM but I have not yet experimented with that. ETOPO1 has a little less resolution than GLOBE but it still works well for basic plotting, and it will be easier to support since I don't need to serve the data file. The argument to the command "dem1" is simply "on" or "off" now. It appears that the current plotting code will still work in GMT5 (with one exception) so that is still permitted. The exception is the call to grdcut, and for that there is a special loop for GMT5 there, referencing the old ETOPO1 file in /tables/gmt/dem/ETOPO. I also fixed a long-time annoyance with the .kml file. The symbols only displayed correctly if the directory is still in the working directory, so it could find the image files in the tables/kml directory. Now each cluster directory acquires a directory called "_kml" containing the necessary images and the .kml file refers to those, so it displays correctly as long as the .kml file is in the same folder with the _kml directory.

Operating System

Recent development and most usage of **mloc** has been done under Apple's Mac OS X and MacOS operating systems, but it has been installed under several Linux distributions successfully, and should be rather easy to port to any Unix-like operating system. **MLOC** is compiled from Fortran source code, runs in a terminal window and uses only a few basic shell commands. All plotting is done through the Generic Mapping Tools (GMT) and plotting can be turned off by the user if GMT is not available.

Hardware Requirements

Disk storage requirements for the program itself, supporting data files and utility programs are modest. The basic distribution will require ~135 MB on disk. If managed by GMT (version 6) the ETOPO1 digital elevation model requires about 270 MB of storage; if the user hosts the corresponding ~.grd file in order to plot topography/bathymetry under GMT5, the storage requirement is ~1 GB. The data files for a typical cluster of ~100 events are unlikely to exceed a few MB in size, but each run of **mloc** produces many output files, some of which are fairly large. If all files from all runs are retained (recommended practice), a cluster directory can easily expand to several hundred MB. On the other hand those files compress well.

Demands on RAM and CPU power go up quickly with the number of events included in a cluster. Current desktop and laptop machines with reasonably high specifications (e.g. a mid-range MacBook Pro) can handle clusters with up to about 200 events before the execution times become unacceptable. The maximum number of events is set at 200 in the code, but it can be easily changed. The code has never been optimized to take advantage of multiple processors or multiple cores.

With careful arrangement of windows **mloc** can be used effectively on a 15-inch laptop with a good screen, but a large external monitor is preferable.

Installation Links

The following links cover all aspects of obtaining and installing the software components for **mloc**:

- Directory Structure
- Source Codes
- Configuration
- External Resources
- Data Files

The distribution package of **mloc** contains the necessary directory structure and supporting files, as well as an executable for **mloc** (compiled under macOS v10.15 Catalina). Source codes and a *makefile* are included if the user needs to recompile for a different architecture. However, even if the executable in the distribution is functional, some critical environmental variables need to be set before **mloc** can be run. This is done with a configuration file, a short text file named *mloc.conf* that is read by **mloc** when it is launched. The configuration is generally installation-specific, so a configuration file copied from another installation would usually need to be edited.

The configuration file uses keywords to define the nature of each line, rather than requiring a specific number and order of lines. All elements have default values defined in the code itself, so it is not necessary to set all possible keywords if the default values are correct for your installation.

Filename and Path of the Configuration File

The name of the configuration file is *mloc.conf* and it must be found in the same directory as the executable **mloc**.

Keywords

The currently-defined keywords are listed here. The keyword starts in column 1 of the configuration file and is always followed by a colon and a blank, before the argument.

WORKING_DIR

Defines *mloc_path*: The full pathname of the directory in which the **mloc** executable and the configuration file reside. I refer to this directory as the *working directory*. Maximum 100 characters. The default value is ‘ ’, which will almost never be correct, so all configuration files should carry a line with the *WORKING_DIR* keyword.

AUTHOR

Defines *mloc_author*: a code for the person running **mloc**. This code will be written into the *.datf* file as the author of the location (“origin”) in the hypocenter record (MNF v1.3 format), and it will show up in the *.summary* file and perhaps some other output files. Maximum 8 characters, case-insensitive. The default value is ‘default’; it is highly recommended that all users of **mloc** set an author code for themselves.

STATION_MASTER

Defines the name of the file that is to be used as the master station file. This is only the name of the file, not the path. The master station file is expected to be found in the */mloc_working/tables/stn* subdirectory. If necessary, the path can be changed in *mloc.f90* and the program will need to be re-compiled. The default name of the master station file is *master_stn.dat*. The master station

file is distributed with **mloc**. Therefore, in a standard installation there is no need to use this keyword.

Station information (i.e., stations codes, coordinates, etc) can be read in several formats that are distinguished by an integer value in the first column of the first line of the file. The integer 0 defines the format used for the master station file so even if the *STATION_MASTER* keyword is used to select a file other than the standard distribution, it must use the same format. Supplemental station files can use the master station file format.

GMT_VER

Defines the version of the Generic Mapping Tools (GMT) that will be used for plots. This keyword is needed during times of transition between major releases of GMT that effect plotting functions used by **mloc**. At this time the only allowable values are "5" and "6". The default is '6', i.e., GMT v6 is the expected GMT environment; GMT5 is deprecated. For now, a standard installation of **mloc** will still work with GMT5 for all plotting except topography in base maps, and there is a work-around by which that can still be made to work.

SHELL

Defines the operating system shell to be used in formulating GMT scripts. Supported arguments are *csh* and *bash* (default)'.

A Sample Configuration File

The *mloc.conf* file that I use is distributed with **mloc** but it must be customized for each new installation. My configuration file consists of only two lines:

WORKING_DIR: /Users/eab/Documents/Seismology/Projects/Software/mloc/mloc_working

AUTHOR: EBergman

All other values (master station file, GMT version and shell) that could be set in the configuration file are satisfied by the defaults for my installation.

Data Files

This section discusses the details of the data files that support **mloc**. All necessary files are included in the distribution package, but there circumstances where the user may wish to edit, replace or augment some of them. All data files reside in subdirectories of the */mloc_distribution/mloc_working/tables* directory.

/tables/crust/

The only function of this directory is to hold custom crustal velocity models, but you also have the option of storing them in the cluster directory itself. The only difference is the pathname of the crustal model file assigned with the lmod command in the command file. My convention is to give crustal models the filename suffix *~.cr* but you can name them any way you like.

My strategy is to keep most custom crustal models with the individual clusters (same directory as the event data files) because they are usually cluster-specific. I reserve the *tables/crust* directory for crustal models that may be used across many clusters. It is nearly mandatory to develop a custom crustal model for any cluster that is undergoing a calibrated relocation, but it's usually not possible to place strong constraints on a crustal model for an uncalibrated relocation study, so a reasonable regional model may be used. In particular I store a model (*ak135.cr*) of the crust and upper mantle from the 1-D global model ak135 (Engdahl et al., 1998) in this directory. This is normally the starting model for a new cluster, from which a cluster-specific model is developed through trial and error. Specifically, I make a copy of *ak135.cr*, rename it for the intended cluster and move it to the cluster directory.

The file *simple.cr* provided in the distribution package is an example of the simplest possible crustal model (one layer crust of constant velocity over a pseudo-halfspace upper mantle) which, nevertheless, would provide a good fit to many datasets after adjusting the velocities and crustal thickness.

The details of the file format for crustal models and many aspects of working with crustal velocity models are discussed in the Crustal Models section.

/tables/ellipticity/

Corrections to theoretical travel times for the ellipticity of the Earth are a standard part of traditional earthquake location programs because the location is usually based mainly on the fit of (corrected) theoretical travel times to observed regional and teleseismic phases. Ellipticity corrections are usually on the order of a tenth of a second. For a calibrated location in **mloc**, however, these corrections are irrelevant since only the *relative* times of regional and teleseismic phases are being used to determine the relative locations of the events in the cluster. The absolute location of the cluster (hypocentroid) will be determined from near-source data for which ellipticity corrections are not relevant. Nevertheless, ellipticity corrections are made in **mloc** with the same algorithm used in Bob Engdahl's single event code (i.e., the EHB catalog), based on the

Dziewonski and Gilbert (1976) representation with further work by Brian Kennett and Wim Spakman.

Ellipticity corrections are based on a single data file: */mloc_distribution/mloc_working/ellipticity/tau.table*.

/tables/faults/

The map plots (e.g., the base plot) produced by **mloc** can display faults (command fmap). The data files follow the format for GMT, *with coordinates given in the order longitude, latitude*. They can be stored in this directory or in the cluster directory itself. Data files that cover a large region are probably best stored here. The example is a file of major faults in Turkey: */mloc_distribution/mloc_working/tables/faults/turkey_faults.dat*.

/tables/gmt/

Two types of data are stored in this directory (in sub-directories) for use in plotting: color palettes and digital elevation models (DEMs). These are only used for the map-like plots, e.g., the base plot.

/tables/gmt/cpt/

The default color palette for showing topographic data in **mloc** plots is *topo.cpt*. It works well in most circumstances, but there are many others and GMT documentation includes detailed instructions about designing your own.

/tables/gmt/dem/

In versions of **mloc** prior to 10.5.0, DEM files for several different models (e.g., GLOBE and GINA) were installed in subdirectories of the */tables/gmt/dem* directory. GMT6 introduces support for DEMs that can be automatically downloaded from the GMT server, as documented in Section 3.7 of the GMT Cookbook. **mloc** now makes use of that capability, specifically calling ETOPO1 if the argument to command dem1 is "on". The DEM (earth_relief_01m.grd, 270 MB) is downloaded the first time plotting of topography is called for, and stored in the invisible directory *~/.gmt/server/*, where "~" is the user's home directory.

The */tables/gmt/dem* directory will therefore be empty in the standard **mloc** distribution, but it is an appropriate place to store custom high-resolution DEMs if they are ever used (see Plotting Topography).

mloc uses ETOPO1 to plot topography (command dem1), regardless of whether GMT5 or GMT6 is used, but under GMT5 the ~.grd file for ETOPO1 (bedrock version, not ice) must be stored in a subdirectory of */tables/gmt/dem/* named "ETOPO" that must be created by the user. The zipped archive of the ETOPO1 ~.grd file (*ETOPO1_Bed_g_gmt4.grd.gz*) can be downloaded from NOAA. When unzipped it will be ~1 GB is size.

If GMT5 is used without installing the ETOPO1 DEM you should accept the default ("off") state of the command dem1.

/tables/kml/

The ~.kml file that is a standard output file of **mloc** uses icons in two shades each of five colors for earthquakes at different depths. The icons are in PNG format and must be stored in this directory.

mloc uses a copy of the */tables/kml* directory ("_kml") which is made in the cluster directory on the first run. The ~.kml files created in the cluster directory always reference the locally-stored icons so that the display will still work if the cluster directory is moved to a new location.

/tables/spread/

One of the more important aspects of using **mloc** is the use of empirical reading errors, estimated for each station-phase pair from the actual data. For the first run (at least), when those estimates are not yet available, **mloc** uses a set of phase-specific default reading errors, read from a file named *psdre.dat* in this directory.

Current values in this file are listed in the table:

Phase	Reading Error (s)
Pg	0.4
Pn	0.8
P	0.5
pP	1.0
sP	1.5
Sg	0.8
Sn	2.0
S	3.0
Lg	4.0
T	5.0
S-P	0.4

The user can edit this file to change the default values or to add or remove phases.

/tables/stn/

This directory holds several data files related to seismograph stations, especially the geographic coordinates and elevation for a given station code. The most important file here is the master station file (*master_stn.dat*), which is meant to carry information *only* on stations in the International Registry of Seismograph Stations (IR) at the ISC. The reason for this policy will

become clear(er) when the user encounters the full suite of problems related to station codes and the strategy used in **mloc** to manage conflicts. In some cases the precise coordinates (especially elevation) of a station from the IR are over-ridden by more trusted information from other sources, but non-IR-registered stations should be handled with supplemental station files (command sstn), *not* entered into the master station file.

The other two data files in this directory are related to specific commands:

- bdps.dat: a list of seismic stations that are suspected of reporting bogus depth phase readings. Specifically, it is suspected that some station operators take focal depths from preliminary locations by agencies such as the NEIC, and report depth phase arrivals taken from theoretical travel time calculations. The command bdps takes the pathname of the file relative to the **mloc** working directory so it does not have to be stored here, but this is the natural place for it.
- neic_stn.dat: This file is used by the command nsmd which causes a search to be made for station codes that were not found in the master station file. It is only needed (possibly) if an arrival time dataset includes data obtained from the NEIC. Many stations of the U.S. regional networks that report to the NEIC use codes that conflict with stations registered in the IR, and for that matter, with each other. NEIC solves this problem by carrying FDSN network codes along with station codes (and a lot more) in their metadata server and location software. This data file is a recent download of the entire contents of the metadata server, so a search for a given station code may well return multiple hits from different networks. The strategy used in **mloc** to manage these issues is discussed elsewhere.

Supplemental station files (command sstn) may be stored here also, especially when one may be working regularly with data from a regional or local network whose stations are not all registered in the IR. Then it may be worth building a supplemental station file named for that network. Supplemental station files that are carefully tuned for a specific cluster are best kept in the relevant cluster directory.

/tables/tau-p/

Two binary data files (*ak135.hed* and *ak135.tbl*) are needed by **mloc** to use the tau-p formulation of the 1-D global travel-time model ak135. These files should work on a Macintosh system, and perhaps other OS's, but if not, they will need to be built for your system. Software to create the binary data files from the ak135 model is widely distributed on-line.

The etopo5 digital elevation model is also stored in this directory for use by **mloc** in calculating bounce-point corrections for depth phases, following the algorithms used in the EHB Catalog.

Directory Structure

This section describes the arrangement of directories and files for use with the multiple event relocation program **mloc**. The following schematic shows the main directory structure as it is configured in the distribution package:

- mloc_distribution
- clusters
- documents
- mloc_gfortran
- mloc_src
- mloc_utilities
- lres
- rstat
- xdat
- mloc_working
- tables
- crust
- ellipticity
- faults
- gmt
- cpt
- dem
- kml
- spread
- stn
- tau-p
- utilities
- mnf_utilities
- mnf_search
- to_mnf
- isc_ims2mnf

- seisan2mnf

The location of the top-level directory */mloc_distribution/* shown above is arbitrary, but it is highly recommended to keep the names of the directory structure as they are, at least until you are experienced with how things work. Updates to the **mloc** ecosystem will be distributed in new distribution packages, almost certainly with the above directory structure.

The following relative pathnames are hard-wired in the main program (*mloc.f90*):

- taup_path = 'tables/tau-p'
- ellip_path = 'tables/ellipticity'
- station_path = 'tables/stn'
- cpt_path = 'tables/gmt/cpt'
- dem_path = 'tables/gmt/dem'
- psdre_path = 'tables/spread'

If you insist on departing from the scheme shown above for these pathnames, you will need to edit *mloc.f90* and recompile.

/clusters Directory

The directories for individual earthquake clusters that will be analyzed using **mloc** are stored here. A *cluster directory* can have a rather descriptive name, e.g. “Qeshm Island” or “Jordan and Sverdrup Region A”. The *cluster directories* (i.e., different clusters) may be further grouped if desired, for example by the name of the country where they occur. The subdirectories of a *cluster directory* hold series of closely-related runs (see below). The series subdirectories will be moved back and forth between here and the mloc_working directory. Here is an example:

```
Iran
   Qeshm Island
      qeshm1
      qeshm2
      ...
      qeshm23
```

Naming Clusters, Series and Runs

An individual run of **mloc** is distinguished by a *basename* (the first interactive input to **mloc**) composed of a *cluster name*, *series number* and *run number*, e.g. “jsa5.1” where “jsa” is the cluster name, “5” is the series number and “1” is the run number. The *basename* is used in the names of all output files (e.g., “jsa5.1.summary”).

Cluster names are normally taken from a geographic feature that exists within the boundaries of the cluster, preferably near the center (hypocentroid). Google Earth is a good tool for exploring possible names. Avoid using names that apply to a region larger than the cluster, (e.g., “zagros”),

and choose a name that is fairly easy to type; you will be typing it frequently! Cluster names are often simplified versions of cluster directory names (e.g., “qeshm” vs. “Qeshm Island”). The cluster name cannot contain blanks.

In practice, a relocation analysis with **mloc** will require a number of runs, and often, several series of runs. Different series may be distinguished by significantly changed data sets (e.g., more or fewer events, new readings for some events) or application of a different relocation strategy.

Example Clusters

The mloc distribution package includes several example clusters with event data files, command files and output files to illustrate various aspects of the usage of **mloc**.

/documents Directory

This directory may contain a variety of documentation about **mloc** and its use.

/mloc_gfortran Directory

This directory is used by the *makefile* that controls compilation and building of the **mloc** executable with the gfortran compiler. Initially only the *makefile* is stored here, but when the executable is built the object files for individual program units will be stored here also. The source code units to which the *makefile* refers are stored in the parallel directory mloc_src/. The executable file (named “mloc_g”) will be created in the *mloc_gfortran/* directory and then must be moved to the mloc_working/ directory for use.

If you have more than one compiler that you wish to use, you can create several of these build directories and name the directory after the compiler, e.g., *mloc_distribution/mloc_absoft* or *mloc_distribution/mloc_intel* rather than *mloc_distribution/mloc_gfortran*. In that case you can set up the makefile of the compiler to name the executable accordingly, “**mloc_a**”, “**mloc_i**”, etc. It can be useful to keep track of version numbers as well, with **mloc_a1050** referring to the executable of version 10.5.0 that was compiled with the Absoft compiler.

/mloc_src Directory

Source code files are stored here. The *makefile* in mloc_gfortran/ must have the correct pathnames to the source codes. Having only the source code files in this directory makes editing easier, and it is essential if more than one compiler is to be used. This directory also contains the Version History (“version_history.txt”). It is highly recommended to review the recent changes when you obtain a new version of **mloc**.

/mloc_utilities Directory

Utility programs related to editing event data files during a relocation analysis are discussed here.

/mloc_working Directory

This is where most of the relocation analysis takes place. The executable **mloc** is stored here. The absolute pathname of this directory must be supplied to **mloc** by the configuration file (*mloc.conf*), which must exist in this directory. While it is being worked on, the cluster directory (e.g., *jsa5*) must be moved to this directory from its permanent home in the clusters/ directory.

/mloc_working/tables

The other essential subdirectory to mloc_working/ that must be present is the tables/ directory and its subdirectories. It contains a variety of data files used by **mloc**, organized in a number of subdirectories, as shown in the schematic above. The contents of these directories are described elsewhere.

/mloc_working/utilities

The subdirectory *mloc_working/utilities/* is used to store the executables of the utility programs whose source codes are stored in the directory mloc_utilities/. In use, the executables are copied into the appropriate cluster-series directory, run there from the terminal, and then deleted when done, so it's convenient to have them close at hand.

/mnf_utilities Directory

Utility programs related to creating MNF-formatted event data files for **mloc** are discussed here.

External Resources

This section deals with software that is required (or highly recommended) for effective use of **mloc**, but that is not distributed from this website.

GMT

mloc makes extensive use of the Generic Mapping Tools (GMT) software for plotting. The code constructs GMT scripts, runs them automatically to create postscript files, and converts postscript files to pdf. Depending on the specifics of the relocation one or more plots are always made and many others are controlled by the user. The main scripts are saved for each run in a folder with a name ending in *_gmt_scripts* in case the user wishes to edit a script and re-run it.

The current production release of GMT is version 6.0.0, released November 1, 2019. As of v10.5.0 **mloc** expects GMT6 to be installed.

Using GMT5

In limited testing GMT5 appears to still work correctly with the standard distribution of **mloc**, with the exception of plotting topography. There is a work-around that will restore the ability to plot topography/bathymetry under GMT5. If you intend to use GMT5 with **mloc**, it would be wise to update to the latest version (v5.4.5). You will also need to edit your mloc.conf file to specify plotting with GMT5.

Google Earth

An output file in Keyhole Markup Language (KML) is produced by each run of **mloc**. Any program that can display a file in KML can be used to view it, but Google Earth is most commonly used. There is no requirement to display the .kml file so it can be ignored if no program is available with which to open it.

A Good Text Editor

Use of **mloc** requires extensive manipulation of text files, so a strong text editor is extremely useful. Some capabilities that are needed for use with **mloc** are:

- Open several hundred files at once and switch quickly between them
- Open individual files up to several hundred MB in size
- GREP-style search and replace capability
- Easy navigation by line number

I have used the commercial program BBEdit for many years in **mloc** analyses, as well as coding, and recommend it highly.

A PDF Viewer

The graphic output from **mloc** is all in PDF format. Any PDF reader should be able to display them.

A Line-Fitting Application

The Lg phase is not included in the tau-p software, but **mloc** can employ a custom travel-time model (linear in epicentral distance) for Lg (command *lgtt*) to better fit the observed data. **mloc** includes a command (*ttou*) to create an output file of the observed Lg travel time data that is suitable for import into any data analysis program that has the capability to fit a straight line, i.e., virtually any data analysis program you can find. I use DataGraph.

Source Code

The original coding of the hypocentroidal decomposition algorithm (mostly in the module *mlocinv.f90*) followed the presentation in Jordan and Sverdrup (1981) extremely closely, to the extent of naming variables as nearly the same as possible and including equation numbers from the paper in comments. The published paper, however, contains a few (minor) typographic errors in equations. The code was based on a corrected version.

MLOC was originally written in Fortran 77 but nearly all code has been updated to use structures and features from Fortran 90/95 that improve legibility and robustness. There are multiple user-controlled levels of warnings for errors and debugging. The code is extensively commented, and a full understanding of the functionality of **mloc** requires at least occasional reference to the source code. Some rarely-needed options are only accessible by editing the code and re-compiling. The code is divided into multiple packages with related functionality. Most variables that are used beyond a given program unit are declared in a master include file (*mloc.inc*). This file also contains a large number of named common blocks and most communication between program units occurs through them.

Aside from the code base borrowed from Ken Creager's LOC program for the initial coding of **mloc**, the major bits of source code that have been adapted from other people's work are:

- Code from Johannes Schweitzer's HYPOSAT program for calculating travel times in the custom crustal models (Schweitzer, 2001)
- Code from Ray Buland's software implementing the tau-p travel time calculations (Buland and Chapman, 1983)
- Code from Bob Engdahl's location program (Engdahl et al., 1998) for various odious tasks, such as ellipticity corrections and bounce-point corrections
- A few routines from Numerical Recipes (Press et al., 1986)
- The code to calculate Sn, the robust estimator of spread, from Croux and Rousseeuw (1992)

Source Code Modules

A list of all the source code modules in the current release of **mloc** and a note about their functionality can be found here.

Version History

Changes to **mloc** are documented in the text file *mloc_version_history.txt*, which is kept with the source code files in the directory */mloc_distribution/mloc_src/*. Recent changes are posted here. Early development of **mloc** was undocumented. Version documentation began in early 2005 with v4.1. The version scheme was expanded to three fields with v9.1.1 in October 2009. The versioning scheme is *major.minor.patch* or sometimes *major.minor.feature*, i.e., the third field increments when a significant bug has been patched or a significant feature has been added (or

occasionally, removed). For lesser changes the release date is updated but the version number stays the same. The version number and release date are hard-coded in the main program and always listed in the .summary output file.

Compilers

The **mloc** distribution package includes a *makefile* for use with the open source Fortran compiler gfortran. Most development of **mloc** has taken place under a commercial compiler from Absoft or Intel.

Warning

*The source code of older distributions of **mloc** (including v10.5.0) is not fully compatible with the most recent versions of gfortran (e.g., v9.2.0).* Versions of gfortran up to about v5 should be compatible with that older code base. The offending sections were some very old codes in the tau-p library *taulib.f90*. That code has been rewritten for version 10.5.1 of **mloc** and it is now compatible with gfortran v9.2.0.

MLOC Source Code Modules

This page lists the code modules supplied in a distribution of the source code for **mloc**, with a brief description of their functions. If you want to get a quick understanding of how **mloc** works, look at *mloc.f90*, *mlocset.f90*, *mloc_commands.f90* and *mloc.inc*.

Main Program

- *mloc.f90*: Assembles information on the details of the relocation and launches the relocation process. Version number and release date are set in a *block data* section at the end

Subroutine Modules

- *cal_shift.f90*: Implements indirect calibration
- *dcal.f90*: Output for direct calibration
- *dsvd2.f90*: Singular value decomposition
- *hyposat_loc.f90*: Travel times through a custom crustal model
- *libtau.f90*: Tau-p travel time models
- *mloc_commands.f90*: All commands processed
- *mlocinv.f90*: Hypocentroidal decomposition algorithm
- *mlocio_mnf.f90*: Read and write the various flavors of MNF (MLOC Native Format) event data format
- *mloclib.f90*: Utility routines

- *mloclib_date_time.f90*: Utility routines related to date and time
- *mloclib_geog.f90*: Utility routines related to geographic manipulations
- *mloclib_gmt.f90*: Utility routines related to GMT plots
- *mloclib_inv.f90*: Utility routines related to hypocentroidal decomposition, inversion
- *mloclib_messages.f90*: Utility routines related to messaging, feedback from the program
- *mloclib_phases.f90*: Utility routines related to processing seismic phase names
- *mloclib_set.f90*: Utility routines used mainly in *mlocset*
- *mloclib_stations.f90*: Utility routines related to station codes and coordinates
- *mloclib_statistics.f90*: Utility routines related to statistical analysis
- *mloclib_tt.f90*: Utility routines related to travel time calculations
- *mlocout_bloc.f90*: Output for BayesLoc
- *mlocout_comcat.f90*: Output for GCCELand NEIC's ComCat database
- *mlocout_cv.f90*: Output of the full covariance matrix
- *mlocout_gmt.f90*: Output plots using GMT
- *mlocout_hdf.f90*: Output of one-line-per-event location summaries
- *mlocout_kml.f90*: Output of the KML file of locations
- *mlocout_phase_data.f90*: Output file with all phase data
- *mlocout_puke.f90*: Output of a "contains everything" file format similar to GCCEL
- *mlocout_rderr.f90*: Output of data on empirical reading errors for every station-phase
- *mlocout_summary.f90*: Output of the summary file for a run
- *mlocout_tomo.f90*: Output of a file format intended to support tomographic studies
- *mlocout_tt.f90*: Output of data for specific phases
- *mlocout_ttsprd.f90*: Output of data on the spread and baseline of residuals for each phase
- *mlocset.f90*: Carries out the actual relocation

Include Files

- *cal_shift.inc*: Indirect calibration
- *ellip.inc*: Ellipticity calculations
- *hyposat_loc.inc*: Custom crustal model
- *mloc.inc*: Main include file, most variable definitions and named common blocks

- *ttlim.inc*: Tau-P travel-time calculations

Utility Codes

Several types of utility programs are essential to the efficient use of **mloc**. These include several programs for editing event files efficiently as part of the so-called cleaning process:

- rstat: an interactive program to explore cluster residuals (normalized, demeaned travel-time residuals), an essential aid to manual editing (flagging) of outlier readings.
- lres: batch processing to flag a set of readings with large cluster residuals across multiple event files listed in the ~.lres output file of a previous run of **mloc**. Flagged readings will not be used in future runs.
- xdat: batch process to flag arrival time readings with grossly large residuals, listed in the ~.xdat file from a previous run of **mloc**.

The source codes for these utilities are contained in the **mloc** distribution in the directory */mloc_distribution/mloc_utilities/*. Executables for macOS are included as well, or they can be easily re-compiled using a simple command, e.g.:

gfortran lres.f90 -o lres

See MNF Utility Codes for some other important utility programs for creating event data files in the MNF format.

Editing Event Files

A proper relocation analysis with **mloc** is never completed in one run of the program, because arrival time data sets nearly always contain some (usually many) outlier readings that must be identified and flagged and because the empirical reading errors for each station-phase pair are never known *a priori*. The task of satisfying these requirements is known as cleaning and it must be done iteratively, i.e., one deals with the worst outliers first, re-runs to obtain improved estimates of empirical reading errors, and repeats until the relocation satisfies the various criteria for completion.

Therefore, the usual pattern of a relocation analysis is to make a run of **mloc**, followed by the use of one or more of the utility programs discussed here, and repeat. These utilities are designed to be run in the cluster directory, where the event data files are stored, so it is convenient to keep copies of the executables in the */utilities* subdirectory of the working directory for easy access. The executables are then copied from there into the cluster directory when the analysis begins and deleted when it is done.

rstat

The utility program *rstat* provides access to detailed information and statistics about the arrivals from a specified station-phase which are very helpful in the cleaning process. It is the tool with which to make the most careful investigation of potential outliers and new users of **mloc** are encouraged to make heavy use of it in order to gain intuition about the nature of outliers in arrival time datasets. The other two tools discussed here, *lres* and *xdat* are more efficient but their

proper use depends on the intuition gained through use of *rstat*. In most cluster relocation analyses I make use of all three. Some clusters go very easily and require very little use of *rstat*, but some require what amounts to hand-to-hand combat and in those cases *rstat* is your best weapon.

To use *rstat* it must be located in the cluster directory, with the output files from the latest run of **mloc** and all the event data files. I keep a copy of the executable *rstat* in *mloc_distribution/ mloc_working/utilities/* and copy it into the cluster directory when I start work. The first steps in using *rstat* are shown in this example, using a cluster directory called *salmas2*:

```
$ cd salmas2
$ rstat

Release date June 29, 2018

Enter file name for phase_data: salmas2.2.phase_data
Enter number of iterations: 2
Is there differential time data? y or n: n
Case sensitive station names? y or n: n
Use deployment codes? y or n: n

Enter station name (q to quit):
```

The data that *rstat* will read and process are all carried in a *~.phase_data* output file. *rstat* next requires as input the number of iterations that were performed so it knows which column of residuals to read. The number of iterations is given in the terminal window near the end of each run, or it can be read easily from the *~.phase_data* file itself. The next three questions deal with unusual cicumstances to which the answer is usually *n*; these will be discussed below.

The next step is to specify a station code and phase name; this is done in two steps. Then *rstat* reads through the entire ~.phase_data file, extracts every instance of that station-phase combination, and carries out a statistical analysis of the residuals:

```
Enter station name (q to quit): tab
Enter phase name (* for all phases): Pg

                            rderr  delta    dts   wgt    eci
 21 TAB              Pg      0.28   0.84   1.19  1.00   1.00 ISC          Pg          5  salmas2/19810524.2112.21.mnf
 27 TAB              Pg      0.28   0.96   0.45  1.00  -1.59 ISC          Pg          5  salmas2/19840629.1955.16.mnf
 31 TAB              Pg      0.28   0.82   0.78  1.00  -0.17 ISC          Pg          6  salmas2/19891203.0739.09.mnf
 32 TAB              Pg      0.28   1.04   1.00  1.00   0.35 ISC          Pg          5  salmas2/19920305.0330.15.mnf
 33 TAB              Pg      0.28   1.11   0.87  1.00  -0.06 ISC          Pg          5  salmas2/19930330.2225.19.mnf
 38 TAB              Pg      0.28   0.95   1.18  1.00   1.29 ISC          Pg         22  salmas2/19981123.1111.39.mnf
 39 TAB              Pg      0.28   1.57   0.59  1.00  -0.82 ISC          Pn         26  salmas2/19990219.1800.10.mnf
 22 TAB            x Pg      0.28   0.82   2.45  1.00        ISC          Pg          5  salmas2/19810524.2207.05.mnf
 24 TAB            x Pg      0.28   1.56  -2.50  1.00        ISC          Pn          8  salmas2/19830803.0306.01.mnf
 26 TAB            x Pg      0.28   0.82   3.50  1.00        ISC          Pg          5  salmas2/19840325.0244.57.mnf
 37 TAB            x Pg      0.28   0.91   3.09  1.00        ISC          Pg         22  salmas2/19981118.1137.19.mnf
 40 TAB            x Pg      0.28   1.41  -1.92  1.00        ISC          Pn         51  salmas2/20000226.0818.38.mnf
Mean =  0.866
Sn =  0.400
On    7 readings.

Enter station name (q to quit):
```

Because we did not request case-sensitive stations codes, most station codes can be entered in lower case. There is an exception, however: station codes that include numbers must be entered in the correct case. The phase name must always be entered in the correct case. The output columns are:

- Event number
- Station code

- Deployment code (if requested)
- Phase reading flag (many are skipped, “x” and “s” are displayed)
- Phase name for the current run
- Reading error used for this station-phase in the current run
- Epicentral distance
- Time residual
- Weight (see the wind command)
- Cluster residual (eci), demeaned, normalized residual
- Reading author
- Phase name read from the event file
- Line number of the reading in the event file
- Name of the event file

The listing of all instances of the requested station-phase is followed by output of the mean of the unflagged residuals, the spread of the unflagged residuals, calculated as the robust statistic *Sn* (no relation to the seismic phase) formulated by Croux and Rousseeuw, (1992) and the number of instances used in calculating the statistics.

rstat does not do anything to any of the events files. It only provides information with which the user may decide to flag certain readings, unflag them, or modify the phase name read from the event file. In most cases **mloc** is run with phase re-identification activated (command phid) so the final phase name may be changed by the algorithm, as it has been for the Pn reading at TAB for event 39 in the above listing. Another option is to use the special flag *!* in the prevent phase re-identification field of the event file, if the user judges that the phase re-identification algorithm is making a mistake.

The field of primary interest in the output of *rstat* is the cluster residual (variable *eci* in the code). It displays the distance of a residual (*dts*) from the mean, normalized by the current estimate of reading error (*rderr*). In most cases the reading error will be an estimate of *Sn* from a previous run, read by **mloc** from the *~.rderr* file by the command rfil.

Early in a relocation analysis there will be many instances of large values of the cluster residual, because there are nearly always many outlier readings in typical arrival time datasets. How is “large” defined? We assume that the “good” readings of a certain phase observed at a certain station have a roughly Gaussian distribution, with unknown standard deviation and perhaps with a baseline offset from the predicted arrival time. Outlier readings are those that lie so far from the the mean that they are unlikely to be members of the population of “good readings”. If our measures of the mean and spread of the population are reasonably accurate, we can judge that nearly 100% of the “good” readings will have values of *eci* that are less than 3.0 (see the

68-95-99.7 rule). One can push the target limit below 3.0 to some extent, but the it does not gain much in the accuracy of the results and it threatens to invalidate the statistical assumptions that go into estimates of uncertainties of the hypocentral parameters. It is not recommended.

You *cannot* simply run *rstat* once, flag all readings with eci > 3 and declare victory over outliers. The cleaning process must be done gradually, beginning with the largest outliers and gradually attaining (through many runs) a condition where very few if any readings exceed that limit. There are several reasons why an incremental approach is needed. One is that the locations of the events will change after readings are flagged and the distribution of residuals will change as a result. It is useful to remember that in **mloc** everything effects everything else. Another is that the *Sn* estimator, while powerful, is not very helpful with distributions containing many outliers. The user's intuition will come into play as well, especially as experience is gained with what these distributions look like in practice. A third reason is that the estimate of *Sn* will become smaller as outliers are removed from the problem and thus readings that appeared to be "within limits" before will appear as outliers when the relocation is run again with a smaller empirical reading error for the station-phase of interest. It can sometimes feel like one is chasing one's own tail but in fact the process does converge.

For new users especially it is better to err on the side of keeping outliers in the problem than risk flagging readings which should actually have been kept. The reason for this is that **mloc** will naturally down-weight readings that belong to a distribution containing severe outliers. The empirical reading error will be large in such cases and the data are weighted inversely to empirical reading error.

In use, one should be able to open all the event files, along with one or more of the output files such as ~.lres, ~.phase_data and ~.dcal_phase_data, while running *rstat* in a terminal window, choosing station-phases to investigate on the basic of clues such as large absolute residuals, large values of eci and large values of empirical reading error. Based on the output from *rstat* the user goes to the appropriate event file, jumps to the line number of the reading of interest and flags the reading or edits the phase name. This why the choice of a text editor is very important in **mloc**; it will be heavily exercised by the manual cleaning process employing *rstat*.

It is not uncommon to find readings that have been flagged incorrectly during the cleaning process, so removing flags is also a regular task. Here is an example, from the same salmas2.2 run:

```
Enter station name (q to quit): cldr
Enter phase name (* for all phases): Sg

                            rderr  delta    dts   wgt    eci
 61 CLDR             Sg      0.70   0.22   0.01  1.00  -0.71 ISC          Sg          7  salmas2/20090705.2348.25.mnf
 62 CLDR             Sg      0.70   0.51   0.68  1.00   0.07 ISC          Sg          7  salmas2/20101106.0105.16.mnf
 63 CLDR             Sg      0.70   0.49   0.83  1.00   0.30 ISC          Sg          7  salmas2/20101206.0516.09.mnf
 65 CLDR             Sg      0.70   0.70  -0.63  1.00  -1.82 ISC          Sg         16  salmas2/20110710.0423.02.mnf
 74 CLDR             Sg      0.70   0.41   0.29  1.00  -0.57 ISC          S          19  salmas2/20111029.2224.24.mnf
 74 CLDR             Sg      0.70   0.41   0.71  1.00   0.02 ISC          Sg         17  salmas2/20111029.2224.24.mnf
 75 CLDR             Sg      0.70   0.38   1.71  1.00   1.50 ISC          Sg         21  salmas2/20111106.0243.14.mnf
 75 CLDR             Sg      0.70   0.38   1.51  1.00   1.21 ISC          S          23  salmas2/20111106.0243.14.mnf
 55 CLDR           x Sg      0.70   1.02   1.55  1.00        ISC          Sg         27  salmas2/20060729.0151.11.mnf
 57 CLDR           x Sg      0.70   1.13   1.34  1.00        ISC          Sg         36  salmas2/20061202.0639.37.mnf
 64 CLDR           x Sg      0.70   0.56   0.54  1.00        ISC          Sg          6  salmas2/20110314.1857.11.mnf
Mean =  0.639
Sn =  0.839
On    8 readings.
```

```
Enter station name (q to quit):
```

Both readings for event 75 should be flagged as outliers, but the reading for event 64 should be unflagged.

The effects of flagging (or unflagging) readings with *rstat* are not all felt immediately in the next run of **mloc**. The presence or absence of the edited readings is felt in the next run, but the empirical reading errors that will be read from the *~.rderr* file of the current run will not yet reflect those edits. The *~.rderr* file from the next run will reflect the edits you just made and so they will be available for the run after next.

There are many subtleties about the use of *rstat* which can only be learned by experience and thoughtfulness about the nature of arrival time data and the various factors that can influence the values that show up in a typical seismic bulletin. Correct and effective use of **mloc** cannot be accomplished without having mastered those subtleties to some extent.

lres

If you understand the use of rstat the use of *lres* is trivial. To use it you must first issue the command lres when running **mloc**. The command takes an argument which is the value of cluster residual (eci) above which a reading will be written to the *~.lres* output file. The utility program *lres* simply reads the content of that file (the name of the *~.lres* file is the only input) and flags the corresponding readings in the event files.

During the course of a relocation analysis the threshold value of *eci* in the lres command would be fairly high (say, 6.0) in early runs and would gradually be reduced until an *eci* threshold of 3.0 results in a *~.lres* file with few if any contents. In a difficult analysis one might run *rstat* rather than *lres* and do the editing by hand, especially in the early going. One might keep the threshold value of *eci* at the same level for a number of runs, while dealing with outliers or other issues such as setting focal depths or refining the crustal velocity model.

As with *rstat* the effects of editing readings with *lres* are not fully felt until the second successive run.

xdat

The utility program *xdat* operates very similarly to *lres*, flagging readings in event files according to data in an output file from a previous run of **mloc**. The input file *~.xdat* is based on the windowing algorithm (see command wind) that identifies gross outliers and drops them from the relocation. These readings are listed in the *~.phase_data* file in the "BAD DATA" section for each event, and they are annotated "PRES" under the "Why Bad" column.

Since these readings are automatically dropped by **mloc** it is not essential to flag them with *xdat* but it is a good practice to do so a few times during an analysis so that you end up with a "clean" dataset. Sometimes these readings can fall back into the relocation and create problems. The

~*.xdat* file is always created by **mloc** but it will often be empty after *xdat* has been run a few times.

MLOC Native File Formats

There are several distinct formats of data that can be read or written by **mloc** that are especially important. Although they are quite different in detail they are all considered to belong to the family of **MLOC Native Formats** or **MNF**; they are distinguished by version number:

- MNF v1.3: Format for *input* of individual event files to **mloc**; also used for *output* of bulletins of the initial (automatic) and final (command datf) state of the current cluster.
- MNF v1.4: Format for single-file *output* of all information related to an entire cluster, designed originally for import to the U.S. Geological Survey's National Earthquake Information Center (NEIC) ComCat database, but also used for the GCCEL project
- MNF v1.5 Format for *input* of differential time data

Detailed specifications for each format are found in other documents linked to the version numbers in the list above.

Version Number Family

The basic functionality and structure of the MNF formats listed above is described by the *major.minor* version format, but for the standard format v1.3 **mloc** keeps track of a third field which tracks certain changes in the format. The full version number of the MNF v1.3 "family" is currently v1.3.3 and **mloc** will issue a warning if it reads a file carrying an earlier version of the format. Depending on the details, a data file in an earlier format may be read and processed without a problem. No format check is done for input files of differential time data in v1.5 format but this would be added in the future if the need for a change in the format arises. Output files in MNF v1.4 format always contain a format line with the format version, currently 1.4.2.

Description and Usage

All three members of the MNF family of formats are fixed formats based on the common concept of a small set of distinct record types with different formats, identified by a single-character flag in the first column. Each record is a single line; each record type has a minimum line length determined by the defined fields (even if they are not all required). There are minimal constraints on the order in which different record types may occur. Each type of information (e.g., event, hypocenter, depth, magnitude, phase arrival) has a specific record type, and there are a few utility record types.

MNF v1.3 is by far the most important format version for **mloc** and non-specific references to "MNF" always refer to this version. It can be used to carry data for a single event, an earthquake catalog (multiple events, hypocenters only) or a seismic bulletin (multiple events, including phase readings), but for input to **mloc** only single event files are used. Data files with this format always have the filename suffix *.mnf*. The bulletin form of MNF v1.3 is used for the ~.dat0 output file created automatically by **mloc** to archive the input dataset for the run, and for the optional (command datf) ~.datf file that archives the final state of the cluster.

MNF v1.4 and 1.5 are special-purpose formats, used only for output and input, respectively. Output files in MNF v1.4 format are only created by the ccat command and input files in MNF v1.5 format are only read by the diff command. Files in MNF v1.4 format are created with the filename suffix *.comcat*. Files in MNF v1.5 format may have any naming convention.

Background

In its early development in the late 1980s, **mloc** used an event data format inherited from Ken Creager's LOC program, on which **mloc** was based. In the late 1990s, when **mloc** began to be heavily developed for application in research on calibrated ("ground truth") locations in close collaboration with Bob Engdahl, the ISC's 96-byte fixed format (FFB), on which Engdahl had standardized his processing, was adopted as the default event data format for **mloc**. The FFB format had to be extended in certain ways to meet the needs of **mloc** (as well as Engdahl's codes), and it was always an awkward format for this application, particularly in the fact that very different formats are used for initial and secondary phase readings, on the need to break up station codes longer than 4 characters, and in the strict requirements on ordering of records that is imposed by the concept of carrying in each record the record type for the following record.

As the FFB format was becoming deprecated at the ISC, replaced primarily by IASPEI Seismic Format (ISF), it became necessary and desirable to devise a format that specifically meets the requirements of **mloc** and which is more compatible with current practice in seismological data formats and exchange mechanisms. This results in data files that are reasonably compact and easy to process and edit in the ways required by the typical processing strategies that have proven to be effective with **mloc**.

MLOC Native Format (MNF) v1.3

The current version is v1.3.3, released October 11, 2017 with **mloc** v10.3.4. **mloc** issues a warning if a file with an earlier format is read but depending on the details the file may still be processed correctly.

MNF v1.3 is the only format supported by **mloc** for input of arrival time data for individual events, i.e., the data for each event is stored in a separate file. Event files have the filename suffix *.mnf*. The *strongly* recommended naming convention is to base the body of the filename on some estimate of the origin time of the event, down to the nearest second:

- YYYYMMDD.HHMM.SS.mnf

It is not important that the origin time represented in the filename be especially accurate, only that it be distinguishable from other events in the cluster.

MNF Bulletins

MNF v1.3 also supports the formatting of seismic bulletins, consisting of the concatenation of two or more individual event files in MNF v1.3 format. There is no limit on the size of a bulletin

other than practical concerns related to storage, viewing and editing. I have worked comfortably with a bulletin of ~44,000 Iranian earthquakes that is 260 MB in size.

The only difference between an event file and a bulletin is that the bulletin's first record is a bulletin record and the end-of-file record appears only once, as the last record of the bulletin. In a bulletin the format record is required only once, before any events are read, but it is legal to include one in each event bloc.

Defined Record Types

MNF v1.3 has 11 defined record types, of which 6 are considered "data-carrying":

Record Types of MNF v1.3

Flag	Record Type	Minimum Length	Full Length
B	Bulletin	1	121
F	Format	15	15
E	Event	1	121
I	ID	1	51
H	Hypocenter	74	121
D	Depth	9	121
M	Magnitude	8	121
P	Phase	55	121
#	Comment	1	121
S	Stop event	1	4
EOF	End of file	3	3

Unlike all other record types, which are distinguished by the flag in column 1, the end-of-file record (*EOF*) uses columns 1-3; it has no other arguments. It causes processing of a data file to end, so it would normally only be found once, at the end of the file, whether it holds a single event or multiple events.

The "natural" line length for MNF files is 121 characters, because most of the information-carrying record types have fields defined to this length.

Data for a single event is carried in a block of records that must start with an event record "E" and end with a stop record "S". Within the block, data is carried in a combination of hypocenter "H", depth "D", magnitude "M", and phase reading "P" records. At least one hypocenter record "H" is required, but multiple estimates of hypocenter are permitted. ID, magnitude and depth records are optional and there is no limit to how many can be supplied of each. Phase reading records are also optional, in the case of an earthquake catalog. The MNF format can represent a catalog but it is not designed to do so in an efficient manner, since it requires a minimum of three lines (an event record, at least one hypocenter record, and a stop record) for each event.

Some of the fields in an MNF-formatted dataset are not required for **mloc** processing, but they carry information which is often important to retain for interpretation of results and for maintaining compatibility of data products with the NEIC and ISC and other agencies that have standardized on formats such as the IASPEI Seismic Format (ISF) format for data exchange. Conversely, MNF does not carry some (actually, many) fields which would be considered essential for a general-purpose seismic bulletin format such as ISF, because it is optimized for use in relocation studies using **mloc**.

Several record types carry a "usage" flag (always in column 3) that determines the way (or if) the information in that record will be used. Some records have an "ID" field to carry an ID number that may have been assigned elsewhere, typically in a relational database (e.g., EvID, OrID, ArrID). Event IDs have their own record type because it can be useful in some aspects of **mloc's** processing. Comment records (flag "#") are supported; they can be inserted anywhere in an MNF-formatted file. Obviously, any text that occurs after the first EOF record will not be processed and can be considered as a comment, regardless of the formatting, but the use of the EOF record in this manner is not recommended.

Except for the requirement to start each event block with an event record and end the block with a stop record, there are few requirements on the order of records within an event block. In most cases the recommended order would be event record, ID record(s), hypocenter record(s), depth record(s), magnitude record(s), phase reading records, followed by a stop record.

There must be at least one hypocenter record, but multiple hypocenter estimates can be carried. The usage flag should be used to determine which of several hypocenter records will be honored for the starting location in **mloc** (or is otherwise the "preferred" location); otherwise precedence will be determined by the software reading the file, probably either the first hypocenter record encountered or the last. The same principle applies to depth and magnitude records: the usage flag should be used to specify a preferred value, if there is one, rather than relying on the sequence of records.

Format Versions

Each data file, whether for a single event or multiple events, should contain a format version record ("F") before the first event record ("E"). The format version record provides a version number for the MNF format in which the file is written. A program that reads an MNF file should check the version number to be sure it will correctly interpret the data records. A bulletin or catalog made up of event files written in different MNF format versions could be processed by including the necessary format version records ("F") when the format changes for the next event, but this is not recommended.

Starting with MNF v1.3.3 and **mloc** v10.3.4 (October 11, 2017 release), format versions are expected to be complete, i.e., "1.3.3" rather than "1.3" which was adequate for previous versions of **mloc**.

Depths

The hypocenter record normally (but not always) carries an estimate of focal depth, but there may be additional estimates of depth that should be carried. The depth record is intended to carry information on depth that is considered credible, but was not associated with a particular hypocenter determination. The most common source of such depth estimates is waveform analysis of some kind. Multiple depth records are permitted and one of them can be designated as preferred by the usage flag '=' in column 3. Depth records can carry an optional depth code flag which **mloc** uses to keep track of the nature of depth constraint. It is highly recommended to use the standard flags. For example, the focal depth in the preferred hypocenter record will be taken as the preferred depth by default in **mloc**, but it would be over-ridden if a following depth record meets these requirements:

- The depth record has been designated as preferred by the usage flag "=" in column 3
- The depth record carries a depth code that is considered "constrained"
- The hypocenter record's depth estimate does not carry a depth code that is considered to be "constrained"

Like the hypocenter record, a depth record can carry an asymmetric estimate of uncertainty. The depth uncertainties are optional. Focal depth and both uncertainties can be given to a tenth of a kilometer, but the decimal point should be present even if the value is only given to the nearest kilometer, to ensure correct reading by Fortran formatted read statements. The authorship field can be used for comments about the nature of the depth estimate. There is no concept of author ID for depth.

Magnitudes

Magnitude estimates are carried in a magnitude record, one magnitude estimate per record. Multiple magnitude estimates can be carried. A usage flag can be used to select a preferred estimate from among multiple records, but is not required. If no magnitude record is marked as preferred, **mloc** uses the first one encountered as a measure of magnitude. In cases where it is desired to carry a magnitude associated with a specific reading, a magnitude record could be interspersed with phase reading records; in any case the author field could be used to indicate the desired association. Magnitudes can be carried to two decimal places. In any case the decimal point should always be given, to ensure correct reading by Fortran formatted read statements.

Station Codes

The concept of station codes is evolving away from the traditional strategy of attempting to carry all necessary information about a seismic station (plus the installed instrumentation and who operates it) in a single code of 4 or 5 characters. The MNF format implements the New IASPEI Station Coding Standard:

- Agency.Deployment.Station.Location.Channel

or “ADSLC” formulation, using the fixed format display standard described in the defining documentation. In this format, what used to be known as the “station code” is carried in 3-5 characters.

mloc’s support for the ADSLC protocol is complete in the sense that the phase record has a field for each element, but in fact **mloc** uses only the station field routinely; the deployment field can also be used optionally to resolve station-naming conflicts, but the implementation of this feature is complex, not entirely bug-free and seldom worth the trouble. One reason for this disappointing state of affairs is that the ISC has not yet implemented the ADSLC protocol throughout their database and the ISF data format returned from searches of the ISC Bulletin only carries the station field. Even if that problem were solved, however, the extensive aliasing incorporated in the ADSLC standard makes the simple question “are these two stations the same?” challenging for a program like **mloc** that attempts to integrate data from many sources. Strategies for dealing with station naming issues are discussed here.

Station codes are carried twice in phase records. One instance is required and that is the field read by **mloc**. The station code is normally repeated in the second field (if it appears at all; it is optional), but it can be different in cases where a network established a station with a convenient code and only later decided to register the station with the IR, at which point they discovered that their favorite code was already taken. The station would then be registered in the IR with a different code but it is not uncommon for the network to continue using the original code in their internal processing and if one acquires data directly from the network there will be a station conflict. The solution is to edit the station code that is read by **mloc** to be the one that is registered.

Phase Names

Seismic phase name is carried twice in the MNF format. The first instance (columns 24:31 of a “P” record, see below) is read by **mloc**, and then it is subject to the various procedures within **mloc** that may change the syntax of the phase name or change the phase ID completely. The second instance (columns 66:73) carries the phase name as reported by the original source. It is sometimes useful to change the input phase name from the original phase name, to assist **mloc’s** phase identification algorithm in determining the correct (or at least the desired) phase identification. It is also useful to retain the original phase name unchanged for reference. The MNF format also carries a position for a special flag (“!”) which informs **mloc** that the input phase name should not be changed during relocation.

ID Numbers

For informational and forensic purposes, MNF includes fields for identification numbers for various kinds of data that are provided by seismological centers. The only use that **mloc** makes of such information is to use event IDs (EvIDs), if provided, to correctly match events with the relocation results of a previous run, as carried in an hdf-formatted file.

Conventionally, ID numbers assigned to events (“EvID”), hypocenters (“OrID”), phase readings (“ArrID”) and other kinds of data in relational databases are integer numbers. Standards on how many digits are carried vary from system to system, but 10-digit integers are likely to be necessary before long, especially for ArrIDs. The MNF format does not enforce a data type on those IDs; they are character fields as far as **mloc** is concerned. Even so, ArrIDs and OrIDs should be right-justifed to aid in correct reading of integers by a Fortran code. Although 10-character ArrIDs are specified in MNF it has proven necessary to provide larger fields for EvIDs and OrIDs.

The larger field for EvIDs is driven by NEIC, where current practice places no practical limit on the length of an event ID, which is no longer an integer, but a combination of letters (e.g. network codes) and digits. The 40-character field provided in the MNF format should be adequate to handle these, but **mloc** only reads the first 10 characters from this field, because this is assumed to be adequate to distinguish between events in a cluster. An EvID of less than 10 characters can be placed anywhere in the 10-character field (columns 12:21) that **mloc** reads; it will be right-justified inside **mloc**.

Prior to v1.3.3 of the MNF format, only a single EvID could be entered for each event; it was carried at the end of the event record. In v1.3.3, EvIDs are read from one or more ID records. The format also includes a character field to identify the source of the EvID. The current version of **mloc** can still read an older MNF file in “1.3” format and extract an EvID carried in an event record. The first-encountered ID record will be the preferred value in **mloc** by default, or the usage code (“=”) can be used to specify among multiple records.

The OrID field in a hypocenter record (columns 104:121) is 18 characters in length because it is used by **mloc** to record the cluster name/series code (left-justified, starting in column 104). For magnitude and phase records, the associated 10-character ID field is in columns 112:121. Although 10 digit integer IDs can be carried in the ID field, it should be noted that 32-bit computer systems may have problems processing integers of more than 9 digits.

Defined Record Types

The concept of “optional” fields in the descriptions of record types is specifically in the context of use by **mloc**. Fields that are optional are indicated in the tables below. In writing MNF-formatted data files it is advisable to pad lines to the full length of that record type, which is based on the defined fields, not the required fields, and it is not unwise to pad all lines to 121 characters, the full length of the longest defined record types, regardless of record type.

Bulletin Record

Column	Description
1:1	Record format flag “B”
5:121	Bulletin description, optional (a117)

Format Record

Column	Description
1:1	Record format flag “F”
10:15	Format version (a6)

Note: It is useful for readability to include extra text, such that columns 1:9 read “F MNF v”, but all that is required is the “F” in column 1 and the version number in columns 10:15.

Event Record

Column	Description
1:1	Record format flag “E”
3:3	Usage flag, optional (a1)
5:121	Annotation, optional (a117)

The only non-blank usage flag for an event record is “-“, indicating that there is no phase data.

ID Record

Column	Description
1:1	Record format flag “I”
3:3	Usage flag, optional (a1)
5:10	ID source, optional (a6)
12:51	Event ID, optional (a40)

The only recognized usage code is “=”, which makes this entry preferred, regardless of its position relative to other ID records. Otherwise the first ID record encountered will be used. An ID record in which the Event ID field is left blank is legal. If there are multiple ID records and it is desired to have **mloc** ignore them, an empty ID record with usage code set to make it the preferred ID could be used.

Hypocenter Record

Column	Description
1:1	Record format flag “H”
3:3	Usage flag, optional
5:8	Year (i4)
10:11	Month (i2)
13:14	Day (i2)

16:17	Hour (i2)
19:20	Minute (i2)
22:26	Seconds (f5.2)
28:32	OT uncertainty, optional (f5.2)
35:42	Latitude (f8.4)
44:52	Longitude (f9.4)
54:56	Smin azimuth, optional (i3)
58:62	Error ellipse Smin, optional (f5.2)
64:68	Error ellipse Smaj, optional (f5.2)
70:74	Focal Depth, optional (f5.1)
76:76	Depth code, optional (a1)
78:82	Plus depth uncertainty, optional (f5.1)
84:88	Minus depth uncertainty, optional (f5.1)
90:93	GTCNU, optional (a4)
95:102	Author, optional (a8)
104:121	Origin or cluster ID, optional (a18)

Note: **mloc** recognizes a hypocenter record in which the usage flag is "=" as the preferred hypocenter for setting the starting location. If no hypocenter record carries the usage flag, the first hypocenter record encountered will be taken as preferred by **mloc**. Other software may behave differently.

Both depth uncertainties are provided as positive numbers. "Plus" depth uncertainty is on the deeper side; "Minus" uncertainty is shallower, and therefore should not be greater than the focal depth in absolute value. If only one depth uncertainty is encountered, it should be interpreted as a symmetric uncertainty. It is important to put the decimal place into the depth field, even if precision is only to the nearest kilometer (or more), to ensure correct reading by a Fortran formatted read statement.

No information on the statistical level of the uncertainties of origin time, depth, or epicenter is provided in the MNF format because there is so little standardization at present. Such values are commonly interpreted as ± 1 σ for origin time and depth, but confidence ellipses are usually calculated at 90% or 95% confidence levels.

The "GTCNU" field carries a four-character code relating to calibration status (I prefer this term to "ground truth" level). It could be the GTX formulation (e.g., Bondar et al., (2004)), but I have developed the GTCNU nomenclature to provide much more detailed information on the subject of what hypocentral parameters are considered to be calibrated (i.e., thought to be bias-free). The GTCNU nomenclature is documented fully elsewhere.

The placement of a traditional (10-digit integer) “OrID” value within the 18-character field is optional, but it is probably best to right-justify it (i.e., in columns 112:121) to facilitate reading as an integer. **mloc** writes the cluster name and series number as a character string that is left-justified in the field, beginning at column 104.

Depth Record

Column	Description
1:1	Record format flag “D”
3:3	Usage flag, optional
5:9	Depth (f5.1)
11:11	Depth code, optional (a1)
13:17	Plus depth uncertainty, optional (f5.1)
19:23	Minus depth uncertainty, optional (f5.1)
25:121	Authorship and comments, optional (a97)

Note: Both depth uncertainties are provided as positive numbers. “Plus” depth uncertainty is on the deeper side; “Minus” uncertainty is shallower, and therefore should not be greater than the focal depth in absolute value. If only one depth uncertainty is encountered, it should be interpreted as a symmetric uncertainty. It is important to put the decimal place into the depth field, even if precision is only to the nearest kilometer (or more), to ensure correct reading by a Fortran formatted read statement. The depth code in column 11 is an optional character flag that informs **mloc** about the nature of the depth constraint.

Magnitude Record

Column	Description
1:1	Record format flag “M”
3:3	Usage flag, optional (a1)
5:8	Magnitude (f4.2)
10:14	Magnitude scale, optional (a5)
16:110	Author and comments, optional (a95)
112:121	Magnitude ID, optional (a10)

Note: **mloc** recognizes a magnitude record in which the usage flag is “=” as the preferred magnitude for this event. **mloc** uses only the first two characters of magnitude type. Magnitude ID would normally be the OrID from a hypocenter record; it should be right-justified in the field.

Phase Reading Record

Column	Description
1:1	Line format flag “P”
3:3	Usage flag, optional (a1)
5:9	Station code (a5)
12:17	Epicentral distance, optional (f6.2)
19:21	Azimuth, event to station, optional (i3)
23:23	Prevent phase re-identification flag, optional (“!”)
24:31	Input phase name, optional (a8)
33:36	Arrival time year (i4)
38:39	Arrival time month (i2)
41:42	Arrival time day (i2)
44:45	Arrival time hour (i2)
47:48	Arrival time minute (i2)
50:55	Arrival time seconds (f6.3)
57:58	Reading Precision, optional (i2)
60:64	TT residual, optional (f5.1)
66:73	Original phase name, optional (a8)
75:79	Agency, optional (a5)
81:88	Deployment or network, optional (a8)
90:94	Station, optional (a5)
96:97	Location, optional (a2)
99:101	Channel, optional (a3)
103:110	Author, optional (a8)
112:121	Arrival ID, optional (a10)

Note: It is helpful, but not required, to place a “dot” between the ADSLC fields. The usage flag carries the variable “fcode” (also known as *phase reading flags*) in **mloc**. The defined phase reading flags are listed here.

Comment Record

Column	Description
1:1	Line format flag “#”
2:121	Comment, optional (a120)

Stop Record

Column	Description
1:1	Line format flag “S”

Note: Only the “S” in column 1 is required, but for better readability it is useful to write “STOP” in columns 1:4.

End of File Record

Column	Description
1:3	Line format flag “EOF”

MLOC Native Format (MNF) v1.4

The current version is v1.4.2, released on November 16, 2017 with **mloc** v10.4.0.

This variant of the MNF format was developed in late 2014 to meet the need for an output file that could be easily parsed for inclusion in the U.S. Geological Survey’s National Earthquake Information Center (NEIC) ComCat database. An output file in MNF v1.4 format is a single text file containing an extensive documentation of the results of a relocation, including information on the hypocentroid, arrival time data and residuals for all events, most of the details about how the relocation was conducted, optional commentary on the composition of the cluster and specifics of the relocation process, all necessary station coordinates, and the crustal model used to calculate theoretical travel times for local and regional distance phases. It is also used as the standard data file format for the GCCEL project. Output files in this format are created only though the use of the ccat command.

Like MNF v1.3, MNF v1.4 is a fixed format based on the common concept of a small set of distinct record types with different formats, identified by a character flag in the first column. Each record is a single line. Each type of information (e.g., event, hypocenter, magnitude, phase arrival) has a specific record type, and there are a few utility record types. Although MNF v1.4 has many record types (and flags) in common with MNF v1.3, in most cases the details of the record formats are different.

The currently-defined record types and their flags are given in the following table:

Record Types of MNF v1.4.2

Flag	Record Type	Minimum Length	Full Length
B	Bulletin	1	121
F	Format Version	9	9
E	Event	1	121
H	Hypocenter	121	121

M	Magnitude	8	110
P	Phase Reading	79	79
#	Comment	1	121
C	Station Coordinates	49	49
S	Stop Event	1	4
L	Layer Velocity	34	34
EOF	End of file	3	3

Unlike all other record types, which are distinguished by the flag in column 1, the end-of-file record uses columns 1-3; it has no other arguments. It should only be found once, at the end of the .comcat file.

An MNF v1.4 file (or "comcat" file) always starts with a bulletin record, and it will carry a descriptive comment if it was created by **mloc**. The bulletin record is always followed by a format record. If there is commentary describing the most important features of the earthquake cluster and its relocation, which is highly recommended, it will follow the format record as a series of comment records. If a custom crustal velocity model has been used this section will be followed by a series of layer velocity records. Otherwise it should be assumed that the ak135 travel-time model was used for all phases. There will next be a series of station coordinate records for all stations used in the relocation. This constitutes the header block of a comcat file. This is followed by a set of event blocks. The comcat file is terminated by an end of file record.

Data for a single event is carried in a block of records that must start with an event record and end with a stop event record. Within the block, data is carried in a combination of hypocenter, magnitude, and phase reading records. Only one hypocenter record is permitted. Magnitude records are optional but there is no limit to how many can be supplied.

Optional Fields

All fields defined below will normally be present in a comcat file written by **mloc**. Under certain circumstances the following record types might be absent:

- Comment records, if no commentary text has been provided.
- Layer velocity records, if ak135 was used for all travel-time calculations.
- Magnitude records, if an event has no magnitude estimates.

Defined Record Types

Bulletin Record

Column	**Description**
1:1	Record format flag "B"

5:121	Bulletin description, optional (a117)

Format Record

Column	Description
1:1	Record format flag “F”
5:9	Format version (a5)

Comment Record

Column	Description
1:1	Record format flag “C”
2:121	Comment, optional (a120)

Layer Velocity Record

Column	Description
1:1	Record format flag “L”
8:14	Depth or layer thickness (f7.3)
20:24	Vp (f5.3)
30:34	Vs (f5.3)

Note: **mloc** writes the crustal velocity model with two values for each layer, the depth of the upper and lower interface. Internal interface depths are therefore repeated, giving velocities above and below. This accommodates a model in which a layer can have a linear gradient in velocity. Alternatively, a flat-layered model could be defined with one line per layer, in which the first parameter is interpreted as layer thickness. Models defined this way are not presently supported by **mloc**, but could become relevant in the future. Depth/layer thickness is given in km. Velocities are given in km/s.

Station Coordinates Record

Column	Description
1:1	Record format flag “C”
3:8	Station Code (a6)
10:14	Agency (a5)
16:23	Deployment (a8)
25:32	Latitude (f8.4)
34:42	Longitude (f9.4)

44:49	Elevation (i6)

Note: Station elevation is the elevation of the instrument, given in meters, relative to mean sea level (positive or negative).

Event Record

Column	**Description**
1:1	Record format flag “E”
5:124	Event ID (a116)

Note: Event IDs are built up from the prefix “cec_” (Calibrated Earthquake Cluster), the cluster name, and the event number within the cluster.

Hypocenter Record

Column	**Description**
1:1	Record format flag “H”
5:8	Year (i4)
10:11	Month (i2)
13:14	Day (i2)
16:17	Hour (i2)
19:20	Minute (i2)
22:26	Seconds (f5.2)
28:32	Origin Time Uncertainty (f5.2)
35:42	Latitude (f8.4)
44:52	Longitude (f9.4)
54:56	Smin Azimuth (i3)
58:62	Error Ellipse Smin (f5.2)
64:68	Error Ellipse Smaj (f5.2)
70:74	Focal Depth (f5.1)
76:76	Depth Code (a1)
78:82	Plus Depth Uncertainty (f5.1)
84:88	Minus Depth Uncertainty (f5.1)
90:93	GTCNU (a4)
95:102	Author (a8)
104:121	Cluster ID (a18)

Note: Both depth uncertainties are provided as positive numbers. “Plus” depth uncertainty is on the deeper side; “Minus” uncertainty is shallower, and therefore should not be greater than the focal depth in absolute value.

The “GTCNU” field carries a four-character code relating to calibration status (I prefer this term to “ground truth” level). I have developed a nomenclature called GTCNU to provide much more detailed information on the subject of what hypocentral parameters are considered to be calibrated (i.e., thought to be bias-free), documented as Location Accuracy Codes.

The “depth code” in column 76 defines the nature of the depth constraint. Standard values are defined here.

Magnitude Record

Column	Description
1:1	Record format flag “M”
5:8	Magnitude (f4.2)
10:14	Magnitude Scale (a5)
16:110	Author and Comments (a95)

MLOC Native Format (MNF) v1.5

The current version of the format is v1.5.0, released on March 20, 2016 with **mloc** 10.3.0.

MNF v1.5 format is the only format that can be used to read differential time data in **mloc**, using the diff command. It is read by the subroutine *read_mnf_15* in the module *mlocio_mnf.f90*. Differential time data consists of time differences between onsets of the same phase recorded at the same station for two different events. The formulation used by **mloc** is described here.

Format Description

Like other varieties of MNF, v1.5 uses several types of ‘records’ with defined formats and a character in column 1 that defines the type of record. Unlike the other MNF formats, which are event-oriented (i.e., records are grouped by event), v1.5 data files are cluster-oriented, and each differential time record carries references to two events in the cluster. Only 4 record types are defined:

Record Types of MNF v1.5

Flag	Record Type	Minimum Length	Full Length
F	Format	14	14
D	Differential Time	87	149
#	Comment	1	149

EOF	End of file	3	3

Unlike all other record types, which are distinguished by the flag in column 1, the end-of-file record ("EOF") uses columns 1:3; it has no other arguments. It causes processing of a data file to end, so it would normally only be found once, at the end of the file. If an end-of-file record is placed in the middle of a file, processing will stop there.

The differential time record carries a "usage" flag (in column 3) that determines the way (or if) the information in that record will be used by **mloc**. None of the other record types carries a usage flag.

Comment records (flag "#") can be inserted anywhere in an MNF-formatted file, but should not be the first or last record. Obviously, any text that occurs after the first end-of-file record will not be processed and can be considered as a comment, regardless of the formatting, but the use of the end-of-file record in this manner is not recommended. Information about the nature of the differential time data set can be carried in comment records.

File Structure

The first line in a data file must be a format record. This is followed by some number of differential time records (and comment records, optionally) and the file must end with an end-of-file record.

Defined Record Types

The concept of "optional" fields in the descriptions of record types is specifically in the context of use by **mloc**. In writing MNF-formatted data files it is advisable to pad lines to the full length of that record type, which is based on the defined fields, not the required fields, and it is not unwise to pad all lines to 149 characters, the full length of the longest defined record types, regardless of record type.

Format Version Record

Column	**Description**
1:1	Record format flag "F"
10:14	Format version (a5)

It is useful for readability to include extra text, such that columns 5:9 read "MNF v", but all that is required is the "F" in column 1 and the version number in columns 10:14. Version number is treated by **mloc** as a character string that can accommodate three-level versioning (e.g., '1.5.0').

Differential Time Record

Column	**Description**

1:1	Record format flag “D”
3:3	Usage flag (a1), optional
5:20	Template Event Designator (a16)
22:31	Template Event EVID (a10), optional
33:48	Target Event Designator (a16)
50:59	Target Event EVID (a10), optional
61:66	Station (a6)
68:75	Phase (a8)
77:87	Reduced Relative Arrival Time (f11.4)
89:90	Reading Precision (i2), optional
92:97	Uncertainty, s (f6.4), optional
99:103	Correlation Coefficient (f5.3), optional
105:112	Original phase name, optional (a8)
114:118	Agency, optional (a5)
120:127	Deployment or network, optional (a8)
129:133	Station, optional (a5)
135:136	Location, optional (a2)
138:140	Channel, optional (a3)
142:149	Author, optional (a8)

The usage field is used here for phase reading flags, usually to indicate that the reading will not be used.

