

# Moment Tensor Inversion Toolkit (MTINV) Documentation, Manual and Tutorial

Version 4.0.1 Release Date: April 27, 2023

Gene A. Ichinose, Jorge Roman-Nieves, Gordon Kraft  
(email: [geneichinose@yahoo.com](mailto:geneichinose@yahoo.com))

## License Agreement, Liability Wavier

The MTINV Toolkit source code is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either versions 2 of the License, or any later version. This toolkit is distributed in the hope that it will be useful, but without any warranty, without even the implied warranty of merchantability or fitness for any particular purpose. The comments and claims in this document are solely those of the author and are not endorsed by the author's past and current employers (University of Nevada, URS Corporation, US Air Force, Lawrence Livermore National Security) or the United States Government. To receive a copy of the GNU General Public License, write to the Free Software Foundation, Inc., 59 Temple Place Ste 330, Boston, MA 02111-1307, USA.

## 1. MTINV Toolkit Manual

### 1.1 What is the MTINV toolkit?

The moment tensor inversion methodology originated from *Gilbert* (1970). Since then, there have been numerous studies extending the method from the frequency-domain to the time-domain, solving higher order source terms, and more recently the addition of three-dimensional Earth models. For a general review please see *Jost and Herrman* (1989). The elegance of this method comes from the use of long-period regional distance seismic waves. This simplifies the complex source process to a delta function in space and time. It also simplifies the complex wave propagation because filtering regional seismograms to longer-periods (or lower frequencies), result in waves that have only propagated over a few wavelength cycles and can be predicted using 1-D layered earth velocity models. Eliminating these two complex effects leaves the robust extraction of the seismic source radiation pattern.

The MTINV toolkit (2002-present) is a collection of computer codes and applications written to invert for the moment tensor of a seismic source given the three components of ground motion recorded at regional seismic stations (*e.g.*, *Ichinose et al.*, 2003). The computer codes and workflow are organized to generate moment tensor solutions for a range of source depths and origin times because of the trade-off between these two quantities. The metric used is the variance reduction and variance reduction modulated by the percent double-couple to determine the best-fit moment-tensor solution. We can solve for a deviatoric moment tensor with a constraint added for no isotropic component although this constraint can be lifted for estimating the full moment tensor like mining collapses or explosion sources. MTINV versions 3.0+ contain the corrections based on *Herrmann and Hutchensen* (1993), *Herrmann and Ammon* (2002)-*Computer Programs in Seismology manual on Source Inversion*, and *Minson and Dreger* (2008) needed to compute full moment tensors (6-degree of freedom) that include isotropic components. MTINV version 3.0.3 includes signal to noise ratios, computed and stored to be used as guide to define stations for use in the inversion. MTINV version 3.0.6 introduced helper

commands (`setupMT`, `makepar`, and `mtbestfit`) which autogenerates UNIX C-shell scripts based on user input and MT results. These helper commands make it easier for users so that they don't need to remember all the command-line options required in the scripts (`makeglib.csh`, `run.csh`, and `run2.csh`). Version 4.x includes Network Sensitivity Solutions (NSS), that are variance reductions computed from millions of forward full-MT calculations for a fixed origin-time, depth, and seismic moment based on *Ford et al.* (2010). A new command `mteig` (*Ichinose et al.*, 2020) creates a plot of the variance reductions for the source-type space (isotropic and CLVD) from a spherical projection of eigenvalues on a lune (*Tape and Tape*, 2012).

Ford, S. R., D. S. Dreger, and W. R. Walter (2010). Network Sensitivity Solutions for Regional Moment-Tensor Inversions, *Bull. Seism. Soc. Am* 100(5A), 1962-1970, <https://doi.org/10.1785/0120090140>

Gilbrt, F. (1970). Excitation of the normal modes of the earth by earthquake sources, *Geophys. J. R. Astr. Soc.* 77, 883-914, <https://doi.org/10.1111/j.1365-246X.1971.tb03593.x>

Jost, M. L. and R. B. Herrmann (1989). A student's guide to and review of moment tensors, *Seismological Research Letters*, 86(2), 37-57, <https://doi.org/10.1785/gssrl.60.2.37>