The “event designator” field for the template and target events carries an identifier for each event based on an estimate of the date and origin time, to the nearest second. The format is “yyyymmdd.hhmm.ss” with the character ‘0’ filling any blanks. Although the identifier could be read as numeric values for year, month, day, etc, **mloc** treats it as a character string that is only used to make the correct association with two corresponding events in the cluster being relocated. An equivalent event identifier is declared for each event in the **mloc** command file, using the even command. Therefore it is desirable to ensure that the integer seconds part of the identifiers match. The evid (event id) fields are provided for the same reason. While evids are not always available, if they are they provide a more robust way to make the association with events in the cluster. **mloc** attempts to make the match using evids first.

The definition of ‘reduced relative arrival time’ is discussed elsewhere.

Reading precision is an integer that indicates the number of significant decimal places in the reduced relative arrival time datum. It is used in **mloc** to correctly format output. The values appropriate for differential time data are:

Value	Meaning
0	nearest second
-1	nearest tenth
-2	nearest hundredth
-3	nearest thousandth
-4	nearest ten-thousandth

If no value for reading precision is provided, **mloc** attempts to determine it from the formatting of the datum. No great harm will be done if this process is inaccurate, but specification of a reading precision is strongly recommended as good practice.

If an uncertainty for the estimate of reduced relative arrival time is provided, it will be read and may be used in **mloc**, at least initially, although it can be over-ridden in various ways. It is desirable to have an estimate of uncertainty from the cross-correlation analysis but **mloc** can proceed without one.

If the correlation coefficient field is non-blank, it is read by **mloc**. It can be used on input to filter out differential times with correlation coefficients below a threshold set by the user. It may also be helpful in deciding whether or not to flag a particular datum as an outlier.

The “original phase name” field provides a way for the user to change the phase name of a differential time datum without losing the information on the original phase ID. The same facility exists in the MNF v1.3 format for arrival time data, but this is likely to be a rare need with differential time data.

The next five fields, all optional, identify the seismograph instrumentation from which the measurement of reduced relative arrival time was made, in the new IASPEI Station Coding Standard. Known as the ADSLC code, this identification standard greatly expands the traditional concept of describing the source of seismic data by a single 3-5 character station code. See the documentation for MNF v1.3 for a fuller discussion of this, especially the rationale for carrying station code in two places in the record. Channel information may be of special significance with differential time data because it is common practice to carry out the cross-correlation analysis on more than one channel of a station and select the one with the highest correlation coefficient for use in relocation. Therefore differential time data sets may well include multiple estimates of reduced relative arrival time, differing only in the channel that was analyzed.

The author field identifies the source (usually a person) of the datum. The field is optional but it is strongly recommended that it be utilized. The character string employed to identify someone is completely arbitrary, but it is limited to 8 characters in **mloc**.

Comment Record

Column	Description

1:1	Record format flag “#”
2:149	Comment (a148), optional

The length of the comment field is rather arbitrary but it is convenient to limit it to the length of the main ‘data’ record, in this case the differential time record. **mloc** ignores comment records.

End-of-file Record

Column	Description
1:3	Record format flag “EOF”

Example

Here is an example, a portion of a dataset provided for a study of North Korean nuclear tests by Dr. Michael Begnaud:

```
F   MNF v1.5
D   20090525.0054.42          20061009.0135.27          INCN   Pn        2444.9006 -4 0.0341      Pn          .       .INCN .  .   MBegnaud
D   20090525.0054.42          20061009.0135.27          INCN   Pb        2444.8794 -4 0.0817      Pb          .       .INCN .  .   MBegnaud
D   20090525.0054.42          20061009.0135.27          INCN   Pg        2444.9888 -4 0.0820      Pg          .       .INCN .  .   MBegnaud
D   20090525.0054.42          20061009.0135.27          HIA    Pn        2445.0637 -4 0.0410      Pn          .       .HIA  .  .   MBegnaud
D   20130212.0257.51          20061009.0135.27          HIA    Pn       -4943.2456 -4 0.0395      Pn          .       .HIA  .  .   MBegnaud
D   20090525.0054.42          20061009.0135.27          MDJ    Pn        2444.8804 -4 0.0160      Pn          .       .MDJ  .  .   MBegnaud
D   20130212.0257.51          20061009.0135.27          MDJ    Pn       -4943.4370 -4 0.0180      Pn          .       .MDJ  .  .   MBegnaud
D   20090525.0054.42          20061009.0135.27          MDJ    Pg        2444.8831 -4 0.0382      Pg          .       .MDJ  .  .   MBegnaud
D   20130212.0257.51          20061009.0135.27          MDJ    Pg       -4943.4678 -4 0.0365      Pg          .       .MDJ  .  .   MBegnaud
```

The original data were provided by Dr. Begnaud in a much different format and a conversion code specifically for this dataset was written. Differential time data from other sources almost invariably uses a unique format and will require its own conversion code.

MNF Utility Codes

Several types of utility programs are essential to the efficient use of **mloc**. These include programs for converting arrival time datasets to **mloc's** native data format (MNF) and a program for extracting event files from MNF-formatted bulletins (concatenations of event files). The source codes for these utilities are contained in the **mloc** distribution in the directory */mloc_distribution/mnf_utilities/*. See MLOC Utility Codes for some other important utility programs.

Data Format Conversion Codes

These programs convert files written in two of the most commonly encountered data formats for arrival times. In most cases the input files are bulletins, containing data for multiple events, and the conversion process will include a choice of having the output be a converted bulletin or a set of individual event files. Contact me if you need help with conversion from other formats.

- isc_ims2mnf: conversion of IMS-formatted data from the ISC.
- seisan2mnf: conversion from the "nordic" format produced by SEISAN.

isc_ims2mnf

isc_ims2mnf handles conversion to MNF format from the version of the IMS format returned by the bulletin search function of the ISC website. The process for acquiring data from the ISC Bulletin is described here. This will usually be a bulletin, i.e., many events in a single file. To convert the bulletin to MNF, copy the executable of **isc_ims2mnf** into the same directory and run it. The only input is the name of the IMS-formatted file. The output will have the same name with ".mnf" appended.

The source code for **isc_ims2mnf** and an executable for macOS will be found in */mloc_distribution/mnf_utilities/mnf_search/*.

Warning! If the search of the ISC Bulletin has used a condition on magnitude there will be a line near the top of the file listing that condition. The conversion program will try to process that line as being part of an event and crash. The solution is simple: add an "#" to the beginning of that magnitude condition line.

Once the bulletin is converted to MNF format, use the program mnf_search to extract individual events for relocation in **mloc**.

seisan2mnf

This program handles conversion to MNF from the "nordic" format produced by SEISAN. SEISAN is widely used by local and regional seismograph networks around the world. Unfortunately, the formats of event data files produced by these many installations do not always agree in detail, causing errors in conversion or outright crashes. In such cases it is necessary to edit the code of **seisan2mnf** (and rename it appropriately) for use with data files from specific

networks. In any case it is always advisable to do a careful examination of the converted data files afterward.

The source code for **seisan2mnf** and an executable for macOS will be found in */mloc_distribution/mnf_utilities/mnf_search/*.

Once the bulletin is converted to MNF format, use the program mnf_search to extract individual events for relocation in **mloc**.

Codes to Manipulate Event and Bulletin Files

mnf_search

mnf_search is a program for searching an MNF-formatted bulletin and extracting individual event files for input (using the inpu command, usually in a command file) to **mloc**. It can also create the event definition section of a command file for the selected events. The source code and an executable for macOS will be found in */mloc_distribution/mnf_utilities/mnf_search/*.

To use mnf_search, copy the executable into the directory containing the MNF-formatted bulletin. The user is first asked for the file name of the bulletin, followed by the choice:

```
Create a new bulletin (1) or individual event files (2)?
```

Response is "1" or "2". Input to **mloc** requires individual event files, but in some cases, such as dealing with a bulletin covering a very large area, it may be desirable to first create a new, smaller bulletin with events that are most suitable for the relocation analysis and then select individual events for **mloc** in a second run. If individual events (2) is chosen, the next question will be:

```
Create mloc command file?
```

The response can be "y" or "n". If "y", only the event definition section (commands memb, even and inpu) will be created, with the basename specified in the answer to the next question:

```
Enter command file basename:
salmas1.0
```

Next comes a set of "filter" questions that help narrow the selection process in useful ways:

```
Use lat-lon limits?
n
Use focal depth limit?
n
Use nearest station distance?
n
Use magnitude?
n
```

The "nearest station distance" criterion is useful in assembling a dataset that is likely to be suitable for direct calibration. The user specifies an epicentral distance and the number of

readings an event must have within that distance in order to be selected. After these choices, a list is displayed that is ordered according to the number of readings for each event that passes the search criteria:

```
   110 events read
   110 events pass the search criteria
 110 events  that pass the search criteria and have  10 or more phase readings
 110 events  that pass the search criteria and have  20 or more phase readings
 109 events  that pass the search criteria and have  30 or more phase readings
 106 events  that pass the search criteria and have  40 or more phase readings
 103 events  that pass the search criteria and have  50 or more phase readings
 103 events  that pass the search criteria and have  60 or more phase readings
 102 events  that pass the search criteria and have  70 or more phase readings
 101 events  that pass the search criteria and have  80 or more phase readings
  94 events  that pass the search criteria and have  90 or more phase readings
  93 events  that pass the search criteria and have 100 or more phase readings
  91 events  that pass the search criteria and have 110 or more phase readings
  84 events  that pass the search criteria and have 120 or more phase readings
  78 events  that pass the search criteria and have 130 or more phase readings
  74 events  that pass the search criteria and have 140 or more phase readings
  73 events  that pass the search criteria and have 150 or more phase readings
  68 events  that pass the search criteria and have 160 or more phase readings
  64 events  that pass the search criteria and have 170 or more phase readings
  63 events  that pass the search criteria and have 180 or more phase readings
  61 events  that pass the search criteria and have 190 or more phase readings
  61 events  that pass the search criteria and have 200 or more phase readings
Minimum number of phase arrivals:
```

In general, events with more readings will perform better in an **mloc** analysis because there are more station-phase instances in common with other events in the cluster. Given that **mloc** is limited to 200 events in a cluster it makes sense to prefer events with more readings. In fact a number of events around 50-75 is a comfortable compromise between having enough data for a robust inversion but not so much that all aspects of the processing (and runtime) are burdensome. So in many cases the user might decide how many events she wants to work with, find the corresponding number of readings and enter that number in answer to the question.

If a source region has a dense local network one might find that even a request for events with, say, 100 or more readings returns too many events (the example above is borderline in that regard). In that case an easy way to further filter the selection is with the magnitude criterion, asking for events with magnitude greater than 4, for example (same bulletin as above):

```
Use magnitude?
y
Enter minimum magnitude:
4.0
   110 events read
    70 events pass the search criteria
  70 events  that pass the search criteria and have  10 or more phase readings
  70 events  that pass the search criteria and have  20 or more phase readings
  69 events  that pass the search criteria and have  30 or more phase readings
  68 events  that pass the search criteria and have  40 or more phase readings
  68 events  that pass the search criteria and have  50 or more phase readings
```

```
  68 events  that pass the search criteria and have  60 or more phase readings
  68 events  that pass the search criteria and have  70 or more phase readings
  68 events  that pass the search criteria and have  80 or more phase readings
  65 events  that pass the search criteria and have  90 or more phase readings
  64 events  that pass the search criteria and have 100 or more phase readings
  64 events  that pass the search criteria and have 110 or more phase readings
  62 events  that pass the search criteria and have 120 or more phase readings
  59 events  that pass the search criteria and have 130 or more phase readings
  57 events  that pass the search criteria and have 140 or more phase readings
  56 events  that pass the search criteria and have 150 or more phase readings
  53 events  that pass the search criteria and have 160 or more phase readings
  53 events  that pass the search criteria and have 170 or more phase readings
  53 events  that pass the search criteria and have 180 or more phase readings
  52 events  that pass the search criteria and have 190 or more phase readings
  52 events  that pass the search criteria and have 200 or more phase readings
Minimum number of phase arrivals:
```

A good rule of thumb is that events with 30 or more readings usually behave well in a cluster. Events with fewer readings can work, in the sense of being stable, but events with fewer than 10 readings are often unstable and they almost never have acceptable levels of location accuracy. When the user provides the minimum number of readings, **mnf_search** completes the extraction process:

```
Minimum number of phase arrivals:
30
Event number selection: beginning and end numbers:
1 110
EOF reached after    110 events
  70 events selected
```

The selection of beginning and end numbers is only relevant if dealing with a very large bulletin that you want to process piece-wise. Otherwise, as in this case, the user just enters "1" and the number of events in the bulletin in order to search the entire bulletin. The event files (and command file, if requested) are written into the same directory as the bulletin. From there they should be copied into a directory ("cluster working directory" or "cluster series directory", e.g. "salmas1") under the mloc_working directory for relocation.

The sharp-eyed will notice that 70 events were selected when the listed said 69 would meet the search criteria. Minor discrepancies like this are common with **mnf_search** and simply reflect weaknesses in the coding. Perhaps the next editing session will correct it, but it has a fairly low priority.

Input

This section summarizes what sorts of input is needed to set up a cluster for relocation with **mloc**, focusing on the basic types of files and interactive input as well as some of the more specialized data files that may be needed. It is assumed that the user has installed the basic **mloc** distribution.

Event Data Files

For a basic relocation analysis **mloc** needs a set of event files and a command file. Even the command file could be bypassed if the user were willing to enter all the event-definition commands (e.g., memb, even and inpu) interactively. It is far more convenient to let the utility code mnf_search create the event definition bloc of the command file during the process of extracting a set of events from an MNF Bulletin.

Command File

With a command file consisting only of the event definition blocs of the events to be relocated (or interactively-entered event blocs), **mloc** could be run in default mode. The main features of this mode would be:

- No calibration is done
- Inverse weighting of data, using default values of reading errors
- Phase re-identification would be done
- Hypocentroid would be located using teleseismic (30-90°) P arrivals only
- All travel times would be computed with ak135
- Arrivals at all distances would be used for cluster vectors
- All parameters (latitude, longitude, depth and origin time) would be free
- Starting locations would be taken from the preferred hypocenter of each event file
- A basemap plot will be made, but no other plots

Rather than depending on the default values of the major commands, it is wise to specify them explicitly in the command file. Also there are some (actually a great many) useful features of **mloc** that are not implemented by the defaults. A slightly expanded basic set of commands for the "header" section of a command file for an uncalibrated cluster analysis might be:

- pltt 1 2 3 5 6 7 8
- dem1 on
- bdps tables/stn/bdps.dat
- dcal off

- phyp on
- hlim 30. 90.
- clim 0. 180.
- wind 3 4
- frec 1 1 0 1
- freh 1 1 0 1
- skip * T *

Note that in this command set, depth is a fixed parameter (commands freh and frec), a better choice unless you have an exceptionally good dataset. If direct calibration were to be done on this cluster several commands would be changed:

- pltt 1 2 3 4 5 6 7 8 ! We definitely want the near-source TT plot for direct calibration
- dcal on
- phyp off
- hlim 0. 1. ! The exact distance range depends on the dataset

Other Issues

Beyond the basics mentioned above, there are several other broad classes of concerns related to input to **mloc**. These will involve changes to the command file and/or the interactive command input as well as creation or editing of files (event files, station files, crustal model files, etc) that will be referenced by one of **mloc's** commands.

Crustal models

A custom crustal model with which to calculate travel times for local and regional phases is nearly always required when performing a direct calibration analysis and it may be useful even in an uncalibrated analysis or an indirect calibration analysis. In this case the input to **mloc** must include the lmod command to specify the crustal model file:

- lmod /my_cluster1/my_model.cr ! pathname is relative to the mloc_working directory

Differential Time Data

To supplement the direct arrival time data normally used in **mloc** and contained in the event files with differential time data requires preparation of a special data file in MNF v1.5 format for the cluster of interest. The file is specified in **mloc** using the diff command.

S-P Data

S-P differential phase data are event-specific and can be carried by the standard MNF v1.3 event file format. They normally need to be entered by hand, however. There is a specialized plot type for S-P data

Station Files

The seismic station coordinate information needed for many clusters can be found in the master station file distributed with **mloc**. It is not uncommon, however, to require one or more supplemental station files to make use of all the available arrival time data for a cluster (command sstn). The nsmd command can be helpful in creating such files, especially for clusters featuring data from one or more of the U.S. regional networks. Several of the formats supported by **mloc** for supplemental station files are those established by other organizations (e.g., ISC) or software packages (e.g., SEISAN).

High-Resolution Topography

User-supplied digital elevation models (DEMs) can be read by **mloc** for use in making certain plots with topography at higher resolution than the default DEMs distributed with **mloc**.

Faults

Plotting of fault traces in **mloc's** map-like plots requires the user to provide a data file which will be read by the command fmap. A sample file is included in the mloc distribution.

Arrival Time Data

mloc uses several types of arrival time data in relocation. The most important data are traditional phase arrival times, which are carried in individual event files formatted in MNF v1.3.The event files used in mloc nearly always must be converted from a different format. Several utility programs for converting from widely-available data formats are provided in */mloc_distribution/ mnf_utilities/* and discussed here.

Event files have the filename suffix *.mnf*. The *strongly* recommended naming convention is to base the body of the filename on some estimate of the origin time of the event, down to the nearest second:

- YYYYMMDD.HHMM.SS.mnf

It is not important that the origin time represented in the filename be especially accurate, only that it be distinguishable from other events in the cluster.

In addition to the standard set of body wave phases, **mloc** is able to use Lg arrival times and T-phase arrival times. Commands lgtt and tptt are provided to allow the user to adjust the travel time models used to generate theoretical travel times for these phases.

S-P Data

The MNF v1.3 format can also accommodate S-P (differential phase) times, as described here.

Differential Time Data

Like all relative location techniques, the Hypocentroidal Decomposition algorithm converts observed phase arrival times from different events to common stations into differential times, but **mloc** was not originally developed to exploit differential time data as a direct input. Such data are now often available through waveform cross-correlation analyses, and a method for using differential time data from other sources has been implemented in **mloc**, using the MNF v1.5 format.

Differential Time Data

Like all algorithms for improving the estimates of relative location for a clustered set of seismic events, **mloc** turns absolute arrival time data into differential times, i.e., the time lag between observations of the same phase at the same station for two different events. The Double Difference algorithm (Waldhauser and Ellsworth, 2000) popularized the use of directly measured differential times, using waveform cross-correlation on digitally-recorded data, for input to a multiple event relocation algorithm.

This approach is especially valuable in the context of routine network operations, where it is possible to automate the processing of consistently-available and well-calibrated digital data streams to obtain the necessary measurements. It is less easily applied to the kinds of analyses typically undertaken with **mloc**, where the set of stations providing data is usually very heterogenous and digital waveforms would be either impossible or very difficult to obtain. Nevertheless, differential time datasets are available for some events of interest, and in keeping with the "omniverous" data usage policy of **mloc** it is desirable to be able to use differential time datasets when they can be obtained.

Since data input to **mloc** is event-based, it is not feasible to reference a second event using the standard MNF v1.3 format. Therefore a special format for reading differential time data in **mloc** was designed: MNF v1.5. No conversion code is supplied with **mloc**; the user will need to write one for each distinct source of differential time data.

Description

Each line of the input file references two events, by the event names assigned with the even command. If an event referenced in a differential time datum is not found in the cluster, a warning is given.

The input datum is "reduced relative arrival time", the time difference between the two arrivals if they are treated as occurring on the same day. This is converted to dummy arrival times for the associated pair of events. For the template event the theoretical TT is added to the origin time of the input data file. For the target event, the reduced relative arrival time is added to the template

event's dummy arrival time. Dummy arrival times derived from differential time data are only used for cluster vectors, never the hypocentroid.

Phase names for differential time data are not altered from the input file and phase re-identification is disabled. There is no testing for duplicate readings. Differential time data for Lg are permitted, but otherwise only phases that are in the tau-p phase list are processed.

Uncertainty

Most (but not all) cross-correlation algorithms produce an estimate of the uncertainty of the differential time datum, but this value underestimates the true uncertainty in the context of relocation even if it is "perfect" in itself because we assume a single theoretical travel time model for the entire cluster, on which the dummy arrival times are based. For a cluster of finite extent (many are 50-100 km across) lateral variations in crustal strucure or even over the full raypath lead to additional scatter.

It is generally true that waveform cross-correlation decreases the scatter in differential time measurements over what is possible by differencing raw arrival time data from standard seismic bulletins such as the ISC Bulletin, but not (in my experience) as much as the formal uncertainties that come with some cross-correlation analyses would imply. This conclusion is based on the application of differential time data to a dozen or so clusters, using measurements from researchers using three different methodologies. Through the **mloc** analysis we obtain empirical reading errors (scatter) for the differential time data just as for "regular" arrival time picks. The empirically-observed scatter (even when the relocation is dominated by the differential time data, is nearly always larger than the formal uncertainty from the cross-correlation analysis.

S-P Data

Development of the facility for handling S-P data in **mloc** was motivated by the availability of picks from a widely distributed network of strong motion instruments in Iran, the ISMN. Because there are so many stations, distributed in the most earthquake-prone regions, ISMN stations are often the closest station to an event of interest, and therefore play a major role in depth constraint. Unfortunately most of the ISMN stations do not have calibrated timing so the absolute arrival times of P and S arrivals (which are often clearly recorded) cannot be used in the location analysis. The differential phase S-P still retains information on the distance of the event from the station, however, so readings at very close stations do provide some constraint on focal depth.

S-P data are carried in the standard event file used by **mloc**. Because there are normally few such readings they are entered by hand. Here is an example, the first few records for an event from the Bojnurd cluster in Iran:

```
F   MNF v1.3
E   Iran-Turkmenistan border region                                                    7391378
H = 2004/08/21 03 32 40.98         37.8979   57.6360                 10.0                    ISC                7439524
M   4.7  mb    ISC                                                                            7439524
M   4.1  MS    ISC                                                                            7439524
P   QUV               Pg       2004  8 21  3 33  2.57  -2   0.4 Pg           .       .QUV  .  .B Z RGhods
P   NJV               Pn       2004  8 21  3 33 21.37  -2   1.9 Pn           .       .NJV  .  .B Z RGhods
```

```
P x KHV              Pn       2004  8 21  3 33 42.82  -2   1.5 Pn           .        .KHV  .  .B Z RGhods
P   BJV              Pg       2004  8 21  3 32 48.44  -2  -0.4 Pg           .        .BJV  .  .B Z RGhods
P   BJV              Sg       2004  8 21  3 32 54.86  -2   0.0 Sg           .        .BJV  .  .B E RGhods
P   BRV              Pn       2004  8 21  3 34  1.91  -2   1.4 Pn           .        .BRV  .  .B Z RGhods
P   A683             S-P      2004  8 21  0  0  3.38  -2  -0.1 S-P          .        .A683 .  .G N RGhods
P   A675             S-P      2004  8 21  0  0  4.99  -2   0.3 S-P          .        .A675 .  .G N RGhods
P   A681             S-P      2004  8 21  0  0  5.55  -2   0.0 S-P          .        .A681 .  .G N RGhods
P   A682             S-P      2004  8 21  0  0  6.19  -2   0.0 S-P          .        .A682 .  .G N RGhods
P   A668             S-P      2004  8 21  0  0  8.08  -2   0.3 S-P          .        .A668 .  .G Z RGhods
```

The first six phase records are from standard seismograph stations with calibrated timing. The next five readings are S-P readings from ISMN stations, all read by Reza Ghods. Notice that the "arrival time" for the S-P readings does carry the year, month, and day, but not hour or minute. The S-P time itself is entered in the seconds field of the arrival time.

If only the S-P time is available the format used above is fine. If the "parent" P and S times are available, however, I recommend a more complete record, in which the P and S arrivals are logged but given the "t" flag to indicate a timing problem:

```
P   B523             S-P      2006  6  3  0  0  3.16  -2       S-P      BHRC .ISMN    .B523 .  .    RGhods
P t B523             Pg       2006  6  3  7 15 53.38  -2       Pg           .        .B523 .  .G Z ARGhods
P t B523             Sg       2006  6  3  7 15 56.54  -2       Sg           .        .B523 .  .G Z ARGhods
```

Output

If any S-P data are used in direct calibration, **mloc** keeps a record of the associated stations in the ~.stn file.

One of the options of the pltt command (plot index 9) creates a plot of S-P times as a function of distance.

Command Files

Control over the processing of **mloc** is managed by commands, of which there are about 70. Most commands are four characters in length (a few are three characters), always given in lowercase. Only a small number of commands are always required for a run. Commands may be entered interactively from the Terminal or read from a text file containing the commands and arguments. In practice we use both approaches, starting with a basic command file that handles commands that are mainly repeated from run to run, and issuing a small number of commands interactively to configure and launch the run. Indeed, a command file cannot be read except via an interactive command (cfil). It is common to edit the command file slightly between runs, but the amount of editing depends on how much control you want to off-load to interactive input. In any case most of the information (notably the event definitions) in a command file stays constant from run to run.

Command files can be given any name, and you could use a simple name (e.g., *mloc.cfil*) and simply edit it from run to run, keeping the same name. A better practice is to make a new command file for each run (i.e., make a duplicate of the command file from the previous run) and name it after the run ID, e.g., *my_cluster1.2.cfil* for the second run in the first series of runs for a cluster named *my_cluster*. Then it is saved with all the output files to provide a clear record of how that run was set up. The filename suffix for a command file *.cfil* is standard, but it is not required. Command files are stored in the cluster directory with the event data files.

Processing a Command File

When **mloc** is launched there are several interactive steps that occur after a few program limits are listed:

- Giving a name for the run (used to name all output files)
- Giving the name of the subdirectory of the **mloc** working directory where the data files for this cluster are stored

```
$ mloc_a1047
mloc v10.4.7, release date 7/14/2019

Current program limits:
  nevmax =   200
  nqmax  =  4000
  ntmax1 = 35000

Enter a basename for this run: my_cluster1.2
Enter the name of the data directory: my_cluster1
```

After this, the program lists all the available commands and asks for interactive command input. Any command can be entered interactively at this point but the normal procedure is to tell **mloc** to process the command file that has been prepared for this run:

```
cfil my_cluster1.2.cfil
```

During processing of the command file (in the main program *mloc.f90*) the user may be asked for some additional input, or warning messages may be issued. The processing of commands is handled by the subroutines in *mloc_commands.f90*. When the command file is finished the program asks for more interactive input. The relocation does not begin until the user issues the run command.

Command File Structure

Command files consist of two primary sections. The first section contains commands that control the procedures that will be used for the relocation, affecting all events. The second section, beginning with the first memb command, defines the events to be included in the cluster for this run and allows the user to issue commands which are event-specific. Some commands can be issued in either section but have a different action, depending on whether they are issued before any events have been defined.

Order of commands

Within the first section there is considerable (maybe total, but it has not been tested) flexibility in the order in which commands are given. In each event definition block, the memb command must come first but otherwise the order of commands is arbitrary. Within the event definition section as a whole it is strongly advised to keep events in chronological order, but it is not required.

Killing events

The easiest way to remove an event from a cluster is to delete the lines of its event definition block from the command file, but there may be reasons to want to keep a record of its existence in the command file even though it is not being relocated. It is also quite common to temporarily "kill" events in order to work with a smaller or better-constrained set of events.

mloc has two methods to meet these needs. To eliminate a contiguous block of events the kill command (with arguments *on* and *off*) is used. To kill a single event, the preferred method is to use *kill* as an argument after the memb command.

Comments

Any text after an exclamation point "!" in a command line is taken as a comment. There is also a "comment" command comm which is useful for comments that are not command-specific and for temporarily preventing a command from running.

Repeated commands

Most commands can be repeated; the last instance is the one that will control the relocation. This feature is used often for the dep_ family of commands that specify the starting focal depth. The fourth character is taken as a flag to indicate the source of the depth constraint. It is very common to have several estimates available for the depth of an event and it is useful to be able to keep them all at hand in the command file.

Defaults

It is possible (but mostly pointless) to make a run of **mloc** with no commands except the ones that define events, because all parameters that control how the relocation is done have defaults. The defaults are defined in the main program *mloc.f90*. In this case the relocation would be done using teleseismic P arrivals and the ak135 travel time model. Data would be weighted inversely to their uncertainty (using the phase-specific default reading errors) and phase re-identification would be done. All four hypocentral parameters would be free. Only a base plot would be made, as well as the core output text files.

Defining Events

Events are defined with a minimum of three commands, for example:

```
memb
even 20090807.1859.26
inpu 20090807.1859.26.mnf
```

The memb command starts the definition of a new event. The even command provides an identifying name for the event that will appear throughout the output files. It is normally derived from the filename of the event data file itself, which is supplied in the inpu command. Event data files are normally named with an origin time, which does not have to exactly match that of any

hypocenter (there can be several) in the event file itself but should be pretty close. Additional commands could be given and they would apply only to the current event until another memb command is encountered.

Except for the very smallest clusters, assembly of the event definition section would be very tedious to do by hand, so it is normally done by one of several utilities, especially ones that search through a concatenated set of event files (i.e., a bulletin) and extract individual events for a cluster.

Terminating a Command File

No special steps are required to terminate a command file. It normally ends with the definition of the last event.

An Example

This is the command file from the ridgecrest3.1 run of **mloc** for the Ridgecrest, California cluster that was used to illustrate the various summary plots, with comments added for many of the commands to explain their function. Most of the event definition section has been removed.

```
pltt 1 2 3 4 5 6 7 8 ! All standard plots except S-P (tt9)
comm ccat ridgecrest_summary.txt ! This command is commented out
comm plot 1: July 4, 2019 sequence ! Comment describing the selected event plot
fdhp on ! Make the plot of a histogram of focal depths
rdpp all ! Make relative depth phase plots for any event that has the appropriate readings
eplt on ! Make an ellipse plot
splt on ! Male a seismicity plot
epap Pn 15. 0 ! Make an Empirical Path Anomaly plot for Pn out to 15°, no lines connecting source and station
tt5e 20190704.1733.49 ! Make a local travel time plot (tt5) for the event 20190704.1733.49
tt5s LRL ! Make a station-specific travel-time plot (tt5) for station LRL
dem1 globe ! Use the GLOBE DEM for topography
sstn ridgecrest3/ridgecrest_stn.dat ! Supplemental station file for this cluster
lmod ridgecrest3/ridgecrest.cr ! Local crustal model
bdps tables/stn/bdps.dat ! List of stations that report bogus depth phases
ttou Lg ! Make an output file of Lg travel times and distances
lgtt 0.03 31.68 1.6 ! Custom Lg travel time model, y-intercept and slope (s/deg), used beyond 1.6°
ppri pP ! Do not re-identify pP phases
ppri sP ! Do not re-identify sP phases
rhdf ridgecrest2.42.hdf_dcal ! Take starting locations from a previous run
rfil ridgecrest2.42.rderr ! Take empirical reading errors from a previous run
tfil ridgecrest2.42.ttsprd ! Take data on the spread of residuals for different phases from a previous run
dcal on ! Use direct calibration
phyp off ! Use both P and S phases to locate the hypocentroid
hlim 0. 0.8 ! Use arrival time data out to 0.8° for the hypocentroid
clim 0. 180. ! Use arrival time data at all epicentral distances to estimate cluster vectors (relative locations)
wind 3 4 ! Windowing limit and taper, in terms of the spread of each phase, to disregard gross outliers
frec 1 1 0 1 ! Free parameters for cluster vectors (lat, lon, OT)
freh 1 1 0 1 ! Free parameters for hypocentroid (lat, lon, OT)
depc 15 ! Cluster default depth, will be over-ridden by rhdf command
memb ! Declare a new event
even 19930520.2014.14 ! Event name
inpu 19930520.2014.14.mnf ! Input file
depm 15.3 1.0 ! focal depth and uncertainty from a free-depth relocation, supercedes rhdf and depc commands
depn 13 3 ! focal depth and uncertainty from near-source data, supercedes all previous depth estimates
memb
even 19950920.2327.35
inpu 19950920.2327.35.mnf
depm 21.6 0.9 ! ridgecrest2.29
depl 17 4 ! focal depth and uncertainty from local-distance data, supercedes all previous depth estimates
...
(more event definition blocks)
...
memb
even 20190711.2345.18
inpu 20190711.2345.18.mnf
depm 12.1 0.6 ! ridgecrest2.32
depn 12 3
plot 1 ! Plot this event in the first selected event plot (there can be multiple such plots)
memb
even 20190712.1311.37
inpu 20190712.1311.37.mnf
```

```
depm 19.0 0.5 ! ridgecrest2.32
depn 19 3
plot 1
```

Commands

This section covers all available commands for controlling the actions of **mloc**. Even if the default values were to be accepted, it would still be necessary to use several commands to define at least one event. Most commands can be issued either in a command file or interactively. The typical usage is to create a command file which is referenced interactively (with the cfil command) and then followed with a few more interactive commands before running the relocation with the command run.

Alphabetical List

When **mloc** is started it provides an alphabetical list of the available commands. As of v10.5.1 that list is:

- anno
- bdps bias bloc bptc
- cal_ ccat cfil clim comm corr cptf ctyp cvff cvtt
- damp datf dbug dcal dem1 dem2 dep_ diff
- ellp epap eplt even
- fdhp flag fmap frec freh
- help hlim
- inpu
- kill
- lat lgtt lmod long lonr lres
- mare mdou memb
- nsmd
- oldr
- pert phid phyp plot pltt ppri pttt
- radf rdpp rels revi rfil rhdf run
- secv shcl skip splt sstn star stat step stop subc
- taup terr tfil tikh time tlog tomo tptt tt5e tt5s ttou
- vect vlog vscr
- weig wind
- xsec

mloc has an interactive help system for commands; give the command help, followed by the name of the command.

Functional Summary of Commands

If the help command is issued without an argument a functional summary of commands is listed:

- Calibration

- cal_: calibration data for current event
- ctyp: treatment of calibration events in indirect calibration
- dcal: direct calibration
- Informational
- anno: annotation
- comm: comment line
- dbug: extra logging for debugging
- help: details about commands
- revi: review current relocation control parameters
- tlog: logging for tau-p calculations
- vlog: set verbose mode for logging
- vscr: set verbose mode for screen display
- Input
- cfil: specify a command file
- diff: differential time data
- even: event name
- inpu: specify an event data file
- kill: kill a block of events
- memb: start a new event (or kill an event with the “kill” argument)
- Inversion
- bias: hypocentroid bias correction
- clim: epicentral distance limits for cluster vectors
- flag: use of data flags
- frec: free parameters for cluster vectors
- freh: free parameters for the hypocentroid
- hlim: epicentral distance limits for hypocentroid
- phyp: set “use only P arrivals for hypocentroid” flag
- pttt: perfect theoretical travel times for hypocentroid
- run : begin the relocation process
- shcl: hypocentroid convergence limits
- step: number of iterations to run
- stop: stop processing
- tikh: Tikhonov regularization
- weig: residual weighting by reading error
- Miscellaneous
- lonr: set longitude range (-180 to 180 or 0 to 360)
- Output
- bloc: BayesLoc output file
- ccat: COMCAT output file
- datf: output MNF bulletin of all events in their final form
- lres: LRES file of large cluster residuals
- mdou: map_dat output file for GMT

- oldr: output of phase readings over a limited distance range
- subc: select a subcluster based on data for direct calibration
- tomo: tomography output files
- ttou: Empirical TTs for a specific phase
- Phase Identification
- phid: toggle phase re-identification
- ppri: phases that cannot be renamed
- skip: skip readings of a given phase (also by station and author)
- Plotting
- cptf: color palette table for topography
- dem1: plot regional-scale topography in GMT
- dem2: plot high-resolution topography in smaller-scale GMT plots
- ellp: plot an ellipse
- epap: plot of empirical path anomalies
- eplt: make a map of locations with only confidence ellipses
- fdhp: make a histogram of focal depths
- fmap: digital fault map for GMT script
- plot: plotting of selected events
- pltt: travel time plots
- rdpp: relative depth phase plots for individual events
- splt: make a seismicity map, same-size symbols for locations
- star: plot a star to highlight a specific event
- stat: plot a triangle to indicate a station location
- tt5e: single-event local distance (tt5) plot
- tt5s: single-station local distance (tt5) plot
- vect: which event shift vectors to plot
- xsec: cross-sections
- Residuals and Uncertainties
- cvff: Cluster vector fudge factor (non-gaussian contribution to uncertainty)
- cvtt: cluster vector travel time error
- mare: minimum allowed reading error
- rels: set reading errors for local stations
- rfil: reading error file (.rderr)
- tfil: travel time spread file (.ttsprd)
- wind: windowing of residuals
- Starting locations
- dep_: set starting focal depth
- lat: set starting latitude
- long: set starting longitude
- pert: perturbation to all starting locations
- rhdf: read starting locations from an HDF file
- time: set starting origin time

- Stations
- bdps: list of stations suspected of reporting bogus depth phases
- nsmd: search NEIC station metadata for missing station codes
- radf: read agency and deployment fields to resolve station code conflicts for a specified station
- skip: skip readings from a given station (also by phase and author)
- sstn: supplemental station file
- Travel Times
- bptc: Bounce point topography correction
- corr: Station elevation correction
- lgtt: Lg travel time calculation
- lmod: local velocity model
- secv: station elevation correction velocities
- taup: Global TT model, using Tau-P formulation
- terr: Timing error correction at a station
- tptt: T-phase travel time calculation

Command Descriptions

The text in these descriptions is taken from the interactive help system.

anno (ANNOtation) is used to provide some extra textual information about an event. The string will be printed at the end of the corresponding line in the HDF files. Maximum 20 characters. This annotation over-writes an annotation given in the event record of an MNF file, in which case a warning will be given.

bdps (Bogus Depth Phase Stations) specifies the pathname of a file containing a list of stations that are suspected of reporting bogus depth phases, i.e., depth phase arrival times that are generated from theoretical arrivals relative to a preliminary hypocenter (e.g., the PDE). Depth phase readings from listed stations will be plotted at a smaller size in "tt6" plots and will have an asterisk next to their entries in the ".depth_phases" file. The generic station file format (type 3) is used for this list because it carries fields for epoch dates "on" and "off". When used for bogus depth phase reporting, however, the dates are interpreted as the epoch during which bogus depth phases have been reported. These dates are optional, and other station information that can be provided in this format is not used by **mloc**. It can be useful for station identification, however. The only required field is station code in columns 1-5, but the first line must have "3" in the first column to confirm the format.

bias (BIAS correction) determines if a correction will be made for a minor source of bias in the hypocentroid, related to the fact that some events have more readings than others. See Jordan & Sverdrup (1981) for details. By default, or if the argument is blank or "on", the correction is made. If the argument is "off" the correction is not made. Correcting the bias increases the variance of the hypocentroid slightly. Compared to the main source of bias in the hypocentroid, i.e., travel time model inadequacies, this is usually negligible.

bloc (BayesLOC) specifies that an output file (with suffix ".bloc") will be written. The format is designed for easy import of mloc calibrated hypocenters into BayesLoc for use as priors. Default is "off". Issuing the command with no argument toggles the current state, or the arguments "on" or "off" can be used to explicitly set the state.

bptc (Bounce Point Topography Correction) makes a correction to theoretical travel times of pP and sP for the topography at the bounce point. For areas above sea level the times of pP and sP are increased. For oceanic areas the travel times of pP and sP are reduced because the reflection point (seafloor) is below sea level. The travel time of the pwP phase is calculated by adding a term for the propagation time in the water column to the pP time. No calculation is done for swP. The command can be toggled on and off by issuing the command without an argument, or set explicitly with arguments "on" and "off".

cal_ (CALibration event) is used to declare a calibration location for the current event. Epicenter is always assumed to be calibrated. The 4th character in the command is used to specify which of the other hypocentral parameters are to be considered calibrated:

- e : the epicenter alone is calibrated (e.g., from InSAR)
- f : focal depth is also calibrated, but not origin time
- t : origin time is constrained by local-distance data but it cannot be considered to be calibrated because focal depth is not calibrated
- h : epicenter, depth, and origin time are calibrated

The command takes 11 arguments: hour, minutes, and seconds of the origin time, plus latitude, longitude and depth, in that order, followed by 3 arguments giving the 90% confidence ellipse and 2 arguments for the uncertainties in depth and OT. The sequence for confidence ellipse is: azimuth of semi-minor axis (clockwise from north) semi-minor axis length, semi-major axis length (both in km). The command can be given multiple times for an event, in which case the last command will define the calibration parameters.

ccat (ComCAT) specifies that two output files will be written. One is a ~.comcat file, a single file similar to the .datf_mnf file, but with additional information needed for uploading to the USGS/NEIC ComCat server. The same file is used in the GCCEL project. The other is a file named ~_plots.pdf, a single PDF file containing all plots created in the run. The command takes a single optional argument, the filename (assumed to be in the data directory) of a text file containing a commentary on the nature of the cluster and the relocation procedures and results. The commentary file should be hard-wrapped at a reasonable line-length (72 characters is good). The commentary will be added to the .comcat file as comment records.

cfil (Command FILe) takes one argument, the name of a command file. All the commands in that file will be processed before control is returned for interactive command processing. Multiple command files can be invoked (by separate **cfil** commands), but they must all be found in the data directory specified when the program starts. The **cfil** command cannot be issued from a command file, only interactively.

clim (Cluster vector LIMits) specifies a set of up to three ranges in epicentral distance that will be used to estimate the cluster vectors. This allows skipping readings, for example, at short distances, in the Pdiff range and near the caustic for PKP phases. Since each range requires two values, this command requires 2, 4, or 6 arguments.

comm (COMMent) declares a comment line. It can be used to disable a command in the command file or simply to add some additional information. It was formerly used as a convenient way to take an event out of the cluster temporarily without losing track of it completely, but that usage has been superceded by the kill command for blocks of events and the "kill" argument to the memb command for single events.

corr (station elevation CORRection) controls station elevation corrections. A single integer argument is required. The allowed arguments are:

- 0: No correction
- 1: Station elevation corrections are made

Regardless of the usage of the **corr** command, ellipticity corrections are always applied. In the case of direct calibration the choice of station elevation correction changes the reference plane for focal depth. If station corrections are made, the reference plane is the reference ellipsoid. If not, the reference plane is, roughly, the surface (average elevation) of the source region.

cptf (Color Palette Table File) defines a color palette table to use for plotting topography. The argument is only the name of the desired color palette table; the path is defined in the main program.

ctyp (Calibration TYPe). This applies only to indirect calibration mode, and only to calibration events. It determines the way in which the uncertainty for calibration events is calculated. There are three options, specified by an integer value:

- 1 = traditional: use the uncertainty of the supplied calibration data
- 2 = systematic: add calibration shift uncertainty to cluster vector uncertainty (default)
- 3 = optimal: traditional or systematic, whichever has smaller semi-major axis

cvff (Cluster Vector Fudge Factor) specifies a radius (km) for additional uncertainty that will be added to the cluster vector covariance matrices to account for bias from non-gaussian components of the arrival time data sets. This is completely *ad hoc*, but it goes in the right direction. A value of about 1.0 km has been inferred from some tests, but further testing is needed.

cvtt (Cluster Vector Travel Time) adds an additional variance to differential time data. When using differential arrival times from waveform cross-correlation, a pure reading uncertainty is often provided, but the actual variance includes a component from inadequate theoretical differential travel times. This command allows that term to be defined. It is only relevant if differential time data are used. The correction is never added to bulletin data because this error term is already absorbed in the empirical reading error. The two parameters are vmr (velocity

model variance), a percent value of velocity variance in the travel time model, (default value 0.05, or 5%), and cscale, a measure of the scale of the cluster in km (default 10 km).

damp (DAMPing) determines if damping will be applied to the cluster vector changes at each iteration. Damping is done by calculating the mean of changes in a parameter for all events and subtracting it from the change for each event. This sometimes helps convergence. The default is off. The state can be toggled by using the command without an argument, or set explicitly with arguments “on” or “off”.

datf (DATa Final) specifies that a .datf output will be written. This is a single file with all the events written in MNF format, but with phase IDs and flags as they are at the end of HD analysis. Flags and phase IDs can be changed during the HD analysis. Arguments “on” and “off” can be used to set the logical state explicitly. Issuing the command without an argument toggles the logical state.

dbug (DeBUG) specifies that extra information about the HD analysis will be written to the output window. Issuing the command without arguments toggles the current state. The arguments “on” and “off” may be used to set the state explicitly. Default is “off”.

dcal (Direct CALibration) specifies direct calibration mode, i.e., the hypocentroid is being located with data (usually near-distance data) that has minimal travel time error and so the hypocentroid can be taken as bias-free. The confidence ellipse of the hypocentroid is added to those of the cluster vectors to obtain final estimate of uncertainty for each events location. Four new output files are created:

- .dcal_phase_data: listing the data used for the hypocentroid
- _dcal.bash: GMT script, showing stations and raypaths used for hypocentroid
- _dcal.pdf: PDF file, from the _dcal.bash script
- .hdf_dcal: hdf with calibrated locations

Confidence ellipses in the .hdf_dcal file are cumulative, from hypocentroid and cluster vector for each event. Can be toggled by using the command without an argument, or set explicitly with arguments “on” or “off”.

dem1 (Digital Elevation Model 1) specifies whether or not commands to plot topography will be added to the GMT script. The argument is “on” or “off” (default). If no argument is given the state is toggled. If plotting of topography is turned on the ETOPO1 1-minute gridded topography and bathymetry model will be used.

dem2 (Digital Elevation Model 2) defines a DEM to be used in the base plot and other plots that benefit from high-resolution topography. It assumes the user has provided a “.grd” file and the argument to the command gives the pathname of that file, relative to the MLOC working directory. Such files are obtainable here. If no argument is given, plotting of topography is turned off.

dep_ (DEPth) specifies a starting focal depth. The 4th character is used to specify the kind of information used to constrain the depth. At least one argument is required, a positive depth value in km, and it can optionally take one or two additional arguments to assign an uncertainty to that depth. Uncertainty in the assigned depth can be symmetric or asymmetric. If a single value is appended to the depth estimate it is taken as a symmetric uncertainty. If two values are appended the first is taken as plus (deeper), the second as minus (shallower) uncertainty, in km. Both values should be positive. Recognized values for the 4th character are:

- c: cluster default depth
- d: depth phases
- e: engineered (man-made explosion)
- f: fault model (InSAR, GPS, etc.)
- i: input data file
- l: local distance readings (more than 2-3 focal depths)
- m: mloc solution (with free depth)
- n: near-source station readings
- r: relocation (outside mloc) with free depth
- u: unknown/unspecified/unconstrained
- w: waveform analysis

The command accepts any character in the 4th position so custom values can be used. The **dep_** command can be issued multiple times, first to set a default depth for all events (**depc**), then to set selected events at different depths. If it is issued before any events are declared (by memb commands), it applies to all events in the cluster that do not have a constrained depth declared in the input file. If issued after a memb command it applies only to the current event, and it overrides the depth read from the input file or from an HDF file.

diff (DIFFerential time data) specifies the pathname of a file containing differential time data for pairs of events in the cluster. The command can be issued more than once, but only the last instance will be processed.

ellp (ELLipse Plot) is used to specify the parameters of an ellipse that will be plotted in the map plots by GMT. The command takes five arguments:

- latitude of the center point
- longitude of the center point
- azimuth of the semi-major axis
- full length of the major axis, in km
- full length of the minor axis, in km

The command can be issued up to 10 times.

epap (Empirical Path Anomaly Plot) controls creation of a map of empirical path anomalies for a specific phase. A symbol is plotted at each station that recorded the phase of interest at least twice. The symbol is color-coded according to the sign of the empirical path anomaly and the size is proportional to its absolute value. Ray paths can optionally be drawn from the hypocentroid to the symbols. The command takes three arguments: phase name, epicentral distance limit (degrees), and a flag (0 or 1) which controls the plotting of raypaths. The epicentral distance is used to set the boundaries of the map. The command can be issued up to 6 times.

eplt (Ellipse PLoT) specifies whether a "confidence ellipse" plot will be made. This plot includes confidence ellipses for relative location but not event numbers or relocation vectors. Issuing the command without arguments toggles the current state. The arguments "on" and "off" may be used to set the state explicitly. Default is "off".

even (event) is used to specify the name of the current event. This normally has a form like YYYYMMDD.HHMM.SS (i.e., 16 characters). Maximum 30 characters. **even** takes one argument.

fdhp (Focal Depth Histogram Plot) specifies whether a plot will be made of a histogram of focal depths. Issuing the command without arguments toggles the current state. The arguments "on" and "off" may be used to set the state explicitly. Default is "off".

flag (data FLAGs) determines if data flags encountered in input files will be honored ("on") or ignored ("off"). Issuing the command with no argument toggles the current state. Default is "on".

fmap (Fault MAP) specifies a file of digitized faults or other linear features to be plotted in GMT. The standard GMT format – longitude, latitude – is used, in free format. The symbol ">" can be used to separate multiple line segments. The pathname relative to the working directory is given. The normal place for these files is the directory */tables/faults*. The command can be issued multiple times.

frec (FREe parameters Cluster vector) is used to specify which location parameters will be free and which will be fixed for cluster vectors. It requires four arguments that are either 0 or 1. 0 indicates a fixed parameter, 1 indicates a free parameter. The order is latitude, longitude, depth, origin time. If **frec** is invoked before any events are declared, the values apply to all events. If invoked after an event is declared, the values only apply to the current event.

freh (FREe parameters Hypocentroid) is used to specify which location parameters will be free and which will be fixed for the hypocentroid. It requires four arguments that are either 0 or 1. 0 indicates a fixed parameter, 1 indicates a free parameter. The order is latitiude, longitude, depth, origin time.

help (HELP) is used to provide detailed information about the purpose and usage of the commands in **mloc**. If it is issued without an argument a list of all commands is returned, organized by general category, and with a short description of the function of each command. A

particular command may be given as an argument, in which case a more detailed description of that command is returned. Only a single command can be given as argument.

hlim (Hypocentroid LIMits) specifies a set of up to three ranges in epicentral distance that will be used to estimate the hypocentroid. This allows skipping readings around the Pg/P crossover, in the Pdiff range, and near the caustic for PKP phases. Since each range requires two values, this command requires 2, 4, or 6 arguments.

inpu (INPUt file) is used to specify the name of the file containing the phase arrival time information for an event. The command requires a single argument. The standard form of the base filename is YYYYMMDD.HHMM.SS. The only supported format is MNF (MLOC Native Format). The maximum length of the input filename is 20 characters.

kill (KILL event) provides a convenient way to ignore ("kill") blocks of events listed in a command file. It requires one argument, either "on" or "off". After the **kill on** command is issued, no other commands will be processed until the **kill off** command is processed. To kill a single event, the memb command with the argument "kill" is preferred.

lat (LATitude) specifies a starting latitude. It requires one argument, a latitude value. If it is issued before any events are declared (by memb commands), it applies to all events in the cluster. Otherwise it applies only to the current event. The **lat** command can be issued many times, first to set a default latitude for all events, then to set selected events at different latitudes. If the **lat** command is not applied to an event, the latitude read from the input data file will be used as the starting value.

lgtt (LG Travel Time) specifies a custom travel-time model for the Lg phase. The model is linear in epicentral distance. It takes three arguments: a zero-intercept (sec), slope (sec/degree) and minimum epicentral distance (degrees) from which to begin calculating travel tiems for Lg. Default values are 0., 31.5 and 2.5.

lmod (Local MODel) specifies a local velocity model to be used to calculate travel times at short epicentral distance ranges. Local velocity models are normally stored in the *tables/crust/* subdirectory. The format of local velocity model files is the format used by HYPOSAT. If a custom velocity model is not specified, the global tau-p model is used for all distances. The command takes one argument, the pathname (relative to the working directory) of the model file. The distance and depth ranges for which it will be used are specified in the file itself.

long (LONGitude) specifies a starting longitude. It requires one argument, a longitude value. If it is issued before any events are declared (by memb commands), it applies to all events in the cluster. Otherwise it applies only to the current event. The **long** command can be issued many times, first to set a default longitude for all events, then to set selected events at different longitudes. If the **long** command is not applied to an event, the longitude read from the input data file will be used as the starting value.

lonr (LONgitude Range) specifies the range to which longitudes should be converted. The command takes a single integer argument that gives the center of the longitude range. Two values are accepted:

- 0 : -180° ≤ longitude < 180° (default)
- 180: 0° ≤ longitude < 360°

A value should be chosen that keeps longitudes for all events in the cluster in the same range.

lres (Large RESidual) specifies that a .lres output file will be written, containing all readings with cluster residuals (*eci*) larger than the value given by the single argument.

mare (Minimum Allowed Reading Error) specifies the the minimum reading errors that will be allowed, regardless of what value is read from a .rderr file. Three arguments are required:

- The minimum value for local phases
- The minimum allowed value for phases beyond local distance
- The minimum allowed value for teleseismic depth phases

Local phases are defined as those with "g" or "b" as the second character of the phase name.

mdou (Map Data OUtput) specifies that a file containing lat-lon data will be written. It will have the filename suffix ".map_dat". This file is designed for easy import by a GMT script for additional plotting. If no argument is given, the status is toggled. The arguments "off" and "on" can be used to set the state explicitly.

memb (MEMBer) is used to define a new event in the cluster or to cause an event in the command file to be skipped (killed). The command takes no argument if a new event is being defined. When the command is issued, the event counter is incremented and specification of a new event starts. To use this command to kill an event, add the argument "kill". Any text following "kill" is ignored, but it can be used for a comment about the reason for killing the event. This is the preferred way to kill a single event; see the command kill to skip blocks of events.

nsmd (Neic Station MetaData) determines if the NEIC station metadata file will be searched for missing station codes after the master station file and any supplemental station files have been searched. Any station codes that are found in the NEIC metadata file will NOT be used in the current run; a supplemental station file must be created and referenced in a subsequent run. The body of a supplemental station file in "NEIC" format (isstn=5) for any matching station codes will be found in the .log file. If there are multiple instances of a station code with different coordinates they are all listed. Default is no search. Logical state is toggled by issuing the command without an argument, or set explicitly with "on" or "off" as the argument.

oldr (Output Limited Distance Range) specifies an epicentral distance range for which an output file will be created to list all phase readings in that range. Two arguments are required.

pert (PERTurb starting locations) is used to specify a perturbation in location parameters that will be applied to all events in the cluster. Four arguments are required, in the order latitude, longitude, depth, and origin time. Latitude and longitude perturbations are in decimal degrees.

phid (PHase IDentification) is used to toggle phase re-identification, which is done after the data are read in and once more after the first iteration. By default, it is turned on.

phyp (P HYPocentroid) specifies that only P phases will be used to estimate the hypocentroid. This is normally used with hlim to specify only arrivals between 30 and 90 degrees, which provides a consistent, (albeit biased) estimate of the hypocentroid. If direct calibration is being used, S-P readings will be retained as well. If no argument is given, the status is toggled. The arguments "off" and "on" can be used to set the state explicitly. Default = "on".

plot (selective PLOTting) specifies that the current event is selected for plotting in a secondary GMT script (_selN.bash). All cluster events are always plotted in the main GMT script (_base.bash). The command takes one argument, an integer which specifies which selected-event plots this event will be included in. Limit is 9 selected-event plots, and an event can belong to more than one by using the **plot** command several times before the following memb command.

pltt (PLot Travel Time) specifies one or more TT vs distance plots to be made. There are nine plot types available (details and examples), which are specified by the indices 1-9 after the command. Up to nine arguments can be given in a single instance of the command:

- 1 = Summary TT plot, full distance range 0-180 degrees
- 2 = Teleseismic P residuals, 16-120 degrees
- 3 = PKP branches around the caustic
- 4 = Near-source residuals
- 5 = Local distance, 0-4.0 degrees
- 6 = Local-regional distances, 0-30 degrees (reduced velocities)
- 7 = Local and regional S phases (reduced velocities)
- 8 = Summary plot of relative depth phases, pP-P and sP-P, for all events
- 9 = S-P times

If no arguments are given, all plots are turned off. If a custom crustal model has been specified for short ranges, the travel times will be calculated from it in the applicable distance range. In all cases, travel times are calculated for the depth of the hypocentroid.

ppri specifies phases that cannot be renamed. It takes one argument, a phase name. The command can be issued multiple times, up to the limit specified by *n_no_phreid_max*.

pttt (Perfect Theoretical Travel Times) allows the variance of the travel time model (ttsprd) to be set to zero ("on") for the purpose of calculating the hypocentroid and its uncertainty. Phase spread values (either default or read from a .ttsprd file) are still used in the windowing algorithm

(command wind). Issuing the command with no argument toggles the current state. Default is "off".

radf (Read Agency and Deployment Fields) specifies that agency and deployment fields in event data files (.mnf files) and station lists for the specified station code will be read and used to resolve station code conflicts. This is only used when a dataset contains readings from two or more different stations with the same code. The command takes a single argument, a station code. The command can be issued multiple times, up to the limit specified by "n_radf_max" (currently 10).

rdpp (Relative Depth Phase Plot) makes a plots of misfit vs. depth for relative depth phases (pP-P, sP-P, pwP). The command takes a single argument. To make a plot for a specific event the argument is the event name (as given in the even command). This command can be issued multiple times, up to the maximum number of events. The argument "all" can be used to make plots of all events for which depth phases are reported.

rels (Reading Error Local Stations) specifies the reading errors that will be used for direct-arriving phases Pg, Pb, Sg, and Sb within a specified distance. This is mainly used for single event location of events with local data, e.g., calibration events. Three arguments are required, the reading error for P and S phases, respectively, and the distance range (epicentral degrees) in which these values apply.

revi (REVIew) is used to provide detailed information about the current events in the cluster and the settings that influence how the relocation will be done. It takes no arguments.

rfil (Reading error FILe) specifies a .rderr file to be opened for reading the empirically-derived reading errors of specific station-phases. The file is expected to be in the data directory, so the argument of the command is just the filename, not the full pathname. Revert to the default values by entering the argument "default".

rhdf (Read HDF file) specifies an HDF file that will be read to obtain starting locations. The hypocentral values read from the HDF file are over-ridden by any of the commands lat, long, time, or dep_ issued later in the command file. If any cluster events are missing from the HDF file, starting locations for those events will revert to those of the corresponding input file. The command takes one argument, the name of the HDF file, which must be stored in the data directory with the input files.

run (RUN the inversion) ends the command processing phase and starts the HD relocation process. There are a few additional queries that provide important controls over the relocation process, but no further opportunities to add events to the cluster or issue any of the standard commands.

secv (Station Elevation Correction Velocities) specifies the velocities that will be used for station elevation corrections. The command takes two arguments, the velocity for P and S. Default values are 5.8 km/s and 3.46 km/s, respectively.

shcl (Set Hypocentroid Convergence Limits) modifies the criteria used to decide if convergence has been reached, i.e., the change in each parameter from one iteration to the next is smaller than the convergence limit. This command only deals with the limits relating to the hypocentroid (origin time, epicenter and focal depth). Actual convergence requires all parameters for the hypocentroid and each cluster vector to satisfy their corresponding convergence criteria, but it is often the case that convergence is prevented by oscillations in the hypocentroid origin time, sometimes coupled with instability in one of the epicentral parameters. The "epicenter" limit is applied separately to latitude and longitude. It should be noted that achieving convergence by increasing the limits may not produce a reliable solution, but there are advantages to having a converged solution for investigating why it was necessary. The command takes three arguments, convergence limits for epicenter (degrees), focal depth (km) and origin time (s). Default values are 0.005 degrees, 0.5 km and 0.1 s.

skip (SKIP) specifies a triplet of station-phase-author for which all readings are flagged (with "s"). Skipped readings will not to be used in the relocation for cluster vectors or hypocentroid. The command takes three arguments, a station code, a phase code and an author code. Wildcards ("*") are supported:

- "skip GRMI * *" means skip all readings from GRMI
- "skip GRMI Pg *" means skip all Pg readings from GRMI
- "skip GRMI Pg ARGhods" means skip all Pg readings from GRMI by the author "ARGhods"
- "skip * * ARGhods" means skip all readings from the author "ARGhods"

Readings with blank phase code can be skipped by giving "blank" (without quotes) as the phase code. The command can be issued multiple times, up to the limit specified by *n_skip_max*.

splt (Seismicity PLoT) specifies whether a "seismicity map" will be made. This type of plot indicates event locations by open circles of 1 km diameter and relocation vectors, but does not carry event numbers. Issuing the command without arguments toggles the current state. The arguments "on" and "off" may be used to set the state explicitly. Default is "off".

spou (S-P OUtput file) specifies that a special file containing all S-P arrivals in the dataset will be written. It will have the filename suffix ".sp". If no argument is given, the status is toggled. The arguments "off" and "on" can be used to set the state explicitly.

sstn (Supplemental STatioN data) specifies the pathname of a file containing coordinates for stations missing from the main station file. These files are often stored in the "tables/stn/" subdirectory, but can be stored elsewhere, including the cluster data directory. Several formats are supported, specified by an integer in the first column of the first line:

- 0 – New master station file format, from April 28, 2014
- 1 – ISC FFB format, (deg-min-sec*10)
- 2 – SEISAN format, (deg-decimal min)

- 3 – simplified “mloc” format, decimal degrees, geographic coordinates
- 4 – China Seismic Bureau format
- 5 – NEIC format
- 6 – MSU format (used by Kevin Mackey)
- 9 – Former master station file format (used index “0”)

There is a support document describing the formats in detail. The command can be called multiple times, up to the value of *n_supp_stn_file_max*.

star (STAR plot) is used to specify the plotting of a solid red star at the location of the current event in the map plots by GMT. This is normally used to highlight events of special interest, such as a mainshock and major aftershocks. The command takes one argument, the size of the circumscribing circle (recommended sizes are in the range 0.2-0.5). The command can be issued up to 10 times.

stat (STATion plot) is used to specify the parameters of a triangle that will be plotted in the map plots by GMT. The command takes three arguments: latitude and longitude of the center point and size of the circumscribing circle (0.30 is a good choice). The command can be issued up to 30 times.

step (STEP) specifies the number of iterations to run. It takes one argument, an integer between 0 and the maximum number of iterations allowed (4). If the convergence criteria are met before the specified number of iterations, the program ends as usual. An “iteration” is counted when the current hypocenters are updated. **step 0** is equivalent to the old **fwd** command. The residuals to the starting locations will be calculated and the program will exit. No inversion or iteration will be done.

stop aborts the current run while still in the command processing phase. It is normally used when some aspect of the reading of a command file (or interactive input) has gone wrong. It is not needed in normal operation, and cannot be issued after the run command has been given.

subc (SUBCluster) specifies the parameters to use to select events that would be most suitable for a high-quality subcluster with direct calibration. Three parameters are required:

- 1 – the maximum epicentral distance
- 2 – the minimum number of readings within that distance
- 3 – the minimum number of station-phases in common with other events in the cluster

The main body (i.e., event definition blocks) of the corresponding command file is written to the .log file.

taup (Tau-P) specifies a global travel time model using the tau-p formulation. Currently, ak135 is the only option. The command takes a single argument, the model name. The associated “.tbl” and “.hed” files must be stored in the “tables/tau-p” subdirectory.

terr (Timing ERRor) makes a static correction to arrival times at a specified station. The correction is applied to all phases. The command takes two arguments, the station code and time correction. The command can be issued up to 4 times.

tfil (Travel-time spread FILe) specifies a .ttsprd file to be opened for reading the empirically-derived spreads of different phases. The file is expected to be in the data directory, so the argument of the command is just the filename, not the full pathname. Revert to the default values by entering the argument "default".

tikh (TIKHonov regularization) specifies the value to be used for Tikhonov regularization of the perturbations for cluster vectors. If a value of 0 is used (default) there is no regularization. For data sets exhibiting convergence problems it may be helpful to set a non-zero (positive) value. Determination of the optimal value is a non-trivial task, but values in the range 0.2-0.6 seem to be about right for this application.

time (origin TIME) specifies a starting origin time. It requires three arguments (hours, minutes, seconds). Unlike other location parameters there is never a need to set all starting origin times to a common value, so it can only be aplied to the current event. If the **time** command is not applied to an event, the time read from the input data file or from an HDF file will be used as the starting value.

tlog (Tau-p LOG) specifies that details of the tau-p calculations will be written to the taup log file (~.taup). This is used for debugging the tau-p code only! The output file can exceed 500 MB for moderately large clusters. Issuing the command without arguments toggles the current state. The arguments "on" and "off" may be used to set the state explicitly. Default is "off".

tomo (TOMOgraphy) specifies output files for tomography. This command requires two arguments, a phase name and a flag for which kind of data to extract:

- 1 = Extract all readings of the specified phase
- 2 = Extract only readings which were used for the cluster vectors
- 3 = Extract empirical path anomalies

The **tomo** command can be issued multiple times, up to the limit specified by the variable *nitomomax*.

tptt (T-Phase Travel Time) specifies an intercept (sec) and slope (sec/degree) to be used in calculating travel tiems for T-phases. Default: 15. and 75., respectively.

tt5e (Travel Time type 5 plot for a single Event) allows a type 5 plot (local distance, out to 4°) to be made for a single event. The command takes a single argument, the event name (as given in the even command. The command can be issued multiple times, up to the limit specified by *n_tt5e_max*.

tt5s (Travel Time type 5 plot for a single Station) allows a type 5 plot (local distance, out to 4°) to be made for a single event. The command takes a single argument, the station code (case-sensitive). The command can be issued multiple times, up to the limit specified by *n_tt5s_max*.

ttou (Travel Time OUtput) specifies that an output file of empirical travel time data will be written for a specific phase. The command takes a single argument, the phase name (case sensitive). The command may be issued multiple times, up to the limit in *n_ttou_max*. S-P is supported but the format of the output file is different.

vect (VECTors) controls the plotting of four types of relocation vectors:

- From data file epicenter to final location, either calibrated or uncalibrated, in black
- From starting location to final location (but not with calibration shift), in green
- Calibration shift, for indirect calibration, in red
- Residual calibration shift, for indirect calibration, in blue

The command takes four arguments which must be either 0 or 1, to set the plotting of these vectors. The default state is TRUE for all four vectors.

vlog (Verbose LOG) specifies that extra information about the HD analysis will be written to the log file. Issuing the command without arguments toggles the current state. The arguments "on" and "off" may be used to set the state explicitly. Default is "off".

vscr (Verbose SCReen) specifies that extra information about the HD analysis will be written to the terminal window. Issuing the command without arguments toggles the current state. The arguments "on" and "off" may be used to set the state explicitly. Default is "off".

weig (WEIGht) determines if the data (residuals) are weighted equally ("off"), or if the data will be weighted inversely to their reading error ("on"). Issuing the command with no argument toggles the current state. Default is "on".

wind (WINDowing) specifies two parameters (and an optional third one) used to define windows for each phase that are used to cut readings with large residuals out of the problem. The parameters given here are multipliers for the travel time spread assigned to each phase. The travel time spreads can be taken from default values or read from a .ttsprd file from a previous run. Default values for the multipliers are 3 and 4. With these values, all residuals within 3*tsprd(phase) will be given full weight. Residuals between 3*tsprd(phase) and 4*tsprd(phase) have weights that taper smoothly to zero with a "1-cosine" function. The .ttsprd file also carries a baseline adjustment for each phase that helps keep the window function centered over the actual distribution of that phase, rather than centering it on the ak135 theoretical time for that phase. The optional third parameter is an epicentral distance below which the windows are expanded by a factor of 2. This is used in direct calibration to help keep good readings from being lost because of a poor starting location. The default value is 1.2 degrees. Weights assigned via the **wind** command are written to the .phase_data output file.

xsec (X-SECtion) controls the plotting of cross-sections. The command takes six arguments:

- Latitude of the first end-point
- Longitude of the first end-point

- Latitude of the second end-point
- Longitude of the second end-point
- Depth (km) of the cross-section (all sections start at zero depth)
- Full width (km) of the cross-section

Multiple cross-sections can be defined, up the limit set by the parameter *n_xsec_max*.

Station Data

Management of information on seismograph stations, especially the geographic coordinates to be associated with a specific station code, is a critical aspect of all earthquake location algorithms but it is often especially complex for the kinds of relocation studies undertaken with **mloc**. Location programs are typically conducted (e.g., by a local or regional network) with a small, stable set of seismograph stations. The U.S. Geological Survey's National Earthquake Information Center (NEIC) uses data from thousands of stations in their global monitoring but they are all real-time digital stations that report their metadata (including station coordinates) automatically. Even the International Seismological Centre (ISC), which collects the most complete and diverse set of arrival time data globally, uses data only from stations that have been registered in the International Registry of Seismograph Stations (IR) that they maintain.

In contrast **mloc** is designed with the philosophy that all possible data should be usable. A successful calibration analysis often depends on combining data from multiple sources, including temporary deployments and data from local and regional networks that are not normally made available to any openly-accessible database. In such datasets, station information may come in a variety of formats and, most important, station code conflicts are common. **mloc** incorporates tools, reporting and logic to assist in managing such issues but the user must become familiar with a number of subtle aspects of the subject in order to ensure full and correct use of the available arrival time data. Some important aspects are:

- Station codes may have up to 5 characters
- Case-sensitive station codes are supported (but not recommended)
- Operational epoch is supported
- Authorship, i.e., the source of the coordinates, is supported
- The concepts of "agency" and "deployment" are supported to help resolve station code conflicts

The most important rule to appreciate regarding station code information in **mloc** is this:

The master station file contains entries only for seismograph stations that have been registered with the International Registry of Seismograph Stations (IR). Station information for unregistered stations is carried in one or more supplemental station files.

The master station file is always read by **mloc**. Supplemental files are usually referenced in a command file with the sstn command, but they can be declared interactively as well.

A limit (currently 24,000) for the total number of stations that can be defined is carried in *mloc.inc*. The master station file presently contains 21,614 entries, leaving 2,386 entries for supplemental station files. A maximum of 8 supplemental station files can be referenced.

The remainder of this section deals with the master station file. Supplemental file formats are described elsewhere. See also the discussion of how to deal with various problems associated with station codes.