Ichinose, G. A., John G. Anderson, Ken D. Smith, and Yuehua Zeng (2003). Source Parameters of Eastern California and Western Nevada Earthquakes from Regional Moment Tensor Inversion, *Bull. Seism. Soc. Am.* 93(1), 61-84, <https://doi.org/10.1785/0120020063>

Ichinose, G. A., S. R. Ford, and R. J. Mellors (2020). Regional Moment Tensor Inversion Using Rotational Observations, *J. Geophys. Res.* 126(2), e2020JB020827, <https://doi.org/10.1029/2020JB020827>

Herrmann, R. B. and C. J. Ammon (2002). Computer Programs in Seismology Manual: Source Inversion. Version 3.30, <https://www.eas.slu.edu/eqc/eqccps.html>

Herrmann, R. B. & Hutchensen, K., 12 March, 1993. Quantification of  $m_{Lg}$  for small explosions, *Report PL-TR-93-2070*, 90 pp., Phillips Laboratory, Hanscom Air Force Base, Mass.

Minon, S. and D. S. Dreger (2008). Stable inversion for complete moment tensors, *Geophysical Journal International*, 174(2) 585-592. <https://doi.org/10.1111/j.1365-246X.2008.03797.x>

Tape, W., and C. Tape (2012). A geometric setting for moment tensors, *Geophys. J. Int.* 190(1), 476-498, <https://doi.org/10.1111/j.1365-246X.2012.05491.x>

## 1.2 Installation, Setup, Software Requirements and Optional Software

MTINV software compilation requires standard C and Fortran77 compilers compatible with all past compiler versions (ANSI C and C89, C90, C95). We recommend GNU `gcc` and `gfortran` compilers. It is best to install and use these together (also same versions). Precompiled `gcc` and `gfortran` binaries are available from <https://hpc.sourceforge.net> for all Apple MacOS Intel and Apple-m1 systems. This installation is easy, only requires “`tar -zxf`” into the `/usr/local/` directory but requires root user permissions. Additional installation tips are noted at the [hpc.sourceforge.net](https://hpc.sourceforge.net) website. Installing `gcc` and `gfortran` on Redhat Linux usually requires downloading and installing development packages (Redhat Package Manager-RPM). Also check out [https://fortran-lang.org/en/learn/os\\_setup/install\\_gfortran/](https://fortran-lang.org/en/learn/os_setup/install_gfortran/) for other OS systems.

We regularly compile and test the toolkit on MacOS and RH Linux operating systems. The Unix C shell (`/bin/csh`) and extended C shell (`/bin/tcsh`) works best. Use of MTINV with other Unix shells is not recommended as the MTINV toolkit code base would need extensive revision (contact me if necessary). Generic Mapping Tools (GMT) <https://www.generic-mapping-tools.org> is also optional; however, many of the plots require GMT, therefore it is highly recommended. MTINV is compatible with GMT version 5 and 6. It is recommended to download and use the Apple Disk Image file (\*.DMG) from github GMT versions 6+. Older GMT

4+ installations are no longer maintained in MTINV version 4 and there are no future guarantees that -gmt4 flags in MTINV applications work in future installations. A new feature added optionally requires python3.

### **Installation:**

1. download `mtinv.version.tar.gz` package file from:
  - a. <https://sourceforge.net/projects/mtinv/>
2. move the package file to installation directory:
  - a. (e.g.) home directory, `/usr/local`, or `/opt`
3. decompress and expand the package file
  - a. `> tar -zxvf mtinv.version.tar.gz`
4. `> cd ./mtinv.version`
5. `> make clean`
6. `> make all`
7. executables and scripts are installed into the `./mtinv.version/bin` directory
8. See C shell script `./mtinv.version/environmental_variables.csh` for required and optional environmental variables
  - a. Add environmental variables `MTINV_PATH` to C shell startup file
  - b. Add bin directory executable path (e.g.),
    - i. `set PATH = ( $PATH /home/user/MTINV.version/bin ... )`
  - c. Set other environmental variables as needed.
  - d. Set the `MANPATH` path variable (NOTE! Since MacOS 10.2 Apple uses `manpath` to automatically set `MANPATH` for `cs`h users).
9. Start a new C shell
10. Type on the command line to see that the usage prints to the screen, also try man page:
  - a. `> mtinv`
  - b. `> man mtinv`

### **Optional Software Installations:**

1. GMT
  - a. <https://www.generic-mapping-tools.org/download/>
2. Ghostscript
  - a. <https://www.ghostscript.com> (PC/Linux also has RPM)
  - b. <https://pages.uoregon.edu/koch/> also see Preview (Mac OS)
3. Python3 and numpy package
  - a. <https://www.python.org/downloads/>
  - b. A package handler like <https://www.anaconda.com/download/>
4. SQLite3, Oracle sqlplus, or mysql
  - a. <https://sqlite.org/index.html>
5. Seismic Analysis Code (sac)
  - a. <https://ds.iris.edu/ds/nodes/dmc/software/downloads/sac/>

### 1.3 Workflow Overview

We recommend MTINV software installation in separate directory from the data analysis. The MTINV toolkit design allows the data analysis in any directory location. Typically, directories are made with the name of the seismic event or project and a unique identifier (e.g., origin date and time) or EventID (a database unique key). This way many events in the same area can be grouped in one directory. Within this directory, subdirectories or “children directories” are made for the seismogram data (in SAC format) and instrument response in the form of simple text files in SAC pole zero format (see tutorial examples in `mtinv4_sampledata.tar`).

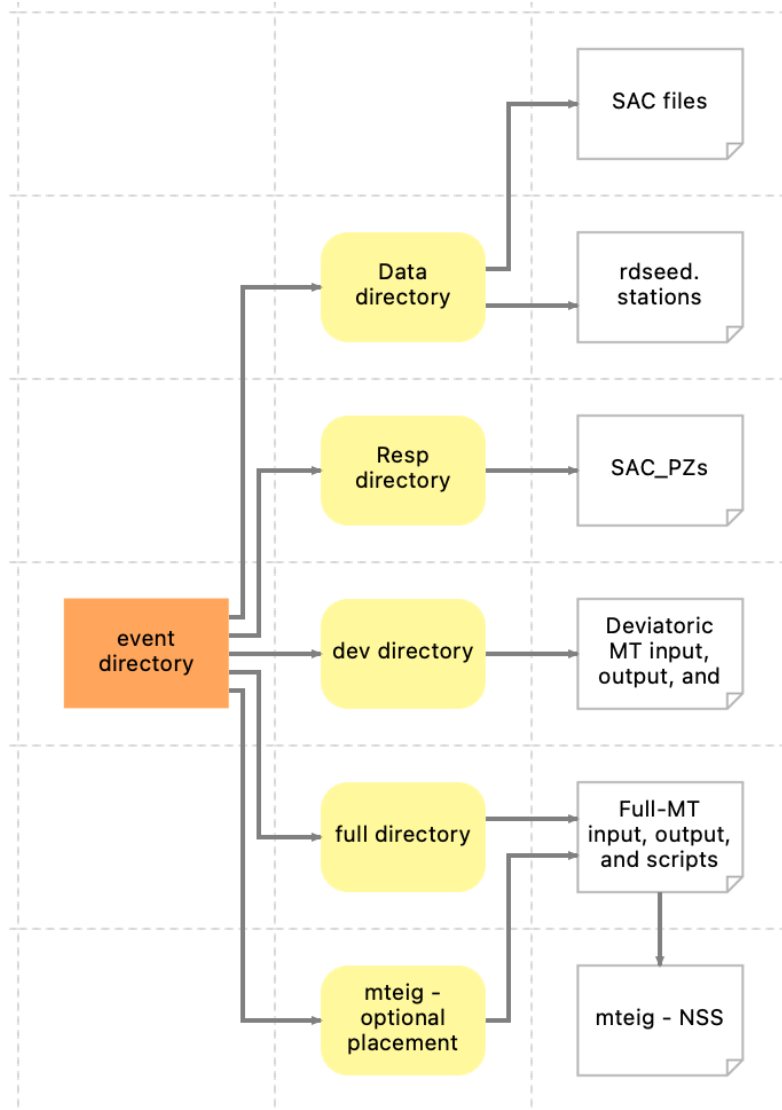


Figure 1. Recommended directory structure for moment-tensor inversion analysis. It is best to keep the data and response directories separate from the MT analysis. We also recommend keeping deviatoric MT and full MT versions in separate directories (`dev` and `full`) as well as versions with different epicenter locations or velocity models. The `mteig` application can be run in the full-MT directory or the full-MT directory can be cloned (`cp`) and `mteig` can be run separately in its own directory.