The New IASPEI Station Coding Standard (ADSLC)

In 2013 IASPEI endorsed a new standard for identifying seismograph stations. A document describing that standard is available at the ISC website. This standard is often referred to as the **ADSLC** standard, in respect of the five defined fields:

agency.deployment.station.location.channel

that take the place of the traditional station code. The MNF data format that is used for event data files in **mloc** has fields for all five elements of the ADSLC standard, but the agency, deployment and station fields are the most important for relocation analysis.

It is important to understand that a given station can have multiple ADSLC codes, governed by a complex system of aliases and umbrella designations, some of which are mentioned below. This unfortunate complexity is necessary to handle the multiple ownership and operational affiliations that exist for many seismograph stations.

Unfortunately, little progress has been made in actually supporting the ADSLC standard at the ISC or most other seismological agencies. Specifically, neither arrival time datasets downloaded from the ISC Bulletin nor station information datasets downloaded from the IR carry information concerning agency or deployment. On the other hand, by definition, any arrival time data downloaded from the ISC Bulletin may be legitimately characterized with the agency code ”ISC” and deployment code ”IR”.

The ADSLC standard provides for a maximum of five characters for the agency and station fields, and eight characters for the deployment field. These field lengths are supported in the MNF format. Although earlier versions of **mloc** supported 6-character station codes as a mechanism for resolving station code conflicts, with v10.4.0 station codes are limited to 5 characters and conflicts must be resolved with another mechanism, e.g., agency and deployment codes.

It is also important to understand that, although MNF format supports the ADSLC formulation, it is usually best *not* to encode arrival time data files (.mnf files) with that information as a matter of course, even if it is available. **mloc** processes most arrival time datasets more efficiently if the agency and deployment fields in the phase lines of the .mnf files are left blank. In other words, situations requiring the use of agency and deployment fields are rare and matching stations solely

by station code works fine most of the time. Therefore it is recommended to use agency and deployment codes only when they actually solve a problem and then they should be applied only to the phase readings that require them.

NEIC Station Metadata

One organization that does support the ADSLC, in a sense, is the NEIC, which maintains a database of station metadata that includes the FDSN network codes. As can be seen in the above-referenced defining document, stations belonging to FDSN-affiliated networks may be legitimately characterized in the ADSLC formulation with agency code ”FDSN” and a deployment code taken from the FDSN network code, a two character alphanumeric code. NEIC processing identifies stations by the combination of station code and FDSN network code.

Complexities arise, however, for several reasons:

- Some stations in the NEIC metadata database are not registered with the IR
- Some of those unregistered stations in the NEIC metadata database have station codes that conflict with the codes of stations (in other places) that are registered at the IR
- Some stations in the NEIC metadata database *have* been registered at the IR, but with station codes that have been modified, typically by adding an extra character to the code

A recent download of the NEIC station metadata file is included in the **mloc** installation, and the command nsmd can be used to recover data from it for a supplemental station file for stations that fail to match the master station file.

Operational Epoch

The period of time during which a set of coordinates is valid for a given station code is called the **operational epoch**. The **mloc** master station file format supports operational epoch, as do several of the supplemental station file formats. It is defined by two variables, *date_on* and *date_off*, which are given as a 7-digit integer composed of the year (left-most four digits) and Julian date (right-most three digits). Either one or both may be left blank, with the obvious meaning.

Seeding of the operational epochs for many codes in the master station file was based on an analysis of data holdings at Lawrence Livermore National Laboratory by Steve Myers, supplemented by information acquired by Bob Engdahl. I have found that there are many errors in these epoch data, especially in the date_off field, such that **mloc** often reports a “date range” failure in processing actual arrival time data against the master station file. In the case of erroneous *date_off* entries the problem appears to be that the field was filled with the date of the most recent data holding at that time, but obviously most stations have continued to operate past that point; therefore the correct course of action is to delete the date_off entry completely. In the case of violations of the *date_on* entry, the preferred course of action is usually to update to the earliest date of the available data.

The definition of the SEISAN station file format (isstn = 2) has been extended to include optional *date_on* and *date_off* fields, because this format is often used for temporary networks that had a limited period of operation, and the station codes for such deployments often conflict with registered stations.

Authorship

The notion of authorship (of station information) has been added to the master station data file to help document discrepancies in reported station coordinates that are sometimes encountered. The author of a set of coordinates is identified in a character variable with up to 8 characters. This use of "authorship" is distinct from authorship for hypocenters and authorship of phase readings in the MLOC Native Format for arrival time data. There are no standard definitions for these authorship entries, but I have used "IR" as the code for entries downloaded from the International Registry, the base set of coordinate information for the master station file. Definitions of authorship codes are carried at the beginning of the master station file in lines that begin with '#' in column 1 (comment lines).

A standard variation on authorship codes in the master station file is the use of the '+' character to indicate a different author for the information on station elevation and depth of burial (if given) than the author appropriate for the latitude and longitude. This is useful because the IR in many cases does not carry any information on station elevation, and in many other cases it is quite inaccurate. An easy way to rectify such lapses is to enter the coordinates in Google Earth and take the elevation from that, in which case the authorship code would be amended from "IR" to "IR+GE". In some cases, where the seismic vault can be clearly identified in Google Earth, I have used the authorship code "EAB+GE" to indicate that I specified all coordinates of the station using Google Earth.

Master Station File

The default set of station information for **mloc** is contained in the master station list, with the pathname */tables/stn/master_stn.dat* relative to the **mloc** working directory. The filename can be changed in the **mloc** configuration file. The master list contains only stations that have been registered at the IR, but the coordinates in some cases have been revised, for two main reasons:

- In some cases stations have been moved without changing the station code. The format includes the concept of epochs that define when a station occupied the given location. Therefore it is necessary to check the date for which coordinates of a given station code are needed to ensure that the correct coordinates are used.
- The coordinates carried in the IR are sometimes found to be in error, for example, by investigation with Google Earth.

When coordinates are modified, the original entry (i.e., the data from the IR) is not deleted. The new entry is placed above the original one in the list; **mloc** selects the first instance it encounters.

The format used for the master station list includes the concept of an author for the station coordinates (eight characters) and this is used to annotate the source of the preferred coordinates.

New stations are regularly registered with the IR, so the master station file must be updated manually when a dataset includes IR-registered stations that are not in the current master list. This seldom involves more than a handful of stations for any one cluster.

Format

The format used for the master station file has changed several times during development of **mloc**. The current master station file format was introduced with v10.0.0, released on April 28, 2014. **mloc** identifies station file formats by an integer (*isstn*) in the first column of the first row of the file. The rest of the first row can be used for a comment about the dataset. The master station file format is identified by *isstn* = 0. This format can be used for supplemental station files too.

Master Station File Format (*isstn* = 0)

Column	Description
1:5	Station code (a5)
7:15	Latitude (f9.5)
17:26	Longitude (f10.5)
28:32	Elevation, m (i4)
34:37	Depth of burial, m (i4)
39:46	Author (a8)
48:52	Agency (a5)
54:61	Deployment (a8)
63:64	Location (a2)
66:72	Date_on (i7)
74:80	Date_off (i7)
82:	Station name (no limit)

Any line with a hashtag (#) in column 1 is treated as a comment line. The following extract from the master station file illustrates several aspects of the format.

```
0 MLOC master station list
# ERE = Bob Engdahl
# GE = Google Earth
# IR = International Registry of Seismograph Stations <http://www.isc.ac.uk/registries/>
ALCN   40.55       0.48      177      IR       ISC  .IR      .                  Alcanar, Spain
ALCS   37.25417   -3.54389  1489      IR+GE    ISC  .IR      .   1982032 1982060 Alfacar, Spain
ALCT   47.6475  -122.0370     55      IR       ISC  .IR      .   2000188         Alcott Elementary School, Washington, U.S.A.
ALD    45.8194  -120.0670    427      IR       ISC  .IR      .   1975305         Alter Ridge, Washington, U.S.A.
ALDR   58.61000  125.40944   682      IR       ISC  .IR      .   1970001         Aldan, Sakha, Russia
ALE    82.4833   -62.4000     65      ERE      ISC  .IR      .   1961272 1990049 Alert, Northwest Territories, Canada
ALE    82.5033   -62.3500     65      IR       ISC  .IR      .   1990050         Alert, Northwest Territories, Canada
```

The elevation for station ALCS has been determined from Google Earth. The station coordinates and operational epoch in the IR entry for station ALE have been superseded by information provided by Bob Engdahl (ERE). Dots are added in the appropriate columns to distinguish the fields for agency, deployment and location for legibility.

Supplemental Station Files

As discussed in the main section on seismograph station data **mloc** can reference (with command sstn) supplemental station files to acquire information on the coordinates of stations that are not found in the master station file, i.e., stations that are not registered with the International Registry of Seismograph Stations (IR). Some clusters, e.g., those using only data from the ISC Bulletin, do not require any supplemental station files.

Supplemental station files are generally assembled for a specific cluster, and it is not uncommon to use several supplemental station files, possibly with different formats, for a cluster. **mloc** supports the use of up to 8 supplemental station files in addition to the master station file. With the current master station file (21,614 entries) and the default limit (24,000) on number of stations that may be defined, up to 2,386 stations can be defined in supplemental station files.

mloc supports several different formats for supplemental station files for convenience. When **mloc** opens a supplemental station file, the format is determined by an integer (*isstn*) in the first column of the first line. The rest of the first line (up to 96 characters total for the line) is considered an optional comment, which can be used to describe the dataset. The supported formats and corresponding values of *isstn* are:

0. Master station file format
1. ISC FFB format (geographic coordinates in deg-min-sec*10, elevation in m)
2. SEISAN format, with station code starting in column 3. Degree-minute-decimal seconds. Extended for operational epoch.
3. Generic format, decimal degrees, geographic coordinates. A basic format for quickly entering station data, supports 5-character station codes with agency and deployment codes.
4. China Seismic Bureau format
5. NEIC format
6. MSU format

Master Station File Format (*isstn* = 0)

The format of the master station file can be used for a supplemental station file. The format is described here.

ISC Station File Fixed Format (*isstn* = 1)

This format is based on the record type 91 used in the ISC's old Fixed Format ("96 Byte") data format, extended to support 6-character station codes. It uses geographic coordinates in deg-min-sec*10, elevation in m. **mloc** no longer supports 6 characters for a station code so only the first 5 will be used.

ISC Station File Fixed Format (*isstn* = 1)

Column	**Description**
15:20	Station code (a6)
62:63	Latitude degrees (i2)
64:65	Latitude minutes (i2)
66:68	Latitude seconds*10 (i3)
69:69	"N" or "S" (a1)
70:72	Longitude degrees (i3)
73:74	Longitude minutes (i2)
75:77	Longitude seconds*10 (i3)
78:78	"E" or "W" (a1)
79:82	Elevation, m (i4)

SEISAN Station File Format (*isstn* = 2)

The standard form of station data produced by the SEISAN software. Limited to 4-characters for station code. Geographic coordinates in degrees-decimal minutes. Elevation in m. The fields *date_on* and *date_off* are an extension of the format and they are optional.

SEISAN Station File Format (*isstn* = 2)

Column	**Description**
3:6	Station code (a4)
7:8	Latitude degrees (i2)
9:13	Latitude minutes (f5.2)
14:14	"N" or "S" (a1)
15:17	Longitude degrees (i3)
18:22	Longitude minutes (f5.2)
23:23	"E" or "W" (a1)
24:27	Elevation, m (i4)
34:40	Date_on (i7, optional)
42:48	Date_off (i7, optional)

Generic Station File Format (*isstn* = 3)

This is a basic format designed to carry all the information that **mloc** might use. **mloc** converts all supplemental station information, regardless of its original file format, into this format and writes it to the ".stn" output file so it can be easily extracted to create a cluster-specific supplemental station file. Geographic coordinates in decimal degrees. It supports agency and deployment codes, operational epoch and both station elevation and depth of burial (positive number), but the only required fields are station code, latitude, longitude and elevation. A comment can be added after column 68, but it is not read by **mloc**.

Generic Station File Format (*isstn* = 3)

Column	**Description**
1:5	Station code (a5)
7:11	Agency (a5, optional)
13:20	Deployment (a8, optional)
22:29	Latitude (f8.4)
31:39	Longitude (f9.4)
41:45	Elevation, m (i5)
47:51	Depth of burial, m (i5, optional)
53:59	Date_on (i7, optional)
61:67	Date_off (i7, optional)
69:	Comment

China Seismic Bureau Station File Format (*isstn* = 4)

A format used by the China Seismic Bureau for their station lists. Uses a 3-character, lower-case station code. Only defined for positive latitude and longitude, but could be easily extended.

China Seismic Bureau Station File Format (*isstn* = 4)

Column	**Description**
1:3	Station code (a3)
5:8	Elevation, m (i4)
10:11	Latitude degrees (i2)
14:15	Latitude minutes (i2)
18:21	Latitude seconds (f4.1)
25:27	Longitude degrees (i3)
30:31	Longitude minutes (i2)

34:37	Longitude seconds (f4.1)

NEIC Station File Format (*isstn* = 5)

This format is used with station coordinate data returned from the NEIC metadata server. If you use arrival time data from the NEIC ComCat server, especially for events within the U.S., you will probably need a supplemental station file to handle the many stations in U.S. regional networks that have not been registered with the International Registry at the ISC and also to solve the many station code conflicts that exist with registered stations.

A recent listing of the entire NEIC Metadata Server is provided as part of the **mloc** distribution, and the command nsmd is provided to help extract entries from it (in this format) that may be needed for a supplemental station file. Another method of building type 5 supplemental station files is to use the **mdget** utility program written and distributed by Dave Ketchum at NEIC (ketchum@usgs.gov). This ensures that the station data returned will be the most up-to-date.

NEIC Station File Format (*isstn* = 5)

Column	**Description**
4:8	Station code (a5)
40:47	Latitude (f8.4)
49:57	Longitude (f9.4)
58:62	Elevation, m (i5)

MSU Station File Format (*isstn* = 6)

A format used by Kevin Mackey at Michigan State University, especially for data from the former Soviet Union.

MSU Station File Format (*isstn* = 6)

Column	**Description**
1:5	Station code (a5)
6:7	Latitude degrees (i2)
9:10	Latitude minutes (i2)
12:15	Latitude seconds (f4.1)
16:16	“N” or “S” (a1)
17:19	Longitude degrees (i3)
21:22	Longitude minutes (i2)
24:27	Longitude seconds (f4.1)
28:28	“E” or “W” (a1)

30:33	Elevation, m (i4)

Travel Time Models

mloc uses three different approaches to calculate theoretical travel-times and derivatives, one based on a flat-layered crustal model that is adjusted to fit the observed data at local and near-regional distances, a global average model using the tau-p formulation to extract travel time information, and a linear model for Lg and T phases. Use of a crustal model is optional. The usage of a custom crustal model is discussed in a separate section.

Global Model

By default, **mloc** uses the tau-p formulation of Buland and Chapman (1983) to calculate theoretical travel-times, based on the 1-D global average model ak135 (Engdahl et al., 1998). If the data set includes arrival times for phases that are not included in ak135, such as PPP, those readings are generally flagged ("p") and ignored in the relocation. There are several cases (pwP and PKP precursors, see below) of commonly-reported seismic phases that do not exist in the ak135 phase set, but which are handled with minor adjustments to standard phases.

A global model other than ak135 could be used, if the user provides the necessary data files for the tau-p software:

- new_model.hed
- new_model.tbl

These files must be stored in the subdirectory */tables/tau-p/*; the command taup would be used to select it. It is a bit difficult to envision a circumstance where the use of a different global model would make a significant difference in the results of a calibrated relocation, because arrival time data at epicentral distances for which the global model would be used are converted to travel time differences. In other words, baseline differences are ignored.

pwP

Multiples of the teleseismic pP phase that reverberate in the water column of oceanic areas, named "pwP", were first reported by Mendiguren (1971). Such phases are not included in the ak135 model, which has no water layer, but Engdahl et al., 1998 found that the ISC Bulletin contains many instances of pwP that are mis-identified as pP, causing over-estimation of the focal depth for many oceanic events. The EHB location protocol includes the pwP phase, calculating travel time by making a correction to the pP phase travel time based on the water depth at the bouncepoint. This capability also exists in **mloc**, through the bptc command. The swP phase is not supported, nor are multiples (pwwP, pwwwP, etc.) of the pwP phase.

PKP Precursors

In the distance range 125-145° precursors to the main PKP arrival are frequently observed (e.g., Cleary and Haddon, 1972). They are sometimes identified as likely precursors with names like "PKPpre", but often appear as PKP with large negative residuals. **mloc** renames "PKPpre" as "PKPdfpre" and calculates a residual relative to PKP but it does not use them for relocation; they are automatically flagged ("p"). Depending on how they are reported in the data set, the phase naming algorithm may fail to catch some precursors and the user may need to edit the incoming phase name in the MNF file so that **mloc** handles these arrivals correctly. If not, they will often end up being flagged as outlier readings in the cleaning process.

Corrections

Several corrections are made to the travel times calculated from the global model. **mloc** does not support traditional station corrections or the patch corrections (regionalized station corrections) utilized in the EHB algorithm (Engdahl et al., 1998), since **mloc** takes a much different approach to the problem of determining hypocenters that are minimally biased by unknown Earth structure.

Ellipticity

Ellipticity corrections (in *real function ellip* in mloclib_tt.f90) are based on the formulation presented by Dziewonski and Gilbert (1976), with code adapted from Engdahl's EHB location code (based in turn on code by D.J. Brown, B.L.N. Kennett and W. Spakman). Ellipticity corrections are always made. They are typically ~0.1 s or less. These corrections have almost no effect on the relative locations of a cluster or on a calibrated relocation, but they are necessary if **mloc** is used to do a "traditional" teleseismic location.

Station Elevation

The code (in *get_elev_corr* in mloclib_tt.f90) used to calculate station elevation corrections is based on the algorithm used in the new ISC locator written by Istvan Bondar. The corrections are based on P- and S-velocities that can be specified by the user with the command secv. Station elevation corrections can be turned off with the corr command.

Bounce Point Corrections for Depth Phases

In the global model, travel times of teleseismic depth phases (pP and sP) are calculated under the assumption that the "short leg" is reflected from the reference ellipsoid (i.e. zero elevation). In developing a refined method of analyzing teleseismic depth phases for the EHB algorithm, Engdahl attempted to reduce the biasing effect of varying elevation at the surface bounce points. Travel time is increased when the bounce point is above sea-level, reduced when it is below sea-level (i.e., in oceanic areas). This correction is available in **mloc**, through the command bptc, with topography (and bathymetry) taken from the ETOPO5 digital elevation model.

Timing Errors

In rare circumstances it may be suspected, or even confirmed, that the timing system of a station was out of calibration when an arrival time reading was made. The command terr is provided to explore the consequences of such errors. If the timing of a station cannot be trusted it would be far preferable to flag the reading ("t") and not use it rather than try to correct it.

Linear Travel-time Models

Neither the continental Lg phase nor the oceanic T phase are included in the ak135 global model, but **mloc** is able to use both phases in a relocation because their travel-times can be predicted fairly accurately with simple linear-with-distance models. Lg is a very commonly reported phase and contributes significantly to many relocations. T phase arrivals have been studied since the early 1950s but have only recently begun to appear in open seismic bulletins such as the ISC Bulletin. They have been shown to have value in localizing seismic sources in the oceans when there is adequate azimuthal coverage by T phase stations (hydrophone arrays) and they play a role in nuclear monitoring (Okal, 2001) but limited experience so far indicates that these data have almost no value (because of large scatter) in high-resolution relocations when combined with traditional "solid Earth" arrival time data (the skip * T * command is often used). Nevertheless, the ability to carry T phase readings through a relocation analysis may eventually lead to improvements in this regard.

Both Lg and T phases are modeled with a linear model (y = ax + b) parameterized by a zero-distance intercept (usually near zero) and a derivative in epicentral distance (s/deg). Default values of these parameters are provided but it is usually desirable (especially with Lg) to derive an improved model from the observed travel times and input them with the commands lgtt and tptt. Use the command ttou to create a file with the distance and travel-time data for a phase of interest, read the data into a program that can do basic line fitting.

Crustal Models

This section discusses the format used for custom crustal velocity models in **mloc**, and some aspects of developing a crustal model for a specific cluster. Although we call them "crustal models" the models discussed here actually consist of one or more layers in the crust over a representation of the upper mantle which can be either a single thick layer (pseudo-half-space) or several layers extending to a depth that is adequate to account for refracted Moho arrivals to a reasonable distance.

The code used in **mloc** for this purpose is borrowed from the single-event location program HYPOSAT (Schweitzer, 2001). HYPOSAT is capable of travel-time calculations for a large number of crustal seismic phases with models constructed of gradient layers as well as constant-velocity layers. For **mloc**, however, many of HYPOSAT's capabilities have been circumscribed.

Usage

A custom crustal model is declared with the command lmod. The use of a custom crustal model is *optional*. If a custom crustal model is not requested **mloc** will determine travel times and

derivatives for all seismic phases using the tau-p representation of the ak135 global model (Engdahl et al., 1998).

If a custom crustal model *is* specified in the command file or interactively by the user when running **mloc**, that model will be used to calculate travel times and derivatives for the crustal phases Pg and Sg and the Moho-refracted phases Pn and Sn, but only to an epicentral distance that is specified in the first line of the crustal model file. Typically that distance will be close to the Pn/P cross-over distance, around 17° epicentral distance. Beyond the specified distance, the tau-p implementation of ak135 is used for all other phases besides the ones calculated with the custom model (so, Pn will not exist beyond 17° in this example). Teleseismic phases that are observed at short epicentral distances (e.g., PcP) are still calculated from ak135 with the tau-p formulation.

Phases

In general, a custom crustal model is used in **mloc** only when doing direct calibration, i.e., estimating the hypocentroid of the cluster using arrival time data mainly from direct crustal phases at epicentral distances where they are the first-arriving phase (some Moho-refracted arrivals sometimes sneak in). Experience has shown that this kind of analysis works best when all reported arrival times for crustal phases are treated as Pg or Sg. Identification and nomenclature of crustal phases in standard catalogs (e.g., the ISC Bulletin) is too haphazard to rely on, so many arrivals must be assigned new phase codes for use in **mloc**. The travel-time differences between different theoretical branches of crustal phases are insignificant, given that the crustal structure is seldom known in any detail and the locations of the sources are, before a calibrated location has been determined, often uncertain by tens of kilometers or more. This philosophy does result in loss (or corruption) of small amounts of data in some cases, but the benefits of a simplified approach to phase identification for crustal phases far outweighs those losses.

Model Structure

Although the implementation of HYPOSAT's code for calculating travel times is capable of handling models with many layers, including layers with gradients in velocity, experience has shown that crustal models used to determine calibrated locations in **mloc** should consist of no more than 3 crustal layers. Gradient layers are supported in **mloc** but I nearly always use constant-velocity layers in order to keep things conceptually simple; I have never seen a dataset that required gradient layers in order to adequately model the arrival time data. Although there is a widespread belief that the key to accurate locations of earthquakes is a very complex (but presumably accurate) velocity model, the reality is that most arrival time datasets do not have the quality and coverage to reliably resolve much more than the average crustal velocity, i.e., a single layer of constant velocity (or a gradient at best), and the average crustal thickness *when the source locations are also uncertain*. In many regions, of course, some additional constraints on gross crustal structure can be derived from other kinds of studies, but arrival time datasets are in general insensitive to structures more complex than an upper and lower crustal layer, and

perhaps a sedimentary layer on top. The strong variations in velocity at very shallow depths that undoubtedly exist in nature are averaged out over an earthquake cluster extending across 100 km or so, and have negligible influence on travel times at local and regional distances in any case.

The details of upper mantle structure in a custom crustal model for **mloc** have very little impact on the relocation. The model needs to extend to a depth that is adequate to generate Pn arrivals to the maximum distance specified in the first line of the file, several hundred km in the usual case where the crustal model is used out to the cross-over distance from Pn to P, around 17°. Since only the relative arrival times of Pn phases would normally be used for relocation, the absolute accuracy of the theoretical Pn times is of minor concern and in any case observed Pn times generally vary by many seconds as a function of azimuth and distance (i.e., at different stations) so the best one can expect with a 1-D crustal model is average agreement. In many cases a single thick layer representing the upper mantle is adequate, but there is no harm in using some version of the layered upper mantle from ak135 or another favored global or regional model, as long as it does not contain any low-velocity zones. This will result in a slight downward curvature of the theoretical travel-time curve of Pn at larger distances that may or may not be observed in the actual data.

Low-velocity zones are not supported.

Format

Formatting of the file defining a custom crustal model for **mloc** is exactly the same as defined for HYPOSAT. The format is very specific and small deviations can lead to unexpected results. The subroutine that reads crustal models is named *rd_loc_mod*, found in the module *hyposat_loc.f90*. The first line is read by:

```
read (iunit,'(a)') line
read (line(1:4),'(f4.1)') rmax
```

The distance limit to which this model will be used is read from the first four characters as a real value in f4.1. The rest of the line can be used for a comment about the model. All following lines in the file are read by:

```
read (line,'(3f10.3,a4)') h(i), v0(1,i), v0(2,i), azo(i)
```

where *h* is a depth in km. *v0* is an array with two columns that hold Vp and Vs for the layer in km/s, and *azo* is a character variable with the value “CONR” or “MOHO” that identifies the major crustal discontinuities. The label is used only in the line that defines the “upper” side of the discontinuity. First-order discontinuities require two lines at the same depth in the input file.

The code recognizes the “CONR” flag but for the reasons discussed above, the declaration of a Conrad Discontinuity in the model is strongly advised against. It is possible (probably, it has not been much tested) to configure **mloc** to use the associated Pb and Sb phases but no case has ever been encountered where it was helpful and it is usually very unhelpful. On the other hand, the Moho discontinuity should always be declared.

Examples

Here is a crustal model based on the ak135 global model:

```
17.0 Crust and upper mantle of ak135
     0.000     5.800     3.460
    20.000     5.800     3.460
    20.000     6.500     3.850
    35.000     6.500     3.850MOHO
    35.000     8.040     4.480
    77.500     8.045     4.490
   120.000     8.050     4.500
   165.000     8.175     4.509
   210.000     8.300     4.518
   210.000     8.300     4.523
   260.000     8.483     4.609
   310.000     8.665     4.696
```

This model uses gradient layers in the upper mantle, following the original, but a simplified version using a single layer of constant velocity would work as well:

```
17.0 Crust of ak135 with simplified upper mantle
     0.000     5.800     3.460
    20.000     5.800     3.460
    20.000     6.500     3.850
    35.000     6.500     3.850MOHO
    35.000     8.040     4.480
   210.000     8.040     4.480
```

The fit to observed arrival times of Pn at larger distances might be somewhat poorer but that would not compromise a calibrated relocation analysis.

Location

As discussed in the section on data files, custom crustal models are normally stored either in the cluster directory (along with the event data files, command files and output files from individual runs), or in the *tables/crust* directory. Because they are referenced by the full path name in a command file they can actually reside anywhere in the user's directory structure.

Forensics

The easiest way to determine the details of the crustal model that was used for a specific run of **mloc** is to have access to the cluster directory, find the ".cr" model in the directory and check in the header section of the ".summary" file that it is indeed the model that was used; the pathname will be given in parentheses after the global model that was used in the line that begins with

"Travel times: ". If for some reason you do not have access to the crustal model file, the full crustal model (if one was used) is also logged in the ".log" file of the run.

Model Refinement

For a calibrated relocation analysis, I normally make the first run with the ak135 model distributed in Data Files. The only difference between this and running without a custom crustal model (using tau-p formulation of ak135 for everything) is that the only crustal phases will be Pg and Sg. Decisions about refinement to the model are based on examination of several output plots of travel-time vs distance (see Summary Plots):

- Focal depths will need to be considered in parallel with the crustal part of velocity model. They don't need to be completely finalized but they need to be in the ballpark. For most clusters a well-chosen default (average) depth for the cluster will suffice. Crustal thickness cannot be established until focal depths are stabilized, and systematic changes to focal depths at a later stage of analysis will require adjustment of crustal thickness.
- Changes to Vp and Vs in the crust are primarily based on observation of distance-dependent residuals in the local distance TT (tt5) plot and to a lesser extent the near-source TT (tt4) plot. It is usually very difficult to constrain the relative thicknesses of crustal layers. Determining whether to make changes in upper or lower layers of the crust can be guided to some extent by attention to the pattern of residuals in the near-source TT plot, but the effects are similar to the signature of errors in assumed focal depth.
- Changes to the Vp/Vs ratio are based primarily on the offset from zero mean of the P and S residuals of near-source TT (tt4) plot and to a lesser extent on the pattern of residuals in the local distance TT (tt5) plot. They should both have offsets with absolute value less than ~0.1 s in the near-source TT plot.
- Changes to crustal thickness are based on the fit to Pn arrivals in the local-regional TT (tt6) plot. When the hypocentroid is constrained using crustal phase alone, as it is in direct calibration, the arrival times of Pn are controlled by crustal thickness. Thicker crust = later Pn arrivals. Sn arrivals (local-regional S plot, tt7) can be used for this too, of course, but they are often a bit inconsistent with the Pn data, which should be preferred.
- Changes to Pn velocity are best made through reference to the local-regional TT (tt6) plot, as upper mantle velocity variations are more clearly reflected at greater distances. However, if there is a discrepancy between closer and farther distances (often around 10°) I would favor the near-regional data for refining the model.

It should be emphasized that in direct calibration the most important aspect of the crustal model is the average crustal velocity, as measured by the fit of the observed arrival time data to the theoretical times over the epicentral distance specified for the location of the hypocentroid, which can range from as close as 0.2° to as much as 1.5° or more. The Pg/Pn cross-over distance (a function of average focal depth and crustal thickness) generally establishes an upper bound on this distance, but if the data permit (sufficient number of readings with good azimuthal

coverage), restricting the distance range reduces the importance of the velocity model for the calibration. Beyond the cut-off distance for estimation of the hypocentroid, the fit of the observed data to the theoretical TT curves is of little consequence for the relocation analysis, but the the average crustal thickness obtained by fitting the Pn data closely is certainly a parameter of broader interest, worth a careful investigation. Although a calibrated relocation analysis with **mloc** can reveal useful information about certain aspects of the velocity structure in the source region, it must be remembered that the model was developed only to predict travel times of certain seismic phases. Those phases sample the crust in only limited ways and do not constitute a credible investigation of crustal structure as a whole.

Output

This section describes the various output files (including plots) that may be generated by a successful run of **mloc**. Plots are always in PDF format. All text files and plots are identified by the *basename* for the run which is the first interactive input to **mloc**, and filename suffixes are used to identify the nature of file. There is a set of default output files that are always produced and many optional files whose creation is controlled by commands, either directly or as a consequence of the type of relocation (i.e., calibration) specified.

The terminal window in which **mloc** is run will contain some textual information that is not repeated in any output file but all information that has been found to be worth retaining is written to at least one output file. It is not necessary to save the terminal window after a run of **mloc**.

Default Output Files

The following files and plots are always created, whether a cluster is to be calibrated or not. The simpler files are described in more detail after the list. The more complex files are described on their own page.

- **~_base.pdf**: Base Plot (map of final locations)
- **~.dat0**: Input Data Bulletin (record of all input data)
- **~.depth_phases**: Depth Phases File (all data relevant to depth constraint)
- **~.hdf**: HDF File (hypocenter data summary, at least one of several flavors)
- **~.kml**: KML File (for plotting in Google Earth or equivalent)
- **~.log**: Log File
- **~.phase_data**: Phase Data File (complete list of phase data and usage)
- **~.plog**: Phase Identification Log (information related to changing phase names)
- **~.rderr**: Empirical Reading Error File (for each station-phase with 2 or more instances)
- **~.stn**: Station Data File (especially useful when there are supplemental station files)
- **~.summary**: Summary File (summary of the input parameters, progress through iterations and final locations)
- **~.ttsprd**: Travel Time Spread File (for each observed phase)
- **~.xdat**: Gross Outliers

Every run of **mloc** also creates a folder named *~_gmt_scripts* containing the GMT scripts used to create any plots (the *~_base.bash* script at a minimum). These are preserved for cases where the user may wish to modify a script for research or publication purposes.

Input Data Bulletin (~.dat0_mnf)

As each event's data file is read it is added line-for-line to a bulletin in MNF format named *~.dat0_mnf* for archival purposes. This is the only reliable record of the input data to a specific run of **mloc**, since event data files are constantly being edited during the course of a relocation analysis. If necessary the *~.dat0_mnf* file could be unpacked into event files using the mnf_search utility program in order to recover the input of a particular run. This has almost never been necessary but it is simply good practice to retain the capability. Differential time data cannot be retained in the *~.dat0_mnf* file since each datum references two events.

KML File (~.kml)

The final locations of the cluster are written in Keyhole Markup Language (KML) to an output file named *~.kml* for plotting in Google Earth or compatible programs. The file references icons that vary in size according to magnitude, and in color according to focal depth:

- 0-9 km: red
- 10-19 km: green
- 20-29 km: skyblue
- 30+ km: blue

A yellow icon is used for events set to the cluster default depth.

The icon image files are stored in the directory */mloc_distribution/mloc_working/kml/*, but on the first run of **mloc** they are copied into a subdirectory of the cluster directory named *_kml* and the relative paths are referenced in the *~.kml file*. This makes it possible to open the *~.kml* file after the cluster directory is moved to the */mloc_distribution/clusters/* directory (or elsewhere) and display the correct icons in Google Earth.

Log File (~.log)

The file *~.log* is used to collect information about the relocation that is normally of little interest, but may be helpful if **mloc** is behaving strangely or giving unexpected results. The main sections deal with:

- Events that were killed (commands memb and kill) in the command file.
- Convergence limits (command shcl).
- The local velocity model (command lmod).
- Starting locations read from a previous run (command rhdf).
- If the bdps command has been set "on", the list of stations read from the file */mloc_distribution/mloc_working/tables/stn/bdps.dat*.
- Progress of reading the input data file for each event and the number of phases re-identified.
- Depth phase readings converted to relative depth phases (e.g., pP-P).

- For each iteration, statistics on the number of readings used to estimate the hypocentroid, the number lost because of epicentral distance, the number lost because of large residual.
- For each iteration, the change in hypocentroid, weighted epicentroid, bias correction (command bias).
- For each iteration, the changes in cluster vector parameters for each event.
- For a calibrated relocation, a list of events with hypocentral data written in the form needed for the cal_ command, so that they could be cut and pasted into another command file to implement indirect calibration.
- A simplified list of hypocentral parameters for each event, useful for some forms of publication and communication.
- For direct calibration, statistics on mean residuals of readings used for the hypocentroid and their use for indicating the need for depth adjustment. Outliers from this list are listed in the terminal output of **mloc**.

There are two commands that will cause additional information to be written to *~.log*. They are both normally used only in a development environment, to track down bugs in the code, but the "lighter" one, vlog, may be of interest to a new user of **mloc** who wants to better understand the inner workings. Most of the extra logging will not be comprehensible without making frequent reference to the source code. The "heavy duty" logging command is dbug. Its use will result in a *~.log* file several tens of MB in size, but it is nearly always possible to isolate the source of a bug through careful inspection of the file. In dire circumstances it can help to enable both logging options, although there is some overlap.

Phase Identification Log (~.plog)

This log file, named *~.plog*, is not, technically, part of the "default" set of output files because it could be turned off using the phid command. The default state is to do phase re-identification. Most data sets would perform poorly if phase re-identification is turned off.

The standard content of the phase identification log is a list of cases where the name of a phase read from the event file has been changed. Phase re-identification is done as event files are read and again after the first relocation iteration; each pass is logged.

Optional Output

Depending on the use of various commands there will be additional output files from a run of **mloc**, in the form of text files or plot files (PDFs). Commands that lead to additional output files are:

- bloc: Output file ~.bloc for export of mloc hypocenters to BayesLoc as priors.
- cal_: One new output file ~.cal and an extra ~.hdf file from indirect calibration.

- ccat: Two output files (~.comcat and ~_plots.pdf) in a subdirectory, containing detailed results of a run for import to the GCCEL or NEIC ComCat databases.
- datf: Output file of the final state of the cluster in MNF v1.3 bulletin format.
- dcal: Two extra output files and a new plot from direct calibration.
- epap: Map of empirical path anomalies for a selected phase.
- eplt: Simplified base plot, showing only confidence ellipses.
- fdhp: Histogram of focal depths.
- lres: Output file ~.lres of readings with cluster residual above a threshold.
- mdou: Output of event locations in a form for easy import to GMT.
- oldr: Output file containing all readings within a specified epicentral distance range.
- plot: Base plot for selected subset of events.
- pltt: Standard summary travel-time plots.
- rdpp: Plots of relative depth phase data for single events.
- splt: Simplified base plot, all events shown with the same symbol (traditional seismicity plot).
- tlog: Output file ~.taup with detailed logging of tau-p calculations.
- tomo: Output files ~.tomo designed for tomographers.
- tt5e: Local distance travel-time plot for a specified event.
- tt5s: Local distance travel-time plot for a specified station.
- ttou: Output files ~.ttou with travel-time and residual data for specified phases.
- xsec: Cross section plot.

Direct Calibration Output

The dcal command tells **mloc** that a direct calibration relocation is being conducted, meaning that the epicentral distance range of data used to estimate the hypocentroid is restricted in order to minimize the biasing effect of unknown Earth structure. The epicentral distance range for this purpose is usually in the range 0.5-1.5°. In this case, the only HDF file produced is the *~.hdf_dcal* file.

When a direct calibration relocation is done, two new files are produced in addition to the default output files:

- ~.dcal: Some extra details about the uncertainties of the calibrated locations.
- ~.dcal_phase_data: Details of the arrival time data used to estimate the hypocentroid.

Direct calibration also produces a new plot, the direct calibration raypath plot, named *~_dcal.pdf*, which makes a base map over a region several hundred km across in order to display the raypaths of the arrival time data used to estimate the hypocentroid. This is very useful for evaluating the azimuthal coverage for direct calibration.

Indirect Calibration Output

The cal_ command (actually, family of commands) tells **mloc** that an indirect calibration relocation is being conducted. In this case the calibration step occurs separately, after the relocation, so two HDF files are written. The first one will be either *~.hdf* or *~.hdf_dcal*, depending on whether the initial relocation was done as uncalibrated or as a direct calibration The second will be *~.hdf_cal*, reflecting the indirect calibration. In the case where both direct and indirect calibration are done, the user must evaluate which outcome is to be preferred; the level of calibration achieved by each method is one criterion, but not necessarily the only one.

The output from indirect calibration also includes a file named ~.cal, containing a detailed record of the analysis used to determine the shift of the hypocentroid that provides the optimal match with the calibration data for a subset of events in the cluster. This analysis takes into account the uncertainties of the relative locations (cluster vectors) of the events with calibration data and the uncertainties assigned to the calibration data itself.

When indirect calibration is done, even if direct calibration was done beforehand, the results from indirect calibration will be reflected in all output files and plots (except *~.hdf* or *~.hdf_dcal* files).

File Types

~.bloc File

A *~.bloc* output file is only created through use of the bloc command. It has a very specialized purpose, to assist in the use of the hypocenters determined in an **mloc** relocation as "priors" in BayesLoc. This is only likely to be of interest in the case of a calibrated **mloc** relocation.

The idea behind this command was that combining **mloc** and **BayesLoc** in this manner might be a useful way to leverage the value of a limited number of calibrated hypocenters to obtain improved catalogs of seismicity over a broader region. This was explored in Ezgi Karasözen's PhD Thesis. If you are interested in this research contact Ezgi to discuss it in more detail.

Conversion of the ~.bloc file

The *~.bloc* file cannot be directly used in **BayesLoc**. It is converted to a **BayesLoc** *origin_prior.dat* file by the command:

```
awk '{print "1000"NR"\t"$2"\t"$3"\t"$4"\t"$5" \t"$6"\t"$7"\t"$8}' *.bloc > mloc_priors.dat
```

In **BayesLoc** these values will be interpreted as:

```
ev_id, lat_mean, lon_mean, dist_sd, depth_mean, depth_sd, time_mean, time_sd
```

It may still be necessary to change the ev_id (event ID). This example uses a simplistic ev_id that starts from 100001. If a parameter is assumed to be known without error, you can assign 0 to the corresponding standard deviation. If you don't trust the information you are inputting, you can assign -1 to the standard deviation. For example, if you want **Bayesloc** to ignore the focal depths from the *~.bloc* file, you would write "-1" to the *depth_sd* column. Here is an example. One line from *~.bloc* is:

```
20180812.1458.53  69.55582  214.79987        1.378   11.0    4.0       1534085933.336  0.20
```

which when converted to **Bayesloc** input is:

```
100001 69.55582 214.79987 1.378 11.0 4.0 1534085933.336 0.20
```

Converting a Station File

It may be desired to use the station information from **mloc** in **BayesLoc**. This is done by extracting the section of **mloc's** ~.stn file that lists all stations used in the relocation and reformatting:

```
awk '{print $1, $2, $3, $4/1000}' mloc_stations.dat > bayesloc_stations.dat
```

Converting Arrival Time Data

If you need to convert **mloc's** arrival time data (MNF v1.3) for use in **BayesLoc**, contact Ezgi Karasözen.

~.cal File

This section describes an optional output file (*~.cal*) that is produced only when indirect calibration is used. It displays a great deal of information related to the problem of shifting the hypocentroid of a relocation (either uncalibrated or calibrated with the direct method) to best match a set of calibration locations that were specified by the cal_ command.

If there is only a single calibration event the calculation is straight-forward. The uncertainty of the calibrated hypocentroid will be the matrix sum of the covariance matrix for the calibration location and the cluster vector of the calibration event. With multiple calibration events the process is much more complicated. To fully understand it, careful inspection of the source code in the module *cal_shift.f90* is required. That code has a lengthy comment section describing the approach, reproduced here:

When calibration locations are available for one of more cluster events, calculate the shift needed to bring the cluster into best alignment (the "optimal" shift vector). The original method (mode = 0) was just to take the average of all the individual shift vectors. this is still supported in the code but to use it you would need to edit the value of "mode" in the call to this subroutine and recompiled. The preferred way (mode= 1, hard-wired in the code) is to take into account the uncertainties of the calibration data and the cluster vectors, through their covariance matrices,

and also to consider the possibility of bias that is not included in the formal covariance matrices. Covariance matrices for calibration locations are input by the user, using the command “cal_”.

The calibration event covariance matrices (rcv) are assumed to be already scaled to a 90% confidence level. This could be done, for example, by applying a utility program that converts from the confidence ellipse calculated in some other code (e.g., a single event location code) to an equivalent covariance matrix. In the case of InSAR analysis or geologic information (fault trace) it may be seat-of-the-pants. The covariance matrices for cluster vectors (ccv) are first converted to 90% confidence ellipses with the normal Bayesian approach, then converted back to raw covariances (subroutine ell2cv), after which they can be added to the calibration event covariances (because they are completely independent estimates), yielding the combined covariances (scv). These are converted (no scaling) directly to 90% confidence ellipses for the combination. The area of these combined confidence ellipses is used for inverse weighting of the shift vectors estimated for each calibration event, to get the estimate of the optimal shift vector for the entire cluster.

The uncertainty of the optimal shift vector (gcv) is based on the weighted (same weights as for the individual shift vectors) combination of the combined covariancess (scv), expressed as a confidence ellipse (90%). This is only part of the uncertainty, however. There is additional uncertainty, which can be thought of as associated with the “unmodelled” or “biased” part of the indirect calibration problem; it arises if any of the calibration event locations are biased. Such bias could arise with local network solutions if the velocity model is poorly chosen, or if readings from too great a distance have been used, or if some phases are mis-identified or outliers are not properly weighted. It can also arise if a serious error (e.g., an outlier reading, with poor azimuthal coverage) has caused bias in the estimate of a cluster vector in the hypocentroidal decomposition analysis. Such bias is seen by comparing the individual estimates of shift vector to the optimal shift vector. If there are departures that are statistically unlikely, given the alleged confidence of the calibration locations and cluster vectors, then there is a need to account for the unmodeled error. Graphically, it is easily seen by plotting residual shift vectors (subtract the optimal shift vector) and comparing to the confidence ellipses of each shift vector (scv).

Instead of taking the RMS of the residual shift vectors to estimate the level of inconsistency (this is too sensitive to outliers and overestimates the uncertainty), I do a test based on the coverage statistic for the residual calibration shift vectors, for which the cumulative binomial probability distribution gives the probability of the observed number of “uncovered” vectors. See ‘rdbt_test2’ for more explanation. If the null hypothesis is rejected, the test is repeated after adding a small amount to each covariance matrix, equivalent to adding a circular uncertainty. When the null hypothesis cannot be rejected, this extra covariance is converted to a “radius of doubt”, which is added to the modelled uncertainty (gcv) to yield an “augmented GT covariance matrix” (agcv). The area of the equivalent ellipse is converted to a circular area, and the radius of that circle is a useful measure of the calibration level of the cluster. The calibration levels of individual events (accv) are found by adding their cluster vector covariances to agcv.

I brought back the older algorithm (rdbt_test1) for radius of doubt (v8.1) that is based on a test of the hypothesis that all residual shift vectors have zero length. I have kept the test based on coverage statistics for now, but it has the problem that we seldom have more than a couple calibration events and the 90% coverage requirement translates into 100% coverage requirement if there are fewer than 10 calibration events. If there are 10 or more calibration events then the larger of the two estimates is used. Otherwise the test based on zero-length residual vectors is used.

These concepts are illustrated with the *~.cal* file from a cluster based on nuclear tests at the Chinese test site Lop Nor. The Chinese government has never released official information on these tests but many research projects have established likely associations between specific tests and test sites (boreholes and adits) observable in satellite imagery. The hypocentroid was estimated from only P arrivals (command phyp) in the distance ranage 30-90°, which normally provides a fairly good estimate of the absolute location, but certainly not a calibrated one. The first part of the file is:

```
 On   15 calibration events:

Event   5
Cluster vector CV for event   5
   kcrit2:    5.463
   90% confidence ellipse:  -62.033   2.661   3.060
   Area:   25.575 km**2
Calibration location CV for event   5
   90% confidence ellipse:    0.000   1.000   1.000
   Area:    3.142 km**2
Weights for event   5
   w12 =    0.348
   w3 =     1.000
   w4 =    21.033
Covariance matrices for calibration event   5
    ccv:      1.622     0.173     0.173     1.388     0.000     0.012
   sccv:      8.860     0.945     0.945     7.581     0.000     0.038
    rcv:      1.000     0.000     0.000     1.000     1.000     0.010
    scv:      9.860     0.945     0.945     8.581     1.000     0.048
```

Event 5 is the first of 15 events in the cluster that are being used for indirect calibration. The confidence ellipses for the event's cluster vector and the associated calibration location are converted to covariance matrices and combined (*scv*), an estimate of the uncertainty of the calibration shift for each calibration event. The first four covariance elements listed refer to the epicenter, the fifth one is focal depth and the sixth is origin time.

The weight factor *w12* will be used for weighting the multiple estimates of epicentral shift when they combined (inverse weighting, so larger area = less weight). Weight *w3* refers to focal depth. Weight *w4* refers to origin time. The same display follows for each of the other 14 calibration events.

Using the weights for each calibration event a weighted mean is calculated for all the calibration shift vectors.

```
Weighted mean (REF-SOL) calibration shift:
   Latitude :   -0.045 deg
   Longitude:   -0.080 deg
   Depth    :    0.000 km
   OT       :    0.252 sec
```

This output is also printed to the terminal window. The depth shift is zero because all events had their depth set to the calibration depth. Next, we need to determine the uncertainty of the estimate of calibration shift. To do that we consider the consistency of the shift vectors from individual events. We start with a weighted average of all the shift vector covariance matrices.

```
Weighted average CV for all calibration events
gcv:      1.900     0.170     0.170     1.660     1.000     0.028
```

Subtracting the weighted mean shift of each parameter from the individual shift vectors produces residual vectors, listed in the next bit of output. If our estimate of uncertainty is reasonable, most of those vectors should lie inside the corresponding ellipse. The coverage parameter *covp* is less than 1.0 in that case.

```
Residual calibration shift vectors and COVP, based on SGCV
iev     dtiev     ddiev       rvl        dx        dy      covp
  5     0.676     0.000     3.360    -2.992    -1.529     0.999
  6     0.093     0.000     2.271    -1.782     1.408     0.713
  8     0.073     0.000     1.675    -1.422     0.885     0.911
  9    -0.035     0.000     0.434     0.202    -0.384     0.053
 10     0.321     0.000     3.468    -1.089     3.293     1.858
 11     0.250     0.000     1.728    -0.056    -1.727     0.894
 15    -0.058     0.000     0.740    -0.408     0.618     0.171
 16    -0.234     0.000     0.646     0.466     0.448     0.125
 18    -0.511     0.000     1.303     1.028    -0.801     0.607
 19     0.444     0.000     2.016    -2.004     0.215     1.167
 22     0.071     0.000     0.940     0.869     0.358     0.290
 23    -0.023     0.000     0.129     0.119     0.050     0.005
 25    -0.125     0.000     1.033     1.026     0.120     0.365
 26    -0.145     0.000     1.599     1.304    -0.925     0.903
 28    -0.104     0.000     1.454    -1.318     0.614     0.758
Coverage:  13 /  15 =  0.867
```

The statistical test on coverage is done a little later. First is a test of the null hypothesis that all these residual vectors have zero length. A variation of this test was introduced in the original paper on hypocentroidal decomposition (Jordan and Sverdrup (1981), where the null hypothesis was that all cluster vectors have zero length, i.e., all events in the cluster could have occurred at the same location.

```
Radius of doubt test based on null hypothesis that all residual cluster vectors have zero length
Critical value =     37.1 at 90%

Individual event contributions to kocs2:
  5     0.98734
  6     0.59863
  8     1.01530
  9     0.37448
 10     2.20194
 11     2.68454
 15     0.10395
 16     0.25038
 18     2.48718
 19     1.35064
 22     0.18115
 23     0.24012
 25     1.67360
 26     3.31006
 28     0.84308
rdbt_test =   0.00; observed value =  18.30; null hypothesis cannot be rejected
rdbt1 =   0.00
```

If the observed value of the statistic *kocs2* exceeds the critical value, the null hypothesis is rejected, in which case we would add a small increment of circular uncertainty ("radius of doubt") to the individual covariance matrices and try again. In this case the null hypothesis cannot be rejected, so we move on to the other test of consistency of the calibration shift vectors.

```
Radius of doubt based on coverage statistics

Threshold precentage of probability for radius of doubt test: 0.10
                rdbt    nr    nrc    k    coverage   P(X  k)
rdbt_test =     0.000    15    13     2     0.867     0.451; null hypothesis cannot be rejected
rdbt2 =   0.00
```

As mentioned in the excerpt from the code documentation above, this test only makes sense if there are at least 10 calibration events. For this cluster there are 15 calibration events, so we can consider this as an alternative to the test of the null hypothesis that all residual vectors could actually be zero length. Both tests result in a zero radius of doubt, but if one test produced a larger radius of doubt than the other we would take the larger one.

In the following output the value for the epicenter is the radius of doubt calculated above (zero in this case). The values for depth and origin time are based on the robust estimate of spread of the residuals. The RMS values are also shown for reference.

```
Inconsistency (between multiple calibration data) terms for augmented GT covariance:
   Epicenter:    0.000 km (rms =    1.786 km)
   Depth    :    0.000 km (rms =    0.000 km)
   OT       :    0.250 sec (rms =    0.284 sec)
```

The previous estimate of the covariance matrix for the calibration shift (*gcv*) is updated by adding the results of these tests on inconsistency. We call this the "augmented GT covariance matrix" (*agcv*).

```
Augmented GT covariance matrix
agcv:      1.900     0.170     0.170     1.660     1.000     0.090
```

Now *agcv* is converted to a confidence ellipse, as our best estimate of the uncertainty of the calibration shift:

```
Uncertainties from the augmented GT covariance matrix:
   Confidence ellipse:     -62.687     1.254     1.410
   Area of ellipse:    5.554 km**2
   Equivalent circular radius (CE level) =    1.330 km
   Depth shift uncertainty:      1.000
   OT shift uncertainty:      0.301
```

The confidence ellipse is defined by three values, the azimuth of the semi-minor axis and the lengths of the semi-minor and -major axes. The final block of output provides a listing of the total uncertainty (including cluster vector) of calibrated hypocenter for each event in the cluster. For calibration events the coverage test (*covp*) now applies to the calibrated epicenter. This test can be done in one of three ways, distinguished by the choice of which uncertainty estimate to apply:

- Traditional: calibration events get the uncertainties of the calibration data.
- Systematic: all events done same way, add calibration uncertainty to cluster vector uncertainty.
- Optimal: use traditional or systematic method, whichever has shorter semi-major axis.

The user can control which mode is used with the command ctyp. Only the first 10 events are shown here:

```
Cumulative uncertainty of calibration-shifted cluster events
iev     alpha       xl        yl      area      eqr      ddep       dot      covp     caltype
  1    83.769     3.527     4.618    51.161    4.035     1.000     0.381     0.000
  2   -54.243     2.504     2.989    23.512    2.736     1.000     0.355     0.000
  3   -44.588     2.907     5.222    47.692    3.896     1.000     0.358     0.000
  4   -71.458     2.061     2.492    16.131    2.266     1.000     0.340     0.000
  5   -62.134     2.941     3.369    31.130    3.148     1.000     0.358     1.097  systematic
  6   -64.958     2.472     3.064    23.794    2.752     1.000     0.343     0.828  systematic
  7   -42.097     1.805     2.155    12.217    1.972     1.000     0.341     0.000
  8   -66.582     1.434     1.944     8.762    1.670     1.000     0.330     1.351  systematic
  9   -49.256     1.556     1.873     9.157    1.707     1.000     0.330     0.075  systematic
 10   -11.258     2.335     2.745    20.129    2.531     1.000     0.356     2.198  systematic
```

~.comcat File

A *~.comcat* file is one of two files (a ~_plots.pdf file is the other) created when the ccat command has been used to create output files for importation to the GCCEL database. The two output files are stored in a subdirectory of the cluster series directory named *~_comcat*. See the Mangyshlak cluster for an example.

The *~.comcat* file is a text file containing a bulletin of the cluster for the current run in MNF v1.4 format.

~.dcal File

When a relocation analysis in **mloc** is done with direct calibration (command dcal), it means that the hypocentroid will be estimated with arrival time data over a restricted epicentral distance in order to minimize the biasing effects of unknown Earth structure. Usually the cutoff in epicentral

distance is in the range 0.5-1.5° so that only (or mainly) direct-arriving crustal phases are used. This would also normally be done with a carefully developed local crustal model (command lmod).

Therefore a direct calibration analysis looks very much like an uncalibrated relocation analysis as far as output files go, and there is no need for an extensive supplemental estimation process as there is for indirect calibration. Therefore, although there is a special output file (*~.dcal*) for direct calibration, it is much simpler than the ~.cal file produced in an indirect calibration. The first few lines of a *~.dcal* file are shown as an example:

```
Cumulative uncertainty of direct-calibrated events
iev     alpha       xl        yl       area       eqr      ddep       dot
  1   -77.265    2.748     4.769    41.162     3.620     0.000     0.267
  2   -78.380    3.303     5.400    56.028     4.223     0.000     0.296
  3   -64.416    2.785     3.297    28.850     3.030     0.000     0.256
  4   -76.395    3.728     5.288    61.939     4.440     0.000     0.382
  5    68.269    2.442     3.454    26.497     2.904     0.000     0.310
  6   -19.484    2.706    12.653   107.575     5.852     0.000     0.651
  7     5.891    4.157     4.939    64.507     4.531     0.000     0.346
  8   -68.576    2.483     2.933    22.876     2.698     0.000     0.201
  9   -79.697    5.652     7.650   135.825     6.575     0.000     0.516
 10    80.105    3.866     4.266    51.811     4.061     0.000     0.259
```

After the event number the 90% confidence ellipse for cumulative uncertainty (cluster vector plus hypocentroid) is given by three values, the azimuth of the semi-minor axis, and the lengths of the two semi-axes, the area of the ellipse, the radius of a circle with the same area, and uncertainties in depth and origin time. All these values except equivalent circular radius are also given in the ~.hdf_dcal file.

~.dcal_phase_data File

A *~.dcal_phase_data* file is written whenever the dcal command has been issued to perform a direct calibration analysis. While writing the standard ~.phase_data output file, the *~.dcal_phase_data* file is created by writing to a separate file any reading within the epicentral distance range specified for the estimation of the hypocentroid (command hlim). It is written from the module *mlocout_phase_data.f90*.

The format of the *~.dcal_phase_data* file is very similar to that of the parent *~.phase_data* file but the header section for each event is simplifed to one line with the event number, event name and a note about the focal depth and how it was set:

```
STA   NETWORK    PHASE    READ  DELTA AZIM  RAY     WGT   STA  P RESIDUALS FOR ITERATION #
CODE                        ERR             PARAM         CORR  *INP*   *0*    *1*    *2*    *3*    *4*         DTMPH  DTMPC    ECI AUTHOR   CHA PHASE0     MNF IDIF

CLUSTER EVENT   1          19300506.2234.23                  GOOD DATA      Depth =   8.0, fixed, from depth phases

CLUSTER EVENT   2          19300508.1535.24                  GOOD DATA      Depth =  15.0, fixed, from cluster default depth

CLUSTER EVENT   3          19340222.0807.13                  GOOD DATA      Depth =  15.0, fixed, from cluster default depth

CLUSTER EVENT   4          19401018.1225.44                  GOOD DATA      Depth =  15.0, fixed, from cluster default depth

CLUSTER EVENT   5          19520930.0250.40                  GOOD DATA      Depth =   7.0, fixed, from depth phases

CLUSTER EVENT   6          19581026.1240.30                  GOOD DATA      Depth =  15.0, fixed, from cluster default depth

CLUSTER EVENT   7          19670517.0428.50                  GOOD DATA      Depth =   7.0, fixed, from depth phases

CLUSTER EVENT   8          19700314.0151.43                  GOOD DATA      Depth =  10.0, fixed, from depth phases

CLUSTER EVENT   9          19740308.0848.16                  GOOD DATA      Depth =  15.0, fixed, from cluster default depth

CLUSTER EVENT  10          19740312.0653.50                  GOOD DATA      Depth =  15.0, fixed, from cluster default depth
```

All events are reported, even if they do not have any readings that were used for direct calibration, as in this example. This cluster (in the vicinity of Salmas, Iran) contains an unusual number of older events that are well connected by regional and teleseismic stations to the modern events that carry the data used for direct calibration:

```
CLUSTER EVENT  53          20060526.1459.30               GOOD DATA      Depth =  17.0, fixed, from near-source readings
ISHB            Pg      0.59   0.46  105  18.1   1.00  0.13  -0.1   0.10   0.43   0.30                    0.1078 0.0253   1.52 UTIG     S Z Pg          21    0
ISHB            Pg      0.59   0.46  105  18.1   1.00  0.13   0.4   0.00   0.33   0.20                    0.1078 0.0253   1.35 ISC          Pg          26    0
IMRD            Pg      0.39   0.59   58  18.5   1.00  0.10   0.5  -0.04   0.22   0.23                    0.2817 0.0986   0.68 UTIG     S Z Pg          16    0
ITBZ            Sg      0.94   0.88  101  29.6   1.00  0.18   0.0  -0.23   0.47   0.30                    0.0438 0.0227  -0.52 UTIG     S N Sg          23    0
ITBZ            Pg      0.41   0.88  101  17.1   1.00  0.13  -0.3  -0.29   0.02  -0.09                    0.0713 0.0479   1.04 ISC          Pg          30    0
ITBZ            Pg      0.41   0.88  101  17.1   1.00  0.13   0.1  -0.18   0.12   0.01                    0.0713 0.0479   1.29 UTIG     S Z Pg          22    0
MAKU            Pg      0.70   1.00  343  17.1   1.00  0.10   0.7   0.01  -0.38  -0.25                    0.1174 0.0366   1.12 ISC      B Z Pg          32    0
IAZR            Pg      0.45   1.03  134  17.1   1.00  0.18   0.2   0.05   0.23   0.04                    0.0160 0.0586  -1.00 UTIG     S Z Pg           7    0

CLUSTER EVENT  54          20060609.1801.24               GOOD DATA      Depth =  12.0, fixed, from near-source readings
HAKT            Sg      1.68   0.08  213  19.5   1.00  0.49  -0.9   1.34   1.12   1.02                    0.0295 0.0296  -0.23 ISC          Sg           8    0
HAKT            Pg      0.98   0.08  213  11.3   1.00  0.30  -2.1   0.42   0.24   0.15                    0.0343 0.0212   0.27 ISC          Pg           7    0
CUKT            Pg      0.59   0.40  198  18.4   1.00  0.06  -1.9   1.15   0.96   0.84                    0.0139 0.2987   0.37 ISC          Pg          10    0
VANT            Pg      0.74   0.89  334  19.0   1.00  0.04  -2.6  -0.41  -0.60  -0.60                    0.0199 0.0765  -1.35 ISC          Pg          12    0
TVAN            Sg      0.95   0.95  343  32.9   1.00  0.12  -1.6  -0.44  -0.64  -0.59                    0.0488 0.1523  -0.32 ISC          Sg          14    0

CLUSTER EVENT  55          20060729.0151.11               GOOD DATA      Depth =  19.0, fixed, from local-distance readings
ISHB            Pg      0.59   0.63  102  17.1   1.00  0.18   0.3  -0.10   0.24   0.07                    0.1078 0.0364   1.20 UTIG     S Z Pg          14    0
ISHB            Sg      1.13   0.63  102  29.6   1.00  0.26   0.0  -0.59  -0.09  -0.33                    0.0893 0.0208  -0.68 UTIG     S E Sg          15    0
ISHB            Pg      0.59   0.63  102  17.1   1.00  0.18  -0.5  -0.20   0.14  -0.03                    0.1078 0.0364   1.04 ISC          Pg          20    0
IMRD            Pg      0.39   0.74   66  17.1   1.00  0.17  -0.2  -0.11   0.02  -0.09                    0.2817 0.0880   0.06 UTIG     S Z Pg          12    0
MAKU            Pg      0.70   0.94  353  17.1   1.00  0.10  -1.2   0.09  -0.13  -0.12                    0.1174 0.0400   1.32 IIEES    B Z Pg          23    0
CLDR            Pg      0.41   1.02  316  17.1   1.00  0.16  -2.6   0.14  -0.10  -0.06                    0.0422 0.0711   0.47 ISC          Pg          26    0
ITBZ            Pg      0.41   1.05  100  17.1   1.00  0.13   0.2  -0.18   0.14  -0.03                    0.0713 0.0707   1.20 UTIG     S Z Pg          16    0
ITBZ            Pg      0.41   1.05  100  17.1   1.00  0.13  -1.3  -0.28   0.04  -0.13                    0.0713 0.0707   0.95 ISC          Pg          25    0
VANB            Pg      0.45   1.12  275  17.1   1.00  0.10  -2.6   0.99   0.90   0.91                    0.1329 0.0958   0.10 ISC          Pg          33    0
TVAN            Pg      1.51   1.12  276  17.1   1.00  0.16  -2.3   0.97   0.87   0.88                    0.0587 0.0084   0.84 ISC          Pg          30    0
IAZR            Pg      0.45   1.17  129  17.1   1.00  0.18  -2.2  -0.75  -0.32  -0.52                    0.0160 0.1115  -2.07 ISC      S Z Pg          28    0
IAZR            Pg      0.45   1.17  129  17.1   1.00  0.18  -2.3  -0.85  -0.42  -0.62                    0.0160 0.1115  -2.29 ISC          Pg          29    0
```

The format for individual readings is identical to that used in the ~.phase_data file, but headers are deleted to improve readability.

Like the ~.phase_data file the *~.dcal_phase_data* file is divided into "good" data and "bad" data sections. Here is the "bad" data for the three events shown above:

```
CLUSTER EVENT  53          20060526.1459.30               BAD DATA       Depth =  17.0, from near-source readings
ISHB          d Pg      0.59   0.46  105  18.1   1.00  0.13   0.5   0.10   0.43   0.30                                                    ISC      S Z Pg          25    0
IMRD          d Pg      0.39   0.59   58  18.5   1.00  0.10   1.5  -0.04   0.22   0.23                                                    ISC      S Z Pg          27    0
ITBZ          d Pg      0.41   0.88  101  17.1   1.00  0.13  -0.2  -0.18   0.12   0.01                                                    ISC      S Z Pg          28    0
ITBZ          d Sg      0.94   0.88  101  29.6   1.00  0.18   1.0  -0.33   0.37   0.20                                                    ISC          Sg          31    0
ITBZ          d Sg      0.94   0.88  101  29.6   1.00  0.18   1.1  -0.23   0.47   0.30                                                    ISC      S N Sg          29    0
IAZR          d Pg      0.45   1.03  134  17.1   1.00  0.18  -1.3   0.05   0.23   0.04                                                    ISC      S Z Pg          33    0

CLUSTER EVENT  54          20060609.1801.24               BAD DATA       Depth =  12.0, from near-source readings
HAKT          d Pg      0.98   0.08  213  11.3   1.00  0.30  -2.1   0.36   0.18   0.09                                                    ISC          Pg           9    0
CUKT          x Sg      1.09   0.40  198  31.9   1.00  0.04  -0.1   2.69   2.47   2.32                    ?                               ISC          Sg          11    0
TVAN          x Pg      1.51   0.95  343  19.0   1.00  0.05  -3.9  -1.45  -1.62  -1.63                    ?                               ISC          Pg          13    0

CLUSTER EVENT  55          20060729.0151.11               BAD DATA       Depth =  19.0, from local-distance readings
ISHB          d Pg      0.59   0.63  102  17.1   1.00  0.18  -0.4  -0.10   0.24   0.07                                                    ISC      S Z Pg          18    0
ISHB          d Sg      1.13   0.63  102  29.6   1.00  0.26   0.8  -0.59  -0.09  -0.33                                                    ISC      S E Sg          19    0
ISHB          d Sg      1.13   0.63  102  29.6   1.00  0.26   0.7  -0.69  -0.19  -0.43                                                    ISC          Sg          21    0
IMRD          d Pg      0.39   0.74   66  17.1   1.00  0.17   0.1  -0.11   0.02  -0.09                                                    ISC      S Z Pg          22    0
CLDR          x Sg      0.70   1.02  316  29.6   1.00  0.24  -0.4   1.92   1.43   1.55                    ?                               ISC          Sg          27    0
ITBZ          d Pg      0.41   1.05  100  17.1   1.00  0.13  -1.2  -0.18   0.14  -0.03                                                    ISC      S Z Pg          24    0
VANB          x Sg      0.50   1.12  275  29.6   1.00  0.14  -1.3   3.39   3.16   3.23                                                    ISC          Sg          34    0
TVAN          x Sg      0.95   1.12  276  29.6   1.00  0.23   1.0   4.61   4.37   4.44                                                    ISC          Sg          31    0
```

Most of the entries are duplicate readings (flag = "d"), and there are a few outlier readings, flagged with "x". In several cases, marked with "?" in the "WHY BAD" column, the algorithm has suggested that it may be wise to check a reading (with rstat) that is close enough to the "good" readings for that station-phase that it might deserve to be added back into the inversion.

~.depth_phases File

The *~.depth_phases* file is created on every run of **mloc** to collect information relevant to the constraint of focal depth. This is of greatest interest in the common case where there is insufficient data to perform a free-depth relocation and focal depth is specified by the user (dep_ command) and held fixed during the relocation (commands freh and frec).

It is important to recognize that all the output in this file is advisory in nature. No changes are made automatically to the arrival time dataset or to the inversion. To implement any change to the relocation analysis based on this file, the user must edit individual event files in some way or make changes in the command file for the next run.

The information written to the *~.depth_phases* file is of two types and written from two different modules of the code. The first type of output relates to teleseismic depth phases (pP, sP and pwP)

and their analysis as relative depth phases. This output is driven by the subroutine *rdp_depth_test* in the module *mloclib_tt.f90*. The second part of the output is driven by the module *mlocout_phase_data.f90* and displays data at epicentral distances at which the arrival time of a direct phase provides depth constraint.