The following program's applications are modularized, and the workflow is split into five stages to provide optimal flexibility. There are two major shell scripts in stage-2 and stage-3, which first runs `mkgrnlib` that generates the Green's function libraries. The second shell-script combines the functions of `glib2inv`, `sacdata2inv`, and finally `mtinv` in a loop (See Figures 2-5) for the MT inversion. These shell scripts can be made in a text editor or autogenerated by `MTINV` toolkit helper applications.



Figure 2. A flowchart of all five stages of the moment tensor inversion analysis. Stage-1 requires unpacking the data. IRIS/EarthScope-GAGE used to distribute waveform data as SEED files but now uses archived SAC files. Stage-2 generates the Green's functions. Stage-3 is the MT inversion stage. The parameters are fine tuned in this stage to estimate the best fitting moment-tensor soliton. Once the best-fit is finalized, Stage-4 is run to generate publication quality figures, maps, and updates to a SQL database. A new stage-5 was added for estimation of the Network Sensitivity Solution (NSS) using `mteig`. The following Figures 3-7 expand more on details of each stage.

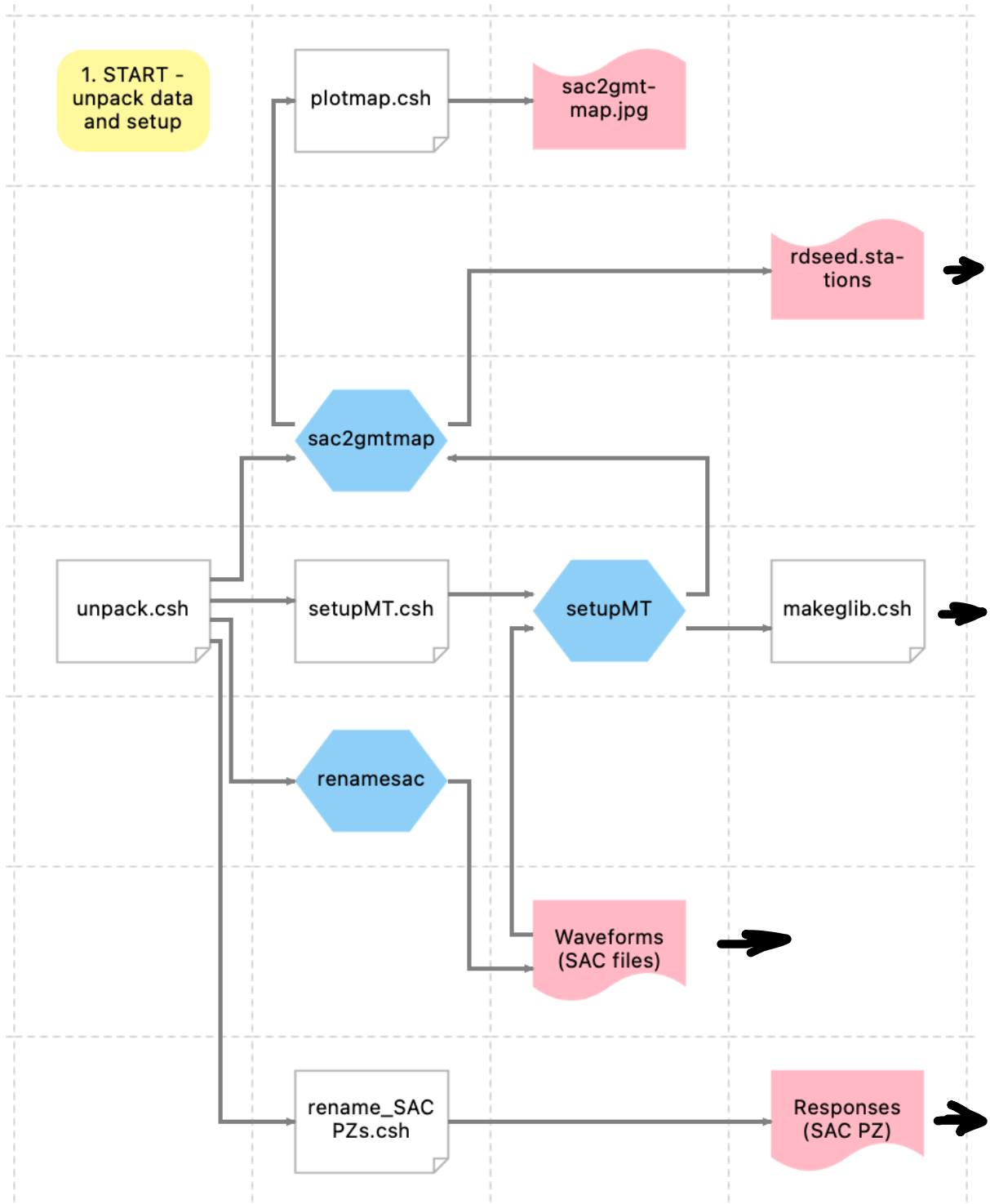


Figure 3. Stage-1 unpacks waveform and instrument response files (e.g., from IRIS/SAGE) and organizes files into two separate directories ( `./Data/*.SAC` `./Resp/SAC_PZs_*` ). The script `setupMT.csh` contains a 1-line command that autogenerates `makeglib.csh` script for computing Green's function libraries based on all available waveform data. Previously RDSEED provided a station location listing; however, IRIS/SAGE now deprecated RDSEED so we use `sac2gmtmap` to generate a file called `./Data/rdseed.stations`.

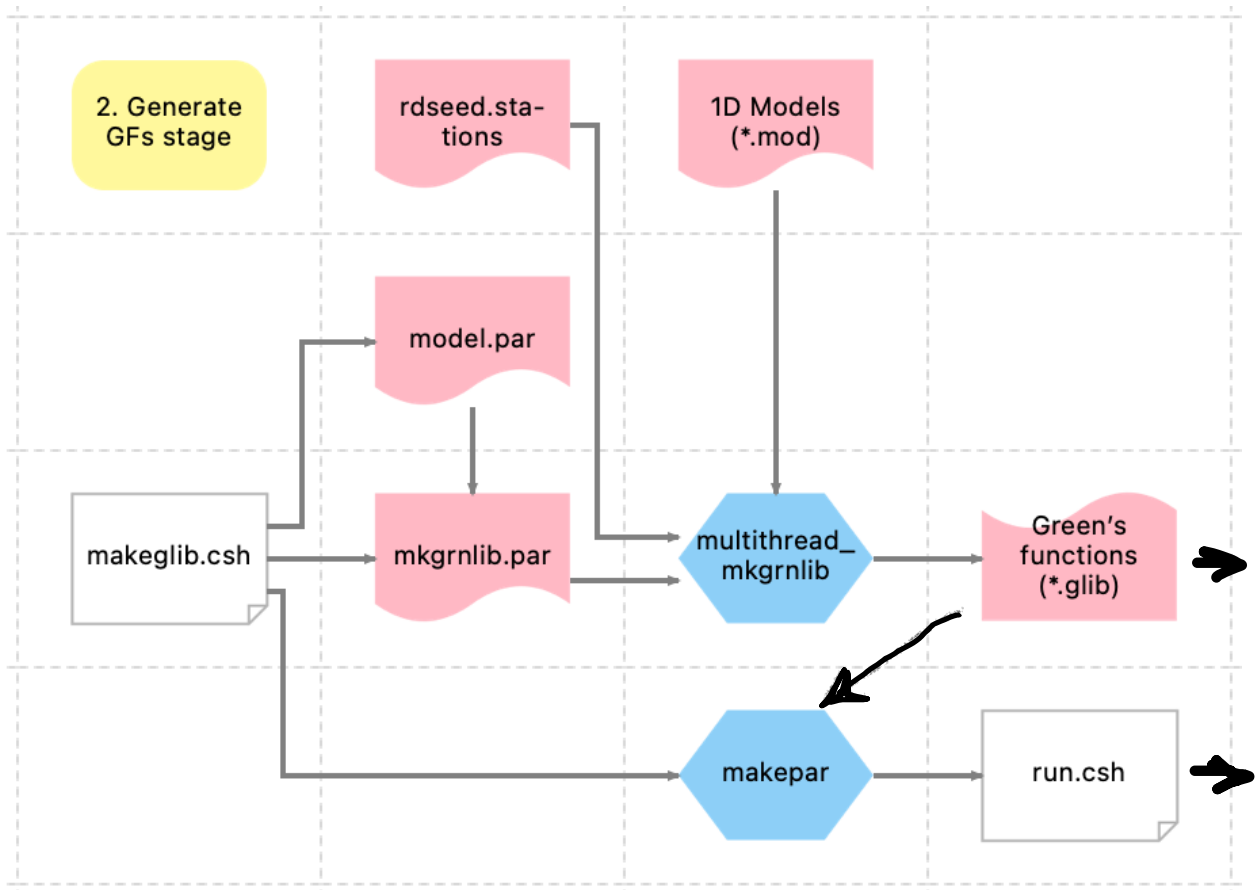


Figure 4. Stage-2 runs `makeglib.csh` from the previous stage to calculate Green's function libraries. The user can edit this C-shell script to change velocity models, remove stations, or change sampling rates. `multithread_mkgrnlib` is a helper application that launches multiple instances of `mkgrnlib` through fork process, with one fork per station. Afterwards, another helper application called `makepar` autogenerates the `run.csh` script based on the Green's function libraries (\*.glib) created by `mkgrnlib`.

`mkgrnlib` generates the Green's function libraries as a function of depth. They are stored in station specific binary files in the form of the ten fundamental faulting orientations (RSS is radial component strike-slip, RDS is radial dip-and-strike-slip, RDD is radial component dip slip, REP is the radial component explosive component, ZSS is vertical strike-slip, ZDS is the vertical component dip-slip, ZDD-vertical strike-slip, ZEP-vertical isotropic, TSS-transverse component strike-slip, TDS-transverse dip-slide). Note that the transverse does not contain a DD or EP terms (isotropic sources do not radiate SH-waves). The program inputs include the station and source locations, directory path pointer to the `modeldb= (*.mod)` files and station database files (files come `./mtinv.version/data/`). However, model files and station listing files can be placed anywhere and only pointers in the PAR files need to be specified.