Relative Depth Phases

The unusual approach used in **mloc** to analyze teleseismic depth phases is discussed elsewhere, but the data behind that analysis are carried in the first section of the *~.depth_phase file*. First is a summary of the results of the analysis of relative depth phases; only events with one or more (unflagged) relative depth phases are listed. Here's a sample of the first six events in a cluster:

```
  1 19300506.2234.23              preferred depth =   5 km (  1 to  13) on   1 samples (depd   5  8  4 ! on   1 samples)
  5 19520930.0250.40              preferred depth =   5 km (  1 to  14) on   1 samples (depd   5  9  4 ! on   1 samples)
  7 19670517.0428.50              preferred depth =   7 km (  2 to  14) on   2 samples (depd   7  7  5 ! on   2 samples)
  8 19700314.0151.43              preferred depth =  10 km (  4 to  16) on   5 samples (depd  10  6  6 ! on   5 samples)
 11 19761124.1222.15              preferred depth =  31 km ( 17 to  38) on   2 samples (depd  31  7 14 ! on   2 samples)
 12 19761124.1511.02              preferred depth =   8 km (  4 to  18) on   1 samples (depd   8 10  4 ! on   1 samples)
```

The part in parentheses after the number of samples can be copied and pasted directly into a command file to implement the depd command, including the part after the "!" symbol, which the command parser will treat as a comment. After the summary section is a detailed listing of the relative depth phase data for each event, shown here for the same six events:

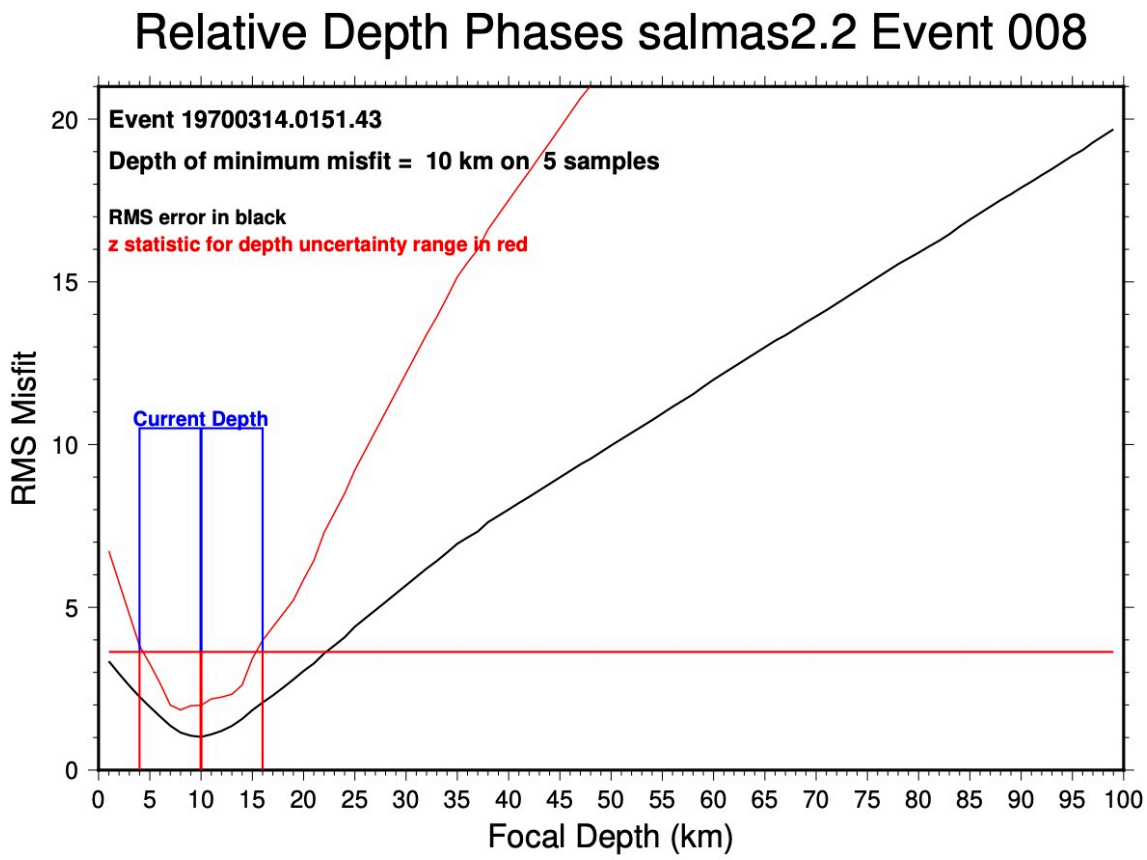
```
Residuals against each theoretical phase for the depth with minimum misfit
                MNF      pP-P      sP-P     pwP-P     Err**2
Event   1 19300506.2234.23              5 km
  VIE   22.94 pP-P       95      1.04      0.16      1.04      0.02 p sP-P
  TYK   69.32 pP-P      338      0.78     -0.04      0.78      0.00   sP-P
Event   5 19520930.0250.40              5 km
  TAM   36.40 pP-P       98      0.78     -0.12      0.78      0.01   sP-P
Event   7 19670517.0428.50              7 km
  NIE   20.20 pP-P       42      1.40      0.22      1.40      0.05 p sP-P
  NUR   25.13 pP-P       57     10.22      8.99     10.22     80.80 p sP-P
  FUR   25.58 pP-P       63      4.22      3.08      4.22      9.49 p sP-P
  FUR   25.58 sP-P       64     11.22     10.08     11.22    101.60 p
  KLS   25.85 sP-P       66     10.42      9.28     10.42     86.06 p
  KJN   27.34 pP-P       73      0.91     -0.32      0.91      0.10   sP-P
  UDD   28.86 pP-P       79      0.20     -1.03      0.20      0.04
  TAM   36.48 pP-P       92     74.18     73.05     74.18   5336.23 x sP-P
Event   8 19700314.0151.43             10 km
  NIE   20.59 pP-P       71      0.78     -0.56      0.78      0.31 p sP-P
  NIE   20.59 sP-P       72      5.78      4.44      5.78     19.74 p
  NUR   25.35 pP-P      102     -1.53     -2.95     -1.53      2.34 p
  FUR   26.00 sP-P      112      3.34      2.07      3.34      4.27 p
  MOX   26.29 pP-P      119     -1.17     -2.44     -1.17      1.37
  KJN   27.50 pP-P      125      0.45     -0.96      0.45      0.20
  OUL   28.74 pP-P      136      1.24     -0.17      1.24      0.03   sP-P
  BNS   29.11 sP-P      140     -2.69     -3.95     -2.69      7.23 x pP-P
  TRO   34.05 sP-P      167      3.62      2.21      3.62      4.89 x
  BAB   39.34 pP-P      181      2.77      1.51      2.77      2.29   sP-P
  BAB   39.34 sP-P      182     12.77     11.51     12.77    132.59 x
  AVE   42.13 pP-P      188     15.25     14.00     15.25    195.93 x sP-P
  LAO   91.24 pP-P      261     -1.14     -2.43     -1.14      1.31
  LAO   91.24 sP-P      262      5.86      4.57      5.86     20.84 x
Event  11 19761124.1222.15             31 km
  BRA   21.43 pP-P      241     -5.95    -10.21     -5.95     35.37 p
  BRA   21.43 sP-P      242     -0.95     -5.21     -0.95      0.90 p pP-P
  NKM   39.12 pP-P      499     -0.26     -4.36     -0.26      0.07
  NKM   39.12 sP-P      500      6.24      2.14      6.24      4.57 x
  KBS   42.02 pP-P      540      0.45     -3.54      0.45      0.20
Event  12 19761124.1511.02              8 km
  PSZ   19.68 sP-P       58      6.53      5.26      6.53     27.69 p
  NIE   19.94 pP-P       62     10.83      9.56     10.83     91.43 p sP-P
  BUD   20.12 sP-P       65     15.02     13.76     15.02    189.29 p
  KRA   20.42 sP-P       70      9.03      7.76      9.03     60.18 p
  NUR   24.78 sP-P      101      5.76      4.51      5.76     20.33 p
  CLL   24.99 sP-P      103     15.89     14.61     15.89    213.31 p
  MOX   25.65 sP-P      111      3.89      2.60      3.89      6.77 p
  GRF   25.67 pP-P      113      6.48      5.20      6.48     27.04 p sP-P
  KJF   27.07 pP-P      118      1.26      0.01      1.26      0.00   sP-P
```

The pwP phase is always listed but if the bouncepoint is on land it will be the same as pP. The phase name sometimes appears twice, first as the phase used in **mloc**, and second as an alternative phase that would fit better at the preferred depth. In general, specifying the correct depth is extremely difficult during routine monitoring, especially for shallow events, so I place very little confidence in the reported phase name for depth phases. To change the phase name in

future runs of **mloc** you would have to edit the event file by hand, but changing the phase name would not have any effect on the depth phase analysis, since it considers all possible depth phase associations as equally likely. The squared error column refers to whatever phase has the smallest residual at the preferred depth. Positive residuals indicate the event should be deeper.

Depth phases are often reported from stations at far-regional distances and distances where upper mantle triplications create tremendous confusion for phase identification. Following the advice of E.R. Engdahl, as employed in the EHB algorithm, depth phases reported at distances less than 26° are automatically flagged ("p") and not considered for constraining the focal depth. The user is free to take these readings into account in making decisions about the remaining depth phase readings. The flag "p" is printed after the column with squared error.

The analysis of depth phases involves a process of cleaning, i.e., identifying and flagging readings that are judged to be outliers. For depth phases this is done manually and in a rather *ad hoc* manner, but the number of samples is usually too small to make any statistical testing worthwhile. We are looking for a depth at which the largest possible number of readings agree (i.e., have small residuals) to within some poorly-known time. As a rule of thumb, I have found that when there are a considerable number of depth phase reports that give a strong minimum, those readings all generally have squared error values less than about 2.0. You can be rather conservative, keeping readings with squared errors of 3-4, and it will usually not effect the estimate of preferred depth very much. Of course it will expand the range of uncertainty a bit. As with the cleaning of standard arrival time data, an incremental approach is recommended, because deleting a single reading with large residual is likely to change the preferred depth and the pattern of the remaining residuals significantly on the next run. It is very helpful and instructive to refer to the plots associated with relative depth phases during the cleaning process, these are requested with the rdpp command. Here is an example (from event 8 in the list above):



The red line is a statistic that is used to determine the preferred depth and uncertainty (the black one is just the squared error sum, a much weaker guide). The details of the statistical analysis are described here.

IMPORTANT During the normal cleaning process (using the automated utility codes lres and xdat) depth phases may be flagged because of their residuals when what is actually needed is a better estimate of focal depth. Therefore I recommend doing the analysis of relative depth phases early in the relocation analysis, before much cleaning has been done. The depth phase analysis is insensitive to location accuracy, but location accuracy can depend a lot on reliable depth constraint. Once a preferred depth has been determined it can be implemented in the command file and after that you don't need to worry if some of the readings get flagged inadvertently.

Direct Phase Readings

The format of this section is nearly a duplicate of the ~.dcal_phase_data file, but the epicentral distance selection criterion for which readings are included is different. The threshold for epicentral distance is the larger of 100 km or three times the focal depth. In many cases (i.e., when the hypocentral distance limit for direct calibration is set especially close with command

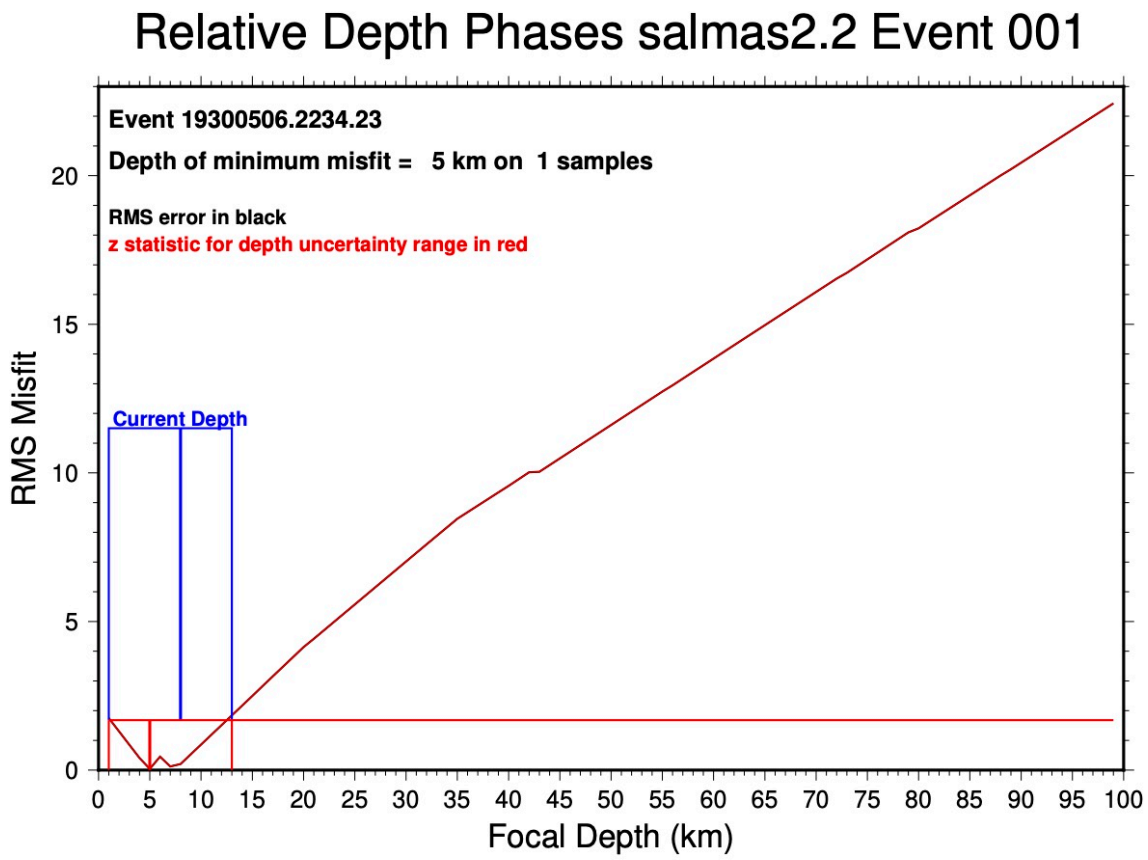
hlim) this will include some readings at greater distance than the *~.dcal_phase_data* file. This is done because those readings are often still useful for helping constrain focal depth, using the local distance formulation (command depl). This example is for the same set of events as the example in the section on the ~.dcal_phase_data file, none of which have any direct arrivals within the epicentral distance range for this listing:

```
Cluster default depth:   15.0
Median of constrained depths:   14.0
Median of input file depths:   10.0
 STA   NETWORK    PHASE    READ  DELTA AZIM  RAY     WGT   STA  P RESIDUALS FOR ITERATION #
 CODE                       ERR             PARAM         CORR  *INP*   *0*    *1*    *2*    *3*    *4*          DTMPH  DTMPC    ECI AUTHOR   CHA PHASE0     MNF IDIF

 CLUSTER EVENT   1          19300506.2234.23                GOOD DATA      Depth =   8.0, fixed, from depth phases
 TYK             pP-P     1.00  69.32   61   6.2   1.00  0.00  -2.2  -0.19  -0.20  -0.20                      0.0000 0.0000   0.00 ISC          pP         338    0

 CLUSTER EVENT   2          19300508.1535.24                GOOD DATA      Depth =  15.0, fixed, from cluster default depth

 CLUSTER EVENT   3          19340222.0807.13                GOOD DATA      Depth =  15.0, fixed, from cluster default depth

 CLUSTER EVENT   4          19401018.1225.44                GOOD DATA      Depth =  15.0, fixed, from cluster default depth

 CLUSTER EVENT   5          19520930.0250.40                GOOD DATA      Depth =   7.0, fixed, from depth phases
 TAM             pP-P     1.00  36.40  255   8.5   1.00  0.00  -3.0   0.15   0.16   0.16                      0.0000 0.0000   0.00 ISC          pP          98    0

 CLUSTER EVENT   6          19581026.1240.30                GOOD DATA      Depth =  15.0, fixed, from cluster default depth

 CLUSTER EVENT   7          19670517.0428.50                GOOD DATA      Depth =   7.0, fixed, from depth phases
 KJN             pP-P     1.00  27.34  344   9.0   1.00  0.00   0.0   0.90   0.90   0.91                      0.0000 0.0053   0.22 ISC          pP          73    0
 UDD             pP-P     1.00  28.86  328   8.9   1.00  0.00   0.0   0.20   0.19   0.20                      0.0000 0.0000   0.00 ISC          pP          79    0

 CLUSTER EVENT   8          19700314.0151.43                GOOD DATA      Depth =  10.0, fixed, from depth phases
 BUC             pP       1.00  15.16  299  11.1   1.00  0.01   0.0  -2.79  -2.92  -2.98                      0.0000 0.0000   0.00 ISC          sPn         44    0
 BAC             pP       1.00  15.38  307  11.1   1.00  0.02   0.0  -1.22  -1.42  -1.41                      0.0000 0.0000   0.00 ISC          pPn         48    0
 MOX             pP-P     1.00  26.29  308   9.0   1.00  0.00   0.0  -1.17  -1.16  -1.17                      0.0000 0.0000   0.00 ISC          pP         119    0
 KJN             pP-P     1.00  27.50  344   9.0   1.00  0.00   0.0   0.46   0.42   0.45                      0.0000 0.0053  -0.22 ISC          pP         125    0
 OUL             pP-P     1.00  28.74  343   8.9   1.00  0.00   0.0   1.25   1.21   1.24                      0.0000 0.0000   0.00 ISC          pP         136    0
 BAB             pP-P     1.00  39.34  272   8.4   1.00  0.00  -7.1   2.77   2.78   2.77                      0.0000 0.0000   0.00 ISC          pP         181    0
 LAO             pP-P     1.00  91.24  341   4.6   1.00  0.00   3.4  -1.14  -1.05  -1.14                      0.0000 0.0000   0.00 ISC          pP         261    0

 CLUSTER EVENT   9          19740308.0848.16                GOOD DATA      Depth =  15.0, fixed, from cluster default depth

 CLUSTER EVENT  10          19740312.0653.50                GOOD DATA      Depth =  15.0, fixed, from cluster default depth
```

The first lines give information on the statistics of focal depths in the cluster and the current default depth (command depc) that is applied to any event with no other form of constraint. It would be natural to set the cluster default depth close to the median of constrained depths.

In contrast to the ~.dcal_phase_data file, this listing includes any available depth phases, even though they are beyond the hypocentroidal distance limit that is used to select direct arrivals. In some cases the event files list depth phases that for some reason cannot be turned into relative depth phases, like stations BUC and BAC in event 8, so they are listed simply as the depth phase. They won't contribute to the analysis for focal depth.

The residuals for the relative depth phases seen in this "all available depth-related phases" listing may differ from the ones shown in the earlier "relative depth phase preferred depth" section of this file because the focal depth used in the relocation may not have been the preferred depth shown above. For example the residual for TYK(pP-P) in event 1 listed immediately above is different from the instance listed in the "preferred depth section" because the relocation was run with the depth of this event at 8 km rather than the 5 km preferred depth (but in that case the preferred phase identification would be sP-P). Reference to the "rdpp" plot for this event shows that both depths fit the (limited) data well:



~.hdf Files

Each run of **mloc** creates one or more output files that summarize the relocated hypocenters of all events in the cluster in a "one line per event" format. The reason for multiple instances of the file is explained below. These files are known generically as HDF files and they all contain *hdf* as part of the file name suffix. The three possible filename extensions are:

- *~.hdf*
- *~.hdf_dcal*
- *~.hdf_cal*

Different Flavors of HDF

The need for several variations of the HDF file in **mloc** arises from the different scenarios that are possible:

- No calibration. Only a *~.hdf* is created. Uncertainties are for relative location only (cluster vectors).

- Direct calibration. Only a *~.hdf_dcal* is created. Uncertainties are for absolute location (cluster vector plus hypocentroid).
- Indirect calibration. Two HDF files are created, a *~.hdf* file (with uncertainties for relative location) and a *~.hdf_cal* file (with uncertainties for absolute location).
- Direct calibration, followed by indirect calibration in the same run. Two files are created, a *~.hdf_dcal* file and a *~.hdf_cal* file. Indirect calibration takes precedence.

All HDF files, regardless of flavor, may be read with the same code. The interpretation of fields dealing with uncertainty of a parameter vary, depending on the flavor.

Format Description

HDF File Format

Columns	Description
1:4	Origin year (i4)
6:7	Origin month (i2)
9:10	Origin day (i2)
12:13	Origin hour (i2)
15:16	Origin minute (i2)
18:22	Origin seconds (f5.2)
24:32	Geographic latitude (f9.5)
34:43	Geographic longitude (f10.5)
45:50	Focal depth, km (f6.2)
52:52	How starting depth was set (a1). See depth codes.
53:53	Free depth flag (a1) (Note)
54:59	Depth from input file (f6.2) (Note)
61:63	Magnitude (f3.1) (Note)
64:65	Magnitude scale (a2) (Note)
67:76	Event ID (a10) (Note)
78:81	Number of observations contributed to hypocentroid estimation (i4) (Note)
83:86	Number of observations used for cluster vector (i4) (Note)
88:91	Number of observations flagged as outliers, fcode = ‘x’ (i4) (Note)

93:98	Normalized sample variance for the cluster vector (f6.2) (Note)
100:104	Uncertainty in origin time, sec (f5.2) (Note)
106:109	+ uncertainty in depth (deeper), in km (f4.1) (Note)
111:114	– uncertainty in depth (shallower), in km (f4.1) (Note)
116:120	Epicentral distance of nearest station for cluster vector (f5.1) (Note)
122:126	Epicentral distance of farthest station for cluster vector (f5.1) (Note)
128:132	Largest open azimuth for cluster vector (f5.1) (Note)
134:136	Semi-axis azimuth (i3) (Note)
138:142	Semi-axis length, km (f5.2) (Note)
144:146	Semi-axis azimuth (i3) (Note)
148:152	Semi-axis length, km (f5.2) (Note)
154:159	Area of confidence ellipse, km2 (f6.1) (Note)
161:164	Calibration code (a4)
166:185	Annotation (a20) (Note)

Free Depth Flag

The column immediately after the focal depth constraint flag is used for a special flag “f” which indicates that the focal depth has been a free parameter in the relocation.

Depth from Input File

This variable holds whatever focal depth was specified for an event in the input data file. Depending on what other commands are issued in the command file or interactively this value may or may not be used as the starting depth for relocation. It is sometimes useful to have this for comparison with the depth actually set for (or determined in) relocation. Large discrepancies, more than, say, 10 or 15 km, may warrant investigation.

Magnitude and Magnitude Scale

A single representative (or preferred) magnitude is carried through the **mloc** analysis to assist in interpretation. It has no bearing on the relocation. MNF input files can carry multiple estimates of magnitude, but one will be specified (or selected by default) as the preferred magnitude. If no

magnitude estimate is available, these two fields will be blank. In some cases a magnitude may be available but not a scale, in which case the magnitude scale field will be blank.

Event ID

This field is provided to carry an EVID that was assigned by some relational database, but as far as **mloc** is concerned it is simply a character variable of 10 characters. If the field has an entry it will have been read from an MNF input file. If an integer EVID is stored in the field it should be right-justified.

Number of Observations (phase readings)

The HDF format carries three different counts of number of readings:

- Number of observations contributed to hypocentroid estimation
- Number of observations used for the cluster vector
- Number of observations flagged as outliers

It is wise to review carefully the reliability of the relative location (cluster vector) for an event that contributes many readings to the hypocentroid in a direct calibration study. Conversely, events with relatively small number of readings contributing to the cluster vector should be reviewed carefully, because the relative location of such events is likely to be poorly determined.

The third variable in this set carries the number of flagged readings that are due to being judged to have a large residual (so-called "cluster residuals" in which the residuals are compared on the basis of mutual consistency rather than absolute value). These are the readings for which fcode = 'x'. It is not uncommon for the number of outliers to be a significant fraction of the number of readings used for the cluster vector, but when the fraction becomes notably large (say, more than 25%) investigation is warranted.

There are other reasons why a reading may be flagged to prevent its being used in the relocation but these are not tabulated in the hdf file. Examples include duplicate readings, unknown phase types or phase types that are not used for relocation, and stations for which no coordinates are available.

Normalized Sample Variance

This field carries a measure of the statistical self-consistency of the error budget related to the cluster vector, the normalized sample variance. If the data followed our statistical model perfectly (i.e., data drawn randomly from a normal distribution with spread equal to our estimated empirical reading error) the expected value would be 1.0. Values both larger and smaller are actually observed, naturally, and the range of variation declines as the analysis proceeds and outlier readings are identified and removed by flagging.

A Bayesian term in the calculation of the normalized cluster sample variance represents our *a priori* state of knowledge about its expected variability and prevents it from going unrealistically

small for events with few data points. This expectation on the spread of values for the normalized sample variance (about 0.35) provides a check on the internal consistency of the statistical model for each event. Values larger than about 2.0 (~3σ) may reveal the presence of readings that violate the statistical model (e.g., outliers). By the end of a relocation analysis there should be few such cases.

Uncertainty of Origin Time

In seconds. For uncalibrated clusters and direct calibration, it is taken from the covariance matrix of the relocation, including uncertainty of hypocentroid and cluster vector. For indirect calibration it is based on the uncertainty of the calibration shift plus the uncertainty of relative origin time from the cluster vector.

Uncertainty in Focal Depth

Uncertainty in focal depth can be read from the input file, from the command file, or from the relocation. Uncertainty in focal depth is carried in two fields because it is not uncommon for the uncertainty to be asymmetric. This can arise when inferring depths from teleseismic depth phases and there is uncertainty about the correct identification of the phase (pP, sP, or pwP). It can also arise in waveform analyses where the error vs depth curve is not symmetric. If focal depth has been a free parameter in the relocation it will be symmetric, and it will over-ride any specification in the input file and command file. For uncalibrated clusters and direct calibration with a free depth solution, it is taken from the covariance matrix of the relocation, including uncertainty of hypocentroid and cluster vector. For indirect calibration it is based on the uncertainty of the calibration shift plus the uncertainty of relative depth from the cluster vector. If no estimate of uncertainty in focal depth is available, the fields are blank.

Epicentral Distance Range

The epicentral distance, in degrees, of the nearest and farthest reading used for the cluster vector of an event is carried in these fields. This is useful for judging how an event is connected to the cluster, i.e., through local readings or teleseismic readings, or both.

Open Azimuth

The largest open azimuth for the readings used to estimate the cluster vector is carried in this field. The reliability of the relative location of an event should be questioned if this value is much greater than 180°.

Confidence Ellipse

The confidence ellipse (90% confidence level) for the epicenter is defined in four columns which give the azimuth and length of each semi-axis (half-length). The shorter semi-axis is given first. Azimuth is in integer degrees, clockwise from North. Semi-axis lengths are given in decimal km.

A fifth column carries the area of the 90% confidence ellipse, in km^2. This is a convenient metric to monitor when searching for events with problems. A circle of 5 km radius (the canonical GT5 location) has an area slightly greater than 75 km^2.

As discussed above, the interpretation of the confidence ellipse (relative vs absolute location) depends on whether the cluster has been calibrated or not, which is indicated by the file name suffix.

Annotation

It is possible to declare a comment or annotation for an event using the anno command. An annotation can also be read from an MNF input file which will take precedence over an annotation given in the command file. Any annotation will be appended to the end of the line in all HDF files. The comment is limited to 20 characters.

Example

The Ahar, Iran cluster was calibrated using direct calibration followed by indirect calibration so the run produced both a ~.hdf_dcal and a ~.hdf_cal file. The shift from "direct" to "indirect" results was small (0.9 km at 20° for the epicenters) and uncertainties are quite similar in both results, providing additional confidence in the results. Here are the first dozen lines from both HDF files:

ahar12.17.hdf_dcal

```
2012  8 11 12 23 14.07  38.39981   46.83729  12.60 m  12.60 6.2mb          8 1988  401   0.71  0.12  1.2  1.2   0.2 164.7  12.2 275  0.70   5  1.75    3.8 CH02
2012  8 11 12 30 11.39  38.44147   46.75668  17.30 m  17.30 4.5            8   75   28   1.35  0.16  1.3  1.3   0.1  78.3  57.3 275  0.99   5  2.12    6.6 CH02
2012  8 11 12 34 32.89  38.44719   46.76897  19.00 m  19.00 6.1mb          2 1884  192   0.56  0.11  1.2  1.2   0.1 133.5  12.8 274  0.69   4  1.74    3.8 CH02 Iran-Armenia-Azerbai
2012  8 11 12 49 14.32  38.41455   46.68159  15.90 m  15.90 4.8mb          9  197   76   1.21  0.14  1.2  1.2   0.1  92.5  29.2 275  0.85   5  1.89    5.1 CH02 Iran-Armenia-Azerbai
2012  8 11 13  5 53.12  38.43239   46.70070  16.00 m  16.00 4.3mb          5  100   64   1.00  0.14  1.3  1.3   0.3  76.6  47.9 276  0.89   6  1.88    5.3 CH02
2012  8 11 13 14  4.36  38.43790   46.68469  17.00 n  15.10 4.7mb          5  305   75   0.74  0.12  3.0  3.0   0.1  94.8  22.4 277  0.80   7  1.83    4.6 CH02
2012  8 11 13 42 10.33  38.42986   46.68102  16.90 m  16.90 4.0mb          5  107   65   1.04  0.14  1.3  1.3   0.3  76.6  22.5 274  0.88   4  1.88    5.2 CH02
2012  8 11 13 54 20.69  38.43172   46.80892  17.00 n  17.80 3.4mb          8   36   17   0.84  0.14  3.0  3.0   0.2  27.2  74.2 273  0.88   3  1.68    4.7 CH02
2012  8 11 14 10  2.64  38.45308   46.66595  19.00 n  18.80 3.6mb          8   29   29   0.86  0.14  3.0  3.0   0.3  26.3  77.3 278  0.88   8  2.04    5.6 CH02
2012  8 11 14 16 47.52  38.44926   46.70688  20.00 n  17.90 3.6            8   43   21   1.04  0.15  3.0  3.0   0.3  27.3  74.9 275  0.94   5  2.23    6.6 CH02
2012  8 11 14 25 14.24  38.43574   46.68371  15.00 n  15.40 4.5mb          4  202   67   0.80  0.12  3.0  3.0   0.3  85.7  22.5 277  0.82   7  1.87    4.8 CH02
2012  8 11 14 33 52.34  38.45821   46.80797  17.20 m  17.20 3.8            7   42   13   0.46  0.12  1.2  1.2   0.2  14.4  71.6 276  0.76   6  1.91    4.6 CH02
```

ahar12.17.hdf_cal

```
2012  8 11 12 23 14.05  38.40744   46.84089  11.60 m  12.60 6.2mb          8 1988  401   0.71  0.20  1.2  1.2   0.2 164.7  12.2 271  0.94   1  1.98    5.9 CH02
2012  8 11 12 30 11.37  38.44911   46.76028  16.30 m  17.30 4.5            8   75   28   1.35  0.23  1.3  1.3   0.1  78.3  57.3 272  1.18   2  2.31    8.6 CH02
2012  8 11 12 34 32.88  38.45483   46.77257  18.00 m  19.00 6.1mb          2 1884  192   0.56  0.20  1.2  1.2   0.1 133.5  12.8 271  0.94   1  1.97    5.8 CH02 Iran-Armenia-Azerbai
2012  8 11 12 49 14.31  38.42219   46.68519  14.90 m  15.90 4.8mb          9  197   76   1.21  0.22  1.2  1.2   0.1  92.5  29.2 272  1.06   2  2.11    7.0 CH02 Iran-Armenia-Azerbai
2012  8 11 13  5 53.10  38.44003   46.70430  15.00 m  16.00 4.3mb          5  100   64   1.00  0.21  1.3  1.3   0.3  76.6  47.9 272  1.10   2  2.09    7.2 CH02
2012  8 11 13 14  4.35  38.44554   46.68829  16.00 n  15.10 4.7mb          5  305   75   0.74  0.20  3.0  3.0   0.1  94.8  22.4 273  1.03   3  2.04    6.6 CH02
2012  8 11 13 42 10.31  38.43749   46.68462  15.90 m  16.90 4.0mb          5  107   65   1.04  0.22  1.3  1.3   0.3  76.6  22.5 271  1.08   1  2.09    7.1 CH02
2012  8 11 13 54 20.67  38.43935   46.81252  16.00 n  17.80 3.4mb          8   36   17   0.84  0.22  3.0  3.0   0.2  27.2  74.2  89  1.08 179  1.92    6.5 CH02
2012  8 11 14 10  2.63  38.46071   46.66955  18.00 n  18.80 3.6mb          8   29   29   0.86  0.22  3.0  3.0   0.3  26.3  77.3 274  1.09   4  2.24    7.7 CH02
2012  8 11 14 16 47.50  38.45690   46.71048  19.00 n  17.90 3.6            8   43   21   1.04  0.23  3.0  3.0   0.3  27.3  74.9 272  1.13   2  2.41    8.6 CH02
2012  8 11 14 25 14.22  38.44337   46.68731  14.00 n  15.40 4.5mb          4  202   67   0.80  0.21  3.0  3.0   0.3  85.7  22.5 273  1.04   3  2.09    6.8 CH02
2012  8 11 14 33 52.33  38.46585   46.81157  16.20 m  17.20 3.8            7   42   13   0.46  0.21  1.2  1.2   0.2  14.4  71.6 272  0.99   2  2.12    6.6 CH02
```

The input_depths shown here are uncharacteristically close to the final depths, because the event files for this cluster used an earlier calibrated relocation run of **mloc** for the preferred hypocenters.

No event IDs were used in this cluster. The annotations shown were taken from the geographic description provided with the ISC data, but they were truncated by the 20-character limit in the HDF format.

~.lres File

This page deals with the ~.*lres* file that is read as input to the lres utility code, one of several utility codes that assist in the cleaning process of an **mloc** relocation analysis. **mloc** only creates this file if the command lres has been issued.

In addition to providing input to the **lres** utility code a ~.*lres* file is very useful as a guide in the use of the rstat utility program when an especially careful approach is taken to the cleaning process.

The format of the ~.lres file is documented in the code module *mlocout_phase_data.f90*. Here is an example:

```
5.00
  6 IAS                ISC      P          27     6.45 19581026.1240.30.mnf  Pn           0
 36 TIC                ISC      P         191     5.09 19980807.0136.15.mnf  P            0
 41 RDF                ISC      Sn         79     5.31 20020228.0046.31.mnf  Sn           0
 43 PYA                ISC      Pg         62    -8.01 20030811.2012.06.mnf  Pg           0
 48 WTSB               ISC      P         523     5.20 20050125.1644.12.mnf  P            0
 48 HGN                ISC      P         527     8.67 20050125.1644.12.mnf  P            0
 48 PBEJ               ISC      P         675     5.52 20050125.1644.12.mnf  P            0
 49 KARS               ISC      Pn         30    -5.53 20050125.1752.12.mnf  Pn           0
 53 ERZM               ISC      Pn         72     5.09 20060526.1459.30.mnf  Pn           0
 55 ERZM               ISC      Pn         63     5.03 20060729.0151.11.mnf  Pn           0
 62 IKOM               ISC      Pn        257    -5.51 20101106.0105.16.mnf  Pn           0
 66 PLD                ISC      P         734    -8.15 20111023.1041.23.mnf  P            0
 66 KURBB              ISC      PcP      1449    13.06 20111023.1041.23.mnf  PcP          0
 66 BER                ISC      P        1679    -7.55 20111023.1041.23.mnf  P            0
 66 PVRL               ISC      P        1983     5.58 20111023.1041.23.mnf  P            0
```

The first line contains only the threshold value of the cluster residual (eci) provided as input to the lres command. In the following lines the value of the cluster residual for each reading is listed in the field just before the event file name.

The phase name is given twice in this format, first as the original phase name read from the event file, second as the re-identified phase name used by **mloc**. If you are using a ~.*lres* file as a guide for manual investigation/flagging of problematic station-phases, you would use the latter phase name as input to rstat.

The last column, which is always zero in this example, carries the line number of the reading in a differential time data file (in MNF v1.5 format). These readings must be flagged by hand. Because of the way **mloc** handles differential time data there will be two entries for each differential time reading, one with positive cluster residual, the other having the same absolute value but negative.

~.phase_data File

Every run of **mloc** produces a ~.*phase_data* file that carries information about every reading in all event files, tracking the travel-time residual through each iteration, and displaying the data importance and cluster residual (see the discussion of the utility program rstat) of the reading, as well as some other information that is useful for evaluating the results of a run. The file is written by the module *mlocout_phase_data.f90*.

The GOOD Data Section

The *~.phase_data* file is divided into two parts. The first part contains the "good" data for each event. These are readings that were not flagged for any reason, and that have phase names for which **mloc** could determine a theoretical arrival time to which the observed arrival was close enough to avoid being dropped by the windowing criterion (command wind). This does not mean the reading actually contributed to the relocation however. That status is conveyed in two columns, labelled "DTMPH" and "DTMPC", giving the data importance of the reading for the hypocentroid and cluster vector, respectively. The concept of data importance is discussed in Jordan and Sverdrup (1981). Reasons why a reading might not contribute are:

- It is the only instance of that station-phase, so it cannot contribute to the cluster vector. It might still contribute to the hypocentroid, depending on how the relocation has been set up (command hlim).
- The reading has an epicentral distance that falls outside the limits set with commands hlim or clim.

It is not unusual for an event to have no readings that contribute to the hypocentroid, especially in direct calibration, but every event must have an adequate number of azimuthally-distributed readings in common with other events to establish connectivity in the cluster.

Here are the first few lines from the good data section in a *~.phase_data* file:

```
**********************************************************************************************************************************************************************
CLUSTER EVENT   1          19300506.2234.23                 GOOD DATA
salmas2/19300506.2234.23.mnf
Input 1930  5  6 22 34 26.8  38.094   44.797  15.0 7.2
Final 1930  5  6 22 34 20.2  38.011   44.682   8.0 7.2

STA   NETWORK    PHASE    READ  DELTA AZIM  RAY     WGT   STA  P RESIDUALS FOR ITERATION #
CODE                        ERR              PARAM         CORR  *INP*   *0*    *1*    *2*    *3*    *4*         DTMPH  DTMPC    ECI AUTHOR   CHA PHASE0     MNF IDIF

BAK             Pn       2.30   4.64   58  13.9   1.00  0.01  -0.5  -0.60  -0.75  -0.81                    0.0000 0.0091  -1.34 ISC          Pn          46    0
KSA             Pn       1.27   8.26  242  13.9   1.00  0.11  -0.8   1.77   1.63   1.49                    0.0000 0.0327   0.16 ISC          Pn          47    0
FEO             Pn       1.49   9.87  318  13.9   1.00  0.00  -1.1  -0.62  -0.73  -0.76                    0.0000 0.0023  -0.67 ISC          Pn          48    0
SEV             Pn       4.03  10.55  312  13.9   1.00  0.00   1.6   2.04   1.93   1.89                    0.0000 0.0002   0.56 ISC          Pn          52    0
HLW             Pn       1.15  13.73  238  13.1   1.00  0.02  -1.5   1.58   1.43   1.29                    0.0000 0.0512   1.06 ISC          Pn          53    0
SAM             P        0.98  17.46   78  11.1   1.00  0.10  -4.4  -1.04  -1.20  -1.29                    0.0000 0.0307  -1.21 ISC          Pn          57    0
SAM             S        1.68  17.46   78  20.4   1.00  0.16  -1.7   1.38   1.20   1.12                    0.0000 0.0142   0.99 ISC          S           58    0
```

The first line carries the event number and event name (command even). The second line gives the file name of the event file (command inpu). The event name and input file name are usually coherent but it is not required. The next two lines list the hypocenter and magnitude of the event read from the event file and at the conclusion of the relocation.

The first half-dozen or so columns of the format for individual readings are easily understood. Ray parameter is given in sec/deg. "WGT" is based on the windowing criterion (command wind). Station correction is the elevation correction, in s. The label "P RESIDUALS" is a mistake that has somehow been carried through 30+ years of code development. I can only conclude that I am fond of it. These columns apply to any phase, not just "P". The first one "INP" is taken from the event file, but in many cases that information is not provided and the column will be blank. The "0" column is the residual with the starting location for **mloc**. In most runs it will be the hypocenter read from a *~.hdf file* of the last run (see command rhdf). The other residual columns track the changes through the iterations of the relocation. In the example **mloc** converged after 2 iterations.

The “DTMPH” and “DTMPC” columns are discussed above. “ECI” is the cluster residual, discussed in the section on the utility program rstat. Before *rstat* was written I evaluated outliers by scanning through the entire *~.phase_data* file. The “AUTHOR” field is the code for the source of the reading. “CHA” is instrumental channel, sometimes supplied in the event file and possibly relevant to evaluating differences between readings. “PHASE0” is the phase name read from the event file; it may be changed by **mloc’s** phase re-identification algorithm. “MNF” is the line number of the reading in the event file, which helps with manual editing and is also picked up and displayed by rstat. “IDIF” is a line number reference for a reading that is part of a differential time datum.

The BAD Data Section

The second part of the *~.phase_data* file carries the “bad” readings for each event. The format is similar to the one used for the “good data”. Here is an example, for the same event used above:

```
**********************************************************************************************************************************************************
CLUSTER EVENT   1          19300506.2234.23                BAD DATA
salmas2/19300506.2234.23.mnf
Input 1930  5  6 22 34 26.8  38.094   44.797  15.0 7.2
Final 1930  5  6 22 34 20.2  38.011   44.682   8.0 7.2

STA   NETWORK    PHASE    READ  DELTA AZIM  RAY     WGT   STA  P RESIDUALS FOR ITERATION #                WHY
CODE                        ERR             PARAM        CORR *INP*    *0*    *1*    *2*    *3*    *4*       BAD                AUTHOR   CHA PHASE0     MNF IDIF

FEO           x Sn       1.60   9.87  318  24.4   1.00  0.01   7.5   6.80   6.73   6.75                                        ISC          Sn          49    0
YAL           x Sn       0.22  10.23  313  24.4   0.00  0.00  18.5  18.17  18.10  18.11                                        ISC          Sn          51    0
YAL           x Pn       0.74  10.23  313  13.9   0.00  0.00   8.9   9.46   9.36   9.32                                        ISC          Pn          50    0
HLW           d Pn       1.15  13.73  238  13.1   1.00  0.02  -1.5   1.58   1.43   1.29                                        ISC          Pn          55    0
HLW           p UNKNOWNS 1.60  13.73  238   0.0   0.00  0.00  -6.2 371.96 371.88 371.78                                        ISC          S           56    0
HLW           p UNKNOWNS 1.60  13.73  238   0.0   0.00  0.00  -6.2 371.96 371.88 371.78                                        ISC          S           54    0
KUC           p UNKNOWNS 1.60  18.31  348   0.0   0.00  0.00  -7.6 455.31 454.88 454.78                                        ISC          Sn          62    0
KUC           p UNKNOWNS 1.60  18.31  348   0.0   0.00  0.00  -7.6 455.31 454.88 454.78                                        ISC          Sn          60    0
```

When **mloc’s** phase identification algorithm fails to find any plausible phase for a reported arrival time, the phase name is changed to “UNKNOWN”, and if there is a hint about the type of phase from the event file, “UNKNOWNP” or “UNKNOWNS”. Such readings are flagged “p”. In this case the value printed in the residual column is the observed travel time. Any reading flagged with an “x”, either by hand or through the utility programs lres or xdat, will be in this section, as will readings specified in the skip command. In recent years it has become common for ISC datasets to have many duplicate readings, which will show up in this section with a “d” flag.

The “WHY BAD” field is used for two situations. It carries the code “PRES” (another ancient typo from when relocations were done only with P phases) if the reading was dropped because it fell outside the window defined for that phase by the wind command, probably from reading the ~.ttsprd file from a previous run. If the utility code xdat is run it will flag these readings.

The other use of the “WHY Bad” field is to print a “?” when a reading has been flagged as an outlier (“x”) but the residual is small enough (according to a fairly crude algorithm) that it may be worth re-checking (with rstat) that it truly does seem to be an outlier. A standard part of a careful relocation analysis is to review these cases late in the process and add readings back in (by removing the flags in the event files, by hand) when it is judged that a mistake was made.

~.rderr File

A *~.rderr* file is produced by every run of **mloc**. The file is written by the module *mlocout_rderr.f90*. It carries the calculation of empirical reading error and baseline offset (from the theoretical travel time) for every station-phase pair that has two or more unflagged instances in the inversion. There is no choice in the first run of a sequence but to use the default values for reading errors, or else run with weighting off (command weig), but it is normal practice to begin using the estimates of empirical reading error from the previous run (command rfil) very soon in the subsequent runs.

~.rderr files are not normally inspected by the user, so there are no headings for the columns:

```
Hypocentroid:     38.413    44.339    14.739
BAK          Pn        9     2.204     2.213     2.301    40.372    49.818     4.674    63.543
KSA          Pn        7     1.277     2.073     1.270    33.823    35.890     8.227   238.737
FEO          Pn        3     0.240     1.429     1.489    45.019    35.390     9.394   317.476
SEV          Pn        2    -0.367     4.005     4.026    44.545    33.668    10.077   310.824
HLW          Pn        8     0.071     1.129     1.146    29.858    31.342    13.725   235.434
SAM          P        12    -0.103     1.033     0.981    39.673    66.990    17.638    78.809
SAM          S         4    -0.539     3.564     1.685    39.673    66.990    17.638    78.809
LVV          S        12    -1.101     4.421     4.538    49.819    24.031    18.445   314.722
TAS          P        25     3.081     1.551     1.570    41.329    69.286    19.351    73.466
BEO          P        11     3.447     1.090     1.155    44.809    20.471    18.921   297.351
BEO          S         5     6.008     1.588     1.685    44.809    20.471    18.921   297.351
BUD          P        35     4.603     1.606     1.587    47.484    19.024    20.555   304.305
BUD          S         2     5.934     0.102     0.150    47.484    19.024    20.555   304.305
ANR          P        19     3.823     0.614     0.740    40.755    72.360    21.689    74.957
```

The first line in the file carries the hypocentroid's latitude, longitude and depth, for use in the plot type known as an empirical path anomaly plot implemented by command epap. The remainder of the file is simply the sequence of station-phase pairs for which an empirical reading error could be calculated, in the order in which they are encountered in the event data files.

The columns are:

- Station code (followed by deployment code if that is being used, see command radf)
- Phase
- Number of samples
- Baseline offset from the theoretical travel-time model
- Empirical reading error calculated from this run
- Reading error used in this run
- Latitude of the station
- Longitude of the station
- Epicentral distance from the hypocentroid
- Azimuth, hypocentroid to station

The last four columns are used to make an empirical path anomaly plot (command epap).

~.stn File

A ~.*stn* file is created every time **mloc** runs. It is written by routines in a number of modules, including:

- mloc.f90
- mlocset.f90
- mlocio_mnf.f90
- mloclib_stations.f90
- mlocout_gmt.f90

The information carried in a ~.*stn* file is of many types and the exact contents of a ~.*stn* file vary according to specifics of the arrival time dataset and the relocation strategy. Some of it is included for archival purposes but the most important function is to provide information about station codes and their coordinates that is helpful in diagnosing problems.

If supplemental station files have been used in a relocation analysis the first section of the ~.*stn* file deals with them:

```
Supplemental station files:
SSTN-1 salmas2/makhin_stn_bhrc.dat
SSTN-2 salmas2/makhin_stn_dda.dat
SSTN-3 salmas2/makhin_stn_neic.dat

salmas2/makhin_stn_bhrc.dat
3 BHRC stations for Makhin
    1 stations read

salmas2/makhin_stn_dda.dat
3
    7 stations read

salmas2/makhin_stn_neic.dat
5 Supplemental station file from NEIC Metadata
   13 stations read
```

Note that the number in column 1 of the first line of each supplemental station file, which is printed after the file name, is not the number of station codes in the file, but rather an index specifying the format of the file. The number of stations read is printed on the next line.

The next section deals with reading the master station file. While it is read, cases of duplication between the master station file (annotated as “IR” for International Registry) and one or more of the supplemental station files are compiled:

```
tables/stn/master_stn.dat
0 MLOC master station list
22113 stations read

22134 total stations read
ALKP                  -29.9418            22.2467     1.065 SSTN-3
ALKP                  -29.9423            22.2480     1.066 IR
```

```
                      0.0005            -0.0013    -0.001 significant difference in coordinates

SKN                  61.9800          -151.5317     0.581 SSTN-3
SKN                  42.9915           -76.4672     0.226 IR
                     18.9885           -75.0645     0.355 conflicting station codes

KHWEE               -21.9464            25.4469     1.080 SSTN-3
KHWEE               -21.9463            25.4468     1.083 IR
                     -0.0001             0.0001    -0.003 minor difference in coordinates

HSNB                 47.2769           -66.0601     0.337 SSTN-3
HSNB                 47.2769           -66.0601     0.337 IR
                      0.0000             0.0000     0.000 pure duplicate

POIN                 72.7012           -77.9620     0.021 SSTN-3
POIN                 72.7012           -77.9620     0.021 IR
                      0.0000             0.0000     0.000 pure duplicate

EDA                   3.7789            10.1534     0.140 SSTN-3
EDA                   3.7791            10.1533     0.000 IR
                     -0.0002             0.0001     0.140 significant difference in coordinates

ACRG                  5.6415            -0.2072     0.075 SSTN-3
ACRG                  5.6415            -0.2072     0.075 IR
                      0.0000             0.0000     0.000 pure duplicate


Duplicate stations:
    3 pure duplicates
    1 with minor differences
    2 with significant differences
    1 station conflicts
    7 total
```

Each case is characterized, depending on how different the coordinates are; the difference in each parameter (latitude, longitude, elevation) is shown. *Pure duplicates* can be ignored (the coordinates from the first occurrence will be used by **mloc**), or the instance of that station can be deleted from the corresponding supplemental station file to simplify things. Likewise, *minor differences in coordinates* can be ignored, or resolved by deleting the one from the supplemental station file, unless that station is being used for direct calibration. In that case it is advisable to investigate further.

Significant differences in coordinates also require investigation; in many cases they are caused by discrepancies in the elevation (see station EDA, above). An entry with zero elevation (like the master station file for EDA) is probably in error. The solution could be to let **mloc** use the entry from the supplemental station, or to check the elevation with Google Earth and correct it in the master station file.

Conflicting station codes may represent a problem or the solution to a problem, depending on whether or not the arrival time dataset contains readings from a station that has the same station code as one in the master station file but different coordinates. If it does then it is necessary to introduce the correct coordinates in a supplemental station file, which will lead to the report of a conflict. If it does not, however, the conflicting station code should be removed from the

supplemental station file. This often happens when a supplemental station file is for an entire network, of which only a few are actually needed for the arrival time dataset.

The next section reports on any cases where the date of an arrival in the dataset falls outside the operational epoch specified for that station in whichever station file was used for its coordinates; usually it is the master station file:

```
   stafind: failed date_range:   11 527 MAG    1976329 1952001 1975259
   stafind: failed date_range:   11 528 MAG    1976329 1952001 1975259
   stafind: failed date_range:   11 529 MAG    1976329 1952001 1975259
   stafind: failed date_range:   19 461 MAG    1977146 1952001 1975259
   stafind: failed date_range:   19 462 MAG    1977146 1952001 1975259
   stafind: failed date_range:   66 572 DMT    2011296 1979213 1998001
mlocset:    6 readings failed the station file date range
```

The first entry is interpreted as : “For the reading of MAG on line 527 of event 11 the observed date (day 329 of 1976) is outside the operational epoch (day 1 of 1952 to day 259 of 1975) currently specified for this station”. As discussed elsewhere the normal solution is to edit the operational epoch information in the station file.

This is followed by a report on missing stations:

```
Stations in the dataset that are missing from the station files:
TDG                   2 instances
MAG                   5 instances
AHLTE                 1 instances
```

Note that the 5 instances of MAG missing are caused by the failed date range condition. Cases where there is a single instance can usually be ignored, since such a reading will not contribute to the cluster vectors. If it would contribute to a direct calibration, however, it would be important to resolve the problem.

The next section is helpful in cases where one or more large supplemental files have been referenced to obtain coordinates for only a few stations:

```
Stations from supplemental station files that are present in the dataset, generic format:
3 Supplemental station file for salmas2.2
QOT1                  38.4800   44.4000  1946                      SSTN-1
3001                  37.5744   43.7877  1732                      SSTN-2
3604                  40.3313   42.5899  2098                      SSTN-2
4702                  37.4171   41.3574   933                      SSTN-2
4904                  38.7356   41.7742  1300                      SSTN-2
6506                  39.0196   43.3380  1681                      SSTN-2
6508                  38.6573   43.9767  1994                      SSTN-2
OLTU                  40.5460   41.9734  1350                      SSTN-2
ALKP                 -29.9418   22.2467  1065                      SSTN-3
KHWEE                -21.9464   25.4469  1080                      SSTN-3
BLKN                  64.3185  -96.0024    40                      SSTN-3
HSNB                  47.2769  -66.0601   337                      SSTN-3
POIN                  72.7012  -77.9620    21                      SSTN-3
YKAW3                 62.5616 -114.6099   196                      SSTN-3
EDA                    3.7789   10.1534   140                      SSTN-3
ACRG                   5.6415   -0.2072    75                      SSTN-3
SHAA                  37.5620   68.1228   868                      SSTN-3
```

The text provided here could be copied and pasted into a new cluster-specific supplemental station file, eliminating the need for the original supplemental station files and greatly reducing the chance of inadvertent station conflicts. The next section is also helpful in trimming unnecessary entries from supplemental station files:

```
Stations from supplemental station files that are not present in the dataset:
SKN                  61.9800 -151.5317   581                      SSTN-3
ZAA0B                53.9481   84.8188   229                      SSTN-3
DHK1                 36.8606   42.8665    16                      SSTN-3
E28B                 46.5748 -100.6934   704                      SSTN-3
```

The next section is quite long, so only the first few entries are shown below. This is a section of the *~.stn file* that is mainly included for archival and/or forensic purposes:

```
  3762 stations in the arrival time dataset for which coordinates were found:
BAK                  40.3720   49.8180    83       1903001         IR
KSA                  33.8233   35.8900   920               2001365 IR
FEO                  45.0190   35.3900    40       1927001         IR
YAL                  44.4875   34.1547    23       1928001         IR
SEV                  44.5450   33.6680    42       1928001         IR
HLW                  29.8583   31.3417   135       1899244         IR
SAM                  39.6733   66.9900   704       1913001         ERE
KUC                  55.7500   37.9667   155                       IR
```

The cases shown have all taken the coordinates from the master station file (including one case where coordinates from Bob Engdahl "ERE" are preferred). In the full listing there will be some entries referencing the supplemental station files (e.g., "SSTN-1", "SSTN-2", etc). For any cases that showed up in the station conflicts section of the *~.stn file*, the nature of the conflict will be added for annotation.

For a direct calibration relocation the last section of the *~.stn file* lists the stations used for the hypocentroid, breaking out any S-P data separately:

```
Seismic stations used for direct calibration.
ERE                  40.1700   44.4700   998
TAB                  38.0675   46.3267  1430
GRS                  39.5000   46.3333  1399
NAK                  39.2050   45.4150   887
KDR                  39.1500   46.1000  2155
ISHB                 38.2833   45.6192  2300
VANT                 38.4210   43.2653  1750
ITBZ                 38.2348   46.1498  1600
IMRD                 38.7132   45.7022  2100
VANB                 38.5090   43.4058  1227
TVAN                 38.5286   43.4061  2008
HKR                  37.5780   43.7408  1750
IAZR                 37.6775   45.9837  2300
MAKU                 39.3549   44.6834  1234
HAKT                 37.5579   43.7071  2153
CUKT                 37.2474   43.6076  1298
CLDR                 39.1431   43.9170  2087
DYDN                 39.5436   43.6889  2010
GEVA                 38.3122   43.0586  1672
```

```
TUTA                39.4019   42.8137  2154
AGRB                39.5755   42.9920  1820
NAX                 39.1740   45.4950   927
VMUR                38.9894   43.5716  1717
BASK                38.0523   44.0040  1931
EKAR                39.2559   42.0640  2129
ERCV                39.0198   43.3380  1681
ADCV                38.8080   42.7245  1774
TATV                38.5080   42.2672  1831
OZAP                38.6614   43.9928  2058
YOVA                37.5871   44.2896  1946
3001                37.5744   43.7877  1732
6508                38.6573   43.9767  1994
6506                39.0196   43.3380  1681
AKDM                38.3285   42.9800  1662
MLAZ                39.1410   42.5495  1581
GURO                38.5509   42.0322  1388
QOT1                38.4800   44.4000  1946

S-P stations used for direct calibration.
GEVA                38.3122   43.0586  1672
```

~.summary File

This section describes the *summary file* (~.summary), one of the most important output files, which is produced by every run of **mloc**. It has several sections:

- Header section
- Hypocentroid section
- Hypocentroid shift section
- Cluster statistics section
- Cluster vector changes section
- Data importance distribution section
- Event section

Header Section

The header section lists basic information and many of the parameters that were set for the run. Its use is mainly forensic: if you are inspecting a relocation by someone else or if you go back after some time to review one of your own runs, the information in the header section will provide answers to most of the basic questions about how it was done. Here is an example, from the Ridgecrest, California cluster (ridgecrest3.1):

```
Program: mloc v10.4.7, release date 7/14/2019
Run: ridgecrest3.1
Author: EBergman
Date: 20190824 132053.523
Travel times: ak135 (ridgecrest3/ridgecrest.cr)
```

```
Lg travel times:   0.03 +  31.68 * delta
T-phase travel times:  15.00 +  75.00 * delta
Station elevation correction: on (focal depth referenced to geoid in direct calibration)
Supplemental station file: ridgecrest3/ridgecrest_stn.dat
Data flags used: on
Residuals weighted by reading errors: on
Assume perfect theoretical TTs for hypocentroid (ttsprd = 0): off
Reading errors: ridgecrest3/ridgecrest2.42.rderr
Minimum allowed reading errors:  0.10 (local) / 0.15 (general) / 1.00 (depth phases)
Reading errors for local phases: Not set
Cluster vector fudge factor (non-gaussian):   0.00
Travel time spread: ridgecrest3/ridgecrest2.42.ttsprd
Windowing:   3.0  4.0
Hypocentroid bias correction: on
Data flags: used
Starting locations: ridgecrest3/ridgecrest2.42.hdf_dcal
Direct calibration: on
Distance ranges:
   Hypocentroid   :    0.0   0.8
   Cluster vectors:    0.0 180.0
Hypocentroid location parameters: (lat, lon,      , origin time)
```

Hypocentroid Section

This section tracks changes in the hypocentroid of the cluster through the relocation.

```
                      TIME        LATITUDE     LONGITUDE       DEPTH          E**2/NSTA   SHAT  PRMAX  DELT PRES FLAG
START              14:18:47.53      35.787      -117.570        15.0

ITER 0                   0.00 S    -0.02 KM       0.01 KM       0.00 KM     104.0/ 1197   0.89    4.0  9235  656 1915
ITER 1                   0.00 S     0.02 KM       0.00 KM       0.00 KM     104.8/ 1197   0.89    4.0  9235  609 1915
ITER 2                   0.00 S     0.01 KM       0.00 KM       0.00 KM     104.2/ 1197   0.89    4.0  9235  608 1915

CUMULATIVE               0.00 S     0.01 KM       0.01 KM       0.00 KM    AZIM =    0 DEG    DIST =  0.0 KM

SOLUTION           14:18:47.53      35.787      -117.570        15.0
STANDARD ERRORS          0.02 S     0.001 DEG     0.002 DEG     0.00 KM    ELLIPSE: -33.9 DEG   0.3 AND    0.3 KM
```

Hypocentroid Shift Section

This section of a ~.summary file only exists in the case of indirect calibration. In that case, after the relocation is converged, using either an uncalibrated or direct calibration methodology, the locations of events that have been declared as calibration events in the command file are compared with the calibration locations and an optimal shift of the entire cluster in space and time is calculated to bring the cluster into best agreement with the calibration data. In this section the shift required for each calibration event is listed and the final optimal shift. Much more detail about this process is provided in the Cal file (~.cal) file that is produced when indirect calibration is invoked.

The following example is taken from a cluster consisting nuclear tests conducted by the United States at Bikini Atoll:

```
SHIFT                   -0.25 S    -0.032 DEG     0.072 DEG     0.00 KM    AZIM =  114 DEG    DIST =  8.6 KM
CALIBRATED         16:30:20.01      11.670       165.324         0.0

Calibration event correction vectors (weights):
```

```
EVENT   1  19540228.1845.00  -0.08(1.39)  -0.021(1.51)   0.058(1.51)   0.00(1.00) AZIM =  110 DEG    DIST =  6.7 KM
EVENT   2  19540326.1830.00  -0.28(1.47)  -0.027(1.75)   0.088(1.75)   0.00(1.00) AZIM =  107 DEG    DIST = 10.0 KM
EVENT   3  19540425.1810.00  -0.66(0.94)  -0.027(0.90)   0.069(0.90)   0.00(1.00) AZIM =  112 DEG    DIST =  8.1 KM
EVENT   4  19540504.1810.00  -0.23(1.35)  -0.033(1.49)   0.050(1.49)   0.00(1.00) AZIM =  124 DEG    DIST =  6.6 KM
EVENT   5  19560527.1756.00  -0.48(0.71)  -0.041(0.91)   0.060(0.91)   0.00(1.00) AZIM =  125 DEG    DIST =  8.0 KM
EVENT   6  19560710.1756.00  -0.47(0.86)  -0.058(0.89)   0.077(0.89)   0.00(1.00) AZIM =  128 DEG    DIST = 10.5 KM
EVENT   7  19560720.1746.00  -0.02(0.59)  -0.021(0.31)   0.114(0.31)   0.00(1.00) AZIM =  100 DEG    DIST = 12.6 KM
EVENT   8  19580511.1750.00  -0.14(0.65)  -0.061(0.35)   0.114(0.35)   0.00(1.00) AZIM =  119 DEG    DIST = 14.1 KM
EVENT   9  19580712.0330.00   0.04(1.05)  -0.023(0.90)   0.084(0.90)   0.00(1.00) AZIM =  106 DEG    DIST =  9.5 KM
```

The first line provides the calibration shift applied to the cluster, followed by the calibrated hypocentroid parameters. The following lines list the calibration shift required in each parameter (in the order: OT, lat, lon, depth), based on each calibration event, and each parameter is followed in parentheses by the weight given that datum in calculating the optimal shift. The weights are based on the uncertainty of that parameter in the calibration data and in the corresponding event cluster vector.

Cluster Statistics Section

This section tracks the cumulative squared error and number of data used for the cluster vectors through each iteration, and calculates the normalized sample variance *shatc* over all cluster vectors. In the early stages of a relocation *shatc* will usually be greater than 1 but as outlier readings are flagged and empirical reading errors are developed, it should finally be very close to 1.0. From ridgecrest3.1:

```
 cluster vector statistics   ITER      E**2/NSTA      SHATC
                               0       5928.2/ 7510    0.99
                               1       5649.7/ 7556    0.96
                               2       5650.0/ 7556    0.96
 Spread of event normalized sample variances:  0.161
```

The last line of this section is based on an analysis of the normalized sample variances for each event in the cluster, which are listed in the event section. These values are expected to be normally distributed around 1.0 with a spread determined by Bayesian considerations, as discussed in Jordan and Sverdrup (1981). The relevant code is in the module *mlocinv.f90*, in the section “Confidence ellipses for lat-long” beginning on line 402, which contains a lengthy comment on the approach used. As noted therein a conservative choice of k=3 is made for the Bayesian parameter, which leads to an expected spread of 0.4 for the normalized cluster sample variances. One could review the observed spreads of many clusters and perhaps decide to change the value of the Bayesian parameter (probably raise it), but this has not yet been done.

If the spread of event normalized sample variances were noticed to be significantly larger than expected, it means one (or probably more) of the individual cluster vectors have major problems, and investigation is certainly warranted.

Cluster Vector Changes Section

This section lists three columns for the cluster vector of each event, the initial vector, the final vector and (in the middle) the change vector. In the early stages of a relocation, the change vectors are often large but by the end they should be very close to zero, as the relocation is

started from the previous location and converges immediately without any events moving around. During relocation if one or a few events are unstable they can be spotted quickly in the "CHANGE" column. If a preliminary or exploratory cluster contains some events that end up being rather far away from the area of interest, the "FINAL" column of the output is a convenient way to pick them out. Here is the section from the ridgecrest3.1 run:

			INITIAL	CHANGE	FINAL
EVENT	1	19930520.2014.14	36.2 KM AT 340 DEG	0.0 KM AT 0 DEG	36.2 KM AT 340 DEG
EVENT	2	19950817.2239.57	12.4 KM AT 269 DEG	0.0 KM AT 0 DEG	12.4 KM AT 270 DEG
EVENT	3	19950830.1529.53	8.2 KM AT 267 DEG	0.0 KM AT 0 DEG	8.2 KM AT 267 DEG
EVENT	4	19950920.2327.35	10.0 KM AT 249 DEG	0.0 KM AT 0 DEG	10.0 KM AT 249 DEG
EVENT	5	19950925.0447.28	5.7 KM AT 306 DEG	0.0 KM AT 0 DEG	5.8 KM AT 307 DEG
EVENT	6	19951018.1242.04	8.5 KM AT 225 DEG	0.0 KM AT 0 DEG	8.5 KM AT 225 DEG
EVENT	7	19960107.1432.52	12.0 KM AT 250 DEG	0.0 KM AT 0 DEG	12.0 KM AT 250 DEG
EVENT	8	19960108.0857.10	8.1 KM AT 258 DEG	0.0 KM AT 0 DEG	8.1 KM AT 258 DEG
EVENT	9	19960108.1052.29	4.4 KM AT 253 DEG	0.0 KM AT 0 DEG	4.4 KM AT 253 DEG
EVENT	10	19960126.1306.02	7.9 KM AT 263 DEG	0.0 KM AT 0 DEG	7.9 KM AT 263 DEG
EVENT	11	19961127.2017.24	33.4 KM AT 347 DEG	0.0 KM AT 0 DEG	33.5 KM AT 347 DEG
EVENT	12	19961217.0403.22	35.5 KM AT 348 DEG	0.0 KM AT 0 DEG	35.6 KM AT 348 DEG
EVENT	13	20081125.0411.36	29.8 KM AT 46 DEG	0.0 KM AT 0 DEG	29.8 KM AT 46 DEG
EVENT	14	20081202.1123.43	32.2 KM AT 47 DEG	0.0 KM AT 0 DEG	32.2 KM AT 47 DEG
EVENT	15	20081202.1641.19	30.1 KM AT 48 DEG	0.0 KM AT 0 DEG	30.1 KM AT 48 DEG
EVENT	16	20081202.1653.09	30.4 KM AT 48 DEG	0.0 KM AT 0 DEG	30.4 KM AT 48 DEG
EVENT	17	20090131.2109.21	45.3 KM AT 204 DEG	0.0 KM AT 0 DEG	45.4 KM AT 204 DEG
EVENT	18	20090220.1219.13	33.0 KM AT 47 DEG	0.0 KM AT 0 DEG	32.9 KM AT 47 DEG
EVENT	19	20121202.1020.34	42.6 KM AT 323 DEG	0.0 KM AT 0 DEG	42.6 KM AT 323 DEG
EVENT	20	20131223.1339.26	60.0 KM AT 311 DEG	0.0 KM AT 0 DEG	60.0 KM AT 311 DEG
EVENT	21	20140304.2249.31	31.3 KM AT 50 DEG	0.0 KM AT 0 DEG	31.3 KM AT 50 DEG
EVENT	22	20140313.0211.04	57.9 KM AT 310 DEG	0.0 KM AT 0 DEG	57.9 KM AT 310 DEG
EVENT	23	20160809.0424.56	33.6 KM AT 326 DEG	0.0 KM AT 0 DEG	33.6 KM AT 326 DEG
EVENT	24	20170822.1951.58	59.5 KM AT 76 DEG	0.0 KM AT 0 DEG	59.5 KM AT 76 DEG
EVENT	25	20170919.1845.44	32.3 KM AT 325 DEG	0.0 KM AT 0 DEG	32.3 KM AT 325 DEG
EVENT	26	20190704.1702.55	11.1 KM AT 145 DEG	0.0 KM AT 0 DEG	11.1 KM AT 145 DEG
EVENT	27	20190704.1733.49	11.3 KM AT 145 DEG	0.0 KM AT 0 DEG	11.3 KM AT 145 DEG
EVENT	28	20190704.1735.01	15.9 KM AT 177 DEG	0.0 KM AT 0 DEG	15.9 KM AT 178 DEG
EVENT	29	20190704.1737.55	4.8 KM AT 160 DEG	0.0 KM AT 0 DEG	4.8 KM AT 160 DEG
EVENT	30	20190704.1740.18	10.6 KM AT 151 DEG	0.0 KM AT 0 DEG	10.6 KM AT 151 DEG
EVENT	31	20190704.1744.32	12.8 KM AT 151 DEG	0.0 KM AT 0 DEG	12.8 KM AT 151 DEG
EVENT	32	20190704.1806.16	15.1 KM AT 162 DEG	0.0 KM AT 0 DEG	15.1 KM AT 162 DEG
EVENT	33	20190704.1827.59	5.2 KM AT 160 DEG	0.0 KM AT 0 DEG	5.2 KM AT 160 DEG
EVENT	34	20190704.1828.43	13.7 KM AT 180 DEG	0.0 KM AT 0 DEG	13.7 KM AT 180 DEG
EVENT	35	20190704.1839.44	20.4 KM AT 184 DEG	0.0 KM AT 0 DEG	20.4 KM AT 184 DEG
EVENT	36	20190704.1847.06	15.0 KM AT 146 DEG	0.0 KM AT 0 DEG	15.0 KM AT 146 DEG
EVENT	37	20190704.1854.13	21.3 KM AT 186 DEG	0.0 KM AT 0 DEG	21.3 KM AT 186 DEG
EVENT	38	20190704.1856.06	7.9 KM AT 172 DEG	0.0 KM AT 0 DEG	7.9 KM AT 172 DEG
EVENT	39	20190704.1921.32	15.6 KM AT 147 DEG	0.0 KM AT 0 DEG	15.6 KM AT 147 DEG
EVENT	40	20190704.1956.00	14.8 KM AT 162 DEG	0.0 KM AT 0 DEG	14.8 KM AT 162 DEG
EVENT	41	20190704.2212.08	5.0 KM AT 177 DEG	0.0 KM AT 0 DEG	5.0 KM AT 177 DEG
EVENT	42	20190704.2334.00	15.0 KM AT 164 DEG	0.0 KM AT 0 DEG	15.0 KM AT 164 DEG
EVENT	43	20190705.0018.01	4.8 KM AT 248 DEG	0.0 KM AT 0 DEG	4.8 KM AT 248 DEG
EVENT	44	20190705.1107.53	3.2 KM AT 193 DEG	0.0 KM AT 0 DEG	3.2 KM AT 193 DEG
EVENT	45	20190706.0319.52	3.3 KM AT 230 DEG	0.0 KM AT 0 DEG	3.3 KM AT 230 DEG
EVENT	46	20190706.0346.26	14.7 KM AT 143 DEG	0.0 KM AT 0 DEG	14.7 KM AT 143 DEG
EVENT	47	20190706.0407.04	25.8 KM AT 168 DEG	0.0 KM AT 0 DEG	25.8 KM AT 168 DEG
EVENT	48	20190706.0413.07	20.9 KM AT 199 DEG	0.0 KM AT 0 DEG	20.9 KM AT 199 DEG
EVENT	49	20190706.0440.16	19.4 KM AT 148 DEG	0.0 KM AT 0 DEG	19.4 KM AT 148 DEG
EVENT	50	20190706.0832.57	18.8 KM AT 156 DEG	0.0 KM AT 0 DEG	18.8 KM AT 156 DEG
EVENT	51	20190706.0859.55	23.0 KM AT 145 DEG	0.0 KM AT 0 DEG	23.0 KM AT 145 DEG
EVENT	52	20190706.1636.04	24.8 KM AT 166 DEG	0.0 KM AT 0 DEG	24.8 KM AT 166 DEG
EVENT	53	20190706.2350.41	9.4 KM AT 293 DEG	0.0 KM AT 0 DEG	9.4 KM AT 293 DEG
EVENT	54	20190707.0538.15	2.7 KM AT 185 DEG	0.0 KM AT 0 DEG	2.7 KM AT 185 DEG
EVENT	55	20190707.1122.16	2.6 KM AT 323 DEG	0.0 KM AT 0 DEG	2.6 KM AT 322 DEG
EVENT	56	20190710.2009.51	14.4 KM AT 162 DEG	0.0 KM AT 0 DEG	14.4 KM AT 162 DEG
EVENT	57	20190711.2345.18	21.9 KM AT 326 DEG	0.0 KM AT 0 DEG	21.9 KM AT 326 DEG
EVENT	58	20190712.1311.37	16.7 KM AT 184 DEG	0.0 KM AT 0 DEG	16.7 KM AT 184 DEG

Data Importance Distribution Section

The concept of data importance is discussed in Jordan and Sverdrup (1981). This section displays the cumulative data importance of all the readings in 30° azimuthal wedges, for the hypocentroid, for the cluster vectors of individual events and the cluster vectors in total. This is a useful way to investigate the azimuthal distribution of the data, which is a critical aspect of a reliable relocation. It is difficult to quickly comprehend the numbers in the main table, so a graphical summary is shown at the end of each event line. If the cumulative data importance in a wedge is less than 0.010 the symbol "0" is plotted. Otherwise a "-" is plotted. A more precise measure of open azimuth for each event is found in the HDF file.