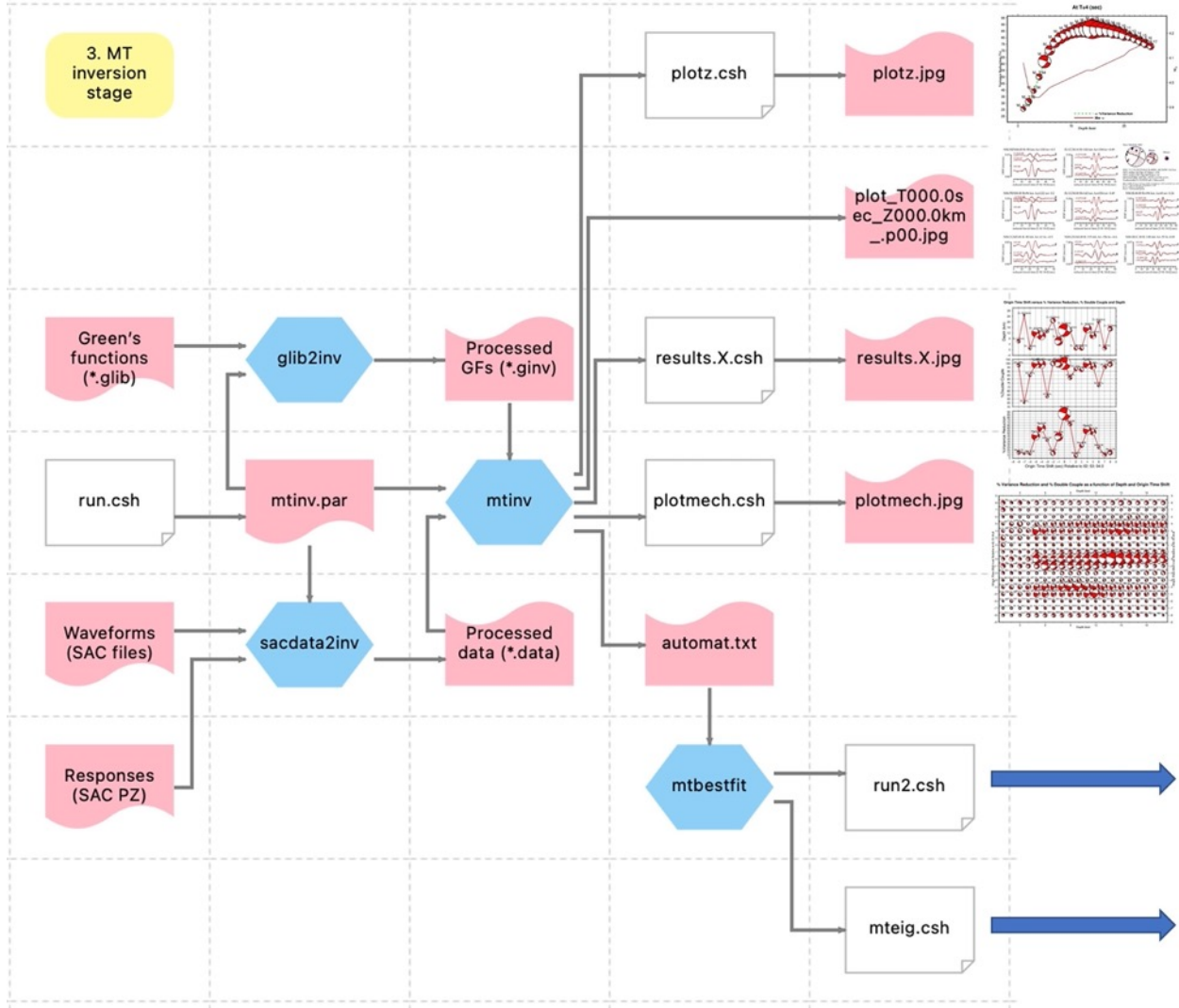


Figure 5. Stage-3 is the MT inversion stage. The `run.csh` script contains the parameter “PAR” file `mtinv.par` which should be edited to change filter bands, use Boolean flag to turn on or off stations, time shifts, and sample points “`npts=`” and sampling rate “`dt=`” to change the time-window duration. The origin-time is entered at the top of the par file “OT” tag in absolute time (in UTC). Inside `run.csh`, there are loops over origin-time shifts “`ts0`” around the origin-time entered at the top of the par file. The default `ts0` is +/- 8 secs in 1 sec increment relative to the origin-time. At `ts0` each iteration, `mtinv` loads the par file `mtinv.par`, processed Green’s function libraries ( `*.ginv` ), and processed data files ( `*.data` ) and writes out to separate files or appends to single file outputs. The file `automt.txt` (shared memory) is appended to after each iteration and then is read by `mtbestfit` to determine the origin-time and depth with the optimal %VR and writes `run2.csh` to finalize the MT solution. If this is a full-MT (`mtdegree=6`), then it also writes a `mteig.csh` script to compute the NSS.

The number of points and sampling rate need to be set to so that the Green’s function length in time is as long or longer than the expected slowest traveling waves. Typically for full



waveforms at regional distances, the slowest waves travel around 1 km/s therefore a station 300 km away would need at least 300 secs. For a `npts=1024` time points, a sampling rate of at least `dt=0.3` samples/sec is needed. One could also over sample at `dt=0.15` samples/sec and `npts=2048` points and later use `decimate` to higher sampling rates. This is accomplished later in the program `glib2inv` that will filter and interpolate/decimate the synthetics to match that of the data. Version 2 of the code also includes a reduction velocity so that the first-time sample of the Green's function can be delayed from the origin time to before the first P-wave arrival that avoids wasted computation time particularly for the larger distance range. The reduction velocity is also handy for the separate inversion of  $P_{nl}$  and surface waves. `glib2inv` reads an input file for the moment tensor inversion (stations used including band-pass corner frequencies, number of samples and sampling rate) and the output Green's function library created by `mkgrnlib` and generates a processed and filter library for the moment tensor inversion. All input files are in text form and keyed to the 1 to 8-character station code and the 1-to-4 character network ID (fully qualified by the FDSN or any dummy code). In MTINV version 3+, location codes were added (e.g., "", "00", "01", "10"), which identifies multiple sensor types at a single station (empty quotes "" is null location code).