In practice, events with an open azimuth up to about 220° are usually stable in **mloc**. In this graphical display 6 or fewer "zeros" in a row (including wrap-around) are usually not worrisome.

```
DATA IMPORTANCE:                NNE    NE   ENE   ESE    SE   SSE   SSW    SW   WSW   WNW    NW   NNW
 HYPOCENTROID:                0.070 0.016 0.096 0.093 0.077 0.059 0.037 0.110 0.060 0.090 0.118 0.175
 EVENT   1  19930520.2014.14 0.054 0.167 0.019 0.001 0.083 0.139 0.000 0.210 0.126 0.113 0.056 0.032   ---0--0-----
 EVENT   2  19950817.2239.57 0.124 0.069 0.085 0.006 0.065 0.170 0.004 0.146 0.035 0.165 0.053 0.078   ---0--0-----
 EVENT   3  19950830.1529.53 0.056 0.093 0.069 0.010 0.049 0.170 0.025 0.044 0.187 0.190 0.047 0.058   ------------
 EVENT   4  19950920.2327.35 0.057 0.073 0.092 0.005 0.088 0.126 0.058 0.093 0.140 0.108 0.059 0.100   ---0--------
 EVENT   5  19950925.0447.28 0.074 0.046 0.098 0.011 0.064 0.113 0.109 0.077 0.072 0.132 0.119 0.085   ------------
 EVENT   6  19951018.1242.04 0.035 0.082 0.065 0.022 0.085 0.144 0.072 0.077 0.143 0.119 0.056 0.101   ------------
 EVENT   7  19960107.1432.52 0.070 0.103 0.068 0.011 0.042 0.149 0.005 0.146 0.035 0.226 0.004 0.139   ------0---0-
 EVENT   8  19960108.0857.10 0.041 0.099 0.007 0.052 0.016 0.183 0.028 0.198 0.059 0.114 0.077 0.126   --0---------
 EVENT   9  19960108.1052.29 0.022 0.074 0.042 0.002 0.089 0.149 0.046 0.105 0.090 0.193 0.033 0.155   ---0--------
 EVENT  10  19960126.1306.02 0.043 0.042 0.059 0.031 0.065 0.136 0.022 0.091 0.165 0.184 0.032 0.129   ------------
 EVENT  11  19961127.2017.24 0.057 0.076 0.091 0.009 0.075 0.123 0.059 0.103 0.177 0.012 0.144 0.074   ---0--------
 EVENT  12  19961217.0403.22 0.103 0.069 0.034 0.002 0.086 0.064 0.127 0.222 0.055 0.057 0.142 0.040   ---0--------
 EVENT  13  20081125.0411.36 0.022 0.096 0.006 0.111 0.143 0.080 0.119 0.080 0.084 0.006 0.207 0.046   --0------0--
 EVENT  14  20081202.1123.43 0.031 0.103 0.029 0.106 0.146 0.075 0.075 0.107 0.042 0.109 0.131 0.046   ------------
 EVENT  15  20081202.1641.19 0.023 0.088 0.005 0.119 0.162 0.080 0.121 0.092 0.042 0.000 0.232 0.036   --0------0--
 EVENT  16  20081202.1653.09 0.030 0.094 0.006 0.121 0.129 0.080 0.140 0.061 0.041 0.006 0.251 0.041   --0------0--
 EVENT  17  20090131.2109.21 0.060 0.079 0.063 0.147 0.091 0.065 0.081 0.019 0.084 0.143 0.009 0.160   ----------0-
 EVENT  18  20090220.1219.13 0.030 0.099 0.026 0.101 0.127 0.130 0.064 0.081 0.081 0.099 0.113 0.049   ------------
 EVENT  19  20121202.1020.34 0.031 0.057 0.071 0.160 0.095 0.114 0.119 0.114 0.034 0.001 0.031 0.173   ---------0--
 EVENT  20  20131223.1339.26 0.026 0.053 0.097 0.134 0.140 0.085 0.098 0.093 0.059 0.051 0.035 0.130   ------------
 EVENT  21  20140304.2249.31 0.024 0.101 0.016 0.090 0.131 0.094 0.052 0.125 0.109 0.063 0.163 0.031   ------------
 EVENT  22  20140313.0211.04 0.013 0.051 0.019 0.106 0.179 0.161 0.129 0.095 0.063 0.024 0.003 0.156   ----------0-
 EVENT  23  20160809.0424.56 0.045 0.049 0.106 0.116 0.117 0.093 0.035 0.169 0.039 0.019 0.160 0.050   ------------
 EVENT  24  20170822.1951.58 0.151 0.075 0.012 0.114 0.044 0.143 0.044 0.093 0.071 0.149 0.009 0.097   ----------0-
 EVENT  25  20170919.1845.44 0.031 0.087 0.076 0.096 0.122 0.086 0.111 0.146 0.037 0.020 0.138 0.049   ------------
 EVENT  26  20190704.1702.55 0.122 0.079 0.111 0.061 0.073 0.076 0.016 0.098 0.080 0.105 0.141 0.038   ------------
 EVENT  27  20190704.1733.49 0.091 0.116 0.067 0.038 0.149 0.070 0.015 0.070 0.123 0.040 0.141 0.080   ------------
 EVENT  28  20190704.1735.01 0.094 0.023 0.024 0.076 0.112 0.011 0.030 0.167 0.114 0.083 0.111 0.154   ------------
 EVENT  29  20190704.1737.55 0.118 0.000 0.032 0.000 0.198 0.016 0.170 0.031 0.138 0.000 0.205 0.090   -0-0-----0--
 EVENT  30  20190704.1740.18 0.022 0.085 0.019 0.023 0.155 0.040 0.003 0.164 0.110 0.072 0.166 0.142   ------0-----
 EVENT  31  20190704.1744.32 0.037 0.121 0.115 0.021 0.119 0.075 0.009 0.112 0.136 0.061 0.050 0.145   ------0-----
 EVENT  32  20190704.1806.16 0.035 0.155 0.028 0.029 0.140 0.067 0.005 0.188 0.053 0.061 0.084 0.154   ------0-----
 EVENT  33  20190704.1827.59 0.119 0.021 0.038 0.043 0.172 0.017 0.106 0.028 0.125 0.091 0.145 0.093   ------------
 EVENT  34  20190704.1828.43 0.091 0.033 0.023 0.072 0.151 0.016 0.141 0.024 0.107 0.063 0.085 0.194   ------------
 EVENT  35  20190704.1839.44 0.135 0.062 0.022 0.075 0.075 0.098 0.030 0.184 0.068 0.056 0.095 0.100   ------------
 EVENT  36  20190704.1847.06 0.072 0.095 0.034 0.065 0.112 0.047 0.003 0.181 0.095 0.109 0.082 0.105   ------0-----
 EVENT  37  20190704.1854.13 0.097 0.140 0.018 0.122 0.052 0.048 0.003 0.170 0.091 0.078 0.078 0.104   ------0-----
 EVENT  38  20190704.1856.06 0.027 0.044 0.105 0.057 0.169 0.023 0.156 0.028 0.163 0.020 0.067 0.141   ------------
 EVENT  39  20190704.1921.32 0.040 0.189 0.028 0.041 0.110 0.073 0.036 0.122 0.058 0.104 0.089 0.112   ------------
 EVENT  40  20190704.1956.00 0.054 0.180 0.043 0.054 0.062 0.087 0.029 0.151 0.068 0.072 0.129 0.072   ------------
 EVENT  41  20190704.2212.08 0.035 0.123 0.053 0.065 0.124 0.073 0.055 0.035 0.126 0.068 0.184 0.059   ------------
 EVENT  42  20190704.2334.00 0.046 0.116 0.035 0.063 0.137 0.057 0.025 0.121 0.146 0.054 0.132 0.070   ------------
 EVENT  43  20190705.0018.01 0.068 0.042 0.141 0.059 0.119 0.051 0.116 0.053 0.078 0.057 0.048 0.167   ------------
 EVENT  44  20190705.1107.53 0.098 0.074 0.059 0.064 0.077 0.096 0.102 0.073 0.107 0.048 0.098 0.103   ------------
 EVENT  45  20190706.0319.52 0.105 0.086 0.057 0.056 0.137 0.086 0.128 0.062 0.048 0.061 0.136 0.038   ------------
 EVENT  46  20190706.0346.26 0.056 0.201 0.009 0.012 0.087 0.102 0.059 0.124 0.129 0.058 0.098 0.066   --0---------
 EVENT  47  20190706.0407.04 0.084 0.127 0.002 0.140 0.066 0.021 0.027 0.165 0.093 0.121 0.075 0.079   --0---------
 EVENT  48  20190706.0413.07 0.089 0.103 0.246 0.038 0.036 0.022 0.000 0.203 0.004 0.098 0.015 0.147   ------0-0---
 EVENT  49  20190706.0440.16 0.118 0.095 0.000 0.092 0.201 0.118 0.020 0.119 0.004 0.184 0.000 0.048   --0-----0-0-
 EVENT  50  20190706.0832.57 0.059 0.097 0.074 0.058 0.075 0.059 0.034 0.197 0.034 0.053 0.159 0.100   ------------
 EVENT  51  20190706.0859.55 0.104 0.040 0.004 0.089 0.121 0.082 0.091 0.128 0.104 0.043 0.097 0.098   --0---------
```

```
EVENT  52  20190706.1636.04 0.138 0.257 0.039 0.050 0.093 0.116 0.099 0.013 0.048 0.018 0.000 0.128   ----------0-
EVENT  53  20190706.2350.41 0.053 0.084 0.029 0.111 0.065 0.085 0.144 0.074 0.052 0.149 0.066 0.089   ------------
EVENT  54  20190707.0538.15 0.118 0.017 0.118 0.098 0.067 0.094 0.106 0.013 0.069 0.096 0.154 0.052   ------------
EVENT  55  20190707.1122.16 0.049 0.064 0.092 0.066 0.090 0.086 0.113 0.105 0.077 0.056 0.099 0.102   ------------
EVENT  56  20190710.2009.51 0.063 0.108 0.082 0.040 0.122 0.085 0.034 0.116 0.075 0.056 0.137 0.082   ------------
EVENT  57  20190711.2345.18 0.070 0.114 0.048 0.108 0.071 0.134 0.103 0.052 0.039 0.141 0.068 0.051   ------------
EVENT  58  20190712.1311.37 0.074 0.067 0.092 0.060 0.065 0.090 0.056 0.119 0.055 0.065 0.133 0.125   ------------
CLUSTER VECTORS:           0.066 0.089 0.054 0.065 0.105 0.089 0.066 0.108 0.084 0.081 0.098 0.095
```

Event Section

The final section of a summary file lists details of the relocation for all events. The first event from the ridgecrest3.1 run is listed:

```
CLUSTER EVENT   1 19930520.2014.14                    ridgecrest3/19930520.2014.14.mnf
MAG    4.2
Cluster vector location parameters: (lat, lon,      , origin time)
           DATE         TIME        LATITUDE     LONGITUDE       DEPTH         EC**2/NSTA    EH**2/NSTA  PRMAX  DELT PRES FLAG
INPUT    5/20/1993   20:14:14.01      36.057      -117.632         0.8
START    5/20/1993   20:14:14.08      36.093      -117.704        13.0

ITER 0                     0.01 S     0.16 KM      -0.20 KM       0.00 KM      51.3/  65      0.3/   6    4.0    94   10   20
ITER 1                    -0.01 S    -0.14 KM       0.18 KM       0.00 KM      50.6/  67      0.4/   6    4.0    94    8   20
ITER 2                     0.00 S     0.01 KM      -0.01 KM       0.00 KM      50.6/  67      0.3/   6    4.0    94    8   20

 CUMULATIVE                0.00 S     0.03 KM      -0.02 KM       0.00 KM    AZIM =    0 DEG    DIST =  0.0 KM
+HYPOCENTROID              0.00 S     0.01 KM       0.01 KM       0.00 KM
 TOTAL                     0.00 S     0.03 KM      -0.02 KM       0.00 KM    AZIM =    0 DEG    DIST =  0.0 KM

SOLUTION  5/20/1993  20:14:14.08      36.094      -117.704        13.0
STANDARD ERRORS            0.05 S     0.004 DEG     0.005 DEG     0.00 KM    ELLIPSE:  55.0 DEG   0.8 AND    0.9 KM

SOLUTION-INPUT             0.07 S     0.037 DEG    -0.072 DEG    12.20 KM    AZIM =  302 DEG    DIST =  7.7 KM

Normalized sample variance for cluster vector:  0.789 (  -1.32)
```

The first line gives the event number, the event "name" and the pathname to the input (~.mnf) file. Cluster vector location parameters (i.e., free parameters) are usually the same as for the hypocentroid. The starting location for most runs (except the first few, until the command rhdf begins to be used) will normally be different from the hypocenter in the .mnf file.

At each iteration the squared error and number of data are listed both for the event's cluster vector and its contribution (if any) to the hypocentroid. The cumulative changes in cluster vector parameters are listed and then the changes to the hypocentroid are added to obtain the final location. Standard errors are those of the cluster vector only. The difference between final location and the input location (.mnf file) are calculated.

Finally, the event normalized sample variance is calculated and the "residual" (relative to an expected value of 1.0 and normalized by the spread of event normalized sample variances), discussed in the Cluster Statistics Section, is shown in parentheses. In this case, the spread is 0.161, so:

(0.789-1.0)/0.161 = -1.32

Interpretation of these numbers is not straight-forward, because the expected value of 1.0 is not always representative. In the ridgecrest3.1 relocation the mean of all event normalized sample variances is ~0.79, so this event is not "better than average", but rather right on the mean for the

cluster. Overall, this cluster displays less error than expected between observed and predicted arrival times, even after empirical reading errors are accounted for. One possible cause is the minimum allowed reading error. If many readings are being given artificially large reading errors, the observed discrepancy may be explained. Some clusters deviate in the other direction. The source(s) of these discrepancies have not been investigated. In any case values of the normalized event sample variance greater than ~2 are worthy of some investigation. The event normalized sample variance for each event is also listed in the HDF file. It is usually more convenient to review these values there.

~.taup File

A *~.taup file* is only produced if the tlog command has been issued. The only reason for this command to be used is in the debugging of the tau-p software in the code module *libtau.f90*. The listing is very cryptic and very large, easily over 500 MB for a moderately large cluster. The output can only be understood by close referral to the code in *libtau.f90*.

~.tomo Files

~.tomo files are produced by the tomo command. The files are written in the module *mlocout_tomo.f90*, in a format intended to provide convenient access to the aspects of a relocation study, especially a calibrated relocation, that are of greatest interest to someone conducting research in seismic tomography. Tomographers normally work with large datasets extracted from more-or-less standard bulletins of (biased) single event locations, so the input data are individual travel times of specific phases. The fact that source hypocenters are biased to variable unknown degrees is conventionally ignored.

For a dataset of (possibly) calibrated events relocated as a cluster by **mloc**, there are several possibilities for the nature of the data set offered for tomography, basically trading quantity of data against quality of data. This is reflected in the three *types* of output that may be selected in the tomo command:

1. Extract all readings of the specified phase
2. Extract only readings which were used for the cluster vectors
3. Extract empirical path anomalies

The type 1 option is very similar to the typical tomographic dataset, especially for an uncalibrated cluster, although the noise in the data may be somewhat lessened by improved relative locations of the events. The type 2 option sacrifices numbers of data in the interest of better quality control. Readings that have survived the cleaning process and are still contributing to cluster vectors should be quite reliable. Readings that are not multiply-observed may still be good readings but they cannot be validated so they are not included. So this dataset will be a bit more sparse but of higher quality. This is even more true for the type 3 option, which reduces all the readings for a give station-phase to a single estimate, known as the empirical path anomaly.

Each ~.tomo file includes an identification of the phase it contains and the type of output file (1-3). All such files are stored in a subdirectory of the cluster series directory named ~_*tomo*.

Types 1 and 2 Format

The only difference between type 1 and type 2 tomo files is the selection criteria, i.e., type 2 is more selective and there will be fewer outliers. Here is an example, the first 10 lines from the file *wells6.1_P_2.tomo* of the Wells, Nevada cluster, with a head row added for clarity:

```
Delta    Res   RdErr   N      ArrTime          Event Origin      TTO   EvLat    EvLon    H     StaLat StaLon   Elev Sta         Phase     Ev# Cal Run
21.384   1.06   0.15  14 20070228 115228.90  20070228 114740.75  288.15  41.143-114.851  12.3   62.493 245.395   0.2 YKA         P          1   1 wells6.1
18.039   1.39   0.33   2 20080221 142014.40  20080221 1416 2.04  252.36  41.141-114.869  11.4   34.545 266.423   0.2 MIAR        P          3   1 wells6.1
18.724   3.37   1.14   2 20080221 142023.93  20080221 1416 2.04  261.89  41.141-114.869  11.4   31.760 265.339   0.2 NATX        P          3   1 wells6.1
18.841   2.36   1.31   2 20080221 142024.18  20080221 1416 2.04  262.14  41.141-114.869  11.4   34.775 267.657   0.1 UALR        P          3   1 wells6.1
19.053   1.98   0.74   2 20080221 142026.10  20080221 1416 2.04  264.07  41.141-114.869  11.4   38.636 269.764   0.2 SLM         P          3   1 wells6.1
19.092   2.73   1.39   4 20080221 142027.30  20080221 1416 2.04  265.27  41.141-114.869  11.4   37.984 269.574   0.3 FVM         P          3   1 wells6.1
19.456   4.08   4.09   2 20080221 142032.66  20080221 1416 2.04  270.62  41.141-114.869  11.4   27.546 262.107   0.0 KVTX        P          3   1 wells6.1
19.804   5.25   1.63   2 20080221 142037.53  20080221 1416 2.04  275.49  41.141-114.869  11.4   58.437 229.973   1.0 DLBC        P          3   1 wells6.1
21.129   4.06   1.81   3 20080221 142050.80  20080221 1416 2.04  288.77  41.141-114.869  11.4   34.512 270.591   0.1 OXF         P          3   1 wells6.1
21.386   1.43   0.15  14 20080221 142050.70  20080221 1416 2.04  288.66  41.141-114.869  11.4   62.493 245.395   0.2 YKA         P          3   1 wells6.1
21.665   3.19   0.41   2 20080221 142055.70  20080221 1416 2.04  293.66  41.141-114.869  11.4   36.130 272.170   0.2 WVT         P          3   1 wells6.1
```

Most of the headers are easily understood. *TTO* is the observed travel time (arrival time – origin time). For a calibrated cluster this will be an empirical true travel time. Calibration state is indicated by the *Cal* column. A number less than 99 will be the calibration level. A value of 99 indicates an uncalibrated cluster, in which case the observed travel time is understood to be biased to an unknown degree, as has been the case in virtually every global and regional tomography study ever done.

The file is written by looping over the events in the cluster and extracting readings that meet the criteria. This example was for the P phase. In the Wells cluster most of the events are rather small; many of them have few or no P observations. Event 1 has a single observation (YKA). Event 2 has none. Event 3 is the mainshock (5.8 Mw) on Feb 21, 2008, and it has the largest contribution to the set of P data in this file. Many of the entries in this file have a small value for number of samples (*N*), so in this case there is perhaps not much advantage to asking for a type 2 output rather than type 1.

Type 3 Format

Here are the first 10 lines from the file *wells6.1_Pn_3.tomo*, from the Wells, Nevada cluster:

```
 1.309  -0.80   0.20  16 99999999 999999.99  99999999 999999.99   24.53  41.173-114.887  10.5   40.719-116.508   1.4 N10A        Pn         0   1 wells6.1
 1.298  -0.68   0.11  27 99999999 999999.99  99999999 999999.99   24.49  41.166-114.882  10.5   41.522-116.540   1.7 M10A        Pn         0   1 wells6.1
 1.427  -1.10   0.25  40 99999999 999999.99  99999999 999999.99   25.90  41.160-114.891  10.4   40.925-113.030   1.6 BGU         Pn         0   1 wells6.1
 1.503  -0.65   0.18  40 99999999 999999.99  99999999 999999.99   27.40  41.163-114.887  10.7   40.292-116.500   1.5 O10A        Pn         0   1 wells6.1
 1.462  -1.21   0.18  33 99999999 999999.99  99999999 999999.99   26.28  41.173-114.888  10.5   42.636-114.903   1.1 K12A        Pn         0   1 wells6.1
 1.496  -0.69   0.28  39 99999999 999999.99  99999999 999999.99   27.29  41.164-114.889  10.5   42.077-116.471   1.5 L10A        Pn         0   1 wells6.1
 1.603  -0.53   0.28  37 99999999 999999.99  99999999 999999.99   28.95  41.162-114.888  10.6   42.649-114.084   1.2 K13A        Pn         0   1 wells6.1
 1.677  -0.81   0.12  24 99999999 999999.99  99999999 999999.99   29.73  41.152-114.892  10.5   39.473-114.908   1.9 P12A        Pn         0   1 wells6.1
 1.702  -0.73   0.21  40 99999999 999999.99  99999999 999999.99   30.18  41.163-114.887  10.5   41.780-112.775   1.6 HVU         Pn         0   1 wells6.1
 1.720  -0.56   0.15  29 99999999 999999.99  99999999 999999.99   30.59  41.145-114.897  10.4   39.553-115.754   1.8 P11A        Pn         0   1 wells6.1
```

The columns in a type 3 tomo file are formatted the same as the type 1 or type 2 file, but some of the columns are interpreted differently. The full travel time associated with an empirical path anomaly is calculated by adding the empirical path anomaly to the theoretical travel time between the hypocentroid and the appropriate station. In this case, the hypocentroid is not of the entire cluster, but the hypocentroid of those events which contribute to this station-phase. The entries for origin time and reading time are ignored. All the events in this cluster are well observed at far regional distances so the number of samples for these entries are much larger than for the type 2 example above. The event number (*Ev#*) is always 0 because the reference point

for each entry is the average location of the events contributing to each station-phase (the *EvLat*, *EvLon* and *H* columns).

~.ttou Files

~.*ttou* files are produced by the ttou command, which can be used up to six times in a run of **mloc**. Each instance of the command specifies a phase for which all corresponding data will be printed to a file named ~_*phasename*.ttou. These files are carried in a subdirectory of the cluster series directory named ~_*tt*.

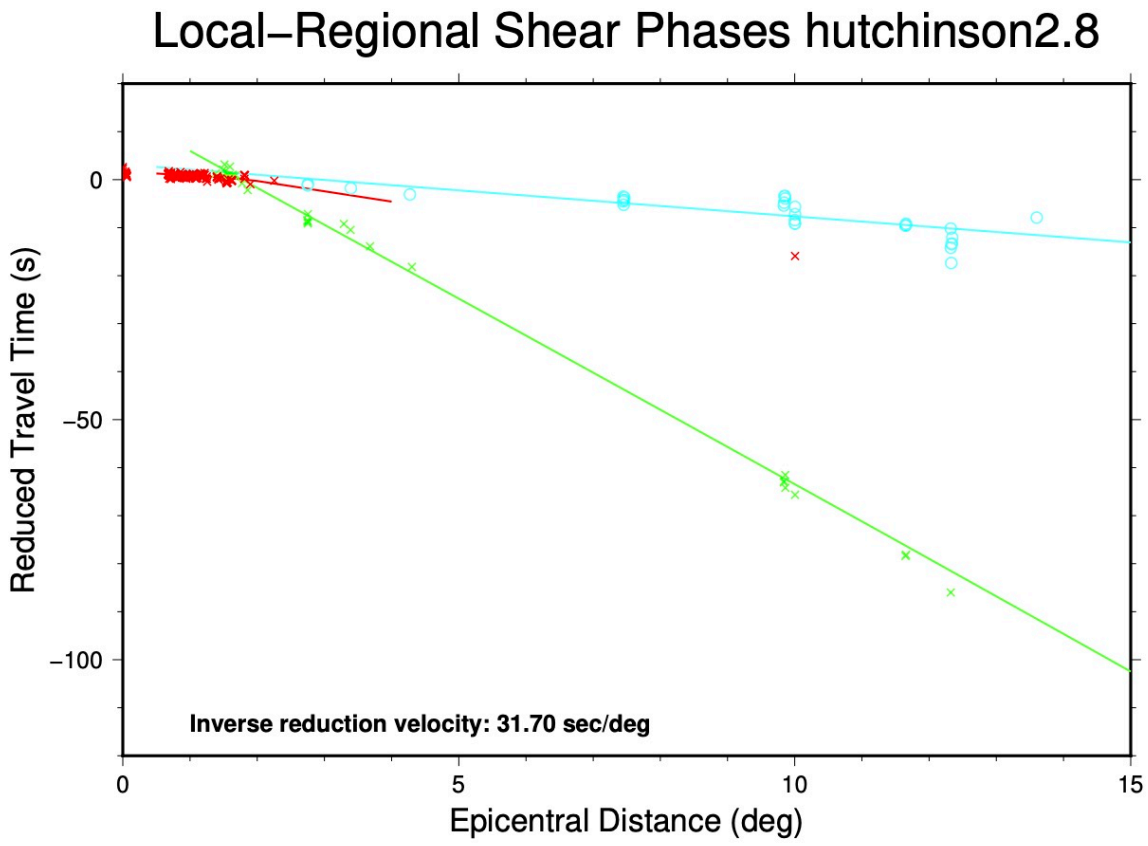
The most common use of this output file is to create a dataset of Lg arrivals that can be quickly analyzed to obtain a model for Lg arrivals (implemented by the lgtt command). Lg travel times can be well-predicted by a model that is linear in epicentral distance with a y-intercept close to zero. The default model in **mloc** is 31.500 sec/degree with intercept = 0. The format of ~.*ttou* files is designed for easy import into a data-modelling program, and the Lg travel-time model is estimated by a simple linear least-squares fit. Here is an example, from a cluster near Hutchinson, Kansas (a row of headers has been added for this display):

No	Event name	Depth	MNF	Station	Delta	Az	Obs. TT	Res.	Read Error	Flag
5	20170730.0013.45	5.0	56	TUL3	2.755	139	86.40	-1.16	0.22	
6	20180301.2026.59	5.0	91	ANMO	7.464	248	232.87	1.17	1.12	
6	20180301.2026.59	5.0	95	TXAR	9.859	210	308.62	3.57	0.86	
6	20180301.2026.59	5.0	98	PDAR	10.001	302	308.60	-0.78	1.14	
6	20180301.2026.59	5.0	99	TKL	11.649	97	363.34	3.50	0.10	x
6	20180301.2026.59	5.0	102	ULM	12.322	6	376.40	-4.03	2.21	
7	20180308.1048.21	5.0	122	TUL3	2.750	139	85.98	-1.40	0.22	
7	20180308.1048.21	5.0	130	BGNE	3.393	358	105.74	-1.33	4.00	
7	20180308.1048.21	5.0	146	OGNE	4.273	315	132.35	-1.66	4.00	
7	20180308.1048.21	5.0	170	ANMO	7.461	248	232.25	0.64	1.12	
7	20180308.1048.21	5.0	173	TXAR	9.853	210	308.95	4.10	0.86	
7	20180308.1048.21	5.0	175	PDAR	10.004	302	308.00	-1.48	1.14	
7	20180308.1048.21	5.0	178	TKL	11.649	97	359.70	-0.13	0.10	
7	20180308.1048.21	5.0	181	ULM	12.329	6	373.45	-7.19	2.21	
12	20180414.0246.34	5.0	136	TXAR	9.860	210	302.15	-2.91	0.86	x
12	20180414.0246.34	5.0	142	ULM	12.323	6	380.46	-0.01	2.21	
25	20190816.1259.10	5.0	245	ANMO	7.449	248	231.58	0.34	1.12	
25	20190816.1259.10	5.0	252	TXAR	9.838	210	306.46	2.07	0.86	
25	20190816.1259.10	5.0	255	PDAR	10.005	302	309.95	0.45	1.14	
25	20190816.1259.10	5.0	262	TKL	11.655	97	360.26	0.25	0.10	
25	20190816.1259.10	5.0	266	ULM	12.343	6	379.19	-1.88	2.21	
25	20190816.1259.10	5.0	270	ELK	13.602	287	423.27	3.64	4.00	
36	20190818.0845.29	5.0	237	ANMO	7.449	248	232.60	1.34	1.12	
36	20190818.0845.29	5.0	240	TXAR	9.841	210	307.15	2.68	0.86	
36	20190818.0845.29	5.0	243	PDAR	10.002	302	307.92	-1.48	1.14	
36	20190818.0845.29	5.0	246	TKL	11.656	97	359.97	-0.09	0.10	
36	20190818.0845.29	5.0	249	ULM	12.339	6	377.83	-3.13	2.21	
37	20190901.1321.06	4.0	50	ANMO	7.455	248	231.09	-0.35	1.12	
37	20190901.1321.06	4.0	53	TXAR	9.846	210	308.68	4.05	0.86	
37	20190901.1321.06	4.0	55	PDAR	10.005	302	311.50	2.00	1.14	
37	20190901.1321.06	4.0	57	TKL	11.651	97	359.79	-0.12	0.10	
37	20190901.1321.06	4.0	60	ULM	12.335	6	377.59	-3.26	2.21	

The Lg travel-time model derived from fitting a line to the TT vs. Delta columns is given in subsequent command files as:

```
lgtt 3.2 30.615 2.5
```

The fit of this model to the Lg data is shown by the turquoise symbols and line in the "Local-Regional Shear Phases" summary plot, one of the summary plots controlled by the pltt command:



~.ttsprd File

A *~.ttsprd* file is produced by every run of **mloc**. It is written by the module *mloc_ttsprd.f90*. Every phase that is present in the dataset at least twice is processed to obtain the robust estimate of its spread and the baseline offset relative to the theoretical travel-time model. These values can be read into subsequent runs with the command tfil. These values are used in the windowing algorithm (command wind) to provide a procedure for rejecting gross outliers that is more sophisticated than a simple single-limit criterion. That is, the width of the window for each phase is based on the observed spread of arrivals for that phase and the window is offset from the theoretical travel-time model by the observed baseline offset to avoid bias.

The example is the full *~.ttsprd* file for a cluster:

```
P          13088     1.592     2.953
Pn          5423     2.179    -0.591
```

```
S            816     4.513     4.471
PcP          173     2.713     2.716
PcS           26     3.016     3.038
PnS           35     2.900    -0.493
SS            88     5.300     2.092
PP           211     3.962     3.770
pP-P          62     1.227     0.246
SKiKP          8     2.677     2.927
SKKSac         9     1.293     0.968
ScS           28     3.592     3.265
SKSac         15     2.789     2.090
PS             7     7.499     5.060
Sn           821     2.637    -1.075
Sg           464     1.355     0.032
Pg          1575     1.346     0.129
Lg           410     3.294     2.857
pP            19     1.298    -1.911
PKiKP        105     1.595     2.329
Pdiff        168     1.062     3.366
PKKPdf        11     3.327     1.861
PKKPbc        19     1.427     3.760
PKPdf         18     1.689     4.479
PKPab          6     1.756    -2.155
sP-P          28     1.220    -0.301
sP            22     1.220    -0.499
ScP           12     2.873     2.034
sS            21     5.003     4.181
P'P'df         8     6.270    -2.066
PKKPab        15     4.855     0.831
```

The columns are:

- Phase
- Number of samples
- Spread
- Baseline offset

Notice that the baseline offset of the P phase is significant: almost +3 seconds relative to the ak135 global mode. On the other hand, the phases Pg and Sg have quite small baseline offsets, because the hypocentroid of the cluster has been determined mainly with these phases (direct calibration), using a local velocity model that has been refined to match the observed data. In calibrated relocation analyses it is very common to observe offsets of this magnitude for teleseismic branches.

~.xdat File

This page deals with the *~.xdat* file that is read as input to the xdat utility code, one of several utility codes that assist in the cleaning process of an **mloc** relocation analysis.

Most arrival time datasets contain readings that cannot be usefully employed in a relocation analysis, for a variety of reasons:

- The pick was made on a noise signal rather than a true phase arrival.
- The pick was made on a phase arrival from a different event.
- The station coordinates are grossly wrong.
- The phase was mis-identified.
- The phase name is not one for which **mloc** can calculate a theoretical arrival time.
- Coordinates cannot be obtained for the reported station code.

The first four cases result in a reading having a large residual against the theoretical arrival time (the others are dealt with elsewhere). All earthquake location codes make some provision to deal with gross outliers. This can be extremely simple, e.g., any reading with a residual greater than N seconds is dropped. **mloc** employs a more complex algorithm to deal with gross outliers, driven by the wind command. Readings that fail the windowing criteria are dropped from the current run of **mloc** (listed in the "BAD DATA" section of the *~.phase_data* output file with "PRES" in the column "Why Bad"), but under some circumstances they can fall back into a subsequent run as gross outliers unless they are flagged. Those readings are also listed in the *~.xdat* output file.

The user seldom needs to look at the *~.xdat* file. The only question is whether or not it is appropriate to run **xdat** after a given run of **mloc**. It is usually desirable to run **xdat** at least once early in the analysis, as there will normally be a large number of entries in the *~.xdat* file, but in later runs there will be few entries, and often the file will be empty.

The format of the ~.xdat file is kept simple, as it is only meant as input for **xdat**, i.e., it really only needs to carry the file name and line number that will be flagged. The format is documented in the code module *mlocout_phase_data.f90*. Here is an example:

```
  7  COP   ISC      XS          71 x 19670517.0428.50.mnf
 15  PYA   ISC      Sn          15 x 19761128.1113.23.mnf
 39  KER   ISC      Pg          28 x 19990219.1800.10.mnf
 41  MLTT  ISC      Sn          35 x 20020228.0046.31.mnf
 41  ZALF  ISC      Sn          50 x 20020228.0046.31.mnf
 41  SALA  ISC      Sn          64 x 20020228.0046.31.mnf
 43  ASAO  IIEES    Sn          11 x 20030811.2012.06.mnf
 55  VRNZ  ISC      Pn          36 x 20060729.0151.11.mnf
 60  BHD   ISC      Sg          89 x 20090310.1313.02.mnf
 61  IBDR  ISC      Sn         179 x 20090705.2348.25.mnf
 66  HOPEN ISC      IVmB_BB   1977 x 20111023.1041.23.mnf
 66  MTDJ  ISC      Pdif      3943 x 20111023.1041.23.mnf
 74  COAL  ISC      P          312 x 20111029.2224.24.mnf
 75  RAYN  ISC      Sn         436 x 20111106.0243.14.mnf
103  KIEV  ISC                  87 x 20200318.2307.22.mnf
```

The phase name listed here is the one that was read in from the event file, not the result of any phase re-identification done by **mloc**. In the example the reading from station HOPEN from event 66 should have been caught by **mloc's** algorithm to identify readings that are not legitimate phase arrivals (and flagged with "p"), but in this case an amplitude measurement slipped through.

Plots

Because of the quantity and complexity of the output from **mloc**, graphic display of information has proven to be of extreme importance in evaluating relocation analyses. All image files (plots) in **mloc** are created by the Generic Mapping Tools (GMT). If **mloc** is run on a computer without an installation of GMT, plotting can be turned off with the command gmtp, but it is a severe handicap.

At the end of a run **mloc** creates the requested GMT scripts and then executes the scripts to produce PostScript files, which are finally converted to PDF. The main GMT scripts are saved in a folder for each run named *~_gmt_scripts*, so that they can be edited and re-run (e.g., for publication purposes) if desired. For display on this website the example plots have been converted to JPEG format.

A variety of other plots are available, in most cases requested by a command. A suite of such plots that are commonly utilized is described in the Summary Plots section. These include the base plot, the direct calibration raypath plot and a set of travel-time plots controlled by the pltt command.

Other important types of plots are:

- Focal Depth Histogram
- Empirical Path Anomalies
- Combined Plots for GCCEL
- Single Event Relative Depth Phases

Summary Plots

This section describes the standard plot types that can be generated by **mloc**. Most plot types are optional (controlled by the command pltt and other commands), but one type of plot (the base plot) is always created and another half dozen or so are routinely used. This section also describes some plots that are variants of one of the basic types. Some additional plot types are referenced here.

Contents

- tt4: Near-Source
- tt5: Local
- tt6: Local-regional
- tt7: Local-regional S
- tt8: Relative teleseismic depth phases
- tt9: S-P
- Base Plot Variants
- Ellipses Only
- Seismicity
- Selected Events
- tt5 Variants
- tt5e: Single event, local distance
- tt5s: Single station, local distance

Base Plot (_base.pdf)

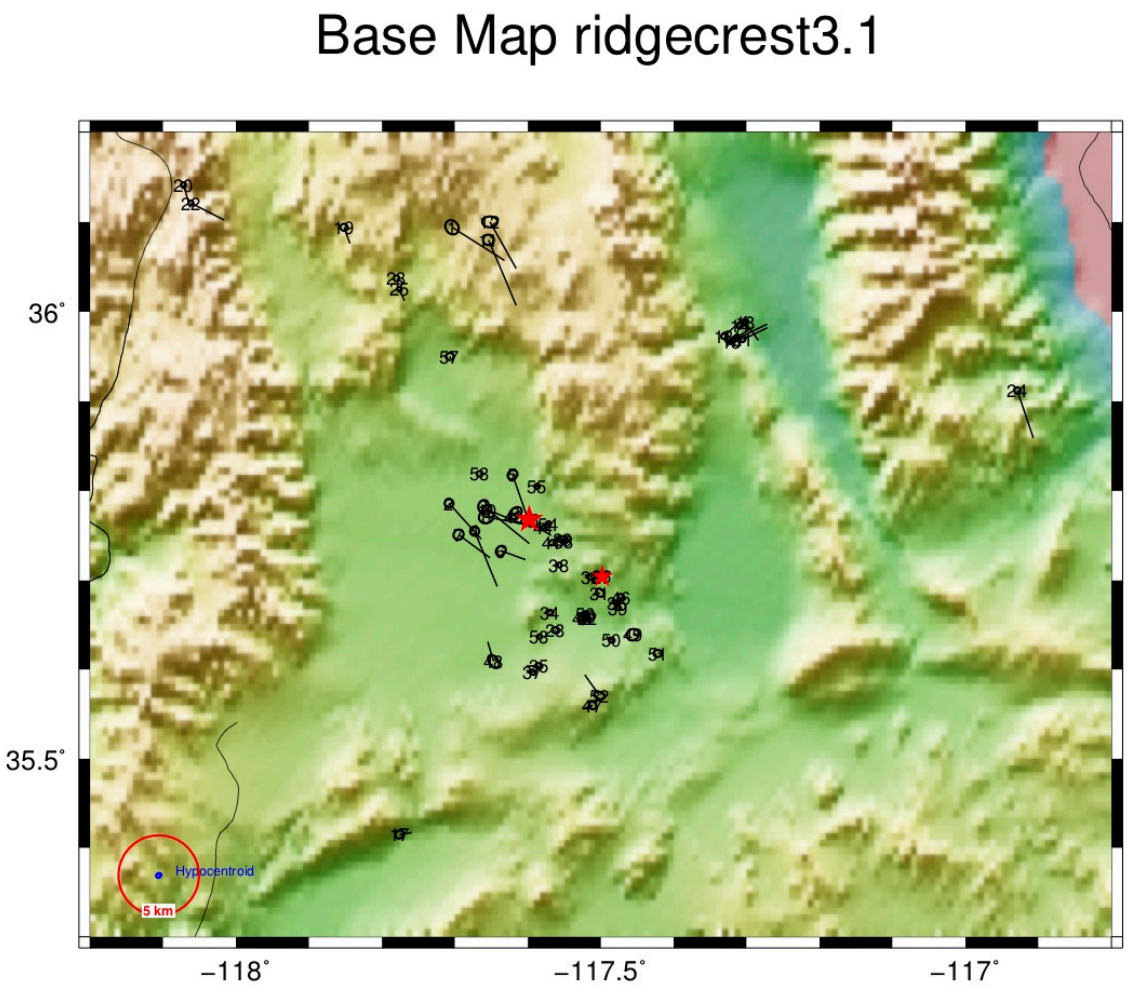
One type of plot is always produced by a successful run of **mloc**, the base plot, which has the file name suffix "_base.pdf". This is a map of the epicenters with confidence ellipses for the cluster vectors (relative locations). Typically each epicenter also carries vectors showing the change in location from the starting location for the current run of **mloc** (in green) and the change from the location given in the event data file (in black), which may or may not be the same position. Each epicenter is numbered. The base plot always carries a red circle in the lower left corner with a radius of 5 km, for scale. Other features of the plot, such as the plotting of topography, depend on the action of other commands and the nature of the relocation.

It is important to note that the confidence ellipses shown in the base plot are not the full uncertainty in epicentral location, whether the relocation is calibrated or not. The full uncertainty for individual events results from the addition (as covariance matrices) of the uncertainty in the hypocentroid to that of the cluster vectors for individual events.

Even in an uncalibrated relocation the cluster vectors have meaning in terms of relative locations, but in that case only very broad bounds can be placed on the uncertainty of the hypocentroid. Adding a large, arbitrary uncertainty to the cluster vectors to represent the uncertainty of the hypocentroid would only degrade the resolution of relative locations.

If, on the other hand, the relocation is of the calibrated type and a suitably small uncertainty (1-2 km is typically sought) has been obtained, the addition of the hypocentroid's uncertainty to that of the cluster vectors does not make much difference. To avoid confusion over the relative contributions of hypocentroid and cluster vector uncertainties I have found it to be more useful to

show only the cluster vector uncertainty in the base plot; the full uncertainty in location is carried in other output files.

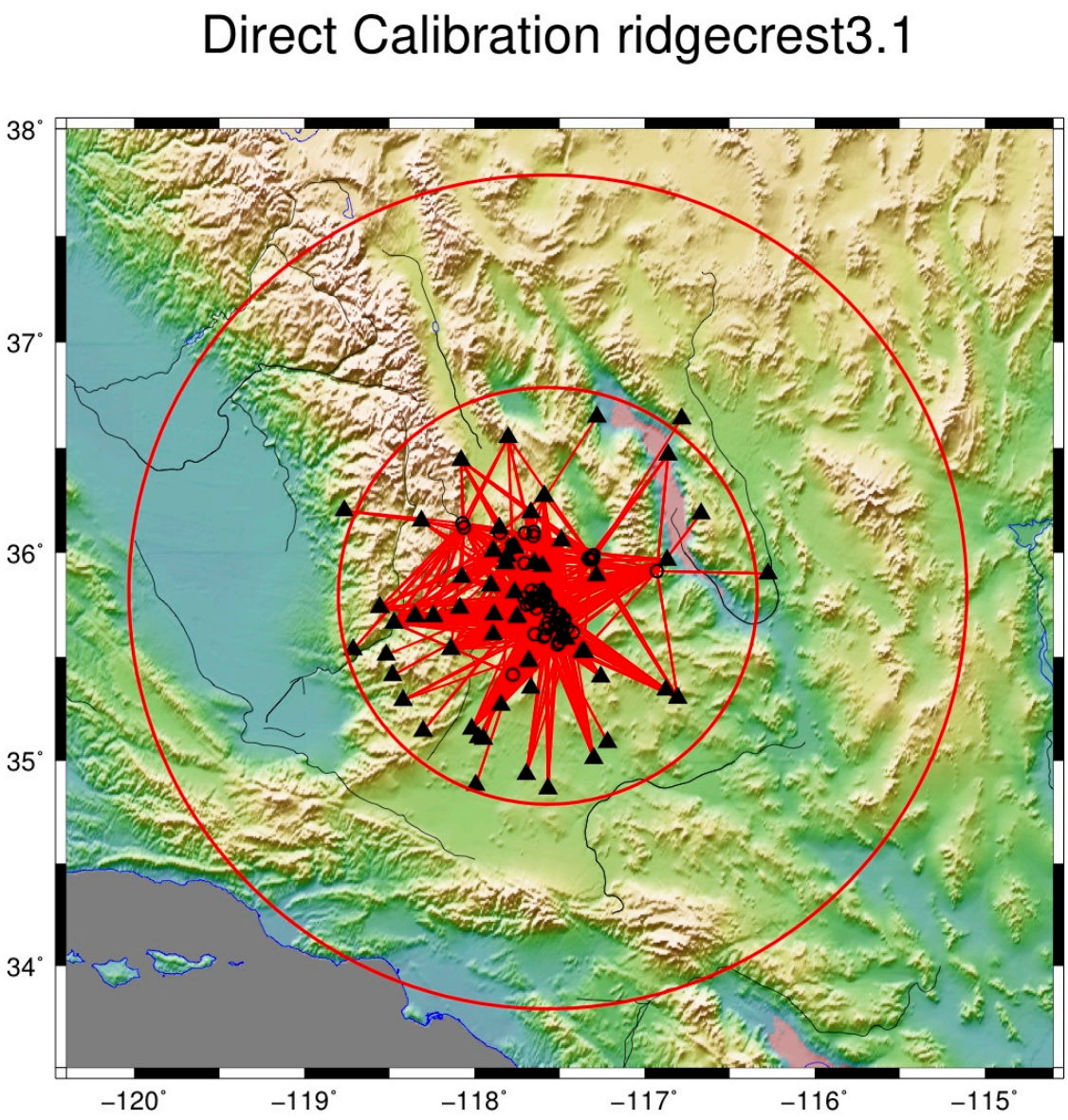


Direct Calibration Raypath Plot (_dcal.pdf)

In **mloc**, calibrated locations can be estimated in two ways, known as “direct” and “indirect” calibration. The great majority of calibrated relocations are done using the direct calibration method, in which the hypocentroid is estimated using only readings from stations at close range, usually less than the distance of the Pg/Pn crossover in the source region, typically around 100-150 km. When direct calibration is used, **mloc** always creates a map figure to show the raypaths used for the hypocentroid, with a file name suffix of “_dcal.pdf”.

In this plot, epicenters are shown as equal-sized open circles, stations as filled black triangles, with red vectors for the raypaths. If there are S-P readings the associated stations are shown as

open diamonds. Two red circles for scale are centered on the cluster's hypocentroid, with radii of 1.0° and 2.0°. Plotting of topography is controlled by the command *dem1*.



Plots Controlled by Command pltt

The command pltt takes one or more integer arguments to control the creation of nine different types of travel time plots, each of which carries a "ttN" nomenclature:

- 1 = tt1, Summary travel time plot, 0-180° (default if no arguments are given)

- 2 = tt2, Entire teleseismic P branch, reduced to the theoretical P time
- 3 = tt3, Around the PKP caustic
- 4 = tt4, Near-source data (range of the hypocentroid data)
- 5 = tt5, local data, 0-4°
- 6 = tt6, local-regional data, 0-30°, reduced velocity
- 7 = tt7, Sg, Sb, Sn, and Lg, 0-15°, reduced velocity
- 8 = tt8, Relative depth phases (pP-P and sP-P)
- 9 = tt9, S-P times

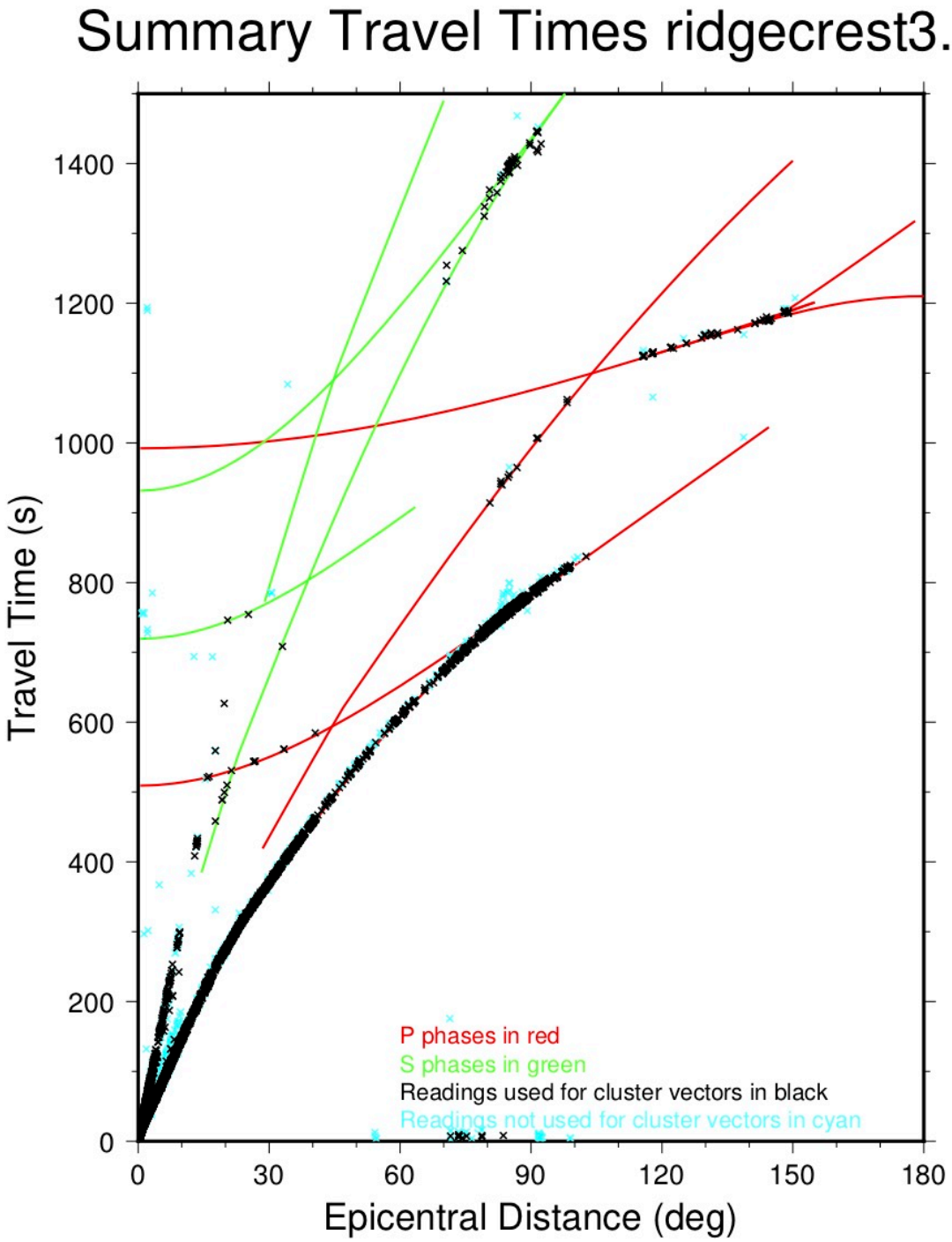
Summary Travel-time Plot (tt1)

This plot displays arrival time data and theoretical TT curves out to 180° and to a travel time of 1500 s. Theoretical curves are from ak135, with depth set to that of the hypocentroid. Curves are yellow for P, PP, Pdiff and core phases; cyan for S, SS, ScP and ScS.

Individual arrival times are plotted in several colors, depending on their use in the relocation:

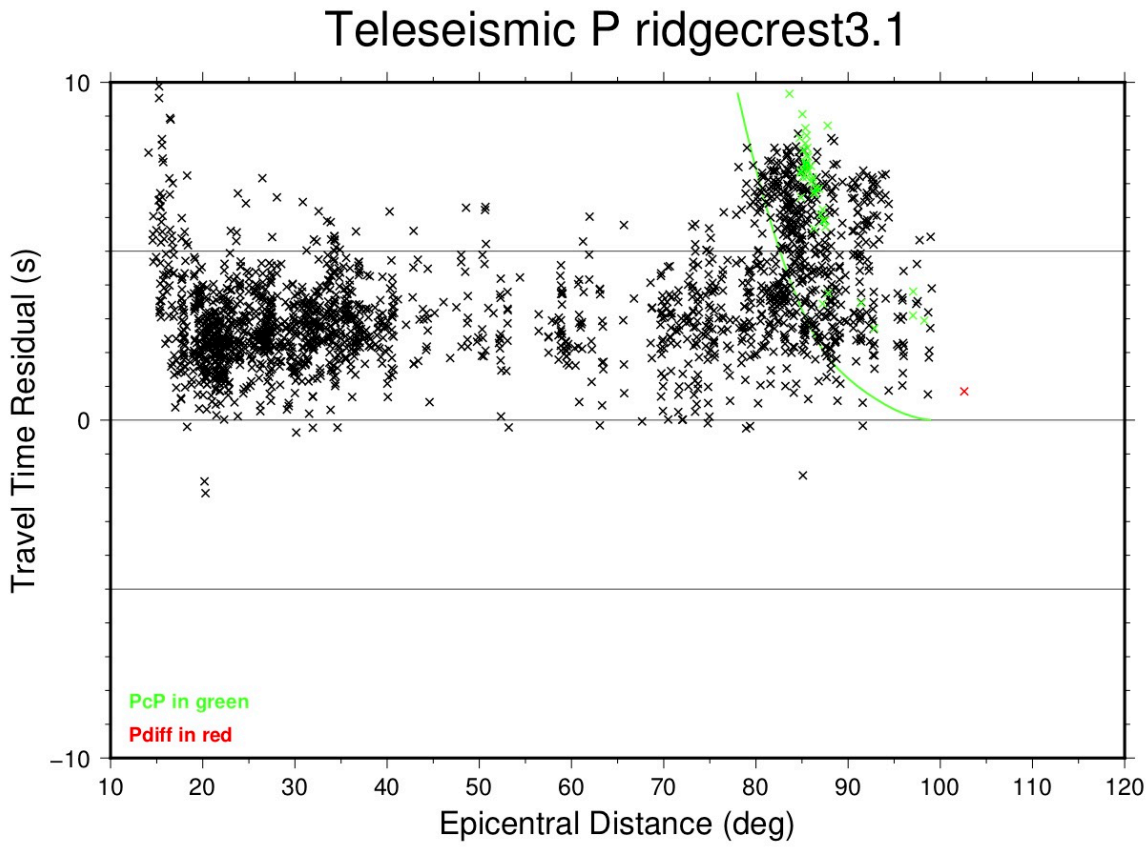
- Travel times used both for hypocentroid and cluster vectors in red
- Travel times used for hypocentroid, but not for cluster vectors in green
- Travel times used for cluster vectors, but not for hypocentroid in black
- Travel times used for neither cluster vectors nor hypocentroid in blue

Because of the scale of the plot, the red and green symbols are usually barely distinguishable.



Teleseismic P Plot (tt2)

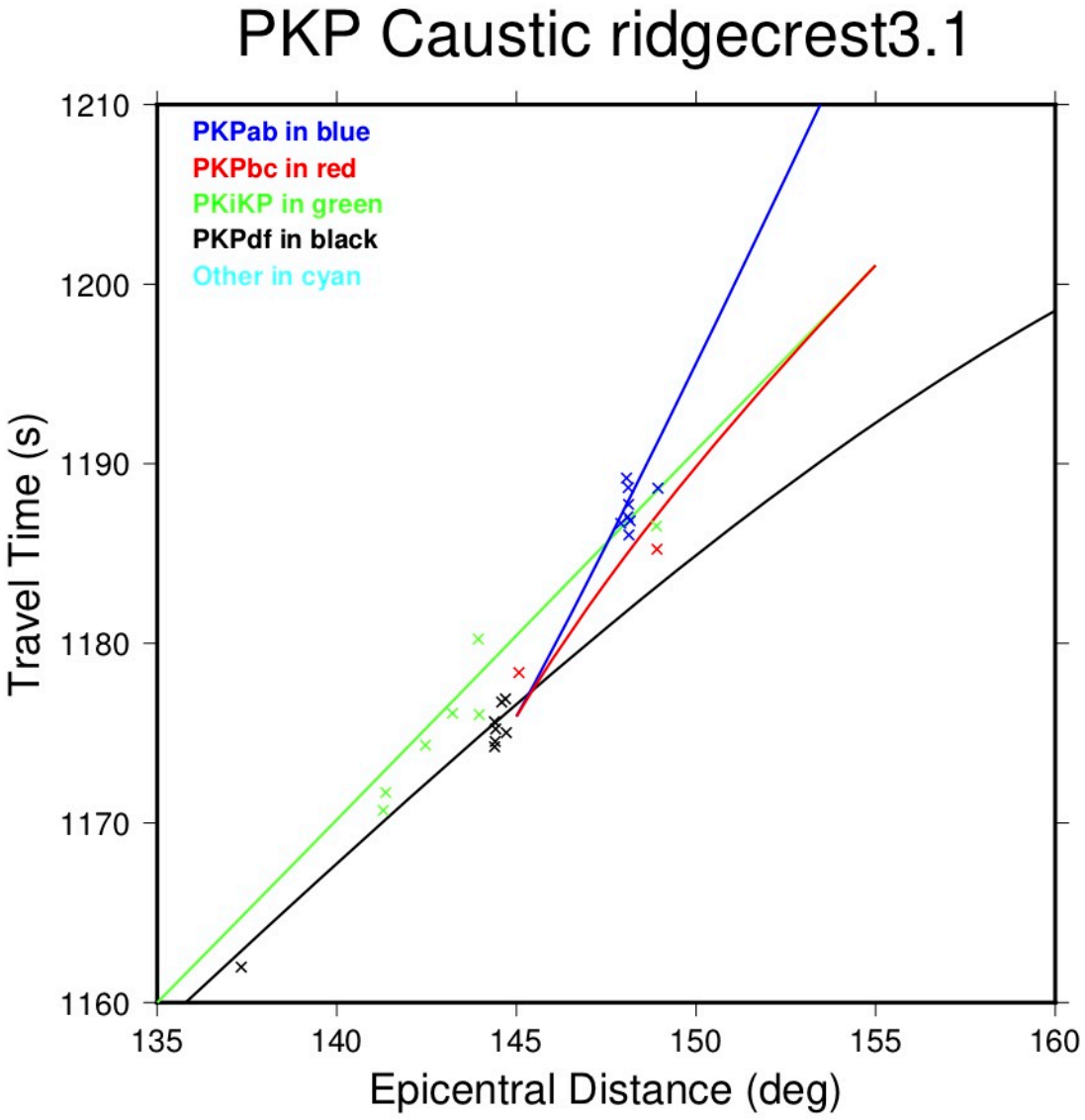
This plot displays P phase arrivals from 10-120°, reduced to the theoretical time from ak135 (for the depth of the associated event). A green line indicates the theoretical arrival time of PcP, with which there can be confusion near a distance of 80-85°. Pdiff arrivals are plotted in red.



PKP Caustic Plot (tt3)

This plot shows arrivals near the PKP caustic between 135-160°. Theoretical curves and symbols for arrivals are color-coded:

- PKPdf in black
- PKiKP in green
- PKPab in blue
- PKPbc in red
- Anything else in cyan



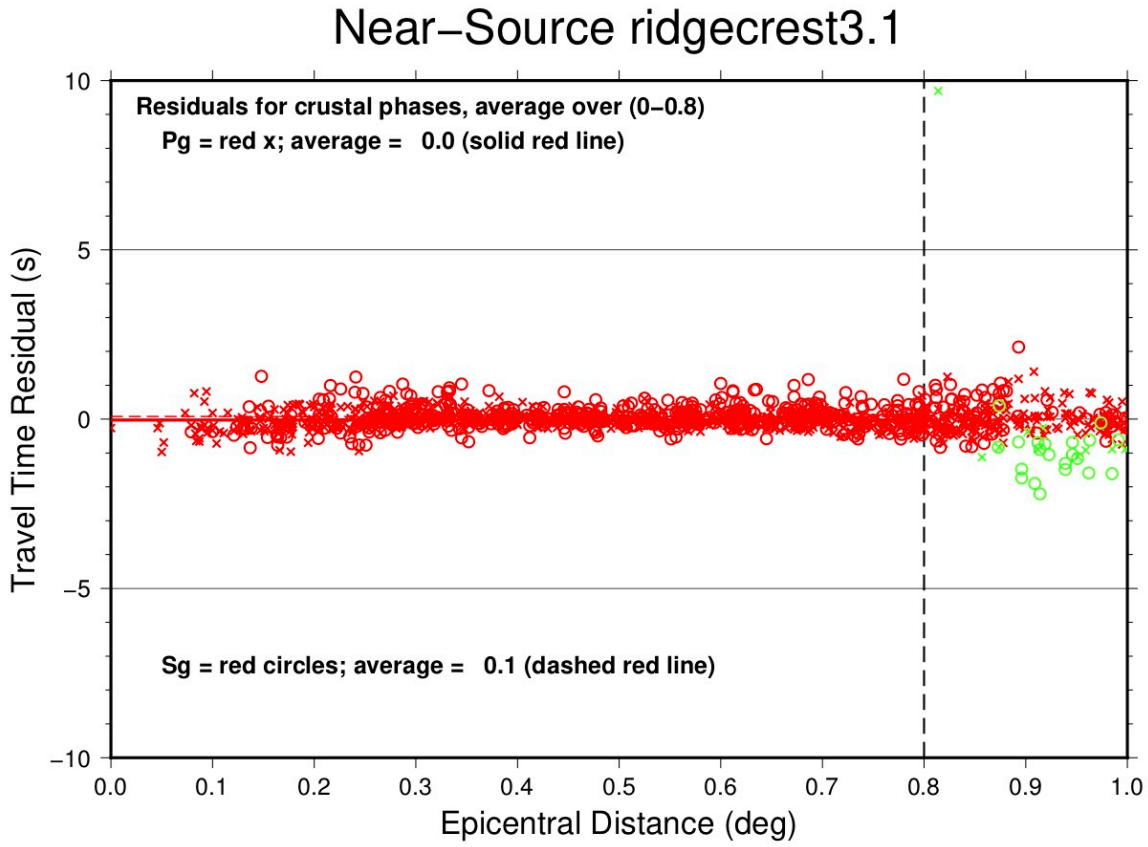
Near-Source Plot (tt4)

This plot displays the residual of each arrival in the distance range used for estimating the hypocentroid (indicated by a vertical dashed line). Symbols for different phases are:

- Pg in red crosses
- Pb in blue crosses
- Pn in green crosses

- Sg in red circles
- Sb in blue circles
- Sn in green circles

For each phase, the average residual (over the distance range used for the hypocentroid) is calculated and listed.



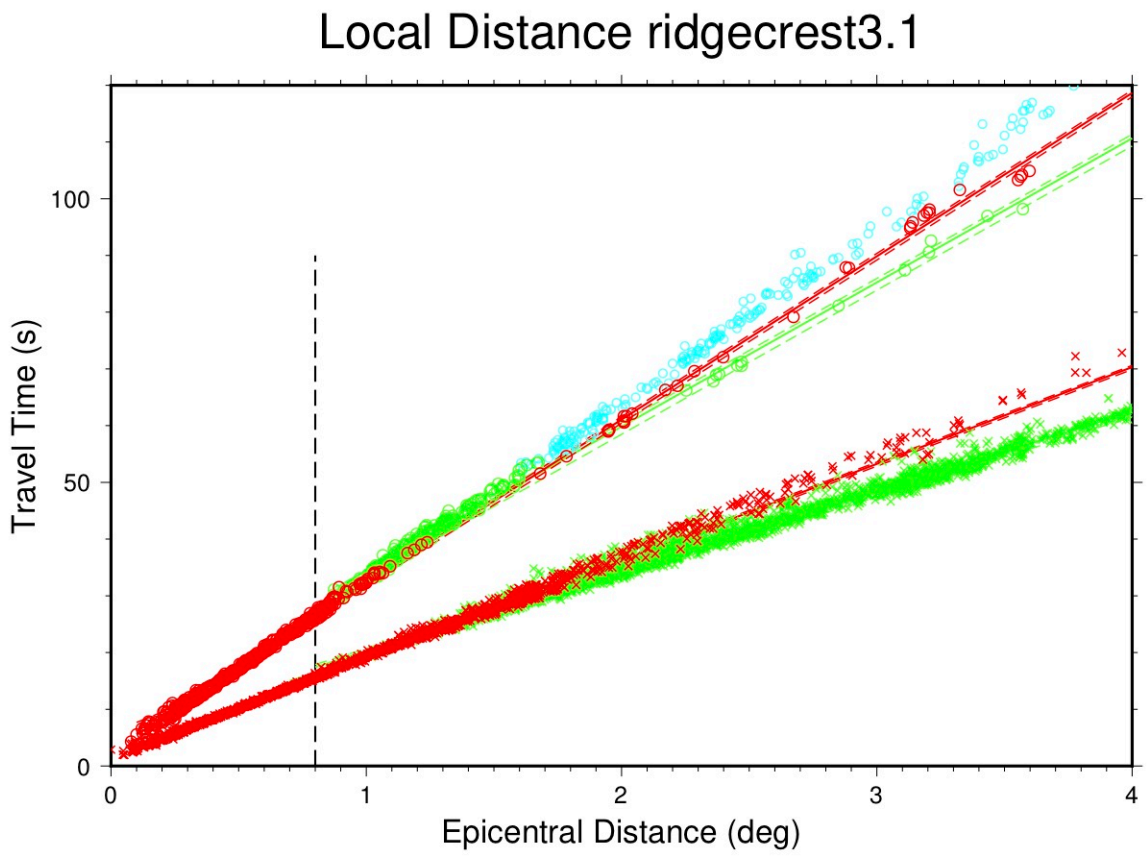
Local Distance Plot (tt5)

This plot displays the travel-times of data over the distance range 0-4°. In most cases the theoretical travel times are based on a custom flat-layered model developed for this particular cluster to match the observed arrival time data. Travel times for Lg are calculated separately within **mloc**, using either a default travel time or one specified by the user, based on the fit to observed data. Otherwise the theoretical travel times would be based on the ak135 global model. Symbols for different phases are:

- Pg in red crosses
- Pb in blue crosses

- Pn in green crosses
- Sg in red circles
- Sb in blue circles
- Sn in green circles
- Anything else (mainly Lg) in cyan circles

A vertical dashed line is drawn at the distance limit used to estimate the hypocentroid.



Local-Regional Plot (tt6)

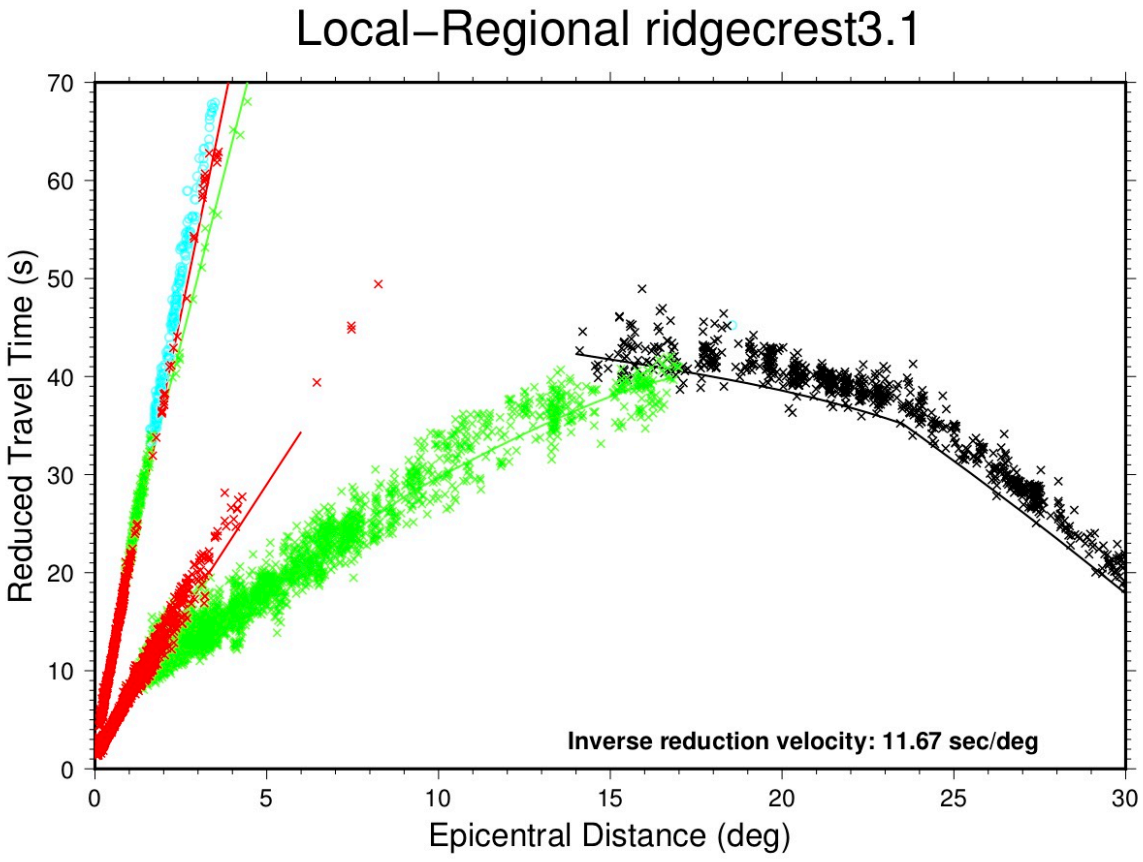
This plot displays the travel times of data out to 30°, in a reduced velocity plot for legibility. The reduction velocity (11.67 sec/deg in inverse form, or 350 seconds at 30°) has no physical meaning. Symbols for different phases are:

- Pg and Sg in red crosses
- Pb and Sb in blue crosses

- Pn and Sn in green crosses
- P in black crosses
- Everything else (mainly Lg) in cyan circles

In most cases the theoretical travel times for local and regional phases are calculated from a custom flat-layered model derived for the particular cluster to fit the observed data. Travel times for Lg are calculated separately within **mloc**, using either a default travel time or one specified by the user, based on the fit to observed data.

The travel time curve for teleseismic P is taken from the ak135 global model. Phase identification at the Pn-P crossover is usually less than optimal, but in most cases it is not worth the effort to manually correct the results of the automatic phase identification algorithm in this distance range, where only relative travel times are used in the relocation.

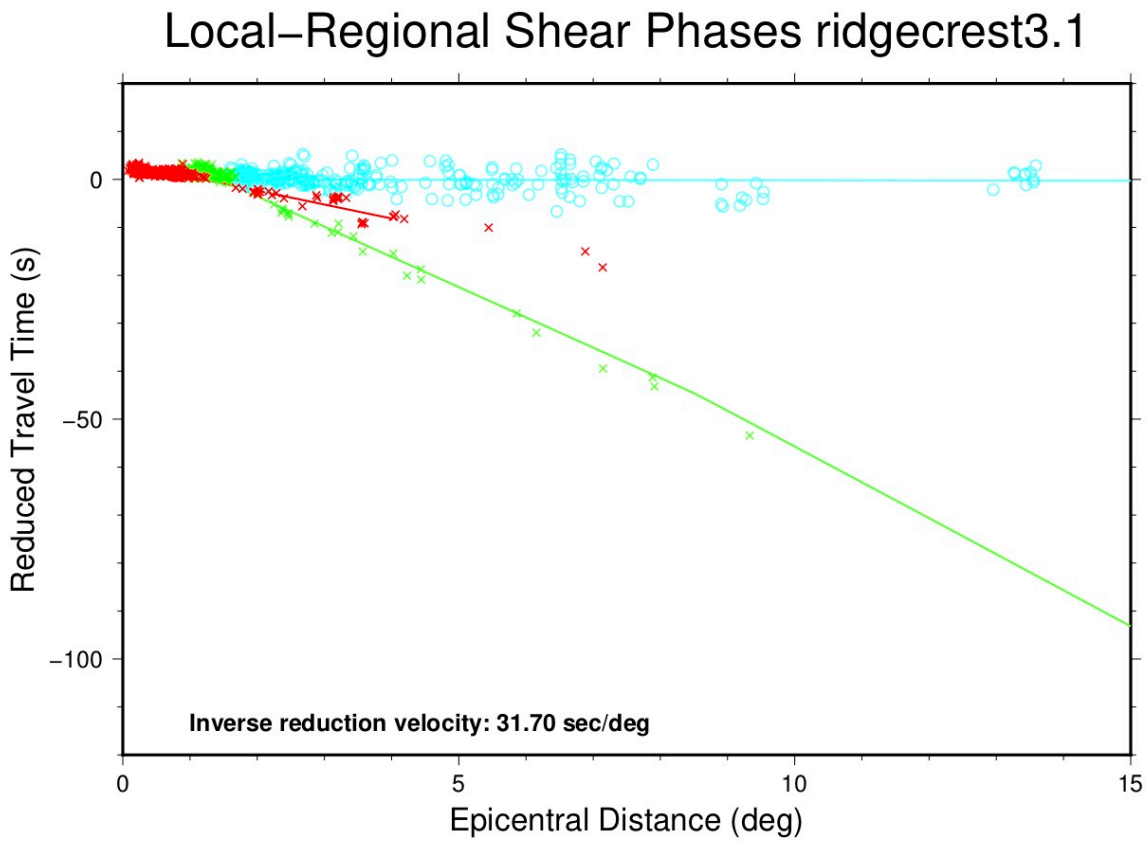


Local-Regional Shear Phases Plot (tt7)

This plot displays the travel times of shear phases out to 15° in a reduced-velocity plot (31.7 sec/deg, ~3.5 km/s). Symbols for different phases are:

- Sg in red crosses
- Sb in blue crosses
- Sn in green crosses
- Lg in cyan circles

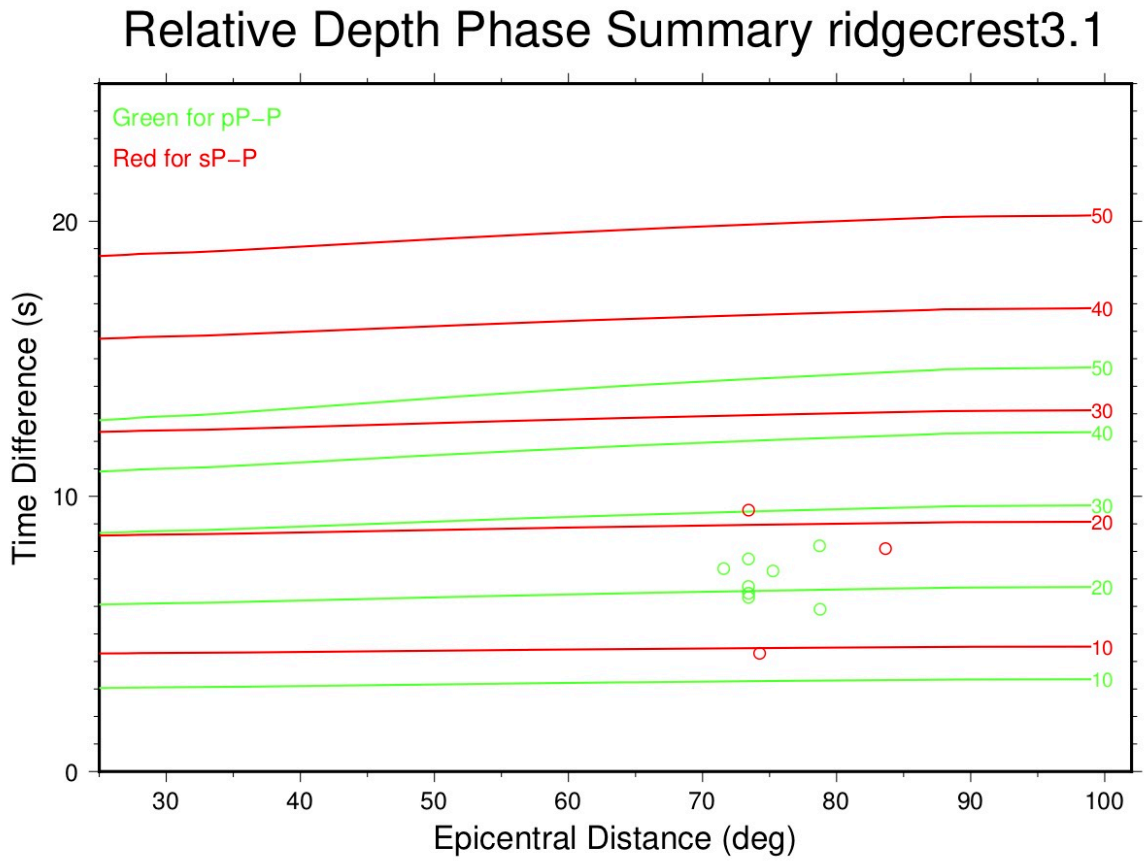
In most cases the theoretical travel times for these phases are calculated from a custom flat-layered model derived for the particular cluster to fit the observed data. Travel times for Lg are calculated separately within **mloc**, using either a default travel time or one specified by the user, based on the fit to observed data. Otherwise the ak135 global model would be used.



Relative Depth Phase Summary Plot (tt8)

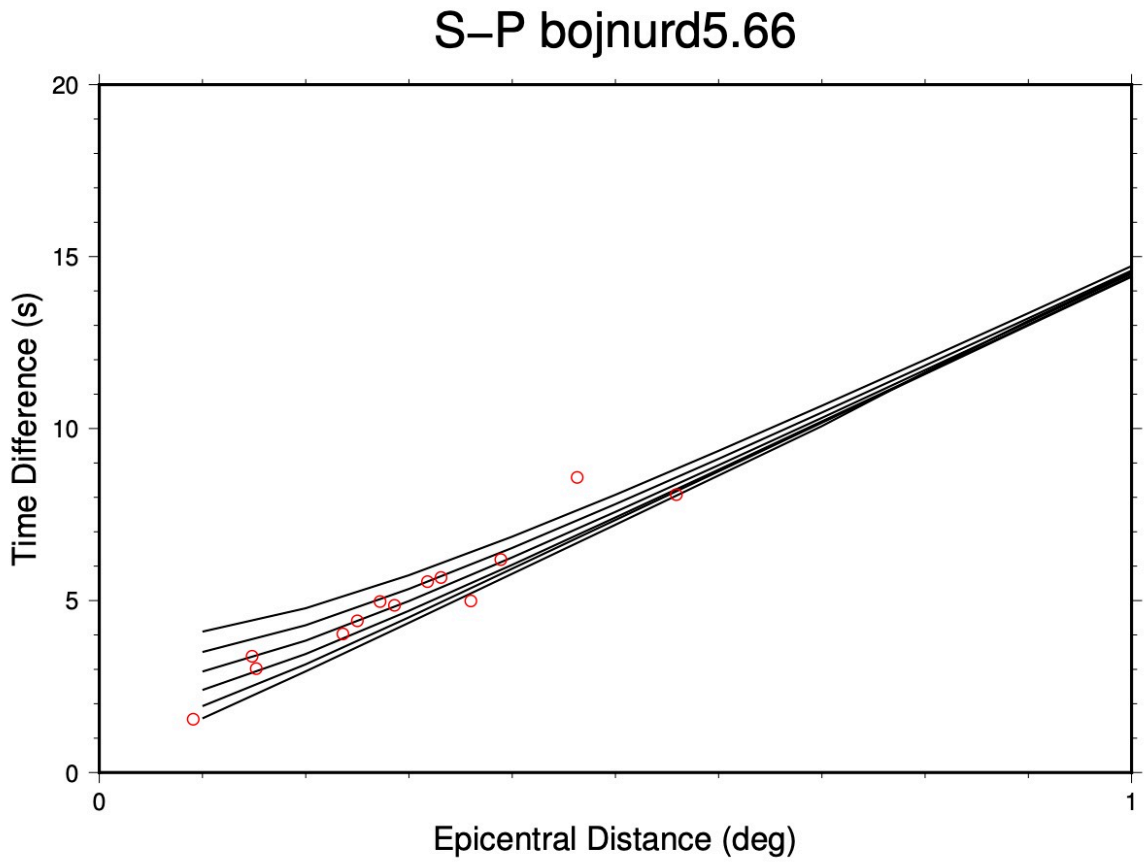
This plot displays all relative depth phase (pP-P, sP-P) observations for all events in a cluster to assess the likely range of focal depths. Because of the baseline offset that is normally observed for the travel times of teleseismic P phases (and depth phases) in calibrated clusters, it is not feasible to use depth phase arrival times by themselves to constrain depth. In **mloc** these data are converted to relative depth phase data, e.g., pP-P times.

The plot displays the value of relative depth phase observations as a function of epicentral distance. Theoretical curves (green for pP-P, red for sP-P) are drawn over the distance range at a range of depths to assist in evaluating the range of depths for the events in the cluster.



S-P Plot (tt9)

These plots display observations of S-P times. S-P data must be entered explicitly in an event's MNF file; it is normally only used when P and S arrivals are available from stations for which the timing systems are known to be uncalibrated or suspected of being miscalibrated. The plot displays S-P time on the vertical axis and epicentral distance (to 1.0°) on the horizontal axis. Theoretical curves for S-P are shown for focal depths of 5, 10, 15, 20, 25 and 30 km. Readings are shown as red open circles.

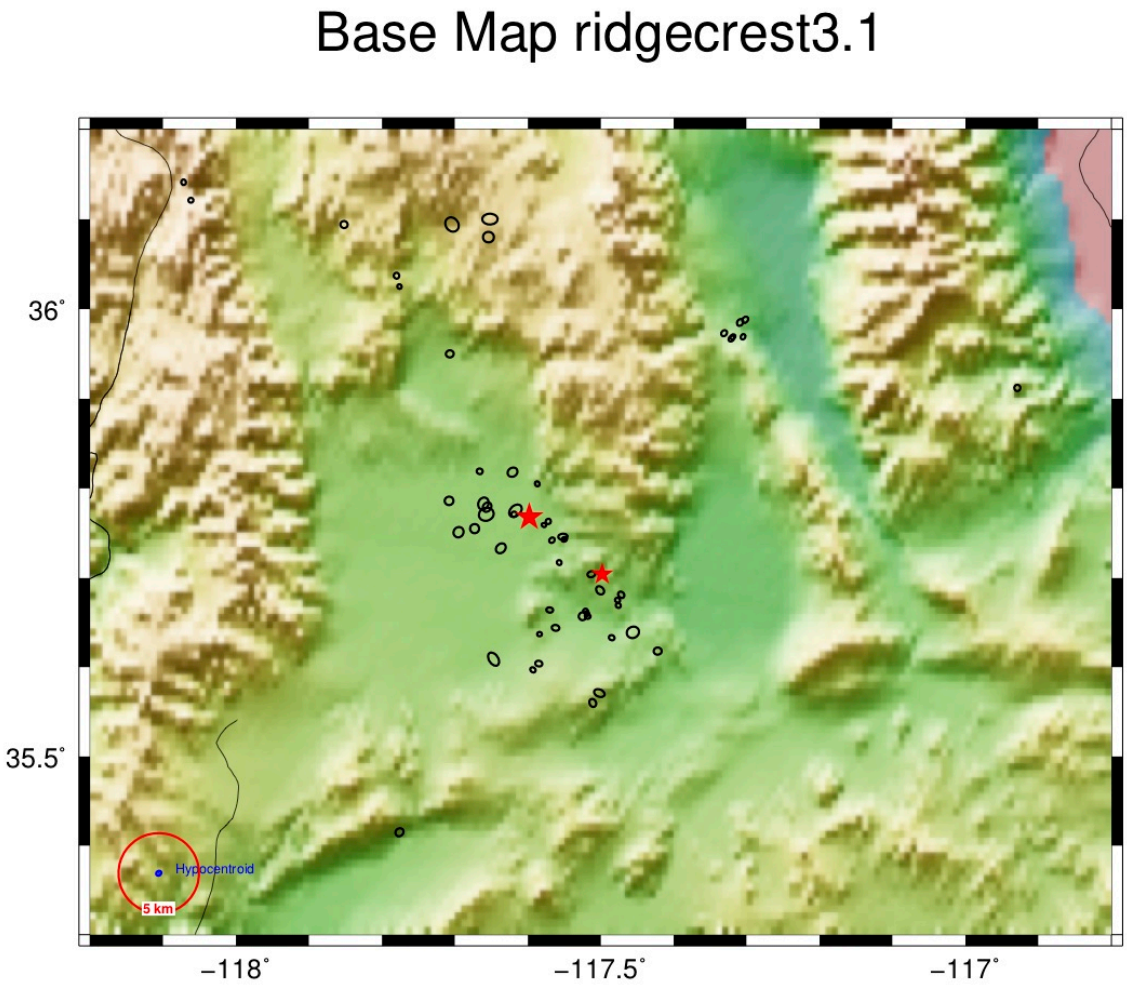


Other Plots

Several other plots can be made from **mloc** with specific commands.

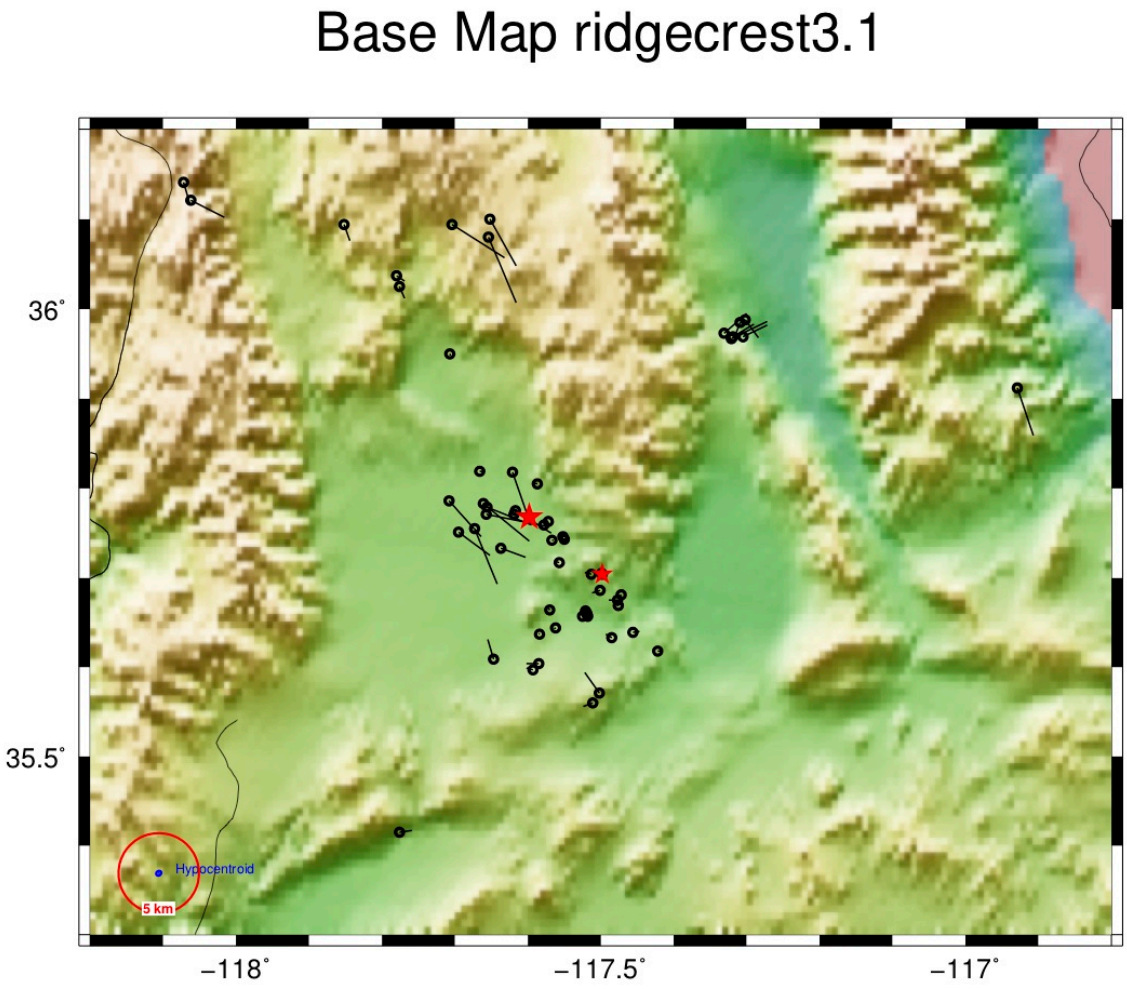
Confidence Ellipse Plot (command eplt)

This plot is based on the base plot but shows only the confidence ellipses, i.e., no event numbers or relocation vectors.



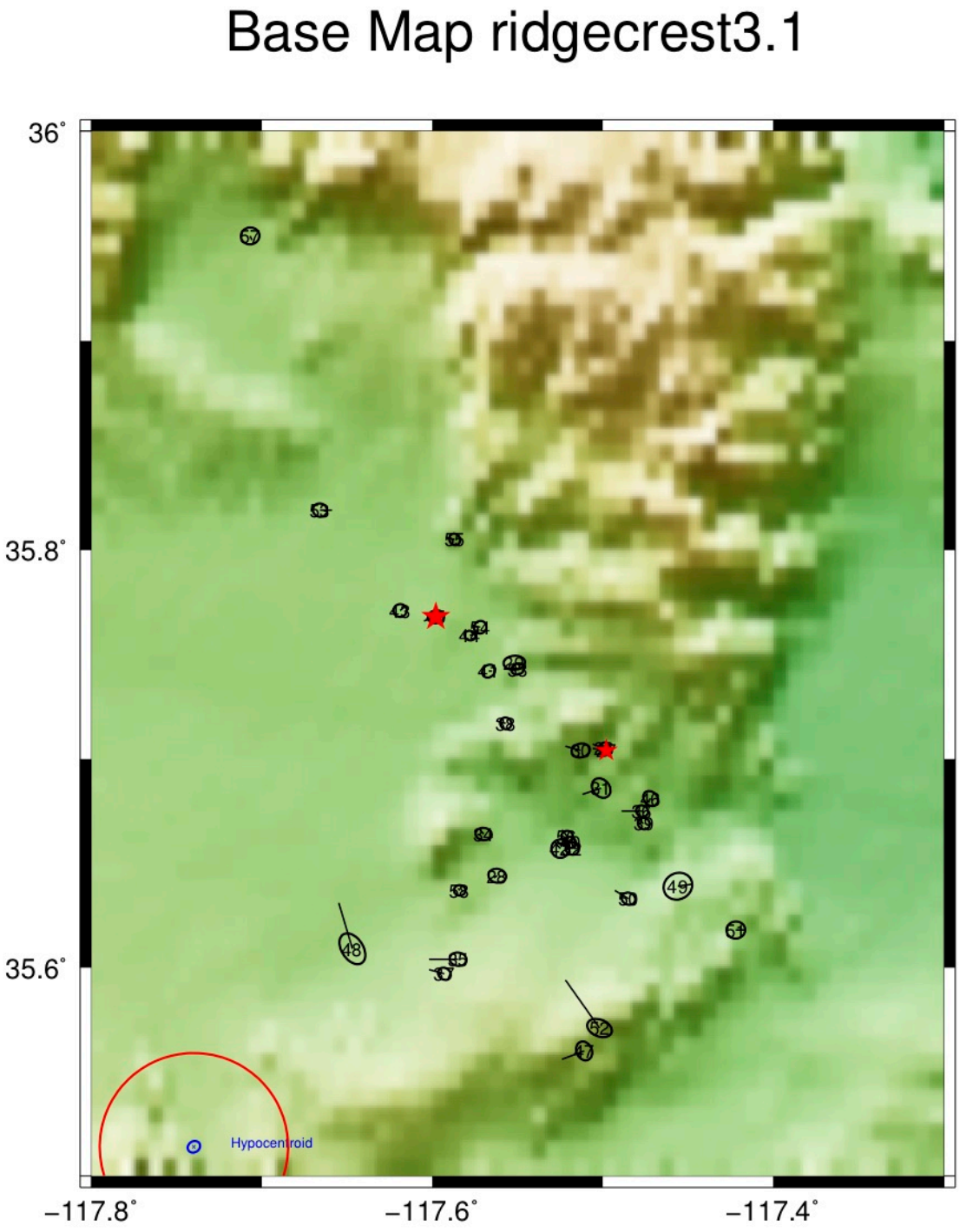
Seismicity Plot (command splt)

This plot is similar to the base plot but shows locations with same-sized symbols and no indication of location uncertainty or relocation vectors. I do not find this plot very useful but it does allow for an easier comparison with the typical presentation of earthquake locations.



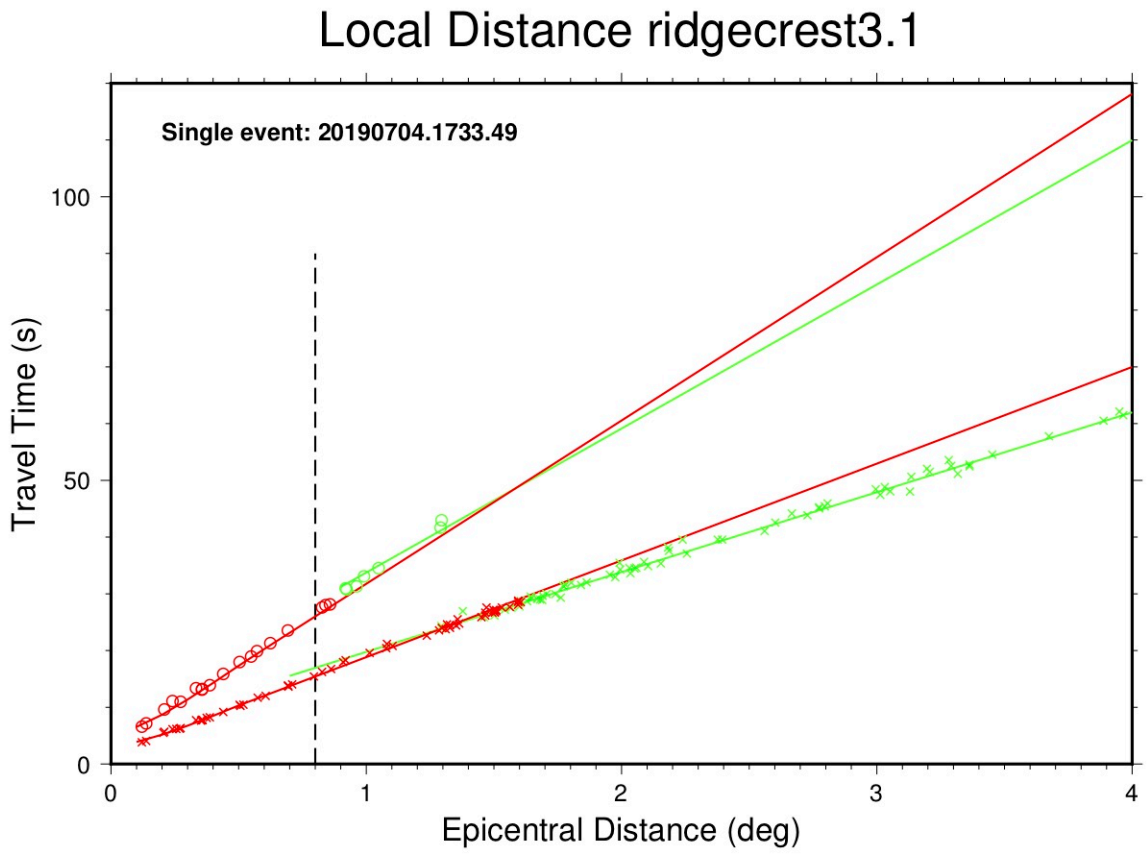
Selected Event Plot (command plot)

Using the command *plot* the user can select individual events from a cluster for plotting in a base map (the boundaries of which will be scaled to the distribution of the selected events). This is often used to provide better visualization of a short series of related events within a large cluster. The command can be used multiple times and the resulting plots have filename suffixes of "_sel1.pdf", "_sel2.pdf", etc.



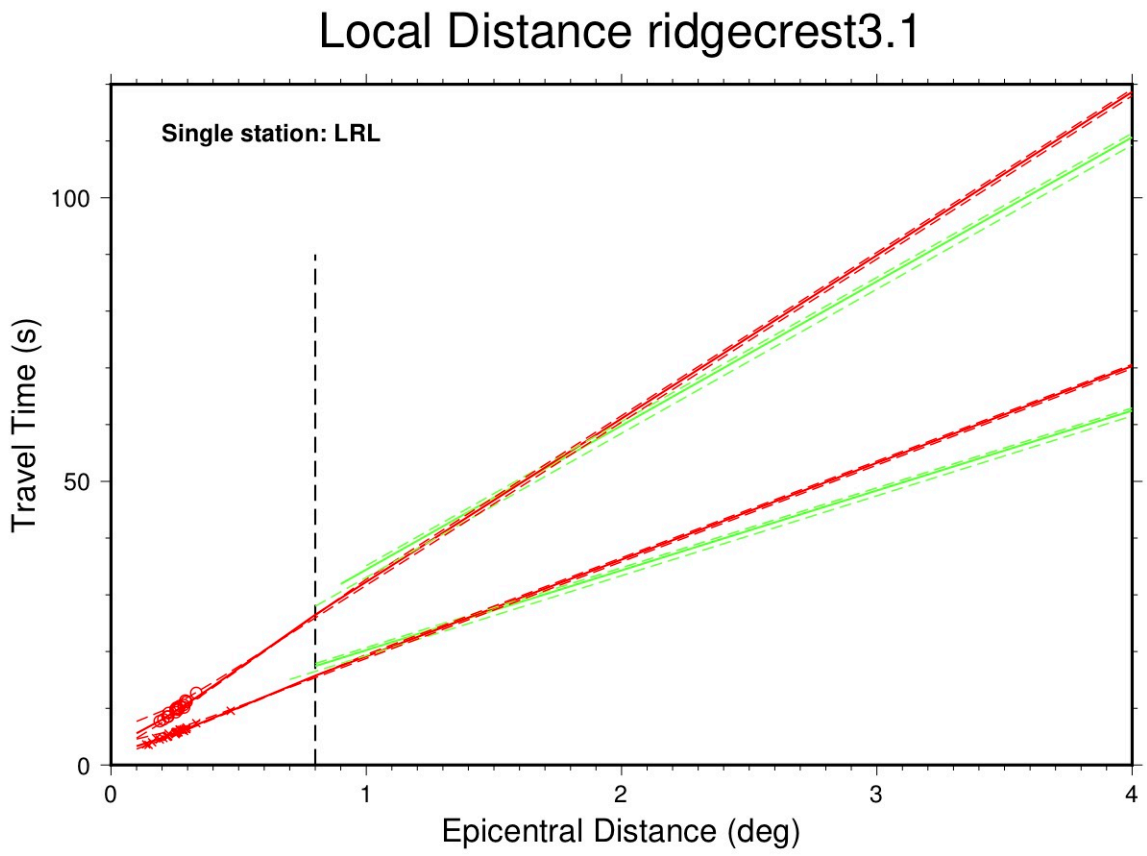
Single Event, Local Distance Plot (command tt5e)

This plot is similar to the local event plot (tt5) but shows arrivals only for a specified event. Useful for investigating an event that is misbehaving, e.g., failing to converge or ending up with a dubious location.



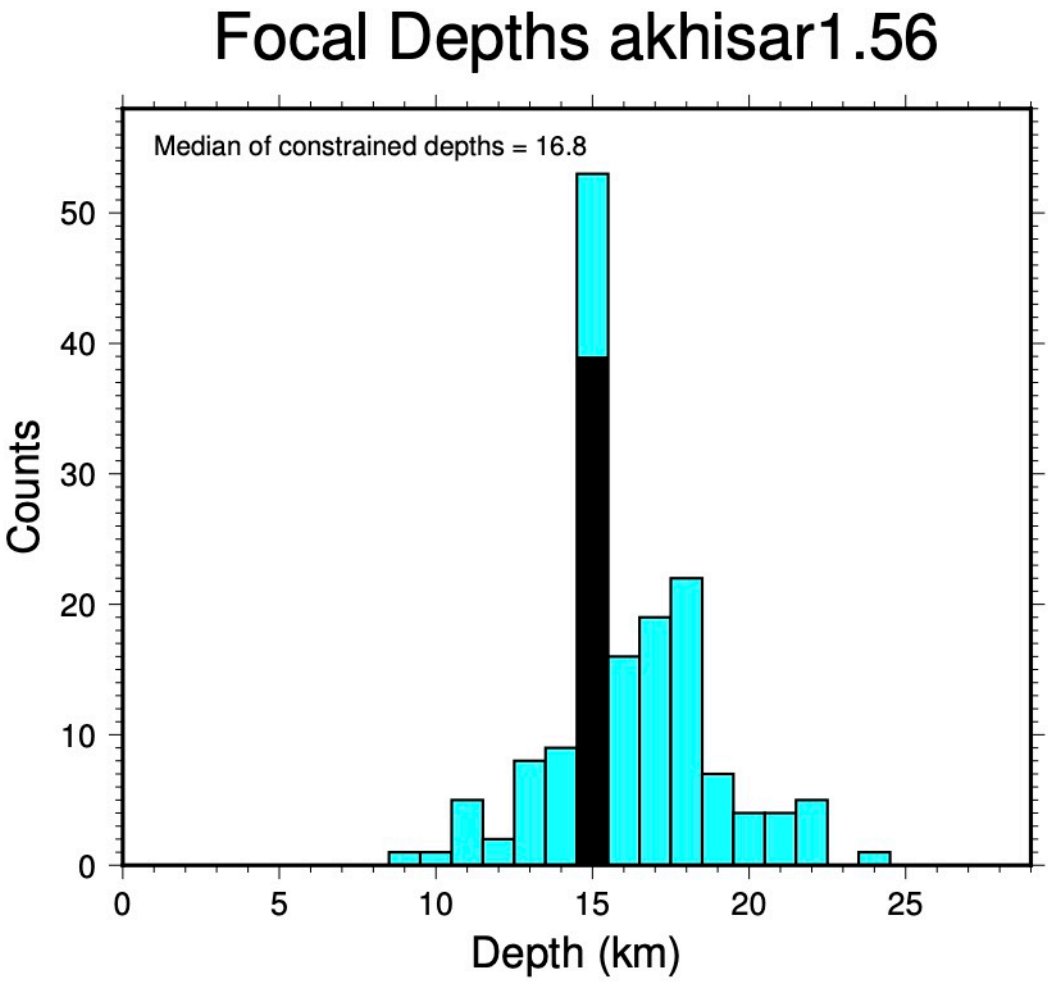
Single-Station, Local Distance Plot (command tt5s)

This plot is similar to the local distance plot (tt5) but shows arrivals only for a specified station. Useful for investigating possible incorrect station coordinates or a timing problem, sometimes for phase identification problems.



~_depth_histogram

A histogram of focal depths in the current run can be made using the command fdhp.



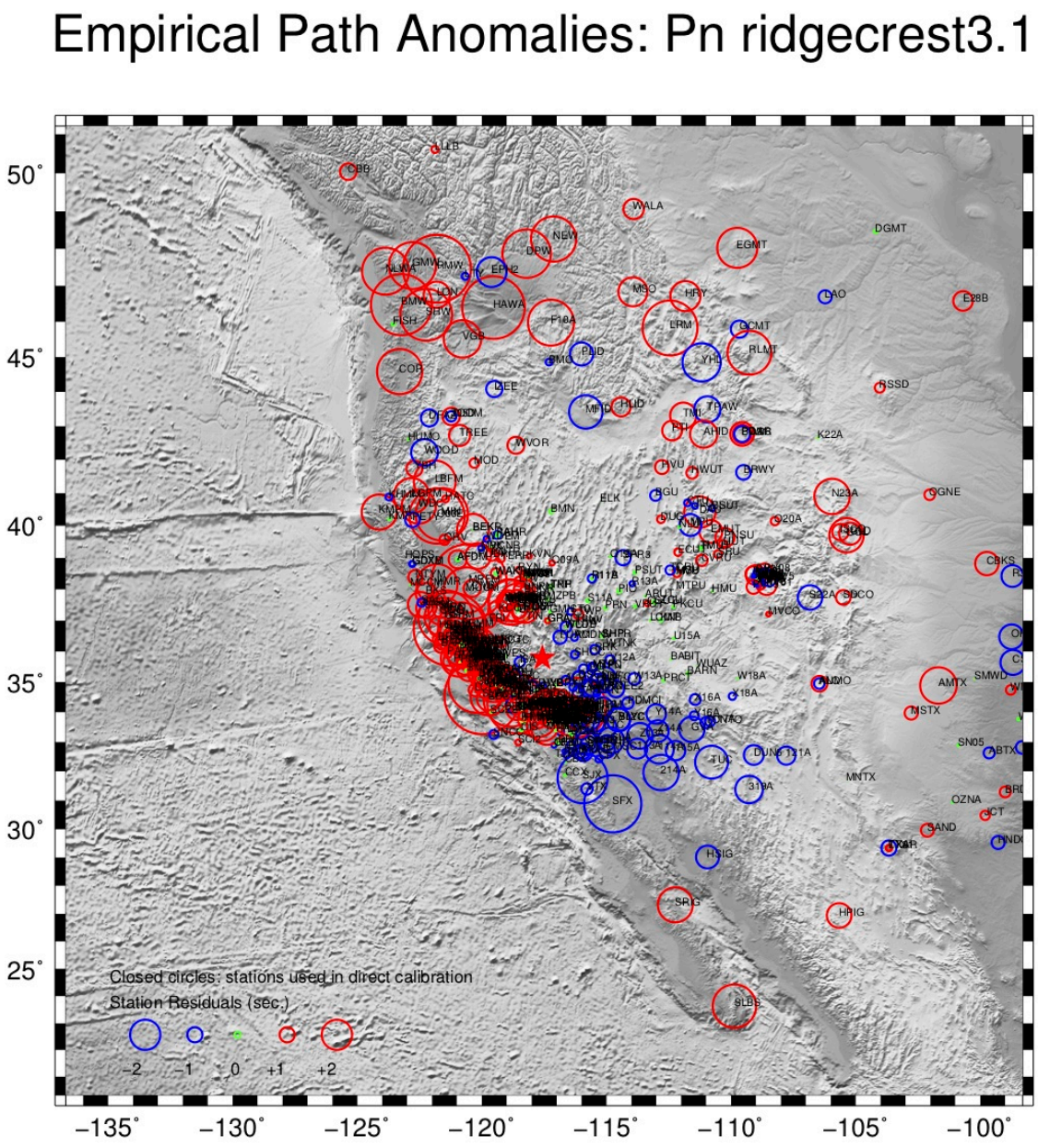
If any events have depths fixed to the cluster default value with the depc command (15 km in the above example), they will be shown in black in the histogram.

~_epa Plots

Plots of empirical path anomalies are made by the epap command. They are gathered in a subdirectory of the cluster series directory named *~_epa* and have names like *~_epa_N.pdf*, where "N" is an index number for multiple instances of the command. This is a map plot that displays the empirical path anomalies for a selected phase over a selected distance range.

Empirical path anomalies are taken as the mean of residuals for a given station-phase pair in a calibrated relocation. When used for Pg or Sg this plot can be helpful in evaluating the data being used for direct calibration, especially if you suspect that a station may have incorrect coordinates or a timing problem. It is very interesting to plot Pn path anomalies with this command, as you

will usually see distinct patterns of early and late arrivals related to gross changes in crustal structure. This is why it is very dangerous to use Pn arrivals for location. Here is an example, showing empirical path anomalies for the Pn phase to a distance of 15° epicentral distance for the Ridgecrest cluster in California.



There is an option to the epap command that controls whether of not raypaths are plotted. When there is a large amount of data, as for Ridgecrest, it is usually better to leave the raypaths off.

Late arrivals are shown in red, early ones in blue, with symbol size scaled according to the magnitude of the empirical path anomaly. Station codes are printed too.

~_plots.pdf File

A *~_plots.pdf* file is one of two files (the other is a ~.comcat file) created when the ccat command has been used to create output files for importation to the GCCEL database. The two output files are stored in a subdirectory of the cluster series directory named *~_comcat*. See the Mangyshlak cluster for an example.

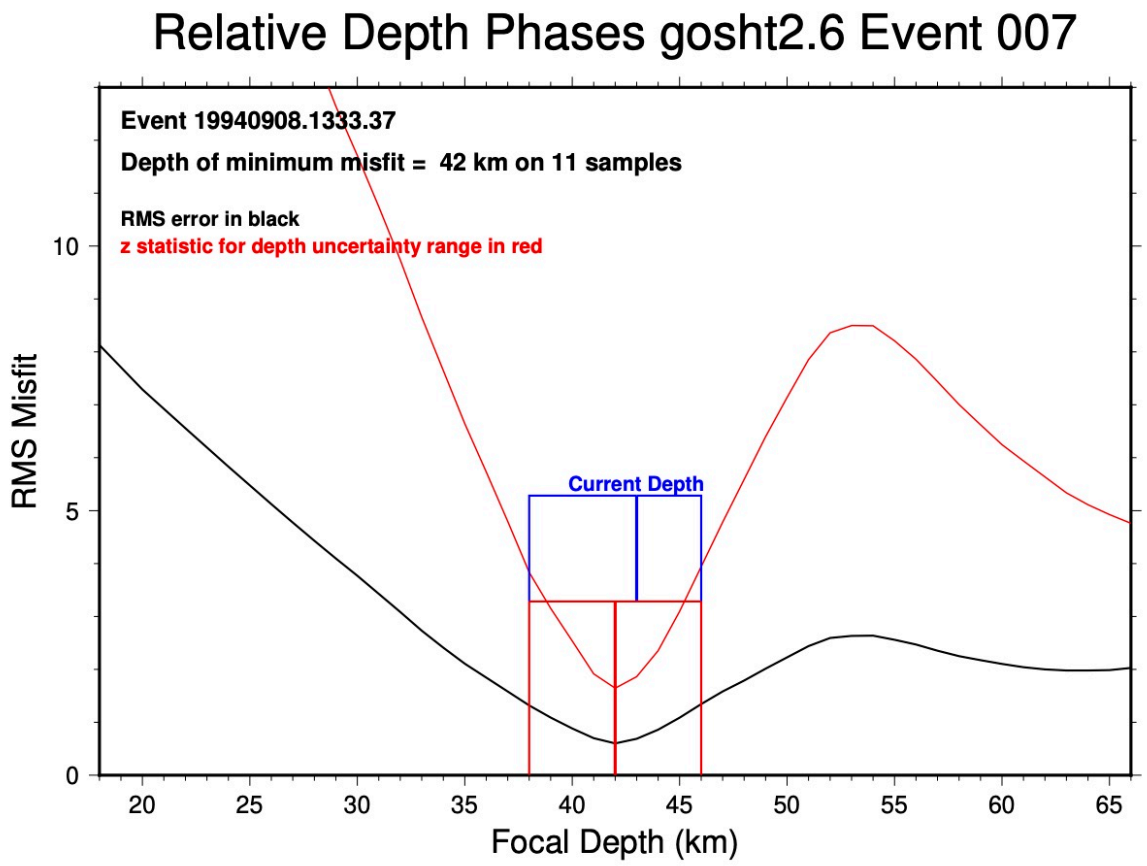
The *~_plots.pdf* file is a single PDF file containing multiple plots, all plots that were created in the run, either by default or by the action of the various commands that create plots.

~_rdp Plots

Plots of relative depth phase (i.e., pP-P, sP-P and pwP-P) information for individual events are made by use of the rdpp command. The plots are gathered in a subdirectory of the cluster series directory named *~_rdp* and the individual plot files (PDFs) are named *~_rdp_NNN.pdf* where "NNN" is the event number.

The content displayed in *~_rdp* plots is discussed at length in the section about the ~.depth_phases output file and in the section on depth constraint using teleseismic depth phases. A brief summary is given here.

The construction of the plot has recently been made more flexible in order to accommodate clusters with deeper events. Previously the plot had a maximum depth of 100 km; now it should work for events as deep as 760 km. As a result of the possibility of having to plot much greater focal depths than before, the depth range used in the plot is now based on the uncertainty limits for the estimated focal depth, taken 20 km further on each side. The underlying statistical test to find the best-fitting focal depth is done over the range of current focal depths in the cluster, extended 40 km both shallower and deeper. Here is an example of an event with best-fitting depth at 42 km:



The black curve is the raw RMS error function. The red curve is a statistic (more sensitive than RMS) used to determine the optimal depth and the asymmetric uncertainty range of that depth, described here. The current estimate of depth (in blue) is also shown.

Usage

This page summarizes some of the more common issues and questions that arise in using **mloc**, especially those encountered while developing a calibrated cluster.

- Creating a cluster
- What Kind of Relocation?
- Do I Need to Repick the Arrival Times to Obtain Better Locations?
- How Do I know When I'm Done?

Creating a Cluster

In selecting events for a cluster, there are several factors to be considered besides the basic choice of a source region, including the geographic extent, range of dates, depth range and the distance ranges represented by the data in different events.

Geographic Extent

A cluster is a collection of seismic events within a limited geographic area. This requirement comes from the notion, underlying all relative event relocation schemes, that if the raypaths from different events to each observing station are very similar, differencing the arrival times can remove most of the "noise" in the signals due to unknown Earth structure, leaving the relative locations as the main source of signal. If your cluster covers too large a region, the raypaths from different events to a station will have too little in common. There is no easy answer to the obvious question: "How big is too big?". In addition to the nature and degree of crustal heterogeneity the answer will depend on the kind of data you have: local-distance data will be more sensitive to that heterogeneity than teleseismic data. Heterogeneity may be azimuthally-dependent.

Experience has shown that 50-100 km is generally a pretty safe target for geographic extent. As clusters become larger than that there is increasing concern about biased relative locations, especially for events near the edges, but there may be little choice in some regions of sparse seismicity. Clusters less than 50 km across rarely evidence any of bias related to crustal heterogeneity.

Through Time

In principle, the date of events in a cluster, or the range of dates in a cluster, is of no relevance to the multiple event relocation problem, but in fact it is quite significant, due to changes in the constellation of observing seismic stations through time. All events must have a minimal level of connectivity, through observations at common stations, or the inversion will become unstable. There are a few seismic stations that have remained operational since the early 20th century and these can sometimes suffice to provide the connectivity needed to include an earthquake from,

say, the 1930s in a calibrated cluster with recent events. This usually works only with rather large events, obviously.

The same issue can arise in more recent time periods, when a previously poorly monitored region gains a dense monitoring network. This occurred in Iran between about 1995 and 2005. A search of the ISC Bulletin may yield what appears to be a nice cluster with some older larger events of interest, mainly recorded at teleseismic and far-regional distances, and many smaller, more recent events recorded by the recently-installed regional network, with abundant readings at close distances and good azimuthal coverage for direct calibration. It can turn out, however, that the older, larger events and the more recent, smaller events have very few stations in common, leading to a failure of **mloc** to converge. There is little to be done with such a cluster until the modern regional network captures a larger event that is recorded at far-regional and teleseismic distances. In cases like this, multiple event relocation can be less capable than single event location at analyzing seismicity over an extended time period. However, when it is possible to include older events in a "modern" calibrated cluster there is considerable value added.

Depth Range

mloc can relocate clusters containing events at any focal depth supported by the ak135 global travel-time model. For the same reason that we wish to keep clusters limited in geographical extent it is wise to limit the depth extent of a cluster. This is usually not an issue; outside of subduction zones the depth range of earthquakes in most continental and oceanic source regions is naturally limited.

Even in subduction zone clusters the range of depths for a cluster that is not too widespread is usually within reason. Even so it may be wise to confine the cluster to a more limited depth range. The reason has to do with **mloc's** crustal models, and whether events are above or below the model's Moho boundary. The simple 1-D crustal model supported by **mloc** already has trouble accounting for local and near-regional arrival time observations in a subduction zone setting with its strong dipping interfaces. The interplay between focal depth and Moho depth has strong effects on the pattern of arrivals and their phase identifications, which are difficult enough to unravel if all the events are fairly shallow (crustal level). If the cluster includes events below the Moho as well, it can become quite challenging to sort out the focal depth vs. crustal structure problem.

Distance Range of Observations

This is another expression of the connectivity problem in multiple event relocation. "Range of observation" can be taken as a proxy for magnitude, obviously. It is possible to have success with clusters containing events with very different ranges of observation, but it will usually require having good representation across the range of magnitudes. There is no objective way to evaluate connectivity problems in advance, but intuition is gained rapidly with experience. If **mloc** is having trouble converging it is nearly always due to weak connectivity.

How Many Events?

mloc can be run with a single event, but some aspects of its operation, such as estimating empirical reading errors, depend on having more than one event. When the number of events is small the statistical power in the dataset, i.e., repeated observations, is weak and the uncertainties in the results will be larger than they would be with a more populous cluster. Experience has shown that this issue stabilizes fairly well when the cluster reaches ~30 events and by 50 events the statistical power is about as good as it will get.

The computational load of a run with **mloc** depends mostly on the number of events in the cluster, i.e., the number of free parameters (usually 3 or 4 per event). The increase in runtime with increasing number of events is greater than linear. I generally try to keep a cluster below ~100 events unless there are strong reasons to include more. **mloc** is configured for a maximum of 200 events. With readily-available desktop and laptop computers a run for a cluster of this size takes many minutes, enough to slow down the overall analysis (which requires many runs) considerably. If you are very patient or have an especially powerful computer on which to run and you really need to analyze more than 200 events in a cluster, it is easy to set the limit higher by editing the *mloc.inc* file and changing the value of the parameter *nevmax*. You will need to recompile afterwards, of course.

What Kind of Relocation?

There are four types of relocation that can be done with **mloc**:

- Uncalibrated
- Direct calibration
- Indirect calibration
- Direct calibration followed by indirect calibration

If there is not enough near-source data to employ one of the calibrated relocation methodologies, the user must accept that the absolute locations and origin times of the relocated events are biased to an unknown degree by unknown Earth structure. The most robust estimate in this case is to use only teleseismic P arrivals to estimate the hypocentroid; the command file would include:

```
phyp on
hlim 30. 90.
```

Although the degree of bias is unknown, it is possible to set some rough limits on it. In most cases the hypocentroid estimated this way is probably not more than about 10 km in error. The origin times are unlikely to be more than about 3 seconds off, and are much more likely to be late than early.

Although it is perfectly capable for performing traditional uncalibrated relocations of clusters of earthquakes **mloc** has been specifically developed to conduct calibrated relocations. My personal

preference is to use direct calibration if possible. It is based strictly on seismological observations. Direct calibration calibrates origin time in addition to epicenter and focal depth, which is extremely important for studies of Earth structure, whereas most other sources of information on "location" rarely constrain origin time. When the data for direct calibration are inadequate to provide a robust result, however, indirect calibration can be very helpful. When direct calibration is successful and other estimates of source locations are available, such as from InSAR or observations of surface faulting, there is much value in having independent estimates of hypocentral parameters to compare.

If a cluster includes one or more sources for which ground truth location information is available, indirect calibration may be the preferred method, but even in this case I would prefer to do direct calibration first (if possible) and then check it with indirect calibration. There are cases in which reported "ground truth" locations are clearly in error (e.g., Mackey and Bergman, 2014).

It is not uncommon, when developing a calibrated cluster, to find it useful to perform a few uncalibrated relocations in the early going. This simplifies the analysis somewhat and permits the user to focus on the basic composition of the cluster and sort out basic issues, such as missing station codes and code conflicts, the geographic extent and depth extent of the cluster and problems with connectivity.

Do I Need to Repick the Arrival Times to Obtain Better Locations?

Not really.

It is usually impossible to know how the arrival times in a dataset downloaded from the ISC or other major data center have been picked. Most have probably been reviewed to some extent by a human but some may be automated picks. The expertise of the human who may have made the pick or reviewed an automated pick is highly variable. **mloc** is designed so that one does not need to worry very much about these issues, by employing empirical reading errors estimated from the arrival time data itself, and using those estimates in a statistical analysis (cleaning) to identify outlier readings and to weight the data in the inversion.

The improvements in location that are likely to be obtained from repicking the arrival times are minor (with one important exception, below), and in fact they can be negative. At one point in **mloc's** development we arranged for one of the most experienced seismic analysts in the world to provide her own readings of arrivals for which we also had the original ISC bulletin data. It turned out that her picks were quite often flagged as outliers because most of the arrival times in our dataset were picked by much less experienced and dedicated analysts. For the most part, **mloc** values consistency rather than absolute correctness.

The exception alluded to above is the case of direct calibration, in which the arrival time data at short epicentral distances (usually less than ~100 km) is used to estimate the hypocentroid. These picks need to be as accurate as possible and careful repicking, even during the **mloc** analysis, can make a difference, especially if the available dataset for estimating the hypocentroid is on the

minimal side. If there is a lot of data, however, repicking is unlikely to make any noticeable difference to the calibration.

How Do I Know When I'm Done?

First of all, **mloc** should be converging within one or two iterations if you are starting the relocation from the locations of the previous run (command rhdf). There should be few if any cases of missing station codes. The cleaning process should be converged, meaning that running with lres = 3 produces a very small or empty ~.lres file and the ~.xdat file should also have no more than a handful of entries.

Review the various output files and plots related to focal depth and convince yourself that there is no more that can be done in that regard. Some events simply do not have any arrival time information that helps to constrain depth, and they will need to be set at some reasonable default depth. I usually take the default depth for a cluster as the median of constrained depths, listed in the ~.depth_phases file.

All of the summary plots are potentially valuable in detecting anomalous results that may indicate the need for more work. On the baseplot, look for events that are still moving significantly from the last run (green vector). Also look for events which have ended up far from their initial location (black vector), say, more than 20 or 30 km away. This could be correct, but it could also be a case where the cleaning process has flagged a number of readings in error. Check the ~.phase_data file to see if there are an unusually large number of readings in the "bad data" section. If so, consider if adding some of them back into the problem might lead to a different solution. Some events do have an unusual number of outliers.

In a direct calibration analysis the near-source travel-time plot is especially important to review. Check the baseline offset of the Pg and Sg phases; they should both be less than one or two tenths of a second in absolute value. There should be little sign of a slope to the residuals of either phase and there should not be any large outliers. For Pg it is rare to see a legitimate residual greater than about 1 s; for Sg the likely limit is ~2 seconds. If the residuals at very short distances (less than ~0.1°) show a noticeable curvature there may be issues with focal depth. If there are many Pn and Sn readings, consider reducing the distance range for the hypocentroid (command hlim).

In an indirect calibration, check the baseplot with respect to the residual calibration shift vectors (blue vectors). Anomalies (failure to cover) can be further investigated in the ~.cal file. Check the statistical tests of "radius of doubt". If it is greater than a couple hundred meters there is probably something amiss. Consider dropping (as calibration events) events that fail the coverage criterion badly, since it is likely that either the original "ground truth" location is bogus or the relative location of the event was mishandled in the **mloc** analysis. Alternatively, consider if the reported uncertainties of the calibration events have been under-estimated. There is a command cvff that can be used to inflate the uncertainties of the cluster vectors such that discrepancies with the adopted calibration locations are reduced in a statistical sense, but this is completely *ad hoc*.

Calibration

In the context of earthquake location, we use the term **calibration** to refer to an analysis that includes procedures to minimize the biasing effect of unknown Earth structure (i.e., velocity variations) on hypocentral parameters and also incorporates procedures to obtain realistic measures of uncertainty of those parameters. Two major challenges in this regard are 1) to reduce the biasing effect of outlier readings in a least-squares estimation procedure, and 2) to minimize the biasing effects of unknown data variances on estimates of the uncertainty of the hypocentral parameters.

Two methods for determining calibrated locations for clusters of earthquakes are implemented in **mloc**. The key to calibrated locations is the use of near-source data to establish the location of the cluster in absolute spatial and temporal coordinates. One method of calibration (**indirect calibration**) is based on external (*a priori*) information on the location of one or more cluster events. Such information can be derived from seismological, geological, or remote sensing data. The second method of calibration (**direct calibration**) is based on analysis of a subset of the arrival time information in the cluster at short epicentral distances.

The second aspect of calibration, realistic uncertainties, is achieved in both direct and indirect methods of calibration by a careful analysis of empirical readings errors (derived from the actual arrival time data) which are used both for weighting in the inversion for location and also for identifying outlier readings, often referred to as "cleaning". These subjects are discussed in other sections of the website. In the rest of this section we describe the direct and indirect methods of calibration and the feature of the hypocentroidal decomposition method of multiple event relocation which makes it especially well-suited to this kind of analysis.

Applicability

The methods of location calibration described here can be used at a range of scales, from a handful of small events recorded locally to several hundred large events recorded globally. The algorithm is best-suited for data sets of about 200 events or less, and it is desirable to restrict the geographic extent of a cluster to approximately 100 km or less. There must be a minimal level of "connectivity" between events in a cluster, meaning repeated observations by the same seismic station. Since calibration depends on data at short range from the source, it is not possible to calibrate the locations of deep earthquakes, although the relative locations can be improved.

Calibrated locations with uncertainties of 2-3 km in epicenter are achievable with either method, as can be seen by reviewing the results posted at the GCCEL website.

Hypocentroidal Decomposition

The location calibration analysis is based on the Hypocentroidal Decomposition (HD) method for multiple event relocation introduced and by Jordan and Sverdrup (1981). The basic algorithm is completely described in that reference. The essence of the HD algorithm is the use of orthogonal projection operators to separate the relocation problem into two parts:

- The **cluster vectors**, which describe the relative locations in space and time of each event in the cluster. They are defined in kilometers and seconds, relative to the current position of the hypocentroid.
- The **hypocentroid**, which is defined as the centroid of the current locations of the cluster events. It is defined in geographic coordinates and Coordinated Universal Time (UTC).

Both methods of calibration described here depend fundamentally on this decomposition of the relocation problem. Similar approaches could conceivably be implemented in other multiple event relocation algorithms but it seems likely to be considerably more difficult than it is with hypocentroidal decomposition.

The cluster vectors are defined only in relation to the hypocentroid. The hypocentroid can be thought of as a virtual event with geographic coordinates and origin time in UTC. The orthogonal projection operators act on the data set of arrival times to produce a data set that includes only data that actually bears on the relative location of cluster events, i.e., multiple reports of a given seismic phase at the same station for two or more events in the cluster.

The hypocentroid is located very much as an earthquake would be, except that the data are drawn from all the cluster events. Thus it is typical for the hypocentroid to be determined by many thousands of readings. Nevertheless, the hypocentroid is subject to unknown bias because the theoretical travel times (typically ak135) do not fully account for the three-dimensional velocity structure of the Earth. Geographic locations for the cluster events are found by adding the cluster vectors to the hypocentroid.

The HD method works iteratively. At each iteration, two inversions are performed, first for the cluster vectors relative to the current hypocentroid, then for an improved hypocentroid. The cluster vectors are added to the new hypocentroid to obtain updated absolute coordinates for each event. The convergence criteria are based on the change in relative location of each event (0.5 km) and the change in the hypocentroid (0.005°). The convergence limits for origin time and depth, for cluster vectors and hypocentroid, are 0.1 s and 0.5 km, respectively. Convergence is normally reached in 2 or 3 iterations.

The data sets used for the two problems need not be (and usually are not) the same. Because the inverse problem for changes in cluster vectors is based solely on arrival time differences, baseline errors in the theoretical travel times drop out and it is desirable to use all available phases at all distances outside the immediate source region. For the hypocentroid, baseline errors in theoretical travel times are more important and one may wish to limit the data set to a phase set, e.g., teleseismic P arrivals in the range 30-90°, to achieve a more stable (but uncalibrated) result. The choice of data set for determining the hypocentroid has great importance in the "direct" calibration method described below.

Similarly, weighting schemes are different for the two inversions, reflecting the different natures of the two problems. Empirical reading errors for each station-phase pair are used in weighting data for estimating both the hypocentroid and cluster vectors, but the uncertainty of the

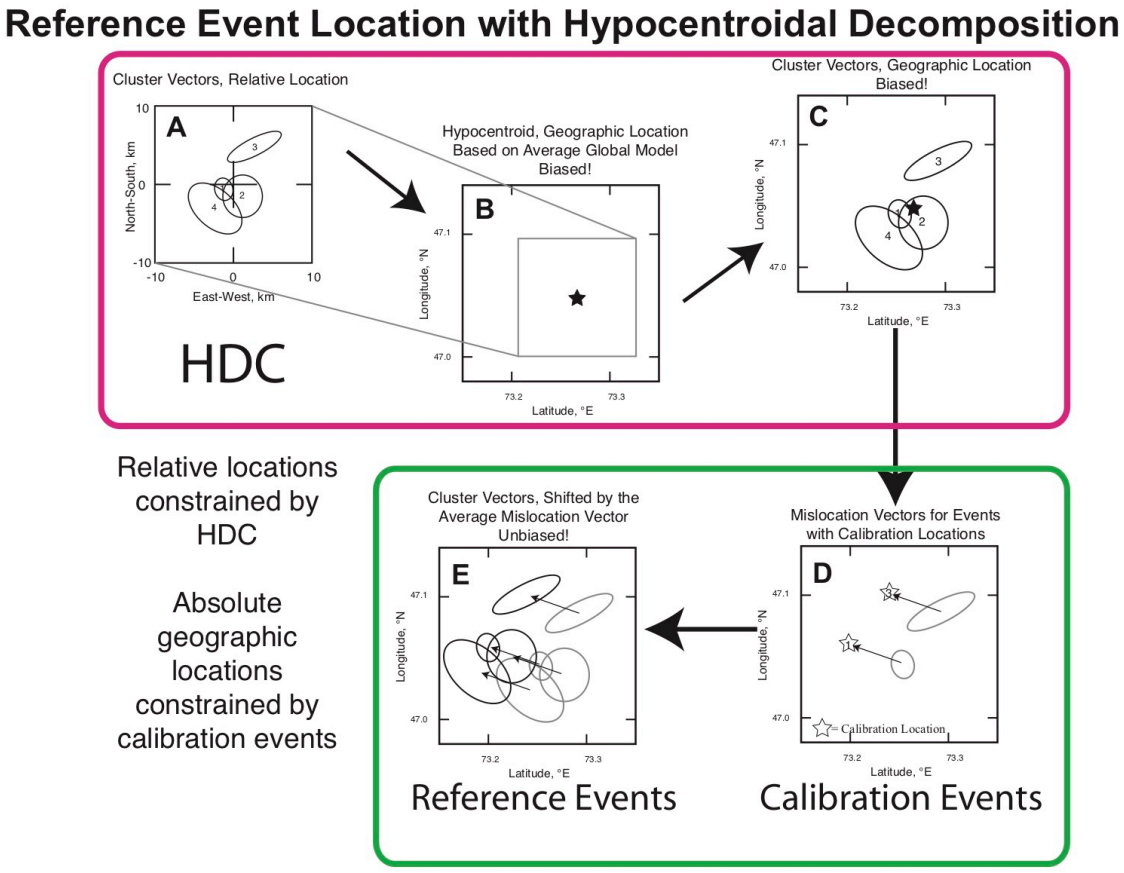
theoretical travel times, which are estimated empirically for each phase from the residuals of previous runs, is relevant only to the hypocentroid.

Until this point the HD algorithm is used only to obtain improved relative locations for the cluster events, with a geographic location for the cluster as a whole (the hypocentroid) that is biased to an unknown degree by unmodeled Earth structure that has been convolved with the (typically) unbalanced geographic distribution of reporting seismic stations. The calibration process attempts to minimize this bias.

Calibration of a cluster is done in two ways, which we refer to as “indirect” and “direct” calibration.

Indirect Calibration

If the location and origin time of one or more of the cluster events can be specified with high accuracy from independent information, we can calibrate the entire cluster by shifting it in space and time to optimally match the known location of the calibration event(s). The approach is illustrated in this cartoon.



The steps inside the red box represent standard, uncalibrated relocation with **mloc**. The steps inside the green box represent the calibration analysis.

- A: Cluster vectors (relative locations) estimated.
- B: Hypocentroid estimated but known to be biased by unknown Earth structure.
- C: Add cluster vectors to hypocentroid, yielding uncalibrated estimates of hypocenters of individual events.
- D: An average "calibration shift" is calculated from comparison with a subset of "known" locations.
- E: All events shifted to obtain calibrated hypocenters.

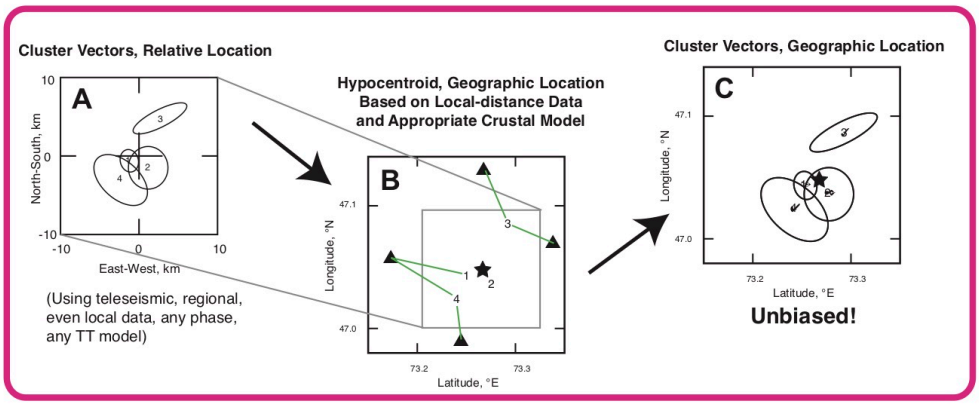
A common source of such independent information is a temporary seismic network deployment that captures an event with a large number of stations at very short epicentral distances, and which is also large enough to be well recorded at the distances at which other events in the cluster were observed (usually, regional and teleseismic distances). Aftershock studies are a frequent source of such data. If it is possible to obtain the temporary network arrival time data a direct calibration analysis would be preferred, but sometimes only the hypocenters are available. InSAR data is also used for this purpose, but this requires the use of at least some seismic data at short distance to calibrate origin time. The other problem with InSAR analysis is that it is difficult to specify where along an extended source the epicenter should be placed. In some cases mapped faulting from large events can be used to help constrain the location of calibration events, but this also suffers from the uncertainty about where along an extended source to place the epicenter.

When we use the indirect calibration approach we must take into account the uncertainty of the calibration locations, and when there is more than one calibration event we also include a contribution to uncertainty to reflect any discrepancy between the relative locations of the calibration events and the cluster vectors of the corresponding events.

Direct Calibration

It is often the case that there are a few permanent seismic stations close to a cluster of earthquakes, but that no single event is well-enough recorded to reach the level of accuracy necessary to serve as a calibration event. On the other hand, the handful of local seismic stations may have recorded many events in the cluster, so that the number of "short distance" readings is rather large, and well-enough distributed to allow the hypocentroid to be located using only these data. The direct calibration method is based on the use of arrival time data from nearby stations to locate the hypocentroid. By keeping raypaths short, accumulated error in the theoretical travel times (and thus, location bias) is minimized.

Reference Event Location with Direct Calibration in Hypocentroidal Decomposition



- A: Cluster vectors (relative locations) estimated.
- B: Hypocentroid estimated from close-in stations, thus minimizing location bias.
- C: Add cluster vectors to hypocentroid, yielding calibrated estimates of hypocenters of individual events.

In any case where we have the data to locate one or more calibration events for the indirect method, we also have the option to use those same data in the direct calibration method. The decision on which to use is made on a case-by-case basis, because characteristics of the data sets may lead to a better result with one method than the other. Of course, if we are using InSAR data or other remote sensing or geological information to constrain the calibration, we must use the indirect method.

In **mloc** it is possible to set up a relocation which uses both methods. Direct calibration will be done as part of the normal relocation process and then the adjustment for indirect calibration will be made, relative to whatever calibration locations have been declared. Some output files will be written for each method, but the *.summary* file will take the indirect calibration as preferred, as

will the summary plots. It is quite useful to have both estimates of calibration as a measure of quality control.

Cleaning

An important aspect of the relocation process with **mloc**, whether calibrated or not, consists of multiple cycles in which the current estimates of empirical reading errors are used to identify outlier readings, which are then flagged so that they will not be used in subsequent relocations. In the following relocation, estimates of empirical reading errors will tend to be smaller because of the filtering of outliers and improvement in the locations of the clustered events. Therefore the process of identifying outliers is iterative and it must be repeated until convergence. In this context, *convergence* means that the distribution of residuals for a given station-phase is consistent with the current estimate of spread. As outlier readings are flagged, the distribution is expected to evolve toward a normal distribution with standard deviation equal to the empirical reading error. We generally continue this **cleaning** process until all readings used in the relocation are within 3σ of the mean for that station-phase, where σ is the current estimate of empirical reading error for the relevant station-phase.

Strategy

In this section I will describe in some detail the specific steps I would normally take in analyzing a new cluster.

For the first run, turn off inverse weighting with command weig ("weig off"). Don't put this in the command file, just issue it interactively, because you may only use it once or twice. You could use the default weighting for this step but turning weighting off completely makes for a more robust inversion when there are likely a lot of outliers. Set the threshold for command lres to 3, again, interactively after reading the command file with command cfil. When using empirical reading errors later in the analysis you would not use a value as small as 3 until the final few runs, but when weighting is turned off every station-phase has an effective reading error of 1 second and 3-second residuals are nearly always safe to flag as outliers.

After the first run of **mloc** with settings as above, it is safe to simply run the utility program lres and flag everything in the ~.lres file automatically. Don't run xdat yet, however, because **mloc** does not yet have the information from the ~.ttsprd file to correct for the offsets of the windows for different phases.

In most cases you can set up your next run of **mloc** to use the empirical reading errors determined in the first run. This means you will add a line to your command file with the rfil command, referencing the ~.rderr file from the first run. Add lines for commands rhdf and tfil as well, referencing the appropriate output files from the first run. Now you are using empirical reading errors that may be rather small already, but there will still be many gross outliers so the threshold for lres should be large, 5 or 6 is usually adequate.

After this run, inspect the ~.lres file. It will probably be a few tens of kb in size. If there are a few cases with very large values of cluster residual (eci) it is not a problem, you can go ahead and run lres on it. However if there are many such, it may be wise to hand edit with the utility program rstat to clean out the very largest outliers. The reason is that very large outliers can induce readings that are actually good to look like outliers in these early stages: they won't be as large and will have the opposite sign. Alternatively, you could re-run with a larger threshold for command lres and then run the utility program lres.

Now you are into a repeating pattern of trimming the largest outliers, getting improved estimates of empirical reading errors and re-running until you find few readings with cluster residuals greater than 3. The reduction in threshold value that you will specify for each run with command lres should be guided by how many outliers you are getting at each run. Small bites are better, but there's no need to waste time, either. It is safest to make at least two runs at each level because it takes two runs before the consequences of flagging outliers to be reflected in the new empirical reading errors. A good approach is to keep running at a given level until there are only a few outliers left at that level; it may take a half-dozen or more runs. From lres=5.0 I would normally drop to lres=4.0, then possibly 3.5 and finally 3.0.

During this process there are other adjustments taking place as well. The two main ones are adjusting the local velocity model, if one is being used, to fit the observe travel time data at local distances and constraining depths. Both activities obviously impact the cleaning process, as well as each other.

Another part of this process is making sure all the data are being used, by checking the ~.stn file for missing stations or conflicts. It is also necessary to occasionally check the "bad data" section of the ~.phase_data file to look for signs of readings that may need to be unflagged.

For a direct calibration analysis it is especially important to keep a close eye on the cleaning process as it applies to the readings being used to locate the hypocentroid. That is the main reason the output file ~.dcal_phase_data exists.

Another way of spotting problems during cleaning is to review the empirical reading errors, either by scanning through the ~.phase_data file or by inspection of the ~.rderr file itself. When the number of samples for a particular station-phase is small, as is often the case, the robust estimator of spread is not very good at distinguishing outliers and will simply produce an overly-large estimate of empirical reading error. It does not take a great deal of seismological expertise to spot these situations, or to know which readings should be flagged. Similarly, scanning the converged residual column of the ~.phase_data file for unusually large residuals will reveal outliers that the automatic process may miss. These kinds of problems are most common among secondary phases, especially regional and teleseismic S-phases, that do not ultimately have much impact on the location analysis, but it is good practice to give them some attention.

Depth Constraint

Focal depth is a notoriously difficult parameter to resolve in any location algorithm because of the strong trade-off in most cases between focal depth and origin time. **mloc** has a number of tools with which to extract information on focal depth, subject to whatever limits the dataset imposes. There are three general cases:

- Focal depth can be resolved as a free parameter.
- Focal depth must be fixed, but a useful estimate can be made.
- No information on focal depth is available.

The most relevant commands in **mloc** are those used to specify which hypocentral parameters will be free in the inversion (freh and frec) and those that establish the starting depth for relocation (the dep_ family).

Free Depth

Focal depth can be included as a free parameter in the relocation of an earthquake when there is arrival time data at a range comparable to several times the focal depth or less. Schematically, what is required is at least one up-going raypath, which will have a partial derivative opposite in sign from that of the down-going raypaths that typically dominate arrival time datasets.

In order to set focal depth as a free parameter for a cluster, every event in the cluster should have suitable data to constrain its depth; otherwise the inversion is unlikely to converge. It is possible, using the frec command, to set depth free for some events and keep it fixed for others, but this has not been carefully tested and it may violate some assumptions of the code. A better strategy is to evaluate the subset of events that have depth control separately in a free depth relocation and then set the depths of those events accordingly (depm command) in a fixed-depth relocation of the entire cluster.

The Wells, Nevada cluster included in the mloc distribution is an example of a free-depth relocation

Fixed Depth, with Constraint

This is the most common scenario in practice. In the command file for a run each event definition section will include a member of the dep_ family that specifies the depth, its uncertainty and the source of the constraint. The different approaches to determining the depth at which an event should be constrained are discussed below. It is not unusual to have a few events in a cluster that must be set to the cluster default value (depc command), which is best determined by reference to those events in the cluster that do have good depth control.

The Mangyshlak PNE cluster included in the mloc distribution is an example of a relocation done this way. Constraints on the depths of the PNEs are available from Russian sources and the range of uncertainty in those values is small compared to the usual uncertainty in earthquake focal depths.

Most calibrated clusters in the GCCEL database have been relocated with fixed depth, but with most events having depth constraint of one sort or another.

No Constraint

Some clusters contain no events with depth control. These are typically clusters composed of relatively small events in remote areas with no local network data. It is often possible, however, to place some broad constraint on the likely range of depths based on seismotectonic considerations, and this in turn can be incorporated in **mloc** using a cluster default depth with the depc command.

The Jordan & Sverdrup Region A cluster included in the mloc distribution is an example of a relocation done this way. Earthquakes in oceanic intraplate settings have been observed as deep as 30 km or so, but in their original application of the Hypocentroidal Decomposition method to clusters in the south-central Pacific, Jordan and Sverdrup fixed all events at 10 km, influenced by the notion that the events were likely related to volcanism. For the purpose of comparison the same procedure is followed in our example cluster. In any case, the effect of moderate errors in depth for events located with teleseismic data is minor; depth mostly trades off with origin time. Since the water depth in the source region is about 5 km, the focal depths are about 5 km below the seafloor, in the oceanic crust.

In most cases, a cluster in which no event has depth control will also be one that cannot be calibrated.

Depth Reference Surface

In the global average travel-time model ak135 the zero level for depth is the reference ellipsoid WGS84. In other words, the zero depth reference surface is some distance below the land surface in most continental regions. Like all global location codes **mloc** normally makes a correction for station elevation (command corr), so the situation is straight-forward for a typical location analysis that uses regional or teleseismic data to estimate the hypocentroid of an event that is deep enough to be below the reference ellipsoid.

The situation gets more complicated if a direct calibration relocation is being done with events that are shallow enough that they may lie above the reference ellipsoid (for a cluster in the Tibet Plateau, for example, the events wouldn't need to be all that shallow). In this case the corr command can be used to turn off station elevation corrections, which makes the reference surface for depth, roughly, the average elevation of the source region.

mloc is not suited to situations, such as volcano monitoring, where an event may be at a higher elevation than the stations being used to locate it.

Starting Depth

The value of starting depth for a relocation in **mloc** can be taken from several sources. The order of precedence (from most to least preferred) is:

- A dep_ command in the command file.
- Depth read from a *.hdf* file from a previous run.
- The depth from the input data file (if it carries a flag as being constrained).
- The value given with the depc command, if it has been given.
- The depth from the input data file (if a depth has been provided).

Methods of Constraining Focal Depth

If the available arrival time data are inadequate to resolve focal depth as a free parameter for all events in a cluster the relocation will probably need to be done with depths fixed, but it is still possible to determine focal depths for many events with a considerable degree of confidence outside the **mloc** relocation. There are four methods used in **mloc** that are based on the available arrival time data:

- Free-depth relocation of a subset of events
- Near-source readings
- Local distance readings
- Teleseismic depth phases (separate section)

There are also "external" sources of information on depth that can be used for constraint and **mloc** includes depth codes for most of the possibilities.

Free-Depth Relocation of a Subset of Events

If some events in a cluster are notable in having readings (say, from temporary stations) at very close range (within several focal depths) it is very worthwhile to form a subcluster for relocation in which depth can be left as a free parameter. In addition to providing very useful information on the likely depth range of the remainder of the cluster, the subcluster can be used to refine a local crustal model which can then be used in the "local-distance" method of depth constraint. The thickness of the crust of the crustal model cannot be reliably constrained unless at least some events have reasonable depth constraint. The depths from the free-depth relocation can be inserted into a subsequent command file as depm constraints in a fixed depth relocation of the larger cluster. **mloc** includes a command subc that helps in extracting the events that are most likely to behave well in such a subcluster.

Near-Source Readings

After a free-depth relocation the most powerful method of constraining focal depth utilizes readings at epicentral distances that are not much greater than the focal depth. Distances of 2 to 3 times the focal depth are still usable. There must be data (at greater distances, but only crustal arrivals) that constrains the epicenter closely for this to work; the near-source stations will not in themselves be sufficient. The relocation is done with depths fixed to match the near-source

readings. Adjustment of focal depth and the crustal model proceed simultaneously. Arrivals at very close range are sensitive to depth more than velocity but the slope of arrival times of more distant crustal arrivals will be sensitive to average crustal velocity. When crustal arrivals (out to the Pg/Pn crossover, say, 100-150 km) are well-matched, the thickness of the crust is then adjusted to correctly predict the arrival times of Pn and Sn.

In many cases, manual adjustment of the focal depth in this manner is a more accurate method of constraining focal depth than a free-depth relocation, because it emphasizes the fit to the closest stations that are the most sensitive indicator of depth. In a free-depth location all the data used contributes to the estimation of depth, but residuals at greater distances are more likely to be distorted by lateral heterogeneity (or simply a less-accurate assumed crustal velocity model) which can be mapped into a biased estimate of depth. This is evident when a free-depth location leaves the closest stations with significant residuals.

Local Distance Readings

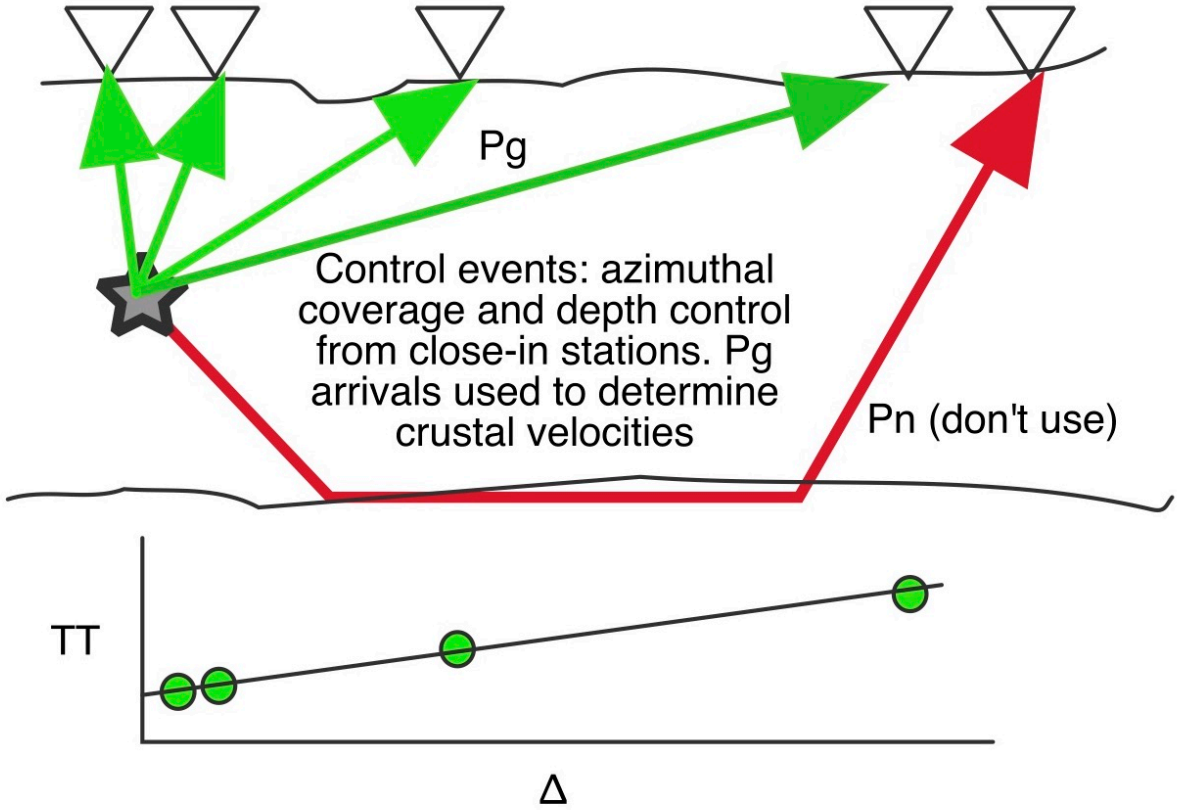
This method of constraining focal depths, unlike free-depth locations or fixing depth on the basis of a few near-source arrivals, will challenge the intuition of most seismologists, but it has been in regular use in **mloc** for over 5 years and has been applied to nearly every cluster in GCCEL (219 clusters and ~14,500 events at this writing). The *local distance method* has proven to be reliable and it greatly expands the number of events in calibrated clusters for which focal depth can be constrained at a useful level.

The key to understanding the method is to appreciate that it is not driven, as most "depth phases" are, by the depth derivative $\partial t/\partial h$, but rather by the derivative of travel-time w.r.t. origin time. A set of circumstances are required that sound rather limiting but which are actually encountered quite commonly in practice. It is assumed that there are no near-source readings, say, within one or two (likely) focal depths, for the event in question or we would be setting depth by the method of fitting the near-source readings in a fixed-depth relocation, if not performing a free-depth inversion. On the other hand we do assume that there is at least one event in the cluster that does have good depth constraint, most commonly from near-source readings or a free-depth relocation of a subset of events that have the necessary data. This event (or preferably a good number of them) is termed a *control event*.

- For the event in question there must be at least one arrival of a direct crustal phase (Pg or Sg) at a distance less than the cross-over distance with Pn/Sn, i.e., the phase identification must be unambiguous.
- The average crustal velocity must have been adjusted to fit the observed slope of direct crustal arrivals as a function of distance, using control events. This is most easily checked with the near-source summary travel-time plot.
- The crustal thickness (Moho depth) must be constrained by considering the Pn arrivals from control events.

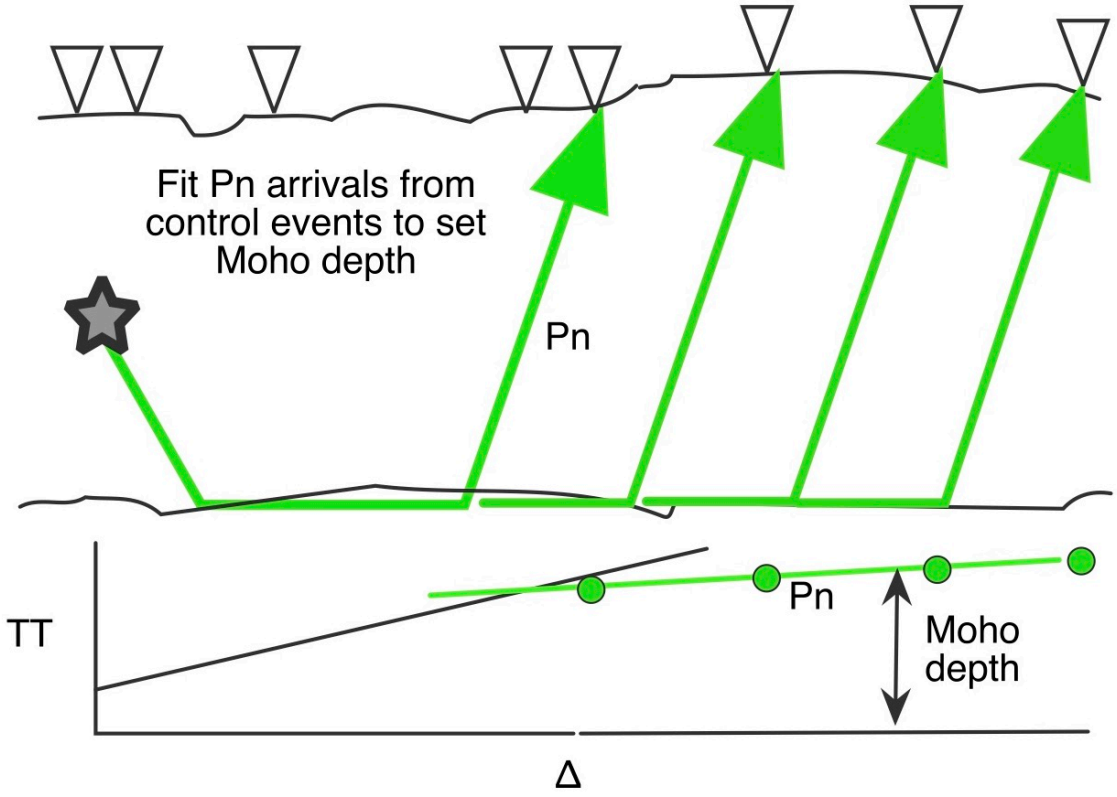
- A healthy population of Pn arrivals is needed for the event in question. Sn also contributes but with far less influence in most datasets.

Treated as a separate cluster in **mloc**, the control events can be accurately located using all readings for the cluster vectors and only direct crustal arrivals for the hypocentroid, i.e., a direct calibration. With Pg and Sg at short distances (within a couple focal depths) as well as greater distances, there is good constraint on depth in the normal manner. With depths set accurately, the direct crustal arrivals from the control events can be used to refine the crustal velocity model.

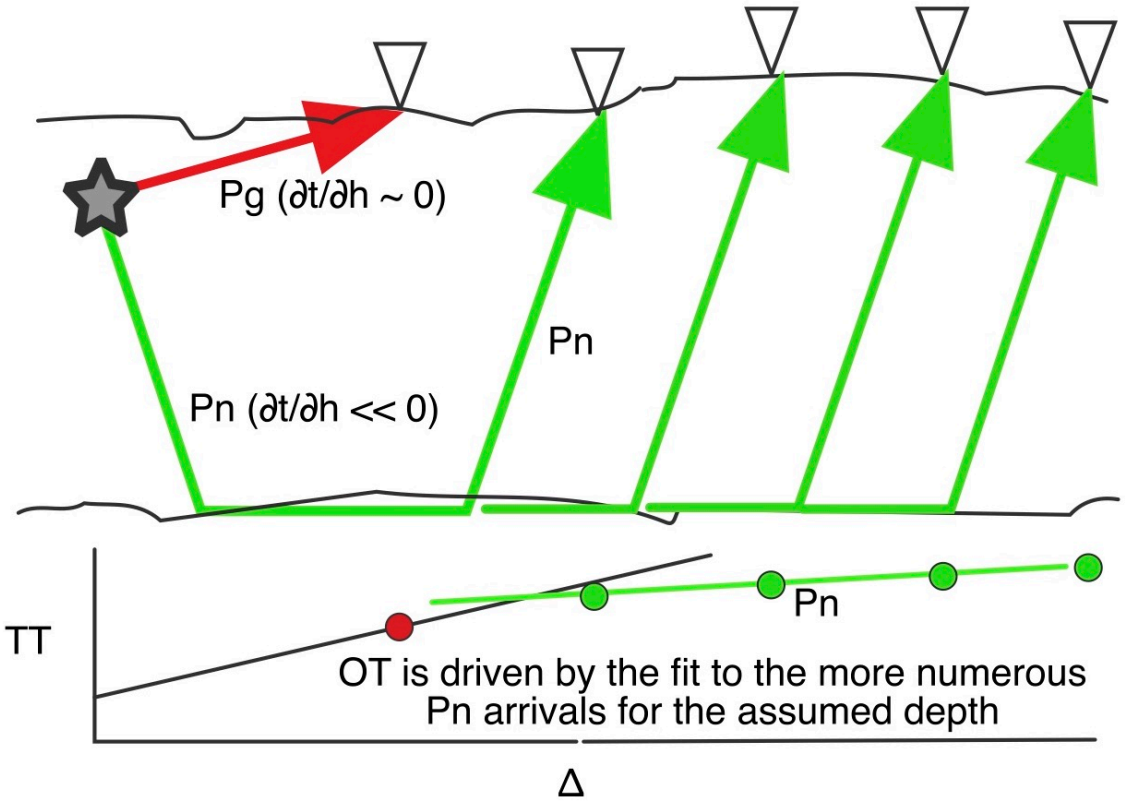


The "don't use" label for the Pn arrival in this cartoon refers to depth control and determination of the hypocentroid. It is fine to use all phases to constrain the cluster vectors. "Refining the crustal velocity model" is not as tricky as it sounds. Very few clusters have ever required more than two layers in the crustal model and a single layer crustal model works quite well for many. It requires much denser data sets than we usually have for **mloc** to resolve more detail in the crustal model and then it still makes very little difference in the final locations.

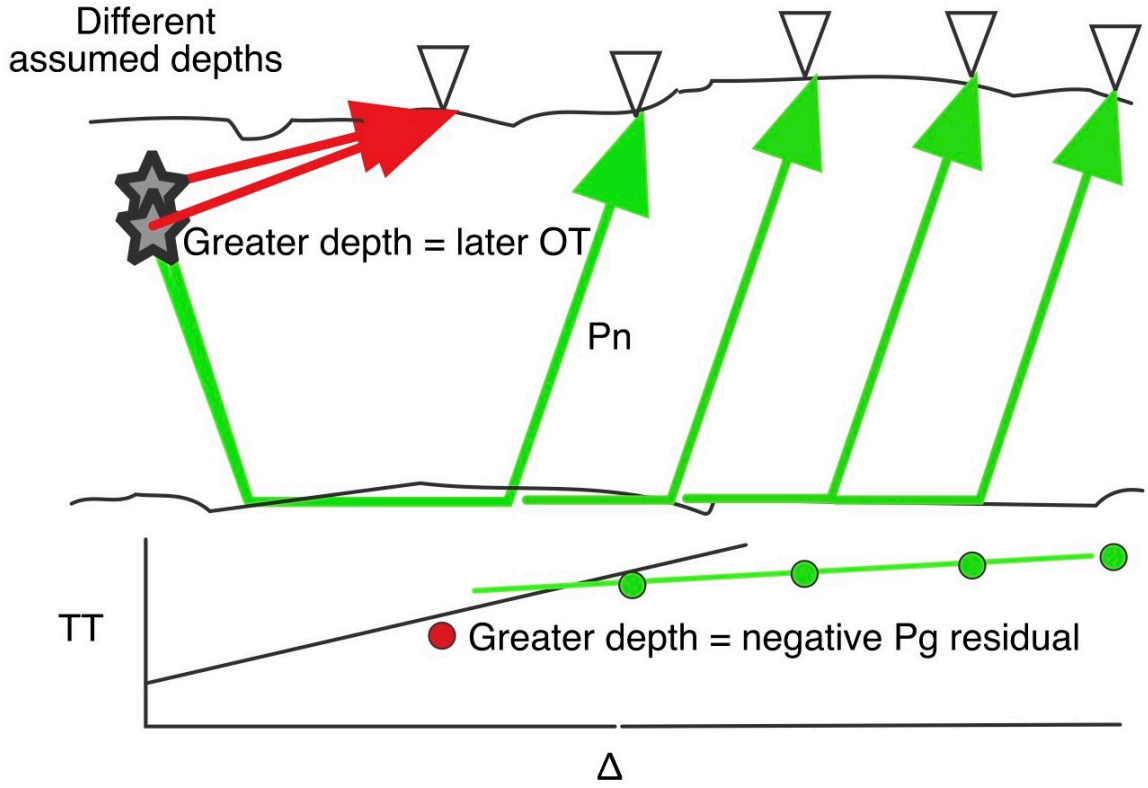
Next, considering the Pn arrivals for the control events, the thickness of the crust (Moho depth) and upper mantle velocity can be constrained. In continental regions it usually needs to be a bit thicker than the 35-km thick crust used in ak135.



Our non-control event may have only a few direct crustal arrivals, and those at distances beyond what is normally needed for depth constraint (so $\partial t/\partial h$ is near zero). If such an event is added to the subcluster of control events, its origin time will be driven by the fit of its Pn readings (steep take-off angles) to those of the control events, for the assumed focal depth.



What happens with different assumed depths for our non-control event? If the assumed depth is too great, the origin time must be later to still fit the Pn arrivals with shorter raypaths. The Pg arrival time has almost no dependence on depth (raypath length stays the same) but its residual *is* affected by changes in origin time. A later origin time makes the theoretical arrival time later, causing a negative residual for the observed arrival, the same sense as if the event were too far from the station. If this were a station on top of the event, we would say the focus should be shallower (closer to the station). The opposite occurs for shallower assumed depths, yielding a positive residual for the observed arrival. Focal depth of our non-control event is adjusted so that the direct arrivals are consistent with the travel-time model for crustal arrivals based on the control events.



The sensitivity of this method of constraining focal depth is not as high as the use of near-source arrivals, but it is still quite useful because it is so widely applicable. As a rule of thumb, when the depth is constrained with the local-distance method (command depl) I usually give it an uncertainty of 4 km, as compared to 3 km for depths constrained by near-source readings. Obviously, confidence will be higher when there are more data points. There are many cases in which events have readings over the entire distance range where direct crustal arrivals are the first-arriving phases, so it does not take long to develop a feel for the sensitivity of the two types of data to focal depth.

It is not just the Pn data that drive the change in origin time when focal depth is changed; all arrivals have some effect on this. For most events, however, the Pn data are the dominant influence. Except for the very largest events they are usually the most numerous and they normally have small empirical reading errors, leading to large importance in the inversion.

It is very important to remember the sequence that must be followed for this method to work:

- Identify control events, usually after an initial run of **mloc**.
- Working only with control events, do additional runs in direct calibration mode (preferably free depth inversions but fixed depth works too), using only direct-arriving crustal phases for the hypocentroid (command hlim), adjusting the crustal velocity model

and focal depths. Cleaning of outliers of the direct-arriving crustal phases is done now also.

- When the crustal model and focal depths of the control events are stable, consider the Pn data and adjust the crustal thickness to match the data. The baseline offset of Pn phase in the .ttsprd file can be helpful.
- Now non-control events that have at least one direct crustal arrival can be added (a few at a time is best) to the cluster, adjusting focal depth to obtain ~zero mean for the residuals of whatever direct-arriving crustal phases are available. Initial cleaning of outliers in the direct-arriving crustal phase dataset is done now.
- Try to avoid changing the crustal model after this point. If you do, you may need to go through these steps again.

Uncertainty of Focal Depth

Among the methods of constraining focal depth discussed here, reasonably objective measures of uncertainty are obtained from free-depth relocations and from the analysis of teleseismic depth phases, with the caveat that these estimates are influenced by the degree to which outlier readings have been handled correctly. For depths set according to the fit to near-source readings or local-distance readings, the uncertainty will be a subjective matter. There is no uncertainty associated with depths set according to a cluster default depth (command depc).

Free-Depth

The calculated uncertainty in focal depth of a free-depth relocation can be found in the ~.summary file and the ~.hdf file. Typical values are 0.5 to 3.0 km.

Teleseismic Depth Phases

For estimates of focal depth based on teleseismic depth phases, which are analyzed as relative depth phases in **mloc**, the uncertainties can vary greatly; in general uncertainty scales with depth because of the near total lack of confidence in reported phase identifications and the increasing separation of pP and sP arrival times with depth. For shallower events it matters less whether an arrival is actually a pP or sP phase. As always, having more samples helps. In practice, the depths determined with 3 or more consistent depth phases from different stations generally agree well with estimates of focal depth from other sources and with common sense.

The uncertainties in focal depths estimated from relative depth phases are usually asymmetric because there are usually two minima in the misfit function that are close to each other. The algorithm tests for the global minimum at 1-km depth increments so in the case of a single depth phase that could be either pP or sP, the choice of one over the other as the “best” is random. In setting a depth constraint with a depd command the user can decide that the other minimum is more likely, given what else is known about the depth range of the cluster’s events, and restate

the uncertainty accordingly. This is most easily done from the relative depth phase plot (rdpp command).

For larger events with many reported depth phases, the ambiguity of phase ID often collapses into a single well-defined minimum with an uncertainty of ±3-4 km.

Near-Source Readings

When judging the uncertainty of a focal depth determined by the fit to arrivals from near-source stations there are several matters to consider:

- For shallow events the raypath will be exclusively in the uppermost part of the crust, whose velocity structure is effectively impossible to constrain with the kind of data used in **mloc**. The velocity in the assumed crustal model may be quite wrong (usually too fast, so the focus will be forced deeper). On the other hand the raypath is very short, so maybe the amount of bias is small.
- Picking the arrival of a larger event at very short distance may be easy for the analyst but it may not be the signal that is observed and picked at stations further away. This tends to lead to negative residuals for the very closest stations, thus estimates of depth that are too shallow.

I usually assign an uncertainty of ±3 km to these estimates of focal depth. When both Pg and Sg readings are present and consistent, or if there are several stations within range, a smaller uncertainty can be contemplated but I think ±2 km is about the limit.

Local Distance Readings

The local distance method of constraining depth is subject to greater uncertainty than the near-source method, especially when the only available readings are at larger distances, say 70-100 km. I routinely assign local distance estimates an uncertainty of ±4 km, but there are certainly cases where a larger value could be justified.

Teleseismic Depth Phases

In most location codes teleseismic depth phases are handled like any other phase; a residual is computed against a standard traveltime model and then the hypocenter (especially focal depth) is adjusted to reduce the misfit. If the theoretical traveltime of the teleseimic P portion of the pP raypath exhibits any bias with respect to the true traveltime, however, the inferred focal depth will also be biased. In fact the teleseismic P branch in calibrated clusters is typically found to be biased (relative to ak135) by several seconds, as seen in this example from the Ridgecrest, California cluster. If the focal depth were based on fitting the observed pP and sP arrivals, it would be over-estimated.

For this reason, **mloc** converts the main teleseismic depth phase readings into *relative depth phases* pP-P and sP-P for analysis, removing most bias from the teleseismic P branch. The water-multiple pwP phase is also processed as the differential phase pwP-P. Any other depth phases

encountered (e.g., sS, pPn, pPKP, etc) will be processed "normally" if they exist in the phase set or flagged (with "p") if not. A section of the ~.log file lists the converted relative depth phases. The main source of information is the ~.depth_phases file.

The analysis described here has to be conducted outside the actual relocation, and the results brought into subsequent relocation runs by setting the depth with the depd command and relocating with fixed depth. Depth constraints from teleseismic depth phases cannot usually be utilized in a free-depth relocation (other than as starting depths) because the data are few in number and sparsely-distributed among events.

Bogus Depth Phase Readings

It has recently become known that certain seismograph stations have been reporting fictitious depth phase readings to the ISC and perhaps other data centers. It appears that analysts at these stations (or perhaps a regional data center) have been reporting theoretical arrival times of depth phases based on a preliminary hypocenter. A characteristic sign of this issue is to find several stations reporting both pP and sP with times that match perfectly with each other. It appears this problem began somewhere around the year 2000 but it may well be station-specific.

In **mloc** this problem is dealt with by maintaining a file of stations suspected of reporting bogus depth phase readings, which is referenced by the command bdps. It is strongly recommended to use this command for any cluster in which depth phase analysis will be employed to constrain depths. Data from suspected stations will be flagged in the depth phase analysis to prevent its use.

As a review of the file bdps.dat will show, the stations so far suspected of reporting bogus depth phase data are all in China, but there have been fears about this ever since routine analysis has come to be done with computer software capable of over-laying theoretical arrival times on observed records. There may well be other sources of spurious readings so vigilance is recommended.

Best-Fitting Depth

The method used in **mloc** to determine the optimal focal depth from reported depth phases is based on an important principle: that the reported phase names for depth phases from standard seismic bulletins cannot be trusted. That the reported arrival is even a proper depth phase is in considerable doubt but the phase names reported are so often in error that they should generally be ignored. This principle was established by E.R. Engdahl during his work on the EHB catalog, in which careful inspection of reported depth phases is a key element of the improved focal depth estimates for which that catalog is known.

After each run of **mloc** a trial is conducted over a range of depths (1-99 km, limits hard-coded in *mloc.f90*) to find the best fit to observed relative depth phases. At each trial depth, and for each relative depth phase time difference, the theoretical traveltime difference is calculated for all supported relative depth phase identifications: pP-P, sP-P and pwP-P. The phase with minimum

residual is selected, whether or not the phase name is consistent with the original report. The total RMS error from all depth phases is calculated at each trial depth, and the depth with minimum cumulative error is taken as optimal for that event in that run of **mloc**. The asymmetric uncertainty in depth (at ~90% confidence level) is calculated using the zM test (e.g., Langley, 'Practical Statistics, Simply Explained', Dover Publications, 1970, p 152.), to find the range of depths around the preferred depth in which the z statistic, relative to the value of z at the preferred depth, is less than the value at which the null hypothesis (no difference) would be rejected at the 10% level of probability.

The z Test for Measurement (hence 'zM') compares a random sample of one or more measurements with a large parent group whose mean and standard deviation are known.

The statistic z is calculated from z = (sqrt(n)·abs(M-m))/S, where:

- n = number of measurements in the sample
- m = mean of the sample measurements
- M = mean of the parent group
- S = standard deviation of the parent group

The probability of no significant difference between M and m at the 10% level of confidence is 1.64. This value is added to the value of z calculated for the depth at which z is a minimum to estimate the uncertainty range for focal depth. We assume M = 0.0. because we expect the relative depth phase residuals to have zero mean at the true depth. The specification of S is more problematic. The code assume S = 1., which seems sensible based on looking at residuals of many depth phase readings, but further investigation is warranted. This analysis is still relatively new in **mloc** and when a larger set of results is available this parameter may be revised.

The output from the test described above often reveals outlier readings which account for an inordinate proportion of the cumulative error. A cleaning process, i.e. flagging the outliers and re-running **mloc**, is used to bring the depth phase data into a self-consistent state. The process is somewhat subjective and experience with many events will lead to a good instinct for what can expected from standard (e.g., ISC-sourced) datasets.

It is certainly true that this analysis for focal depth can be criticized from several points of view but the results are generally consistent with other estimates of focal depth (when they exist) and with our expectations of what can reasonably be expected in terms of precision from the depth phase data extracted from standard global catalogs. Traditionally, this kind of fitting has been done "by eyeball". These are initial steps to try to put the analysis on a more statistically-driven footing.

Depth Phase Analysis Output

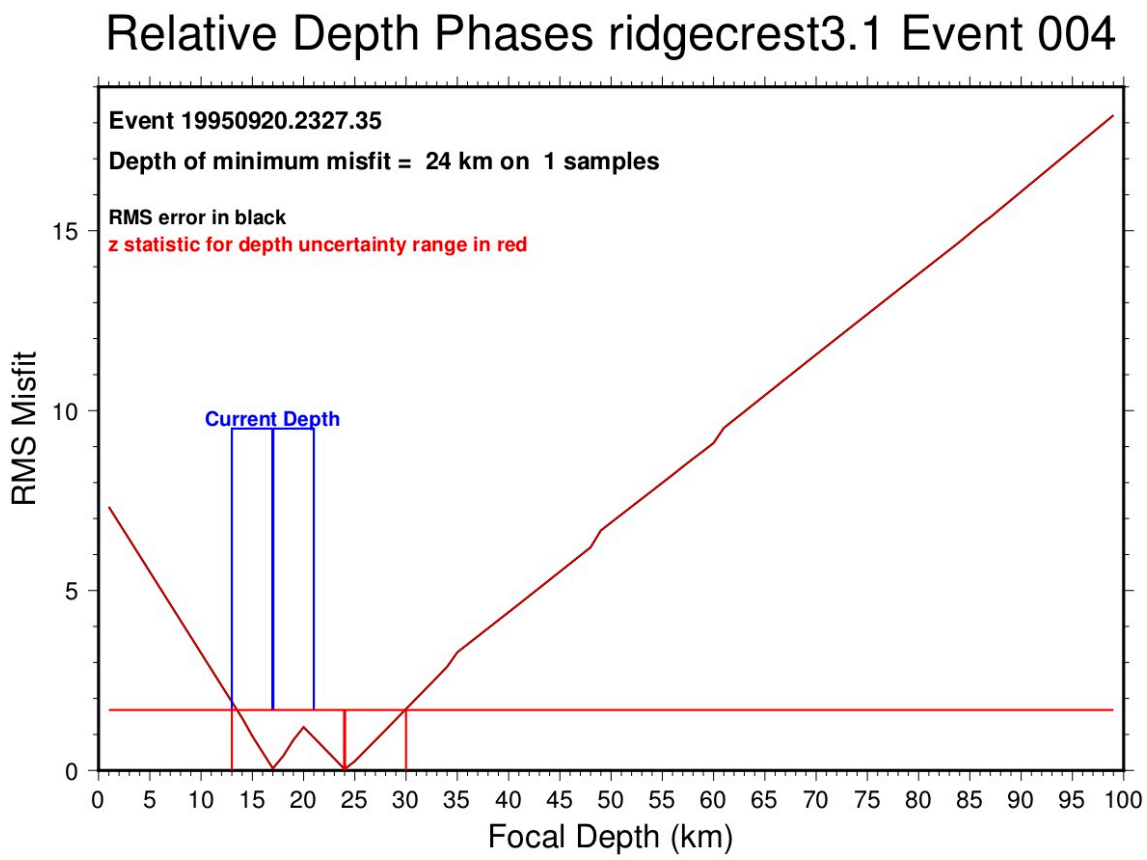
The results of the analysis described above are presented in two ways. Details for each event are listed in the beginning of the ~.depth_phases output file and the error-vs-depth curve is

optionally plotted using the rdpp command. Here is an example of the file output, for the Ridgecrest, California cluster:

```
  4 19950920.2327.35              preferred depth =  24 km ( 13 to  30) on   1 samples (depd  24  6 11)
 11 19961127.2017.24              preferred depth =   8 km (  4 to  17) on   1 samples (depd   8  9  4)
 27 20190704.1733.49              preferred depth =  19 km ( 12 to  23) on   4 samples (depd  19  4  7)
 45 20190706.0319.52              preferred depth =  22 km ( 13 to  26) on   5 samples (depd  22  4  9)

Residuals against each theoretical phase for the depth with minimum misfit
                        MNF      pP-P      sP-P     pwP-P     Err**2
Event   4 19950920.2327.35                 24 km
  GRF   83.65 sP-P      261      0.04     -2.90      0.04      0.00   pP-P
Event  11 19961127.2017.24                  8 km
  LVZ   74.26 sP-P      170      1.12     -0.02      1.12      0.00
Event  27 20190704.1733.49                 19 km
  ARCES 71.62 pP-P      644     -4.06     -6.26     -4.06     16.47 x
  EKA   73.43 pP-P      648     -0.11     -2.48     -0.11      0.01
  EKA   73.43 pP-P      649      0.04     -2.33      0.04      0.00
  EKA   73.43 sP-P      650      3.06      0.69      3.06      0.48
  NB2   74.94 sP-P      664      5.22      3.04      5.22      9.21 x
  HFA0  76.42 pP-P      679     -2.86     -5.04     -2.86      8.16 x
  FINES 78.76 pP-P      682     -0.76     -2.94     -0.76      0.58
Event  45 20190706.0319.52                 22 km
  ARCES 71.58 sP-P      532      7.30      4.67      7.30     21.84 x
  ARCES 71.58 pP-P      535     -0.05     -2.68     -0.05      0.00
  EKA   73.42 pP-P      562     -0.71     -3.50     -0.71      0.50
  EKA   73.42 pP-P      563      0.29     -2.50      0.29      0.09
  EKA   73.42 sP-P      564      5.82      3.02      5.82      9.13 x
  EKA   73.42 sP-P      565      6.12      3.32      6.12     11.04 x
  NORES 75.25 pP-P      583     -0.17     -2.96     -0.17      0.03
  NORES 75.25 sP-P      584      5.57      2.78      5.57      7.71 x
  FINES 78.73 pP-P      611      0.70     -1.90      0.70      0.49
  FINES 78.73 sP-P      612      8.65      6.05      8.65     36.55 x
```

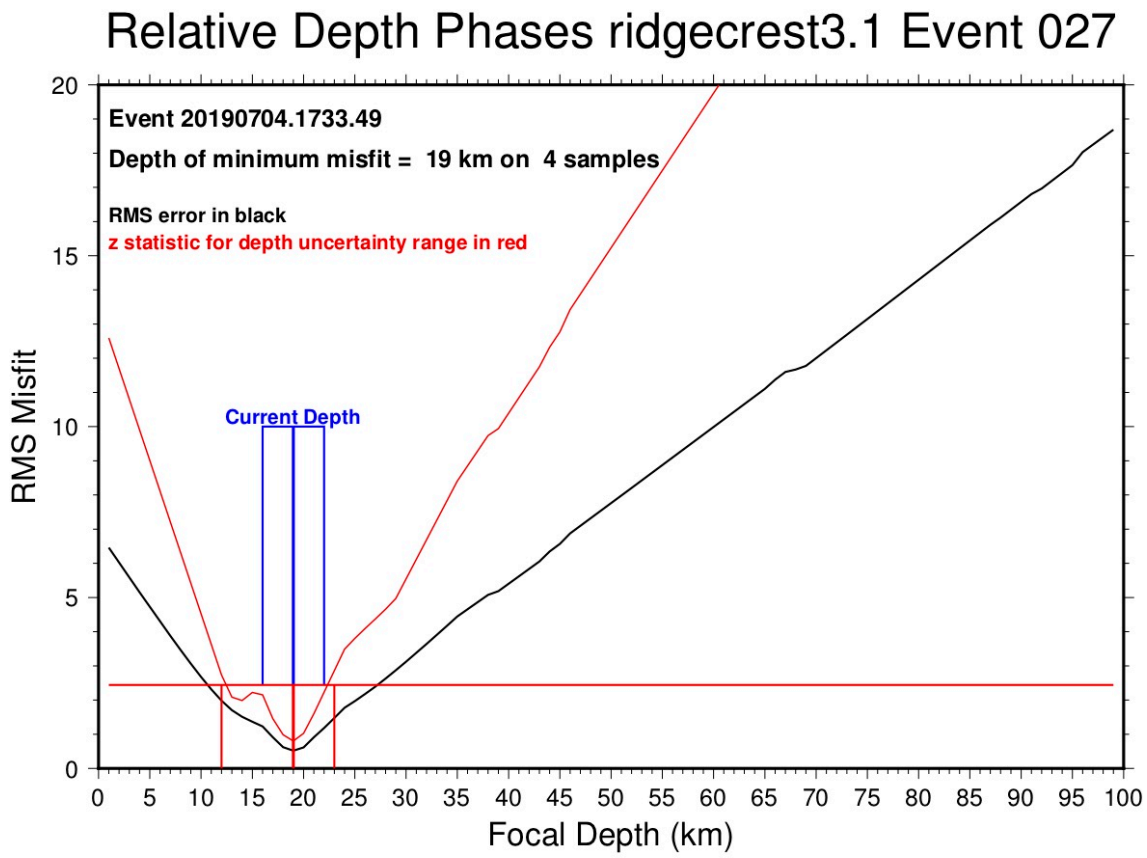
Four events in the cluster have depth phase data. First, the preferred depth for each event is listed, along with the range of acceptable depths (also formulated as a depd command for easy insertion into a command file). The following section shows every depth phase reading for each event, with the residual associated with each possible phase identification. For convenience the pwP phase is processed even for continental events but it has the same residual as pP and will not be proposed as the correct phasename. If the phase association at the depth of the minimum residual differs from the phasename in the input file, the preferred phasename is printed at the end of the line.

For the first two events there is only one depth phase reading. In this case the error curve in the corresponding ~_rdp plot has two minima, corresponding to the identification of the phase as sP or pP. One or the other will be slightly preferred because the test is only done at 1-km increments but the error curve plot will show that the range of acceptable depths includes both:



The plot includes two curves, one of the z-statistic (in red) and one (in black) of the total RMS error. For a single datum they are the same. The current depth and uncertainty range is shown in blue. In this case the depth has been set at the shallower minimum (17 km), but not because of the depth phase data; the depth in this run was based on the depl criteria (local distance readings) with an uncertainty of ±4 km.

When there are multiple reports of depth phases there is the possibility (likelihood, actually) of some instances of incompatibility, such as can be seen in the listings above for events 27 and 45. In each case several readings have been flagged after previous runs and now the remaining data are in fair agreement with each other. Experience with a number of events has led to the conclusion that when the squared error term of a reading exceeds 2-3 that reading should probably be flagged. If there are readings with especially large errors (e.g., the sP-P readings for FINES in event 45) they should be flagged first, leaving the “lesser” outliers for consideration after another run.



When there are multiple readings the minimum error will usually be non-zero. When outlier readings have been flagged the minimum will usually be less than 2-3. The curves of raw RMS error and the z-statistic will be distinct as well, and the z-statistic always gives a sharper minimum. In this case the preferred depth (19 km) has been set on the basic of near-source readings (depn), but that agrees very well with the depth phase analysis.

Example Clusters

This page links to example clusters provided with the **mloc** distribution package to illustrate various aspects of the usage of **mloc**.

- Jordan and Sverdrup Region A (basic uncalibrated teleseismic relocation)
- Mangyshlak PNEs (indirect calibration)
- Wells, Nevada (direct calibration, free depth)

Example Cluster: Jordan & Sverdrup Region A

In Jordan and Sverdrup (1981) the hypocentroidal decomposition method is applied to three small clusters of earthquakes in the south-central Pacific (Line Islands), labeled regions A, B and C. This example cluster is their Region A.

If you are not yet familiar with the naming conventions for clusters in **mloc**, please review the guidelines. The *cluster name* is "jsa" and the *series number* is "5". The output of only one run of **mloc** is included, so the basename is *jsa5.1*.

This is a very simple cluster with only four events. It cannot be calibrated because there is no close-in arrival time data or independent knowledge of the locations, so it is a basic teleseismic relocation that only provides an improved estimate of the relative locations.

The results of running **mloc** on the jsa5 cluster, either the output files included in the distribution or from a run you do yourself, will vary somewhat from what is presented in Jordan and Sverdrup (1981) (J&S). This is due to differences in the arrival time dataset, the travel-time model and the weighting scheme:

- J&S repicked some arrival times themselves.
- J&S used the Herrin (1968) travel-time model (P only).
- J&S used first-arriving P arrivals only, even for cluster vectors. No Pn or PKP phases were used.
- Readings with residuals greater than 5 s were discarded by J&S.
- J&S used reading errors of 0.7 s and 1.0 s, based on subjective judgement of quality.

The dataset in our example consists only of the readings available from the ISC Bulletin. In the example jsa5.1 run the ak135 model is used to estimate the hypocentroid, using P arrivals between 30° and 90°. Default reading errors (0.5 s for P) are used. Cluster vectors are estimated with all available phases at all distances. The windowing algorithm (command wind) is based on the default values for spread of different phases. For the P phase, readings with residuals greater than 1.5 s will be rolled off and readings with residuals over 2 seconds will be dropped from the relocation. In subsequent runs of **mloc**, we would begin using empirical readings errors from the ~.rderr file of a previous run (command rfil) and observed values for the spread of different phases from the ~.ttsprd file of a previous run (command tfil).

Example Cluster: Mangyshlak PNEs

The Mangyshlak cluster illustrates indirect calibration using a set of three Peaceful Nuclear Explosions (PNEs) conducted by the U.S.S.R. in western Kazakhstan, termed the "Mangyshlak" sequence by the Russians. Although nuclear tests are often thought of as "ground truth" the record-keeping for PNEs was not done to the standards typical of a major nuclear test site and there is still quite a bit of uncertainty concerning the precise hypocentral coordinates of many PNEs. Indirect calibration was done in this case as a consistency check on the reported epicenters and shot times of these three explosions. Details of this study can be found in Mackey and Bergman (2014).

The series included in the distribution is mangyshlak4. This cluster has already undergone the cleaning process in previous series of runs, so the command file *mangyshlak4.1.cfil* takes empirical reading errors and the observed spread of various travel-time branches from the previous run (commands rfil and tfil):

```
rfil mangyshlak3.5.rderr
tfil mangyshlak3.5.ttsprd
```

The two output files from mangyshlak3.5 are included in the mangyshlak4 directory.The full command file for mangyshlak4.1 is shown here:

```
pltt 1 2 3 6
ccat mangyshlak_summary.txt
rfil mangyshlak3.5.rderr
tfil mangyshlak3.5.ttsprd
hlim 30. 90.
clim 0. 180.
wind 3 4
frec 1 1 0 1
freh 1 1 0 1
comm elevation in the source area is 200 m
memb
even 19691206.0702.59
inpu 19691206.0702.59.mnf
depe 0.2 0.1
comm OT and depth from Sultanov
calf 7 2 59.85 43.867  54.800  0.2 0.0 0.01 0.01 0.04 1.0 ! Sultanov geodetic
calf 7 2 59.85 43.8625 54.7727 0.2 0.0 0.01 0.01 0.04 1.0 ! Mackey site visit
memb
even 19701212.0700.59
inpu 19701212.0700.59.mnf
depe 0.3 0.1
comm OT and depth from Sultanov
calf 7 0 59.83 43.85   54.80   0.3 0.0 0.01 0.01 0.04 1.0 ! Sultanov seismic
calf 7 0 59.83 43.9096 54.7937 0.3 0.0 0.01 0.01 0.04 1.0 ! Mackey site visit
memb
even 19701223.0700.59
inpu 19701223.0700.59.mnf
depe 0.3 0.1
comm OT and depth from Sultanov
calf 7 0 59.76 44.025  54.993  0.3 0.0 0.01 0.01 0.04 1.0 ! Sultanov geodetic
calf 7 0 59.76 43.8858 54.8973 0.3 0.0 0.01 0.01 0.04 1.0 ! Mackey site visit
```

The presence of two calf commands (a variation meaning the focal depth is calibrated but not the origin time) for each event is notable, and it leads to a warning from **mloc** during the run that there are multiple calibration commands for the event:

```
Enter commands:

: cfil mangyshlak4.1.cfil
  FYI from proc_cal_: event   1 19691206.0702.59 is declared as a calibration event
  FYI from proc_cal_: event   1 19691206.0702.59 has multiple calibration commands
  FYI from proc_cal_: event   2 19701212.0700.59 is declared as a calibration event
  FYI from proc_cal_: event   2 19701212.0700.59 has multiple calibration commands
  FYI from proc_cal_: event   3 19701223.0700.59 is declared as a calibration event
```

```
 FYI from proc_cal_: event   3 19701223.0700.59 has multiple calibration commands
```

mloc will use the last calibration command found for an event, so the warning should simply be taken as a reminder that the user should be sure that the order of commands is as intended. In this case the first calibration hypocenter is from a publication that attempted to compile the most reliable information for all PNEs. The second calibration location (epicenter) is the one measured with GPS by Kevin Mackey at the actual boreholes during a site visit in 2013, guided by the results of the **mloc** analysis.

Nearly all commands in **mloc** can be issued more than once in a command file, and the last instance will take effect. This is quite useful with dep_ commands, for there are often multiple estimates of depth for an event.

Another feature to note is the use of in-line comments (beginning with "!") in the "calf" commands to identify the source of the calibration data. In-line comments can be used with any command.

Even though the residual shift vectors (remaining errors in location after the calibration shift) are small by most standards, ranging from 250-600 m, inspection of the ~.cal file for mangyshlak4.1 reveals that the consistency of the relative locations determined by **mloc** is a bit marginal when compared to the calculated uncertainties. For example here is the first estimate of radius of doubt:

```
Radius of doubt test based on null hypothesis that all residual cluster vectors have zero length
Critical value =      5.2 at 90%

Individual event contributions to kocs2:
  1     4.28889
  2     0.44913
  3     0.12304
rdbt_test =   0.00; observed value =   4.86; null hypothesis cannot be rejected
rdbt1 =   0.00
```

Although the radius of doubt comes out zero, it is a close thing; the observed value of the test statistic (4.86) is almost as large as the threshold (5.2 at 90% confidence) for rejecting the null hypothesis that all residual shift vectors have zero length. The second test, based on "coverage" of the confidence ellipses indicates that the radius of doubt could be non-zero:

```
Radius of doubt based on coverage statistics

Threshold percentage of probability for radius of doubt test: 0.10
                  rdbt    nr    nrc    k    coverage   P(X  k)
rdbt_test =      0.000     3     1     2     0.333     0.028; null hypothesis is rejected
rdbt_test =      0.200     3     1     2     0.333     0.028; null hypothesis is rejected
rdbt_test =      0.400     3     2     1     0.667     0.271; null hypothesis cannot be rejected
rdbt2 =   0.40
```

This test is not actually appropriate in this case because of the number of data. It requires at least 10 calibration events to be applicable. For any number less there would need to be complete coverage in order not to reject the null hypothesis.

All statistical arguments in this case are on shaky ground because with only three events the power of our robust estimator to accurately estimate the empirical reading errors is rather limited.

The final aspect of the mangyshlak cluster to note here concerns the use of the ccat command, used to create a special pair of output files designed for easy import into the GCCEL database. These files are stored in the directory *mangyshlak4.1_comcat*. One, *mangyshlak4.1_plots.pdf* contains all the plots that were requested and the other, *mangyshlak4.1.comcat* is a text file containing all data and results in an MNF v1.4-formatted bulletin.

The argument to the ccat command is the name of a text file that is treated as a commentary on the cluster; it is inserted near the top of the *~.comcat* file. The text file can have any name; here it is named *mangyshlak_summary.txt*. It must be hard wrapped at some reasonable line length.

Example Cluster: Wells, Nevada

This cluster of 49 events is calibrated with the direct calibration method. Because of abundant arrival time data at very short epicentral distances for many of the events in the cluster (thanks to the temporary deployment of seismographs after the mainshock) the relocation could be done with focal depth as a free parameter for all events. This cluster also illustrates the tomo command.

From the ccat commentary file *wells_summary.txt* used in the GCCEL entry for this cluster:

```
The Wells cluster is named for the town of Wells in northeastern Nevada. The cluster is based on
a mainshock-aftershock sequence that began with an Mw 5.8 event on February 21, 2008.
Calibration, especially in depth, is greatly aided by data from temporary seismograph stations
deployed to the source region, beginning about 2 weeks after the mainshock. The relocation was
done with free depth. This cluster includes only the larger events that were located beyond local
distances. For a comprehensive study, see Nealy, et al. (2017).
```

The run of **mloc** included here (wells6.1) follows from extensive work leading to publication as a journal article and also in the GCCEL database, so little further improvement is expected. The ccat command has been disabled with a comm command in the command file *wells6.1.cfil*, so no *~_comcat* directory is created. The starting locations, empirical reading errors and spreads of phases are taken from output files of the last run of the previous series (wells5.12), which are included in the *wells6* directory.

```
rhdf wells5.12.hdf_dcal
rfil wells5.12.rderr
tfil wells5.12.ttsprd
```

The wells6.1 run converged quickly to solutions very close to the starting locations, as can be seen by the small values for the change of each hypocentral parameter of the hypocentroid and the cluster vectors at each iteration. This information is a major part of the output in a ~.summary file. Here is the section on the hypocentroid from *wells6.1.summary*:

```
                    TIME        LATITUDE     LONGITUDE       DEPTH          E**2/NSTA   SHAT  PRMAX  DELT PRES FLAG
START             12:26:17.14      41.166      -114.889        10.1

ITER 0                  0.01 S     0.02 KM      -0.00 KM       0.30 KM      42.8/  761   1.18    4.0  8927  474 1588
```

```
ITER 1                    0.00 S     0.01 KM      0.01 KM      0.08 KM      42.2/  761   1.17    4.0  8927  464 1588
ITER 2                    0.01 S     0.00 KM      0.00 KM     -0.02 KM      42.0/  760   1.16    4.0  8927  465 1588

CUMULATIVE                0.01 S     0.04 KM      0.01 KM      0.36 KM    AZIM =    0 DEG    DIST =  0.0 KM

SOLUTION            12:26:17.16      41.166      -114.889        10.4
STANDARD ERRORS           0.08 S     0.002 DEG    0.003 DEG    0.98 KM    ELLIPSE:  53.3 DEG   0.4 AND    0.5 KM
```

The epicentroid (epicenter of the hypocentroid) shifted by only a few tens of meters and the average depth went 360 meters deeper. Since this is a direct calibration the uncertainty of the hypocentroid sets the baseline for the uncertainty of all the events; the uncertainty of the cluster vector for each event will be combined with the hypocenter's uncertainty to obtain the final location uncertainty. The epicentroidal uncertainty seen here, 0.5 km is about as small as one ever sees for earthquake clusters relocated with standard bulletin arrival times.

Another way of reviewing how things changed is in another section of *wells6.1.summary* file, showing the changes in cluster vectors (overall, not by iteration). The entries for the first 10 events are:

```
                                          INITIAL                CHANGE                 FINAL
EVENT   1  20070228.1147.41     4.1 KM  AT  128 DEG     0.0 KM  AT    0 DEG     4.1 KM  AT  129 DEG
EVENT   2  20070228.1446.10     4.2 KM  AT  136 DEG     0.0 KM  AT    0 DEG     4.2 KM  AT  137 DEG
EVENT   3  20080221.1416.04     3.2 KM  AT  149 DEG     0.0 KM  AT    0 DEG     3.3 KM  AT  149 DEG
EVENT   4  20080221.1420.52     4.8 KM  AT  216 DEG     0.0 KM  AT    0 DEG     4.7 KM  AT  215 DEG
EVENT   5  20080221.1434.41     4.3 KM  AT  141 DEG     0.0 KM  AT    0 DEG     4.3 KM  AT  141 DEG
EVENT   6  20080221.1534.25     5.0 KM  AT  207 DEG     0.0 KM  AT    0 DEG     5.0 KM  AT  207 DEG
EVENT   7  20080221.1855.51     2.1 KM  AT  257 DEG     0.0 KM  AT    0 DEG     2.1 KM  AT  263 DEG
EVENT   8  20080221.1906.43     4.4 KM  AT  187 DEG     0.0 KM  AT    0 DEG     4.4 KM  AT  187 DEG
EVENT   9  20080221.1910.21     6.4 KM  AT  189 DEG     0.0 KM  AT    0 DEG     6.5 KM  AT  189 DEG
EVENT  10  20080221.2026.08     4.1 KM  AT  241 DEG     0.4 KM  AT  341 DEG     4.1 KM  AT  246 DEG
```

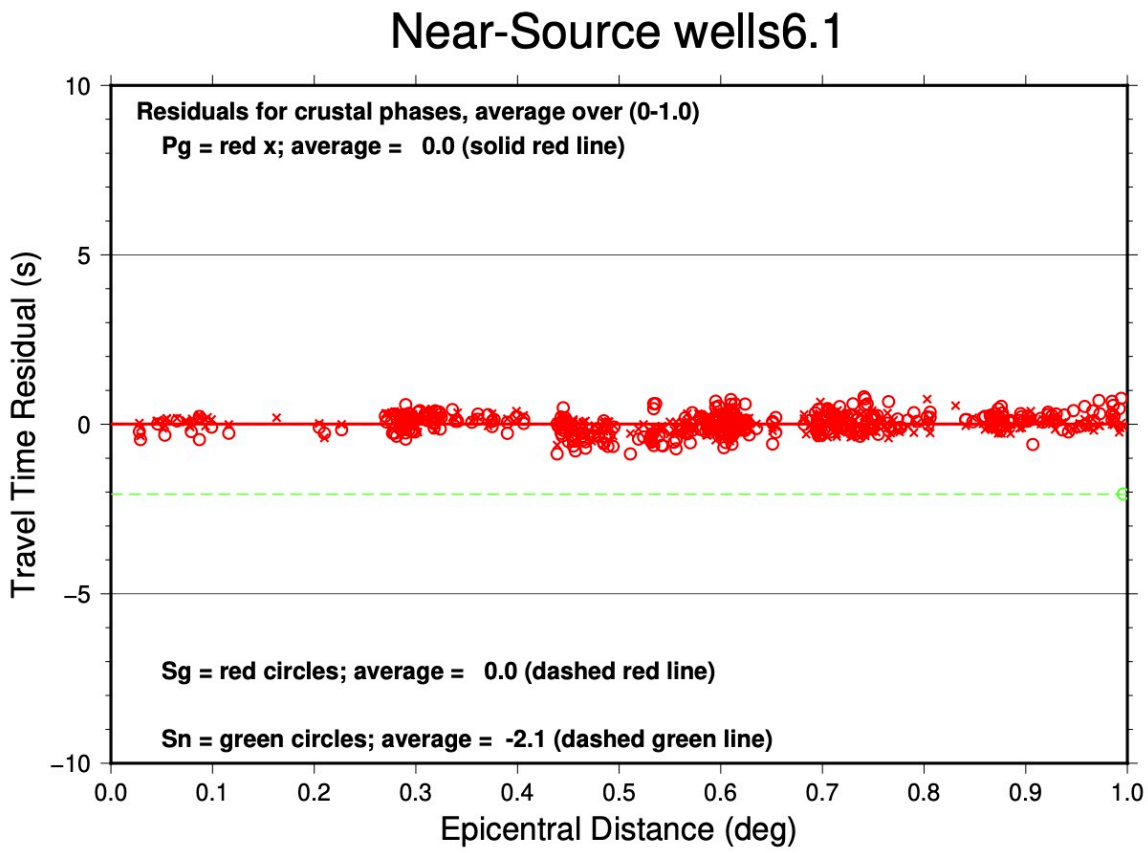
It can be seen that event 10 moved a little bit but none of the others did, and the remaining events all show no change (within the ~50 m resolution of the output format) in epicenter during the location. Further detail is available later in *wells6.1.summary*, in the individual event blocks.

Free Depth

The subject of focal depth is nearly always of interest in **mloc** relocation studies, but when a cluster can be relocated with depth as a free parameter it is of even greater interest. One drawback of the hypocentroidal decomposition algorithm is that it places rather stringent requirements on the nature of the arrival time dataset for a free depth relocation: basically, no event can be relocated with free depth unless they all can be. If any events are poorly connected among the readings that constrain focal depth (mainly, near-source stations) the inversion will probably fail to converge. Therefore, most relocations with **mloc** are done with depth fixed, but there are a variety of methods to independently constrain focal depth, including free-depth relocation of a subset of events that do have data sufficient to drive a free-depth relocation.

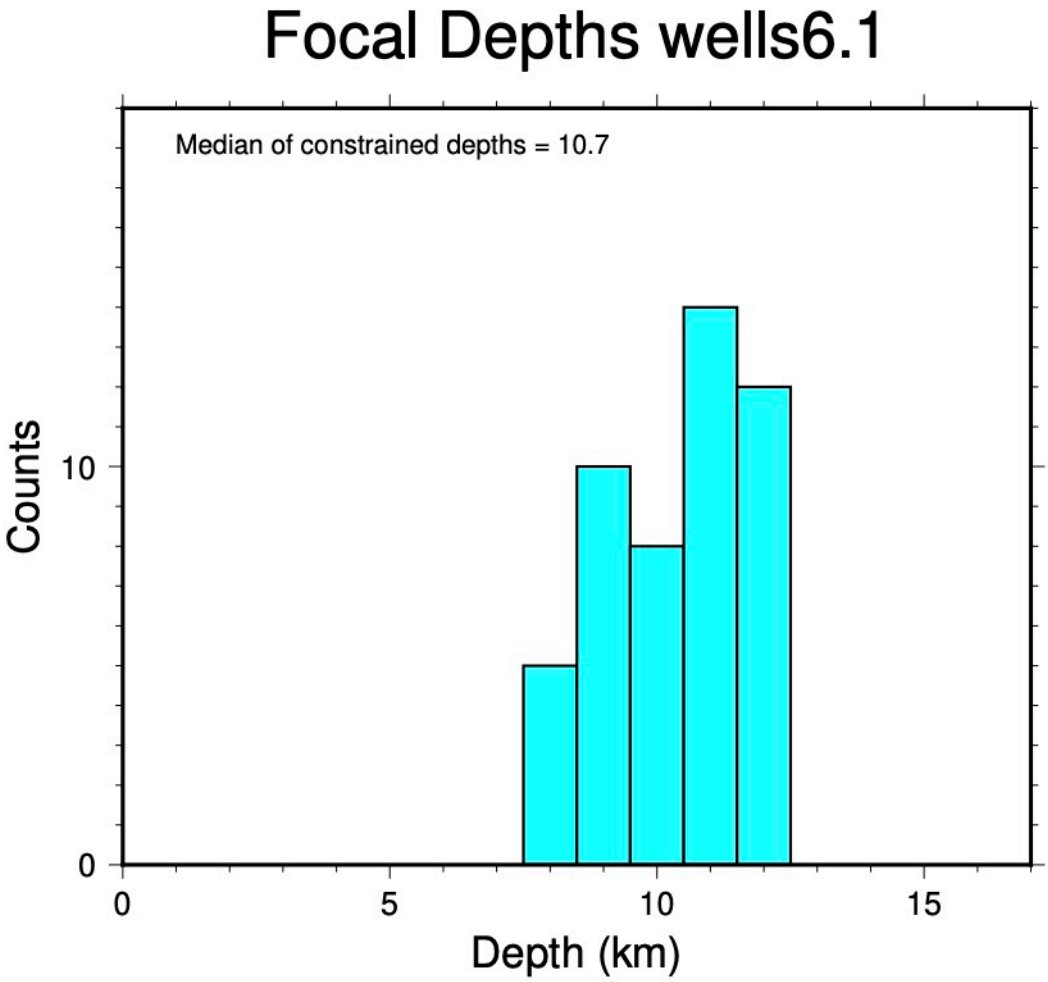
In the version of the Wells cluster presented here, all events are stable in a free depth relocation. A review of the ~.depth_phases file shows that every event has at least 2 readings at a distance range that provides some constraint on focal depth, and most events have many more than that.

There are other output files besides *wells6.1.depth_phases* that bear on the evaluation of the robustness of depth determination. One very powerful tool for quickly reviewing the status is the travel-time plot for near-source readings:



The main features we would like to see (all of which are satisfied here) are few outliers, no slope to the pattern, no curvature near zero and near-zero averages for Pg and Sg phases.

In a free-depth relocation the histogram of focal depths created by command fdhp is of special interest:



The tight pattern of resolved depths gives confidence in the results. Another aspect of this is to review the uncertainties of the depth estimates in the ~.hdf_dcal file. Here are the first 10 entries from *wells6.1.hdf_dcal*:

```
                                        Depth                                                  H+   H-
2007  2 28 11 47 40.75  41.14270 -114.85068  12.25 mf  1.60 3.4mb   11511719   16  234   34   1.00  0.14  1.1  1.1   0.3  21.4  14.6  39  0.51 129  0.56    0.9 CH01 Nevada
2007  2 28 14 46  9.51  41.13826 -114.85400  11.36 mf  4.40 3.1ML   11511822   16  169   15   1.02  0.14  1.1  1.1   0.3   6.3  16.9  40  0.52 130  0.56    0.9 CH01 Nevada
2008  2 21 14 16  2.04  41.14064 -114.86855  11.44 mf 10.00 5.8Mw   12942319    9  511   67   0.67  0.13  1.1  1.1   0.3 145.3   8.2  42  0.49 132  0.55    0.8 CH01 Nevada
2008  2 21 14 20 50.78  41.13145 -114.92061  10.68 mf 10.00 4.5mb   10542164    3   43    6   1.02  0.30  2.4  2.4   0.5 145.3  72.7  50  1.39 140  2.23    9.7 CH02 Nevada
2008  2 21 14 34 41.34  41.13560 -114.85666  12.44 mf  5.80 4.2mb   10623522   10  139   34   1.03  0.15  1.2  1.2   0.3 120.8  15.6  43  0.57 133  0.70    1.3 CH01 Nevada
2008  2 21 15 34 24.84  41.12585 -114.91570  11.05 mf 10.00 3.7mb   10623527   15  223   30   1.02  0.14  1.0  1.0   0.3 120.8  11.9  46  0.52 136  0.58    0.9 CH01 Nevada
2008  2 21 18 55 50.50  41.16363 -114.91386   9.44 mf  5.00 2.6ML   11324643   16   78    2   0.87  0.14  1.2  1.2   0.3   4.1  36.5  48  0.58 138  0.65    1.2 CH01 Nevada
2008  2 21 19  6 42.66  41.12638 -114.89507  10.09 mf 13.90 2.7ML   11324644   12   68    5   1.21  0.17  1.4  1.4   0.3   4.6  30.5  59  0.69 149  0.75    1.6 CH01 Nevada
2008  2 21 19 10 20.91  41.10845 -114.90102  11.13 mf 11.60 2.5ML   11324645   13   79    9   1.27  0.16  1.3  1.3   0.3   4.2  30.0  42  0.63 132  0.68    1.3 CH01 Nevada
2008  2 21 20 26  8.45  41.15103 -114.93307   9.44 mf  5.00 2.7ML   11324657   15   81    5   1.36  0.17  1.3  1.3   0.3   7.2  23.6  42  0.61 132  0.71    1.4 CH01 Nevada
```

I have added a header line to show the columns containing the converged depth and the depth uncertainty. **mloc** supports asymmetric depth uncertainties ("H+" and "H-") but in a free depth solution they are the same. These values are quite good, but it can be seen that the depth constraint for event 4 is a bit weaker than others. This is likely a reflection of the fact that it has

fewer readings than most, both for the connectivity through cluster vectors (43) and in the distance range used for the hypocentroid (3).

Tomography Files

In the example run of the Wells cluster the tomo command was included twice to produce output that is intended to be convenient for those conducting research in seismic tomography:

```
tomo Pn 3
tomo P 2
```

These output files are used as examples in the section on the content and format of ~.tomo files.

ISC Data

The multiple event relocation program **mloc** recognizes only one format for files carrying arrival time data: MLOC Native Format or MNF. No data center provides data in MNF format so it will be necessary for users of **mloc** to convert from other formats into MNF. This document describes the procedure for obtaining data from the ISC website and converting it into MNF format for use in **mloc**. The source codes of the utility programs used in this process are provided in the distribution and may serve as templates for conversion programs for other formats. Please contact me if you need help with a conversion program for some commonly-encountered format, as I may already have written one.

The basic steps are:

- Search the ISC Bulletin and save the output as a bulletin (one file) in ISF format
- Convert the ISF Bulletin into one in MNF format
- Search the MNF Bulletin to extract the events desired for the cluster, creating the event definition portion of the command file at the same time
- Add the "header" section of the command file to control the relocation

Search the ISC Bulletin

- Navigate to the ISC Bulletin webpage or the mirror site at IRIS (may be faster).
- Select the desired database. The "Reviewed ISC Bulletin" is of higher quality but is usually about 2 years behind real time. The "ISC Bulletin" will provide data for some events that are quite recent, often within the last day or so, but the quality of the data and the hypocenters is, well, unreviewed.
- Select "ISF Bulletin" as the output format.
- Choose search region, time period, and optional search parameters to narrow your search (but see the note below).

- Choose output options. The “Prime hypocenters”, “phases”, “magnitudes” and “headers” options must be checked. The “web links” option can be checked but it will not change the content of the output file that will be converted to MNF format. If the “comments” option is checked any comments in the ISC Bulletin output will be converted to comments in the MNF file, but in general I do not find these very useful.
- Click the “Search Bulletin” button. Soon a window will pop up and start filling up with data. It may take a while for all the data to transfer. At the end you will see a STOP line and some other output.
- Select everything in the output window, paste it into a new text document and name the file in some reasonable fashion. I’ll refer to it generically as the “ISC Search Bulletin”.
- Make a new directory in your “clusters” subdirectory for the new cluster, then a further subdirectory “Data”, and store the ISC Search Bulletin there.

The search region can be specified as a range in latitude and longitude or as a circular region. I tend to use the former if I’m just exploring a region and do not have a specific target event in mind. If I am trying to build a cluster around a specific event the circular search option is preferred, with a radius of 50-150 km, depending on the density and nature of the seismicity in the area.

I normally try to make the search as all-inclusive as possible: date range January 1, 1960 to today, all magnitudes, any number of readings, etc. I depend on the search of the converted MNF bulletin with the utility program mnf_search to weed out events that are unsuitable for the cluster, but then I know I have a full representation of all events in the search region if I change my mind about the selection parameters later.

The filename of the new text document created from the search of the ISC Bulletin is arbitrary but I try to pick a distinctive geographic name for the proto-cluster at this point. This can be aided by downloading the .kmz file that accompanies every search and opening it in Google Earth.

If I expect that I will be collecting arrival time data from other sources besides the ISC, e.g., the NEIC, a local network, a temporary network, etc., I will make additional subdirectories under “Data” for the different sources.

Location Accuracy Codes

This section ultimately explains the details of the code system (called GTCNU) used in **mloc** to describe the accuracy of hypocenters, but leading up to that is a rather lengthy and, to most people, tedious diatribe about the subject. All you really need to know is this: **mloc** figures out the appropriate code, based on how you’ve set up the relocation, and lists it with each hypocenter in several output files, notably the ~.hdf file. For **mloc** the calibrated class of codes is most important. Click this link if you want to skip to the description of the GTCNU code: Beyond GTX.

From Ground Truth to GT25

The terminology “ground truth” as it relates to seismology goes back at least to the late 1980s, in the context of developing a monitoring system for a proposed Comprehensive Test Ban Treaty (CTBT). The International Monitoring System (IMS) which has been developed to support the CTBT utilizes multiple technologies, but seismological analysis is a major component of the system, as it is for all national programs for nuclear monitoring. In recent years the concept of ground truth data sets has moved out of the nuclear monitoring community into the wider seismological research community. Although it has been used for several aspects of seismological research, the concept of ground truth is most often used in the context of event location, as a way to solve the “chicken and egg” problem of seismology: an accurate location cannot be determined without an accurate velocity model of the Earth, and an accurate velocity model cannot be determined without accurate source locations. Although the terminology “ground truth” was apparently not in use at the time, this approach was employed by Herrin (1968) in developing a P-wave travel time model for global earthquake location using arrival time data from nuclear tests.

Thus, ground truth events came to be thought of as seismic events for which all hypocentral parameters (origin time, epicenter, focal depth) are known with exceptional accuracy, i.e., having zero bias at the scales of interest to the relevant community, which I take here to be the community of earthquake seismologists, which largely includes the community of seismologists working on nuclear monitoring. In practice, the set of ground truth events is dominated by man-made explosions, especially nuclear tests, with the complication that this information has not always been shared by testing states. Such events are of exceptional value in research, and so there is great pressure (i.e., funding) to extend the geographic coverage of ground truth events beyond the, thankfully, limited number of nuclear test sites.

One way to do this is to relax the requirement that hypocentral parameters must be known with essentially zero uncertainty. A second approach loosens the requirement that all hypocentral parameters be known to equivalently small uncertainty. In between true ground truth events and events with significant, unknown errors in all hypocentral parameters (most seismic catalogs, for example) there is a class of events whose hypocentral parameters (or at least some of those parameters) can be arguably determined with uncertainties that are usefully small, if not zero.

This approach was reinforced by the needs of the portion of the CTBT community that deals with on-site inspections. Such inspections must be limited in geographical extent, (e.g., 1000 km^2) so the monitoring system must be able to pinpoint suspected violations of the treaty to within small, quantifiable limits. As a result the concept of “GT” events with significantly larger uncertainties in their locations than traditional ground truth events became entrenched in the monitoring research community, most notably in the form of the classification “GT5”. GT5 is taken to mean that the epicenter of an event is known within 5 km, and it has assumed the status of a litmus test in the earthquake location business. If an event qualifies as GT5 it’s of interest but there is often little effort made to try to say how much better than GT5 it might be. Nor is there much interest in events that fall just short of achieving GT5 status.

Traditional ground truth events, mainly nuclear tests and large chemical explosions, meet much tougher standards than GT5, of course, but everyone knows that that data set is unlikely to expand much. They are usually categorized as GT0, although GT1 or GT2 are sometimes encountered. The EHB global catalog of earthquakes (using the methodology of Engdahl et al., 1998), which has been analyzed specifically to eliminate the worst location bias, is thought to have an average epicentral accuracy level of about 15 km, except for subduction zones, where location bias can be considerably larger, so the GT5 standard represents a considerable challenge to seismologists. There are several important points to be made about GT5:

- Unlike GT0, where hypocentral parameters are expected to be known a priori, the designation of GT5 involves some kind of estimation or regression process, which always carries uncertainties. Therefore it should carry an indication of the level of uncertainty, such as $GT_{90}5$. This is often neglected in practice. The need for the introduction of such uncertainties to the once rather pure concept of "ground truth" is a huge, ugly camel's nose under the tent.
- The uncertainty of an epicenter is normally treated, not as a circle, but as an ellipse, derived from a covariance matrix in a least-squares analysis, although grid-search methodologies can yield even more irregular shapes. There is no standard on how the single scale length attached to a "GT" designation should relate to that ellipse. Should it be the semi-major axis? The semi-minor axis? The average?
- The concept of GT5 is easily extended to other scale lengths ("GTX"), and thus it has been extended, to as far as GT25. A cynic may wonder if this process has been encouraged by the desire to improve the apparent global coverage of "GT Databases" in Powerpoint presentations. If the EHB catalog can be reasonably thought of as GT15 or so, what does GT25 mean? How far from the original concept of ground truth are we prepared to go?
- GTX classifications deal only with epicenters and have nothing to say about the accuracy and uncertainties of focal depth or origin time. For the narrow application of guiding on-site inspections under a CTBT, that is all that is required. For many other important applications, such as developing improved crustal models, it is a crucial omission, and modelers are prone to assuming the best about their data sets.
- The determination of meaningful confidence ellipses, or other measures of location uncertainty, is critically dependent on correct knowledge of the uncertainties in the individual arrival time data. The gap between original ground truth and GTX is further widened by the degree (extensive in most cases) to which location algorithms fall short of this standard.

The Unfortunate Influence of the Bondar et al. Criteria

Bondar et al (2004), on which I am a co-author, has had the unfortunate effect of exacerbating these issues by presenting what many seismologists have taken to be an easy way to discover

ground truth events, using nothing more than network geometry criteria. I'm quite sure that each of the authors has a somewhat different opinion on the matter of how the results reported in that paper should be used, but there always seemed to be good agreement among us that we were trying to come up with some convenient rule-of-thumb guidelines for estimating epicentral accuracy from seismic bulletins (i.e., containing the arrival time data as well as hypocentral parameters) that normally carry no useful estimate of that property, and in particular those estimates would attempt to account for location bias as well as random uncertainty.

To my mind the criteria were most suitable for use as a screening mechanism to select events which could be considered as good candidates for further analysis to determine location accuracy at levels (with 5 km as a target, but always trying for better) relevant in research that depends on seismic sources with small and well-characterized uncertainties. I view it as extremely unfortunate that we did not propose a different nomenclature than GTX for classifying events processed with network geometry criteria. The information yielded by a network geometry analysis can be quite useful but it should not be co-mingled with either legitimate ground truth data, which are a *priori*, rare and precious, or with the results of detailed relocation analyses carried out specifically to determine locations with the greatest possible accuracy for as many hypocentral parameters as possible.

Beyond GTX

Based on the above considerations I believe the GTX formulation has become an impediment to seismological research and should be abandoned. Therefore I have attempted to design a new nomenclature that more clearly and accurately reflects what is actually known about a hypocenter. These categories and criteria are based on the current practices and capabilities in seismic source location and earth model construction in the nuclear monitoring and seismotectonic research community. These practices and standards will undoubtedly evolve. The categories are not intended to obviate the need for full specification of uncertainties in hypocentral parameters, only to provide a fairly convenient way to categorize seismic sources in ways that are of importance in research, and especially, to prevent the inadvertent assumption that some parameters are known more accurately than they really are.

Of necessity a new nomenclature must be more complex than the GTX formulation. The endpoint of more complexity is a full description of the location methodology and data analysis, especially regarding error analysis, and the means taken to reduce bias in all parameters. There is clearly no practical way to reflect that information in a simple nomenclature. Given the variety of possibilities that deserve to be distinguished I have found that eleven categories are needed to adequately distinguish the different levels of knowledge concerning location accuracy that exist in today's research environment.

- Eight of these categories deal with events that would currently be lumped under the GTX formulation.

- The other three categories deal with uncalibrated events, such that all events in any seismic catalog can be assigned a category that clarifies their status with regard to location accuracy.

A fundamental goal of the new nomenclature system is to re-establish the distinction between traditional ground truth and what are more accurately termed “calibrated” data sets. It is therefore helpful to think of the eleven categories as members of four classes:

- The ground truth class (two categories)
- The calibrated class (four categories)
- The network geometry criteria class (two categories)
- The uncalibrated class (three categories)

In my usage, calibrated data sets are the result of location analyses that aggressively seek to minimize bias in one or more (but as many as possible) hypocentral parameters. The degree to which this can be done depends on the available data and is therefore quite variable. Researchers who would use calibrated data sets as input for other investigations, such as in tomography, should be certain they understand the nature of their source data with regard to location accuracy. Seismologists who generate such data sets should assist those researchers by using a well-designed nomenclature to characterize the accuracy of their results.

Ground Truth, the GT Class

The GT0 nomenclature is reserved for what I have termed legitimate (or original or traditional) ground truth, events for which all four hypocenter coordinates are known *a priori* at levels of accuracy which are negligible for the purpose at hand, which is taken to be among the purposes of most current earthquake seismology and nuclear monitoring research. For the location parameters (epicenter and depth) these uncertainties are typically less than a few hundred meters. At a typical crustal P velocity of 6 km/s 100 meters represents 0.015 s of travel time, so origin time should be known to several hundredths of a second in order to be compatible. These limits may not be suitable for some engineering purposes or specialized source studies.

There is a need for a somewhat relaxed GT category, because even though the hypocentral parameters of a man-made explosion may be given *a priori*, the uncertainties may not meet the stricter requirements given above. This may be the case because of inadequate record keeping or the difficulty in carrying out suitably accurate surveying or timing prior to the availability of GPS technology. The GT1 category is meant for such cases. This still implies near-certain knowledge of location within a kilometer or so, with comparable uncertainty in origin time (several tenths of a second). Industrial explosions and even some nuclear tests may not meet this standard. Such events ought to be treated in the calibrated class of events, as discussed below, rather than being assigned GT status with inflated scale lengths.

GT5 should not ever be used in this classification system, which will help avoid confusion with the current GTX system of classification.

Calibrated Events: the C Class

In contract to the GT class, where the concern is primarily with the scale of random error in the hypocentral parameters, the class of calibrated events is dominated by concern that the estimation process which has been used to determine hypocentral parameters may have introduced significant bias. Therefore we are very much concerned about minimizing bias and understanding which hypocentral parameters may be treated as effectively bias-free. Obviously we also desire to estimate the hypocentral parameters such that the formal uncertainties (driven by uncertainty in the data), usually expressed in Gaussian terms, are as small as possible; this will be handled similarly to the "X" in the GTX formulation, discussed below in the section "Scale Length".

A very important point about the calibrated class of events is that it includes only events for which the epicenter (at least) has been determined in such a way as to minimize bias. Although a bit unsatisfying in a logical sense, this policy reflects the reality that the seismological community overwhelmingly thinks of GT events (using the popular current nomenclature) as referring only to the epicenter. The other important point to be made is that this class requires an actual location analysis, not just the application of some set of network geometry criteria such as those presented by Bondar et al. (2004). In other words, application of network geometry criteria to estimate location accuracy is a precursor to calibration analysis, not a substitute for it.

Given that we do not know the Earth's velocity structure to sufficient accuracy, the only way to reduce bias for an event that was not engineered by man is to keep path lengths through the unknown Earth structure as short as possible. In other words only near-source data should be employed for estimating calibrated parameters. The concept "Near source data" is not restricted to seismological stations at short epicentral distance, although that is by far the most common case. Mapped surface faulting, treated with all due geological sensitivity, may serve as near source data for the purpose of constraining an epicenter, as may InSAR or other types of remote sensing analyses, since the ultimate signal (e.g., surface deformation) is not subject to bias from unknown Earth structure. InSAR analysis, through modeling of the distribution of rupture on a fault plane, may be used to reduce bias in focal depth. Waveform modeling (even at regional or teleseismic distances) may similarly provide useful near-source constraint on focal depth through analysis of the interference of direct and near-source surface-reflected phases.

Unfortunately, there is no methodology for obtaining usefully-calibrated hypocenters for deep earthquakes because every available data type must propagate through an excessive volume of material with insufficiently well-known velocity. The exact definition of "deep" in this context must be evaluated on a case-by-case basis, but it probably includes any event deeper than about 100 km. If uncertainties in velocity structure (and their effect on raypath geometry) are honestly propagated into the uncertainties of the derived location parameters, then the issue will be resolved by the increasing uncertainty of the location, leaving aside the question of bias.

The nomenclature I propose for the calibrated class is based on the following practical considerations about the calibration of the various hypocentral parameters:

Epicenter

Bias in epicentral coordinates can be minimized by means of seismological analysis (typically a location analysis), as well as by other means, including geological and remote-sensing analyses and *a priori* knowledge of human-engineered sources that may be too weak for GT status. It is quite common for the epicenter to be the only hypocentral parameter of an event that can be usefully constrained with minimal bias.

Depth

Focal depth is more difficult to constrain than the epicentral coordinates. In the location analysis, it requires data at epicentral distances comparable to the focal depth itself, a few tens of kilometers for crustal events, a much stricter requirement than that for the epicenter, which can be usefully constrained with stations 100 km or so away. This distance requirement can be ignored for waveform modeling, however, as well as for analyses of teleseismic depth phases, most famously emphasized by the EHB algorithm (Engdahl et al., 1998). Therefore the minimization of bias in focal depth can be part of the general location analysis, coupled with the estimate of a minimally-biased epicenter, or it can be constrained independently, even when the epicenter may be uncalibrated.

Origin Time

Calibration of origin time is only fully possible when both the epicenter and focal depth can be calibrated. Unless it has been specified *a priori* for a human-engineered event it must be estimated from seismic arrival time data at the shortest possible epicentral distances, and any bias in the location parameters would propagate into origin time. It is quite common, however, to encounter cases where the epicenter and origin time of an event can be constrained with near-source data (not necessarily for the event in question but through linkage to other events in a multiple event analysis), but the focal depth of the event cannot be usefully constrained, other than as an average depth for a cluster of events, some of which have well-constrained depths, or through regional seismotectonic considerations. In this case the origin time itself cannot be considered to be unbiased, but since it is reliably coupled to the assumed focal depth, the combined hypocentral coordinates can still provide valuable information on empirical travel times from a specific point in the Earth.

Given the above considerations there are four cases that need to be distinguished in the calibrated class of the nomenclature:

Calibrated Location Codes

Code	**Epicenter**	**Focal Depth**	**Origin Time**
CH	Calibrated	Calibrated	Calibrated
CT	Calibrated		Calibrated
CF	Calibrated	Calibrated	

CE	Calibrated		

CH (“H” refers to hypocenter). All four hypocentral coordinates have either been inferred by means that yield minimally-biased estimates or constrained *a priori* (as in some human-engineered events that don’t quite qualify for GT1 status or better).

CT (“T” refers to travel time). Epicenter has been calibrated; depth has been fixed at some assumed value (e.g., the average depth of nearby events with constrained depths); the estimate of origin time is based on local-distance data, but relative to an uncalibrated depth. Neither the focal depth not origin time can be considered calibrated in themselves but the combination can be used to estimate empirical travel times from the specific point in the Earth. Such events are not quite as valuable as CH events but still have considerable value as input to model-building exercises or as validation events.

CF (“F” refers to focal depth). Epicenter and focal depth have been calibrated, but not origin time. An example could be an InSAR location for an event and depth calibrated either by an additional analysis of surface deformation to infer distributed displacement on a fault surface, or through waveform analysis. The estimate of origin time is not based on near-source readings. These events can be used in validation exercises where their epicenters are compared with locations done with ray-traced travel-times through a model.

CE (“E” refers to epicenter). The epicenter is calibrated. As with the CT category, depth has been fixed at some assumed (albeit reasonable) value. If the calibration of the epicenter has not been based on near-source seismic data (e.g., an InSAR location), the estimate of origin time must be based on regional or teleseismic arrivals and therefore cannot be considered calibrated, nor can it be used for the estimation of empirical travel times. These events can be used in validation exercises where their epicenters are compared with locations done with ray-traced travel-times through a model.

Network Geometry Criteria: The N Class

Events in the N class are not considered to be calibrated in the sense defined here, but the arrival time data set has been processed with some network criteria (e.g., Bondar et al. (2004), but others are developing similar criteria for different source regions) based on simple metrics such as number of readings and distribution of reporting stations, in order to provide an estimate of epicentral accuracy that is expected to account for systematic location bias. The assumption here is that 1) the data do not permit a calibration analysis because there are insufficient near-source data, or 2) that such an analysis has simply not yet been done (i.e., a bulletin has simply been scanned for candidate calibration events). If a careful relocation analysis has been done to standards that can arguably justify classification as a calibrated event, the C class should be used.

NE (“E” from epicenter). The epicentral accuracy has been estimated with an appropriate network geometry criteria. Focal depth and origin time are uncalibrated. Many so-called “GT Catalogs” are dominated by events in this category. Requires a scale length, confidence level optional.

NF (“F” from focal depth). As NE but focal depth is calibrated. Requires a scale length, confidence level optional.

Everything Else: The U Class

All seismic events that do not fit into one of the GT, C or N classifications are considered uncalibrated. That does not mean that none of the hypocentral coordinates are calibrated, only that the epicenter is not considered to be calibrated. The following classifications are defined:

UE (“E” from epicenter). No hypocentral parameters are calibrated but there is a credible estimate of epicentral accuracy from a location analysis (confidence ellipse), leaving aside the question of systematic location bias. Requires a scale length, confidence level optional.

UF (“F” from focal depth). As UE, but focal depth is calibrated. The subset of events in the EHB catalog that carries depth estimates based on analysis of teleseismic depth phases would fall into this category, as would any event that has been the subject of a waveform modeling exercise that solves for focal depth. Requires a scale length, confidence level optional.

U (uncalibrated). Simply a dot on a map. No credible information is available on location accuracy, epicentral or otherwise. No scale length or confidence level is used.

Scale Length

With the exception of the “U” category all classifications should carry a scale length, equivalent to the “X” in the GTX formulation. The ground truth (GT) class categories are defined with specific scale lengths, which refer to the uncertainty in both the epicenter and focal depth.

For the Calibrated (C) and Network Geometry Criteria (N) classes the scale length is related to the uncertainty in epicenter only. For the CH class one would have to refer to a more detailed description of the data set to learn anything quantitative about the uncertainty in focal depth. The scale length is an integer, in kilometers, related to the uncertainty of the epicenter. Network geometry criteria always yield a single value for scale length. For the C class, as discussed above, there is no consensus about how the 2-dimensional uncertainty in an epicenter should be reduced to a single number. Three possibilities that seem reasonable when dealing with an ellipse with semi-minor axis a and semi-major axis b are:

- Nearest integer to the semi-major axis length of the confidence ellipse: nint(b)
- Nearest integer to the average of the two semi-axis lengths: nint((a+b)/2)
- Nearest integer to the radius of the circle with the same area as the ellipse: sqrt(ab)

For a circular confidence region all three methods are equal. As the ellipticity of the confidence region increases, there will be substantial differences between the different scale lengths, but the first method will always yield the largest value. For a confidence ellipse with semi-axis lengths 1 and 5 km, for example, the scale length calculated with the three methods would be 5, 3, and 2 km, respectively. I have adopted the first method for my own work, in order to be more conservative in the “GT5” wars, and I recommend the same to others.

Scales lengths larger than 9 are permitted, but I think they have rapidly diminishing value in the current research environment. When the scale length of confidence ellipses moves into double digits, one ought to begin to worry about the legitimacy of the assumptions underlying the statistical analysis. I would consider moving such events into one of the uncalibrated categories.

Confidence Levels

As Bondar et al. (2004) pointed out, it is necessary to specify the confidence level that has been used in determining epicentral uncertainties, e.g., as a subscript in the form "$GT_{90}5$" to indicate that the confidence ellipse was calculated for a 90% confidence level. Compliance on this point seems to be casual at best. It is admittedly awkward to include the subscript in computer output, and since the nomenclature I am proposing here is primarily intended to be carried in digital files, I leave it as optional in that context, with the strong recommendation to clarify the issue in accompanying documentation. I strongly encourage the use of the subscript in published reports to ensure wide understanding.

Nomenclature of Nomenclatures

It is inevitable that some shorthand will be needed to refer to the set of classifications defined above, in particular with respect to the widely-known GTX system. I suggest "GTCNU" or "GTCNU System of Classification" or "GTCNU System of Classification for Location Accuracy", depending on the degree of abbreviation desired.

Phase Identification

mloc will use any phase for which a theoretical travel time can be provided, but travel time models always come with a set of "legal" phase names. Other than the teleseismic P and S phases, naming conventions for seismic arrivals have varied widely over time and among different organizations. There are also many cases in which arrival times are provided without any phase name. **mloc** contains a number of routines that process the the arrival time data read from event files to put them in a form which is usable for relocation.

The correctness of (most) phase identifications is less important than the consistency of those identifications, because most arrival time data are only being used as time differences for the estimation of cluster vectors.

Phase Set

The canonical set of phase names for observational seismology has been established by a working group in IASPEI. For most phases, **mloc** calculates travel times and derivatives using Tau-P software implementing the global 1-D model ak135 (Engdahl et al., 1998), which supports only a subset of the canonical set. When a custom crustal model is used (command lmod), the set of returned phases is limited to Pg, Pb, Pn, Sg, Sb and Sn, and the "Conrad Discontinuity" phases Pb and Sb are normally suppressed (see below). In addition, **mloc** can process Lg and T phase data, as well as S-P times and relative depth phase (e.g., pP-P).

Probability Density Functions for Phase Identification

When an arrival time reading occurs in a portion of the travel-time curve where only a single phase exists, the assignment of a phase name is trivial. Where two or more phase branches cross or come near one another, however, the problem is more difficult. in the early 1990s Ray Buland proposed using probability density functions (PDFs) for specific seismic phases in deciding, on the basis of arrival time only, what phase name should be assigned to a reading when this occurs. This approach was implemented in a limited way (for depth phases only) in the EHB algorithm (Engdahl et al., 1998). If a reading is in the vicinity of several possible phases, each of which has a separate PDF with varying amplitude and spread, the decision is made by assigning proportional segments of the number line 0-1 to the candidate phases and generating a random number on that range to make the selection. Thus it's possible to select a phase which has lower probability over the most likely candidate. From the point of view of building catalogs in which the distributions of phases are free of cross-contamination around crossing points, this strategy makes great sense, but it is less obvious that it is a good way to obtain the most accurate locations for individual events. In any case, the necessary PDFs have never been developed for most phases.

MLOC's Strategy

Phase identification is done for single readings, i.e., there is no concept of groups of readings (primary and secondary readings at a station) being analyzed simultaneously. This was tried in earlier versions of **mloc** but it was found to be too complex. The current version of **mloc** uses a traditional "best fit" approach, but the code exists to take advantage of information on the probability distribution functions (PDFs) of different phases. The necessary research to establish the appropriate coefficients of the PDFs of all the needed phases has not been done. Therefore, at this time all PDFs are the same and the choice boils down to the classical "smallest residual" criterion.

The target arrival time is tested against all phases in the theoretical TT model for the corresponding focal depth and epicentral distance, regardless of arrival time order. Probability is calculated for each possible association, based on the candidate phases's PDF, and the choice is made on the basis of highest probability. Depth phases should be handled separately if they are to be used for depth constraint. Phase type (P or S) is honored if a phase name has been provided in the event file. If the phasename matches the name of the first-arriving phase at that distance it is not changed. A phase name is not changed unless the probability of the new phase identification is 0.05 or greater.

Depth Phases

The details of how teleseismic depth phases are analyzed in **mloc** to constrain focal depth are discussed elsewhere. With respect to phase identification the recommended approach is to use the ppri command to keep **mloc** from changing the phase name of pP and sP phases that are read

from the event file. If they are to be renamed it should be done manually (editing the event file) after a suitable analysis.

Unknown Phases

Some arrival times come with a phase identification indicating that no phasename could be assigned or a phasename that cannot be translated into a legal one. If **mloc** is unable to associate such a reading with any legal phase, the reading is set to "UNKNOWN", or in the case where a P or S character is indicated, "UNKNOWNP" or "UNKNOWNS". Such readings will receive a "p" phase reading flag as well.

Phase Reading Flags

The MNF file format for earthquake arrival time data files includes the concept of "usage flags", in column 3 of phase reading records. Usage flags are used by **mloc** (in combination with explicit commands and built-in logic) to determine which phase readings are available to be used for relocation. If there is a valid flag in column 3 of a phase reading, the reading will not be used; the availability of different flags is simply informational. Some flags (e.g., most flags for duplicate readings) are added by **mloc** during processing. Much of the work in a relocation with **mloc**, done by the user between runs, is identifying and flagging outlier readings (i.e., cleaning). This section describes the meanings of the available flags.

The absence of a flag (blank in column 3 of an MNF phase readings record) indicates that there are no known reasons to avoid using it, but this does not guarantee that a particular reading will be used in the relocation. For example, the hypocentroidal decomposition algorithm requires that a station-phase be observed more than once among the events in a cluster, so that a travel-time difference can be calculated, in order to use it to estimate the cluster vectors. If a reading is the only sample of a particular station-phase in the data set, it might still be used for estimating the hypocentroid, but if it does not meet the criteria (e.g., epicentral distance) for use in the hypocentroid, it will play no role in the relocation analysis. The usage (or not) of every reading can be found in the ~.phase_data output file.

Any phase reading with one of the flags defined below in column 3 of its phase record will not be used by **mloc** for either the hypocentroid or cluster vectors.

Phase Reading Flags

Flag	**Summary**
d	Duplicate reading
m	Missing station
p	Problematic phase
s	*skip* command
t	Timing problem

x	Outlier reading

Duplicate Readings

There is some rudimentary logic in **mloc** that attempts to identify duplicate readings automatically, but it is far from thorough. Duplicates can also be flagged manually. Multiple samples of the same station-phase from the same event are not duplicates if they are actually independent estimates, but it can be very difficult to determine from the data itself. This has become a serious problem at the ISC in recent years, as the multiplicity of communication channels for seismic data leads to them receiving reports of the same events from many sources, some of which may have made their own pick from the waveform.

Missing Station

If the user knows in advance that she does not have station coordinates for a station in the arrival time dataset, she can use this flag to explicitly state that fact. It is mainly a matter of good record-keeping because **mloc** will drop a reading anyway if it cannot find station coordinates.

Problematic Phase

This flag indicates that the phase name of the reading is problematic (indecipherable) or the seismic phase itself is problematic. It could be that the phase identification algorithm in **mloc** fails to associate the reading with a known seismic phase, or it may be that the phase, while known, is one for which it is not possible to calculate a theoretical travel time (e.g., PPP in ak135). This flag is applied automatically to depth phases reported from epicentral distances less than 26°, which are known to be subject to bias from upper mantle structures, and for depth phases reported by stations suspected of reporting bogus depth phase readings (command bdps).

skip Command

Using the skip command, readings can be flagged on the basis of phase name, station, author or a combination of these parameters.

Timing Problems

This flag is used if the station is known or suspected to have timing problems. There is documentation of timing problems for a few stations during known periods, and **mloc** has logic to check for corresponding readings and flag them automatically. The flag can also be applied manually.

One use of the timing problem flag has been to deal with readings from strong motion instruments that do not have calibrated timing systems. The P and S arrivals can be read from nearby earthquakes and a new phase record for S-P can be added to the MNF file while the original P and S phase records are given the *t* flag.

Another use has been in dealing with suspected one-minute errors (and on occasion, one-hour errors) in reported arrival times, which used to be quite common in the analog record era. To correct a suspicious reading, the original phase record is flagged and a copy of the record is added, in which the arrival time is (hopefully) corrected and the author of the reading is changed to be the user. This provides a clear record of what was done.

It is obviously a very dangerous procedure, but the timing of a specific station can be systematically altered with the terr command. Unless there is information to corroborate its use, terr should only be used in an experimental mode.

Outlier Readings

This is the most important and commonly-used flag. A reading may be determined as an outlier and flagged manually during analysis of empirical reading errors with the utility program rstat or through inspection of the ~.phase_data file, or it may be flagged semi-automatically as a result of running the lres or xdat utilities. The concept "outlier" can be either relative (to other samples of the same station-phase) or absolute (with reference to a theoretical travel time model).

Plotting Topography

This section deals with plotting topography and/or bathymetry in the map-like plots produced by **mloc**. Map-like plots include the base plot and its derivatives (selected event plots, ellipse plots, seismicity plots), the direct calibration raypath plot, and the empirical path anomaly plot.

As of v10.5.0 of **mloc** the only global digital elevation model (DEM) available for plotting topography/bathymetry is ETOPO1. Access to this DEM is different under GMT6 than it is under GMT5, as discussed in the Data Files section.

Calling

Plotting of topography is optional; the default is off. To plot topography in all suitable plots the command dem1 is used.

Color Palette

The default color palette (and the only one supplied in the standard **mloc** distribution) is *topo.cpt*. The command cptf can be used to change the color palette if alternative color palettes have been added to */tables/gmt/cpt/*. Any of the standard color palettes supplied with a GMT installation can be used for plotting topography in **mloc**.

High Resolution Topography

The resolution of the global DEM models is adequate for most clusters but in some cases it is desirable to be able to plot topography at higher resolution for the plots that cover a limited area (i.e., the base plot and derivatives). This can be done using the dem2 command, but it requires the user to provide the necessary grid file. A convenient source for such data files is GMTSAR,

from which SRTM topography (at 90 m resolution) and ASTER G-DEM topography (at 30 m) can be downloaded in GMT format. The requested region cannot exceed four degrees in latitude or longitude. To keep file size as small as possible, select a region slightly larger than the limits of the base plot.

mloc will make a temporary color palette for the high-resolution plots, based on the default color palette or one selected with the command cptf.

High-resolution grid files can be stored in the cluster directory but since they are rather large (a few tens of MB) and the same file would need to be stored for each series of runs, I recommend storing them in */tables/gmt/dem*.

Reading Errors

Regardless of the location alogorithm used, both the hypocentral parameters and the estimates of their uncertainty depend strongly on what is assumed about the uncertainties of the input arrival time data. **mloc** provides several tools with which to handle this difficult issue, commonly referred to as "reading error". This section describes most of them, but there is a separate section for empirical reading errors, a method of processing that is unique to **mloc**.

Most relocations with **mloc** are done with the data (arrival time residuals) weighted inversely to an estimate of uncertainty, but it is possible, and often useful in the early stages of a relocation analysis, to weight all data equally. The next step in complexity would be to use phase-specific default values, i.e., read from the file of default reading errors distributed with **mloc**. The ultimate step is to use an estimate of the uncertainty for a specific phase observed at a specific station (station-phase) that is based on analysis of the actual data, i.e., an empirical reading error. **mloc** does not have the facility to assign uncertainties individually to arrival time readings.

In addition to the above features, **mloc** enforces minimum allowed values of reading error that can be controlled by the user. There is also a capability to specify fixed reading errors for Pg and Sg phases at local distance (over-riding any empirical reading errors that may be available). This is sometimes useful for direct calibration or for locating a single event, for which empirical reading errors cannot be estimated. These issues are discussed in more detail below.

Default Values

mloc always reads a file containing a set of phase-specific default reading errors at the beginning of each run. The user can edit this file to change the values for specific phases or add or delete phases. Unless the user turns off weighting with the weig command these are used for the initial run(s) of **mloc**, until we begin using empirical reading errors from previous runs. They are also used for instances of a single sample of a station-phase, since two or more samples are needed to estimate an empirical reading error. Such readings are not used in the estimation of cluster vectors but they may be used to estimate the hypocentroid. If **mloc** encounters a phase which is not found in the list of default values, a value is determined according to an algorithm in the subroutine *readerr* in *mloclib.f90*.

Fixed Values at Local Distances

Because of the limited number of readings that may be available the use of empirical reading errors is sometimes unsatisfactory or impossible for the local-distance data used to estimate the hypocentroid of a cluster in direct calibration. In such cases the user can specify reading errors for Pg and Sg within a specified distance range, usually a few tens of kilometers (the distance range for data used to estimate the hypocentroid) using the command rels.

mloc can be used to locate a single event, in which case it would not be possible to estimate empirical reading errors from that event alone but one could still read the empirical reading errors from a previous run of a cluster that would be relevant to the event in question. If some estimate of empirical reading errors is not used, the location would be done with the default reading errors (unless weighting is turned off with command weig). Finally, if the location is to be based on local-distance data, one might use the rels command to control the reading errors.

Minimum Values

Minimum values are enforced for reading errors, regardless of their source. The need for such limits is primarily driven by the need to prevent divide-by-zero situations with unreasonably small estimates of empirical reading error that often occur when the number of samples is small (e.g., two samples with the same residual). Default values of minimum reading errors are specified in the main program *mloc.f90* for three cases, local distances (0.10 s), beyond local distances (0.15 s) and teleseismic depth phases (1.0 s). These can be changed with the command mare.

Minimum values are also enforced that are based on the precision of the arrival time datum. Reading errors cannot be smaller than the standard deviation of a uniform rectangular continuous distribution over the range defined by the precision of the reading. The minimum reading error for readings given to a precision of a tenth of a second or less (i.e., most modern data) is too small to be relevant but this constraint can be relevant when using data from very old bulletins.

Minimum Reading Error Based on Reading Precision

Precision	Minimum Reading Error (s)
1 second	0.29
6 seconds	1.7
10 seconds	2.9
1 minute	17.0

Output of Reading Errors

The reading error that was used for every reading is printed in the standard output file ~.phase_data that is created for every run of **mloc**. For readings used to locate the hypocentroid

in direct calibration, reading errors are found also in the ~.dcal_phase_data file. They will be listed also in certain optional output files, such as the ~.comcat file created by the command ccat.

Every run of **mloc** produces a ~.rderr file containing the empirical reading errors calculated from the current run. It also carries the empirical reading errors that were used in the current run and some other statistics. To use these empirical reading errors in a future run, use the command rfil to reference the desired ~.rderr file.

Empirical Reading Errors

This section focuses on the estimation and usage of empirical reading errors in **mloc**. There are other aspects of reading errors that are covered in the main Reading Errors section.

The concept of "reading error" or "picking error" in earthquake location is normally understood as an estimate of the uncertainty of the reading of the arrival time ("pick") of a specific seismic phase on the seismogram of a specific earthquake. Seismic analysts rarely provide their own estimate of that uncertainty beyond a qualitative characterization as "emergent" or "impulsive", and earthquake location codes that employ a quantitative estimate of reading error, e.g., for inverse weighting, normally use an ad hoc value based on phase type.

Empirical Reading Error is a related concept, based on multiple event relocation, i.e., simultaneous location analysis of a clustered group of earthquakes. Many seismic stations observe the same seismic phase for multiple events in the cluster. The resulting multiple observations of the same **station-phase** provide an opportunity to carry out a statistical analysis which leads to an estimate of the uncertainty of those readings that is based on the readings themselves, thus **empirical**. It would be more correct to refer to this as an *empirical reading uncertainty*, but we follow the traditional terminology.

It is important to appreciate that this concept of empirical reading error includes contributions to scatter of readings beyond reading error *per se*. For example it will absorb differences in travel time through a heterogeneous Earth even from events that are not exactly co-located, as well as scatter arising from the different philosophies of arrival time picking used by different analysts, changes in station equipment, irregularities in timing systems, differences in the precision to which picks are reported, etc.

Each arrival time reading of a given station-phase is assigned the same empirical reading error. Although this obviously falls short of the ideal of having a reliable estimate of the uncertainty of each reading, it is a significant improvement over the traditional methods for handling uncertainties in arrival time data. Because the arrival time readings are weighted inversely to their reading errors (whatever the source) in the location algorithm, the specification of reading errors has a major impact on the estimated hypocenters and their uncertainties.

How Empirical Reading Errors are Determined

Empirical Reading Errors are estimated from the distribution of residuals for a given station-phase (for example, the Pn phase at station TUC) for a specific cluster. The number of samples

can range from two to several hundred. It is not uncommon to have multiple independent readings of the same phase at the same station for the same event. The analysis is done on the set of residuals obtained by removing a theoretical arrival time for each reading, based on a standard Earth model and the current hypocenters of the events in the cluster.

The estimate of spread of the residuals must be done with a robust estimator, i.e., one that is not sensitive to outliers, which are very common in arrival time data sets. We employ the estimator *Sn* (no relation to the seismic phase) proposed by Croux and Rousseeuw, (1992). This measure of scale or spread has three desirable properties, 1) it requires no assumptions about the nature of the underlying distribution, 2) it requires no estimate of the central tendency (e.g., the mean or median) of the distribution, and 3) it reduces to the standard deviation if applied to a Gaussian distribution. It is also very easy to compute (see subroutine *croux* in *mloclib_statistics.f90*)

Cleaning

An important aspect of the relocation process consists of multiple cycles in which the current estimates of empirical reading error are used to identify outlier readings, which are then flagged so that they will not be used in subsequent relocations. In the following relocation, estimates of empirical reading errors will tend to be smaller because of the filtering of outliers and improvement in the locations of the clustered events. Therefore the process of identifying outliers is iterative and it must be repeated until convergence. In this context, *convergence* means that the distribution of residuals for a given station-phase is consistent with the current estimate of spread. As outlier readings are flagged, the distribution is expected to evolve toward a normal distribution with standard deviation equal to the empirical reading error. We generally continue this **cleaning** process until all readings used in the relocation are within 3σ of the mean for that station-phase, where σ is the current estimate of empirical reading error for the relevant station-phase.

More detail on the cleaning process is provided elsewhere.

Output File

Every run of **mloc** produces a ~.rderr file containing the empirical reading errors calculated from the current run. It also carries the empirical reading errors that were used in the current run and some other statistics. To use these empirical reading errors in a future run, use the command rfil to reference the desired ~.rderr file.

Starting Locations

mloc requires an initial estimate of hypocentral parameters for all events in the cluster. The commands that are relevant are lat, long, time, dep_ and rhdf.

Because of the number of free parameters in a multiple-event relocation analysis, as compared to a single event relocation, **mloc** is sensitive to the accuracy of those initial hypocenters. In other words, if the initial locations are very poor **mloc** may have trouble converging. Hypocenters for

recent events obtained from the major global and regional seismograph networks are usually adequate, within 20 km or so of the ultimate location, but older events are prone to having much larger errors. In such cases several runs may be needed in which **mloc** does not converge because the relative location (cluster vector) of one or more events continues to shift. In the early stages of a relocation analysis where such issues are being addressed it is highly recommended to leave focal depth as a fixed parameter.

Sources of Initial Hypocentral Parameters

mloc can take the initial estimate of location from several sources, in order of decreasing precedence:

- A command (commands time, dep_, lat or long) in the event definition section of the command file.
- A value taken from the ~.hdf file of a previous run (command rhdf). Depending on the type of relocation this may be a file named *~.hdf*, *~.hdf_cal* or *~.hdf_dcal*.
- A command (usually depc, but in principle, time, lat or long) in the first section of a command file (before any events are defined) that will therefore apply to all events unless specifically over-ridden.
- The preferred hypocenter of the input data file.

For the initial run there is usually no alternative to using the locations carried in the input data file. Normally I will switch to taking starting hypocentral parameters from the ~.hdf file of the last run (command rhdf) immediately after the first run.

Station Code Problems

The correct assignment of geographic coordinates to the station codes in arrival time datasets is an essential aspect of every relocation algorithm. **mloc** is extremely flexible in regard to station data but that very flexibility leads to some rather complex scenarios. Failure to resolve station coordinate problems occasionally leads to data being used incorrectly, but much more often leads to data dropping out of the relocation, with consequences ranging from nil to severe. **mloc** produces a considerable amount of information on the subject and provides tools for helping to solve problems of this type but to use that information wisely it is necessary to have some understanding about how the station coordinate issue is handled in **mloc**. Ultimately it is a responsibility of the user to review the output carefully, especially in the early stages of a new cluster analysis, to ensure the correct usage of the available data. The purpose of this section is to give you the necessary background and a sense of strategies that have proven useful.

Before reading this section it is highly recommended that you read the section on Station Data and the one on Supplemental Station Files. A few main points to remember:

- The master station file distributed with **mloc** includes only station codes registered at the IR (although some of the coordinates have been corrected slightly from what you'll find

at the IR website). If arrival time data came from the ISC and the station code matches an entry in the master station file, it is almost certainly correct.

- New stations are being registered at the IR constantly, so if you are using ISC data for recent events you may find that the master station file needs to be updated with the new codes. Just do it.
- *Do not add non-registered station codes to your master station file.* Put them in supplemental station files.
- **mloc** reads supplemental station files before the master station file and puts all of it in a list. When matching a station code to coordinates, the first successful match is selected. If any code in a supplemental station file conflicts with one in the master station file, the supplemental file wins. For this reason it is wise to keep supplemental station files as short as possible. Unneeded entries may step on entries from the master station file (case 1 below).

During the Run

The first hint that you have station code problems will occur during the terminal session in which you run **mloc**. After the part where the event files are read there will be a summary of the number of station codes that failed the date range (operational epoch) or which were not found at all. If nothing is listed and you are using only the master station file and all your data come from the ISC Bulletin, then you are in the clear and should have no worries about station code problems.

If, however, you are using one or more supplemental station files or if any of your data come from stations that may not have been registered with the International Registry of Seismograph Stations (IR) at the ISC, the absence of warnings at this stage cannot be taken at face value. There could still be problems:

1. The coordinates of an unregistered code defined in a supplemental station file might have been applied to an arrival in your dataset that should be using the coordinates from the IR.
2. Coordinates from a station code registered at the IR (i.e., the master station file) may have been assigned to an unregistered, conflicting station code in your dataset (perhaps from a local network).
3. The same code could be defined in more than one of your supplemental station files.

Case 1 is rare. Case 2 is common when using data acquired from somewhere other than the ISC. Case 3 would be revealed in the station output file (below) as a case of conflicting station codes, but the result of all these scenarios is that the incorrect coordinates are assigned to a certain station code. Calculations of epicentral distance are therefore very likely to be in error, so theoretical travel time is in error, probably by a lot. The reading is likely to fall out of the relocation as a gross outlier. These cases will be hiding among "legitimate" cases of gross outliers in the Bad Data section of the ~.phase_data file, but there are clues to help pick out the

station code problems. The main one is to check the authorship of the pick (an excellent reason for including authorship in your phase records) and consider if it makes sense to find a pick from that source at that epicentral distance. Another clue is to compare the epicentral distance to the one listed (hopefully) in the event file.

The Station Output File

After a run of **mloc**, the first place to check for station code problems is the ~.stn file. There are three main types of problems to look for: cases of failed date range, station conflicts and missing stations.

Failed Date Range

It is not unusual to have a few "failed date range" warnings, even in the plain vanilla case of using only ISC data with only the master station file, because many of the entries for operational epoch are in error. To fix those problems, one can clear the operational epoch fields in the master station file (they are optional), or extend the operational epoch as necessary by pasting in the new epoch-limiting value (listed in the entry in the ~.stn file).

Duplicates and Conflicts

If any supplemental station files have been used, there may be cases where the same station code appears in both a supplemental file and the master station file. These instances are laid out with comparison of coordinates and categorized as "pure duplicate", "minor differences in coordinates" or "station conflict". Duplicates can be left, but I prefer to remove them from the supplemental station file. Minor differences won't make any difference for stations that are only used to estimate the cluster vectors, but if the station is one that will be used for direct calibration the issue should be investigated. Looking at the competing coordinates in Google Earth is sometimes helpful.

If a conflict is found for a station that is in the arrival time dataset from a source other than the ISC, then it's working as it should; you want the entry in the supplemental station file to override the one in the master station file. If the conflict is between two supplemental station files you'll need to ensure the correct supplemental station file is loaded first, or else edit the file with the irrelevant entry.

Missing Stations

The next section of the station output file lists all the cases of missing stations. It is good practice (but admittedly tedious) to take the time to resolve these problems, especially in cases where there are many instances of the missing station; those instances provide information that would help resolve cluster vectors. However, even a case with a single instance could provide important information in a direct calibration if it's in the appropriate distance range. The best strategy is to check first at the IR, and if the code is found, add it to the master station file. If not, you'll need

to check with the source of the data you are using and make an entry for it in a supplemental station file.

The remainder of the station output file lists all stations used in the relocation, their coordinates and the source (i.e., one of the supplemental station files or the master station file) of the coordinates. This is sometimes useful in tracking down a station problem.

The Worst Case Scenario

The discussion so far has been based on the assumption that all station codes in the dataset are unique, i.e., a given station code always refers to a single geographic location. Fortunately this assumption is usually true, but when data from several different sources are being combined in a relocation it may happen that arrival time data exist from two distinct stations (i.e., different locations) with the same code. There are several ways to deal with this awkward situation.

Ignorance is Bliss

By far the easiest way to deal with a station conflict is to ignore it. That means that the coordinates for the station in question that were introduced in the supplemental station file will be applied everywhere, even to arrivals that should be taking their coordinates from the master station file. Those arrivals will fall out of the relocation as outliers. The use of a supplemental station file typically means that those stations are at local distance range and therefore of considerable importance to a calibration analysis. In most cases the readings that are lost will be at teleseismic range and it's likely there will only be one or a few instances in the dataset. If there is only one reading it will not contribute to the cluster vectors in any case, and if it is not in the distance range be used for the hypocentroid in direct calibration its loss is irrelevant. When there are several instances the loss of those data can still probably be accepted with virtually no consequence to the quality of the relocation, but it must be considered.

Change station codes

If a station conflict involves only a small number of readings the conflict can be managed by altering the station code in the supplemental station file (and the relevant event files) in some manner. The provision in the MNF format for carrying the station code in two fields, one of which is treated as archival, makes this approach more palatable than losing the original station code in the dataset completely.

The Right Way

The correct way to handle a station code conflict when you need to use data from both locations is to expand the identification of the station beyond the traditional station code. This is why the so-called ADSLC system was proposed. The MNF v1.3 format used for event files supports the full ADSLC description, but **mloc** uses only the agency and deployment fields to resolve station code conflicts. Only the master station file format (which can also be used for a supplemental

station file) and the generic and NEIC supplemental station file formats carry agency and deployment codes.

To use agency and deployment as well as station code to resolve station code conflicts in **mloc**, the command radf is used. This allows you to turn on the use of agency and deployment codes for specific stations, i.e., the ones for which there is a conflict. For every other station code the agency and deployment fields are treated as blank in **mloc**, regardless of what is in those fields in the event files or station files. It is necessary to ensure that the agency and deployment fields in phase records with the station code of interest are filled in with the values corresponding to the appropriate entry in the master station file and supplemental station files. In the NEIC format the agency is implicit ("FDSN") and the deployment code is the 2-letter FDSN network code. Most entries in the master station file use "ISC" as the agency and "IR" as the deployment, but there are exceptions so it may be necessary to inspect the station entry in the master station file before setting the fields in your event files.

This functionality of the radf command exists only since **mloc** v10.5.0. The command exists in earlier versions but it used a different strategy that proved to be impractical.

Bibliography

This list consists of papers that have used the Hypocentroidal Decomposition (HD) method of multiple event relocation (Jordan and Sverdrup, 1981). Most of them used the program **mloc**. The non-**mloc** ones are mostly early and include Keith Sverdrup as a co-author, using an updated version of the original code implementing HD. The early (pre-2003) **mloc** papers used the program in an uncalibrated mode, i.e. they made use of improved relative locations but did not attempt to improve knowledge of absolute locations.

Original Description of the Algorithm

Jordan, T.H., and Sverdrup, K.A., 1981, Teleseismic location techniques and their application to earthquake clusters in the South-Central Pacific: Bulletin of the Seismological Society of America, v. 71, no. 4, p. 1105-1130. (PDF)

Uncalibrated Locations

Sverdrup, K.A., 1986, Multiple-event relocation of earthquakes on and near the Gorda Ridge: Geophysical Research Letters, v. 13, no. 7, p. 674-677.

Sverdrup, K.A., 1987, Multiple-event relocation of earthquakes near the Gorda Rise-Mendocino Fracture Zone intersection: Geophysical Research Letters, v. 14, no. 4, p. 347-350.

Bergman, E.A., and Solomon, S.C., 1990, Earthquake swarms on the Mid-Atlantic Ridge – Products of magmatism or extensional tectonics?: Journal of Geophysical Research, v. 95, p. 4943.

Cronin, V.S., and Sverdrup, K.A., 2003, Multiple-event relocation of historic earthquakes along Blanco Transform Fault Zone, NE Pacific: Geophysical Research Letters, v. 30, p. 2001, doi: 10.1029/2003GL018086.

Wolfe, C.J., Bergman, E.A., and Solomon, S.C., 1993, Oceanic transform earthquakes with unusual mechanisms or locations – Relation to fault geometry and state of stress in the adjacent lithosphere: Journal of Geophysical Research, v. 98, p. 16187.

Bai, L., Bergman, E.A., Engdahl, E.R., and Kawasaki, I., 2007, The 2004 earthquakes offshore of the Kii peninsula, Japan: Hypocentral relocation, source process and tectonic implication: Physics of the Earth and Planetary Interiors, v. 165, p. 47, doi: 10.1016/j.pepi.2007.07.007.

Calibrated Locations

2003

Ritzwoller, M.H., Shapiro, N.M., Levshin, A.L., Bergman, E.A., and Engdahl, E.R., 2003, Ability of a global three-dimensional model to locate regional events: Journal of Geophysical Research, v. 108, no. B7, p. 2353-2353, doi: 10.1029/2002JB002167.

2004

Bondar, I., Engdahl, E.R., Yang, X.-P., Ghalib, H.A.A., Hofstetter, A., Kirichenko, V., Wagner, R., Gupta, I., Ekström, G., Bergman, E.A., Israelsson, H., and McLaughlin, K.L., 2004, Collection of a reference event set for regional and teleseismic location calibration: Bulletin of the Seismological Society of America, v. 94, no. 4, p. 1528-1545.

Bondar, I., Myers, S.C., Engdahl, E.R., and Bergman, E.A., 2004, Epicentre accuracy based on seismic network criteria: Geophysical Journal International, v. 156, p. 483-496, doi: 10.1111/j.1365-246X.2004.02070.x.

2005

Rastogi, B.K., Bergman, E.A., and Engdahl, E.R., 2005, Improved earthquake locations and estimation of Pn and Sn path anomalies for India, using multiple event relocation and reference events: Current Science, v. 88, no. 10, p. 1586-1591.

Walker, R.T., Bergman, E.A., Jackson, J.A., Ghorashi, M., and Talebian, M., 2005, The 2002 June 22 Changureh (Avaj) earthquake in Qazvin province, northwest Iran: epicentral location, source parameters, surface deformation and geomorphology: Geophysical Journal International, v. 160, p. 707-720.

2006

Biggs, J., Bergman, E.A., Emmerson, B., Funning, G.J., Jackson, J.A., Parsons, B.E., and Wright, T.J., 2006, Fault identification for buried strike-slip earthquakes using InSAR: The 1994 and 2004 Al Hoceima, Morocco earthquakes: Geophysical Journal International, v. 166, p. 1347-1362.

Parsons, B.E., Wright, T.J., Rowe, P., Andrews, J., Jackson, J.A., Walker, R.T., Khatib, M.M., Talebian, M., Bergman, E.A., and Engdahl, E.R., 2006, The 1994 Sefidabeh (eastern Iran) earthquakes revisited: new evidence from satellite radar interferometry and carbonate dating about the growth of an active fold above a blind thrust fault: Geophysical Journal International, v. 164, p. 202-217, doi: 10.1111/j.1365-246X.2005.02655.x.

2007

Tatar, M., Jackson, J.A., Hatzfeld, D., and Bergman, E.A., 2007, The 2004 May 28 Baladeh earthquake (Mw 6.2) in the Alborz, Iran: Overthrusting the South Caspian Basin margin, partitioning of oblique convergence and the seismic hazard of Tehran: Geophysical Journal International, v. 170, p. 249-261.

2008

Bondar, I., Bergman, E.A., Engdahl, E.R., Kohl, B., Kung, Y.-L., and McLaughlin, K.L., 2008, A hybrid multiple event location technique to obtain ground truth event locations: Geophysical Journal International, v. 175, no. 1, p. 185-201, doi: 10.1111/gji.2008.175.issue-1.

2009

Roustaei, M., Nissen, E., Abbassi, M.R., Ghorashi, M., Gholamzadeh, A., Tatar, M., Yamini-Fard, F., Bergman, E.A., Jackson, J.A., and Parsons, B.E., 2009, Vertical separation of surface folding, earthquake faulting, and aftershocks in the Zagros Simply Folded Belt (Iran): Geophysical Journal International, v. 142, p. 1-24.

2010

Nissen, E., Yamini-Fard, F., Tatar, M., Gholamzadeh, A., Bergman, E.A., Elliott, J.R., Jackson, J.A., and Parsons, B.E., 2010, The vertical separation of mainshock rupture and microseismicity at Qeshm island in the Zagros fold-and-thrust belt, Iran: Earth and Planetary Science Letters, v. 296, no. 3-4, p. 181-194, doi: 10.1016/j.epsl.2010.04.049.

2011

Walker, R.T., Bergman, E.A., Szeliga, W., and Fielding, E.J., 2011, Insights into the 1968-1997 Dasht-e-Bayaz and Zirkuh earthquake sequences, eastern Iran, from calibrated relocations, InSAR and high-resolution satellite imagery: Geophysical Journal International,, p. no-no, doi: 10.1111/j.1365-246X.2011.05213.x.

2012

Copley, A. C., Hollingsworth, J., & Bergman, E. A. (2012). Constraints on fault and lithosphere rheology from the coseismic slip and postseismic afterslip of the 2006 Mw7. 0 Mozambique earthquake. Journal of Geophysical Research, 117(B3), B03404. http://doi.org/10.1029/2011JB008580.

Ghods, A., Rezapour, M., Bergman, E.A., Mortezanejad, G., and Talebian, M., 2012, Relocation of the 2006 Mw 6.1 Silakhour, Iran, Earthquake Sequence: Details of Fault Segmentation on the Main Recent Fault: Bulletin of the Seismological Society of America, v. 102, no. 1, p. 398-416, doi: 10.1785/0120110009.

Yamini-Fard, F., Tatar, M., Hessami, K., Gholamzadeh, A., and Bergman, E.A., 2012, Aftershock analysis of the 2005 November 27 (Mw 5.8) Qeshm Island earthquake (Zagros-Iran): Triggering of strike-slip faults at the basement: Journal of Geodynamics, v. 61, p. 138-147, doi: 10.1016/j.jog.2012.04.005.

2013

Aziz Zanjani, A., Ghods, A., Sobouti, F., Bergman, E. A., Mortezanejad, G., Priestley, K., et al. (2013). Seismicity in the western coast of the South Caspian Basin and the Talesh Mountains. Geophysical Journal International, 195(2), 799-814. doi:10.1093/gji/ggt299

Hayes, G. P., Bergman, E. A., Johnson, K., Benz, H. M., & Brown, L. (2013). Seismotectonic framework of the 2010 February 27 Mw 8.8 Maule, Chile earthquake sequence. Geophysical Journal International. doi:10.1093/gji/ggt238

Walker, R.T., Bergman, E.A., Elliott, J.R., Fielding, E.J., Ghods, A.R., Ghoraishi, M., Jackson, J.A., Nazari, H., Nemati, M., Oveisi, B., Talebian, M., and Walters, R.J., 2013, The 2010-2011 South Rigan (Baluchestan) earthquake sequence and its implications for distributed deformation and earthquake hazard in southeast Iran: Geophysical Journal International,, doi: 10.1093/gji/ggs109.

Walker, R. T., Khatib, M. M., Schnabel, C., Rodés, A., Fattahi, M., Talebian, M., et al. (2013). Co-seismic, geomorphic, and geologic fold growth associated with the 1978 Tabas earthquake fault in eastern Iran. Geomorphology, 1-44.

2014

Barnhart, W. D., Benz, H. M., Hayes, G. P., Rubinstein, J. L., & Bergman, E. A. (2014). Seismological and geodetic constraints on the 2011 M w5.3 Trinidad, Colorado earthquake and induced deformation in the Raton Basin. Journal of Geophysical Research, 119(10), 7923-7933. http://doi.org/10.1002/2014JB011227

Hayes, G. P., Herman, M. W., Barnhart, W. D., Furlong, K. P., Riquelme, S., Benz, H. M., Bergman, E., Barrientos, S., Earle, P. S., and Samsonov, S. (2014). Continuing megathrust earthquake potential in Chile after the 2014 Iquique earthquake. Nature, 512(7514), 295-298. doi:10.1038/nature13677

Mackey, K. G., & Bergman, E. A. (2014). Ground truth locations for the Mangyshlak Peaceful Nuclear Explosion sequence, western Kazakhstan. Bulletin of the Seismological Society of America, 104(4), 1-9.

McNamara, D. E., Benz, H. M., Herrmann, R. B., Bergman, E. A., Earle, P. S., Meltzer, A., et al. (2014). The Mw 5.8 Mineral, Virginia, Earthquake of August 2011 and Aftershock Sequence: Constraints on Earthquake Source Parameters and Fault Geometry. Bulletin of the Seismological Society of America, 104(1), 40-54. http://doi.org/10.1785/0120130058

2015

Elliott, J. R., E. A. Bergman, A. C. Copley, A. R. Ghods, E. K. Nissen, B. Oveisi, M. Tatar, R. J. Walters, and F. Yamini-Fard (2015), The 2013 Mw 6.2 Khaki-Shonbe (Iran) Earthquake: Insights into seismic and aseismic shortening of the Zagros sedimentary cover, Earth and Space Science, 2, doi:10.1002/2015EA000098.

Ghods, A., Shabanian, E., Bergman, E. A., Faridi, M., Donner, S., Mortezanejad, G., & Aziz-Zanjani, A. (2015). The Varzaghan-Ahar, Iran, Earthquake Doublet (Mw6.4, 6.2): Implications for the geodynamics of northwest Iran. Geophysical Journal International, 203(1), 522-540. http://doi.org/10.1093/gji/ggv306

McNamara, D. E., H. M. Benz, R. B. Herrmann, E. A. Bergman, P. Earle, A. Holland, R. Baldwin, and A. Gassner (2015), Earthquake hypocenters and focal mechanisms in central Oklahoma reveal a complex system of reactivated subsurface strike-slip faulting, Geophys. Res. Lett., 42, 2742-2749, doi:10.1002/ 2014GL062730.

McNamara, D. E., Rubinstein, J. L., Myers, E., Smoczyk, G., Benz, H. M., Williams, R. A., et al. (2015). Efforts to monitor and characterize the recent increasing seismicity in central Oklahoma. The Leading Edge, 1-8.

2016

Karasözen, E., Nissen, E., Bergman, E. A., Johnson, K. L., & Walters, R. J. (2016). Normal faulting in the Simav graben of western Turkey reassessed with calibrated earthquake relocations. Journal of Geophysical Research: Solid Earth, 121(6), 4553-4574. http://doi.org/10.1002/2016JB012828

McNamara, D.E., Yeck, W., Barnhart, W., Schulte-Pelkum, V., Bergman, E., Adhikari, L.B., Dixit, A., Hough, S., Benz, H., Earle, P., Source Modeling of the 2015 Mw 7.8 Nepal (Gorkha) Earthquake Sequence: Implications for Geodynamics and Earthquake Hazards, Tectonophysics (2016), doi: 10.1016/j.tecto.2016.08.004

Yeck, W. L., Weingarten, M., Benz, H. M., McNamara, D. E., Bergman, E. A., Herrmann, R. B., et al. (2016). Far-field pressurization likely caused one of the largest injection induced earthquakes by reactivating a large preexisting basement fault structure. Geophysical Research Letters, 43(19), 10,198-10,207. http://doi.org/10.1002/2016GL070861

2017

Nealy, J. L., Benz, H. M., Hayes, G. P., Bergman, E. A., & Barnhart, W. D. (2017). The 2008 Wells, Nevada, Earthquake Sequence: Source Constraints Using Calibrated Multiple-Event Relocation and InSAR, Bulletin of the Seismological Society of America, 107(3), 1107-1117. http://doi.org/10.1785/0120160298

Nealy, J. L., Herman, M. W., Moore, G. L., Hayes, G. P., Benz, H. M., Bergman, E. A., & Barrientos, S. E. (2017). 2017 Valparaíso earthquake sequence and the megathrust patchwork of central Chile. Geophysical Research Letters, 44(17), 8865-8872. http://doi.org/10.1002/2017GL074767

2018

Karasözen, E., Nissen, E., Büyükakpınar, P., Cambaz, M. D., Kahraman, M., Kalkan Ertan, E., Abgarmi, B., Bergman, E.A., Ghods, A.R. and A. Arda Özacar (2018). The 2017 July 20 Mw 6.6 Bodrum-Kos earthquake illuminates active faulting in the Gulf of Gökova, SW Turkey. Geophysical Journal International, 214(1), 185-199. http://doi.org/10.1093/gji/ggy114

2019

Aflaki, M., Mousavi, Z., Ghods, A.R., Shabanian, E., Vajedian, S. and M. Akbarzadeh 2019. The 2017 Mw 6 Sefid Sang earthquake and its implication for the geodynamics of NE Iran, Geophysical Journal International, 218, 2, p. 1227-1245, https://doi.org/10.1093/gji/ggz172.

Gaudreau, É., Nissen, E., Bergman, E. A., Benz, H. M., Tan, F., & Karasözen, E. (2019). The August 2018 Kaktovik Earthquakes: Active Tectonics in Northeastern Alaska Revealed With InSAR and Seismology. Geophysical Research Letters, 117(4), B03401–9. http://doi.org/10.1029/2019GL085651.

Karasözen, E., Nissen, E., Bergman, E.A. and A.R. Ghods (2019). Seismotectonics of the Zagros (Iran) From Orogen-Wide, Calibrated Earthquake Relocations, Journal of Geophysical Research, https://doi.org/10.1029/2019JB017336.

Nissen, E., Ghods, A., Karasözen, E., Elliott, J.R. Barnhart, W.D., Bergman, E.A., Hayes, G.P, Jamal-Reyhani, M., Nemati, M., Tan, F., Abdulnaby, W., Benz, H.M., Shahvar, M.P., Talebian, M. and L. Chen (2019). The 12 November 2017 Mw 7.3 Ezgeleh-Sarpolzahab (Iran) earthquake and active tectonics of the Lurestan arc. Journal of Geophysical Research: Solid Earth, 124. https://doi.org/10.1029/2018JB016221

Savidge, E., Nissen, E., Nemati, M., Karasözen, E., Hollingsworth, J., Talebian, M., Bergman, E.A., Ghods, A.R., Ghorashi, M., Kosari, E., Rashidi, A. and A. Rashidi (2019). Geophysical Journal International 217, 2, p. 909-925, https://doi.org/10.1093/gji/ggz053.

2020

Pousse-Beltran, L., Nissen, E., Bergman, E. A., Cambaz, M. D., Gaudreau, É., Karasözen, E., & Tan, F. (2020). The 2020 Mw 6.8 Elazığ (Turkey) Earthquake Reveals Rupture Behavior of the East Anatolian Fault. Geophysical Research Letters, 47(13), 319. http://doi.org/10.1029/2020GL088136.

Soto Cordero, L., Meltzer, A., Bergman, E. A., Hoskins, M., Stachnik, J. C., Agurto Detzel, H., et al. (2020). Structural Control on Megathrust Rupture and Slip Behavior: Insights From the 2016 Mw 7.8 Pedernales Ecuador Earthquake. Journal of Geophysical Research: Solid Earth, 125(2), S225–28. http://doi.org/10.1029/2019JB018001.

References

Following are references for journal articles of particular relevance to **mloc**. In some cases PDFs are linked for convenience in investigating the algorithmic foundations of **mloc**.

Bondar, I. K., Myers, S. C., Engdahl, E. R., & Bergman, E. A. (2004). Epicentre accuracy based on seismic network criteria. Geophysical Journal International, 156, 483–496. http://doi.org/10.1111/j.1365-246X.2004.02070.x

Bondar, I., Bergman, E.A., Engdahl, E.R., Kohl, B., Kung, Y.-L., and McLaughlin, K.L., 2008, A hybrid multiple event location technique to obtain ground truth event locations: Geophysical Journal International, v. 175, no. 1, p. 185-201, doi: 10.1111/gji.2008.175.issue-1.

Buland, R.P., and Chapman, C.H. (1983). The computation of seismic travel times. Bulletin of the Seismological Society of America, v, 73, no. 5, p. 1271-1302. (PDF)

Cleary, J.R., & Haddon, R.A.W. (1972). Seismic Wave Scattering near the Core-Mantle Boundary: a New Interpretation of Precursors to PKP. Nature, 240(5383), 549–551. http://doi.org/10.1038/240549a0

Croux, C., and Rousseeuw, P.J., 1992, Time-efficient algorithms for two highly robust estimators of scale: Computational Statistics, v. 1, p. 411-428. (PDF)

Dziewonski, A.M., and F.J. Gilbert, 1976. The effect of small, aspherical perturbations on travel times and a re-examination of the corrections for ellipticity. Geophysical Journal International, 44(1), p. 7-17.

Engdahl, E.R., van der Hilst, R.D., and Buland, R.P. (1998). Global teleseismic earthquake relocation with improved travel times and procedures for depth determination. Bulletin of the Seismological Society of America, 88(3), 722-743. (PDF)

Herrin, E. (1968). Introduction to “1968 Seismological Tables for P Phases.” Bulletin of the Seismological Society of America, 58(4), 1193–1195.

Jordan, T.H., and Sverdrup, K.A., 1981, Teleseismic location techniques and their application to earthquake clusters in the South-Central Pacific: Bulletin of the Seismological Society of America, v. 71, no. 4, p. 1105-1130. (PDF)

Karasözen, E. (2018). Seismotectonics of Turkey and Iran from calibrated earthquake relocations. PhD Thesis, Colorado School of Mines.

Mendiguren, J. A. (1971). Focal mechanism of a shock in the middle of the Nazca Plate. Journal of Geophysical Research, 76(17), 3861–3879. http://doi.org/10.1029/JB076i017p03861

Okal, E. A. (2001). T-phase Stations for the International Monitoring System of the Comprehensive Nuclear-Test Ban Treaty: A Global Perspective. Seismological Research Letters, 72(2), 186–196. http://doi.org/10.1785/gssrl.72.2.186

Press, W.H., Flannery, B.P., Teukolsky, S.A. and W.T. Vetterling (1986). Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, Cambridge, England, 818 p.

Schweitzer, J. (2001). HYPOSAT—An enhanced routine to locate seismic events. Pure and Applied Geophysics, v. 158, p. 277-289. (PDF)

Waldhauser, F. and Ellsworth, W. L. (2000). A double-difference earthquake location algorithm: Method and application to the northern Hayward fault, California. Bulletin of the Seismological Society of America.

Funding

The following list of projects covers the major sources of research funding that have supported the development of **mloc**, in reverse chronological order.

- *A Global Database of Calibrated Earthquake Locations*, Air Force Research Laboratory FA9453-17-C-0021, E.A. Bergman, 2017-2020.
- *A Two-Tiered Approach to Event Calibration Across Iran*, Air Force Research Laboratory, E. Nissen, E. Karasozen and E.A. Bergman
- *Seismotectonics of the Zagros (Iran) from Orogen-wide Earthquake Relocations*, National Science Foundation EAR 1524815, E. Nissen, E. Karasozen and E.A. Bergman, 2016-2018.
- *Calibration of Earthquake Locations using Multiple Event Processing Methods*, U.S. Geological Survey G16PX01490, E.A. Bergman, 2016-2017.
- *Application of Calibrated Multiple Event Relocation at the National Earthquake Information Center*, U.S. Geological Survey G14AP00131, E.A. Bergman, 2014-2015.
- *Is Continental Collision Thick- or Thin-skinned? Combining local seismicity with Receiver Functions in the Zagros Fold-and-Thrust Belt*, National Science Foundation EAR 1246287, V. Schulte-Pelkum and E.A. Bergman, 2013-2015.
- *Crustal and Upper Mantle Structure from Joint Inversion of Body Wave and Gravity Data*, Air Force Research Laboratory FA9453-11-1-0233, M. Maceira, E.A. Bergman and C. Rowe, 2012-2014.
- *Multiple Event Relocation at the National Earthquake Information Center*, U.S. Geological Survey G11AP20016, E.A. Bergman, 2011-2012.
- *Crustal Structure of the Iran Region from In-Country and Ground Truth Data*, Air Force Research Laboratory FA8718-08-C-0020, E.A. Bergman, E.R. Engdahl, M.H. Ritzwoller and S.C. Myers, 2009-2011.
- *Improved Ground Truth in Southern Asia Using In-Country Data, Analyst Waveform Review and Advanced Algorithms*, Dept. of Energy DE-FC52-03NA99516, E.R. Engdahl, E.A. Bergman and S.C. Myers, 2004-2007.
- *Global Ground Truth Data Set with Waveform and Arrival Data*, Air Force Research Laboratory AFRL-RV-HA-TR-2007-1101, I. Bondar, E.A. Bergman, B. Kohl, Y-L Kung, K. McLaughlin, H. Israelsson and E.R. Engdahl, 2004-2007.
- *Identification and Validation of Reference Events within the Area Regionally Monitored by IMS Stations in Asia and North Africa*, Defense Threat Reduction Agency DTRA01-00-C-0032, E.R. Engdahl and E.A. Bergman, 2000-2003.

The initial phase of **mloc** development in the late 1980s was supported by two grants from the National Science Foundation, EAR 8617967 and 8817173, and one from the National

Aeronautics and Space Administration NAG 5-814, all with Sean Solomon as PI. Support also came from the U.S. Geological Survey External Grants Program in a project to study foreshock-mainshock-aftershock sequences with detailed source studies using body waveform inversion and multiple event relocation (my first funded proposal). All details of this project have disappeared into null space.

No work was done on **mloc** between 1990 and 1998.