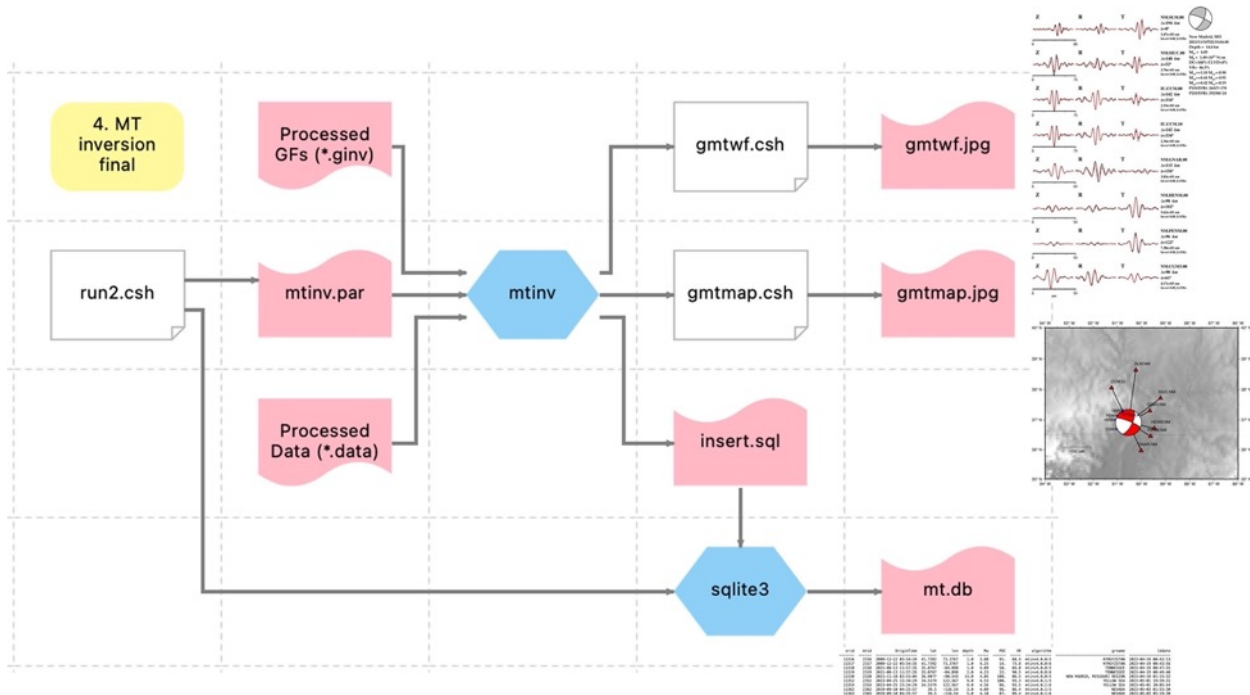


Figure 6. Stage-4 MT inversion final. The script `run2.csh` is like the previous script `run.csh` except that it runs `mtinv` just once at the best origin-time and depth with options to generate a map and publication quality plots. Note that `sacdata2inv` and `glib2inv` are not run here in this stage flow and uses the files from previous stage. A `insert.sql` SQL script is written, and the format depends upon the options used for the database. We now recommend `sqlite3`, since it is free, uses a file format, and does not require a database management system nor a database server.

`sacdata2inv` reads an input PAR file `mtinv.par` for the moment tensor inversion (same as that used for `glib2inv`) and generates a processed and filtered version of the data in the form of a binary input file from SAC files extracted from SEED volume via RDSEED or elsewhere. The program requires the directory path locations for the SAC formatted data files and SAC pole-zero response files. Given a station and network character pair, the program will automatically figure out the appropriate files and load the data and do the instrument correction. The toolkit requires the RDSEED SAC file and PZ file naming convention. A simple tool to rename the sac files is provided in version 2. SAC file also requires some basic header information including `kstnm`, `kcmpnm`, `knetwk`, `cmpaz`, `cmpinc`, and `khole`.

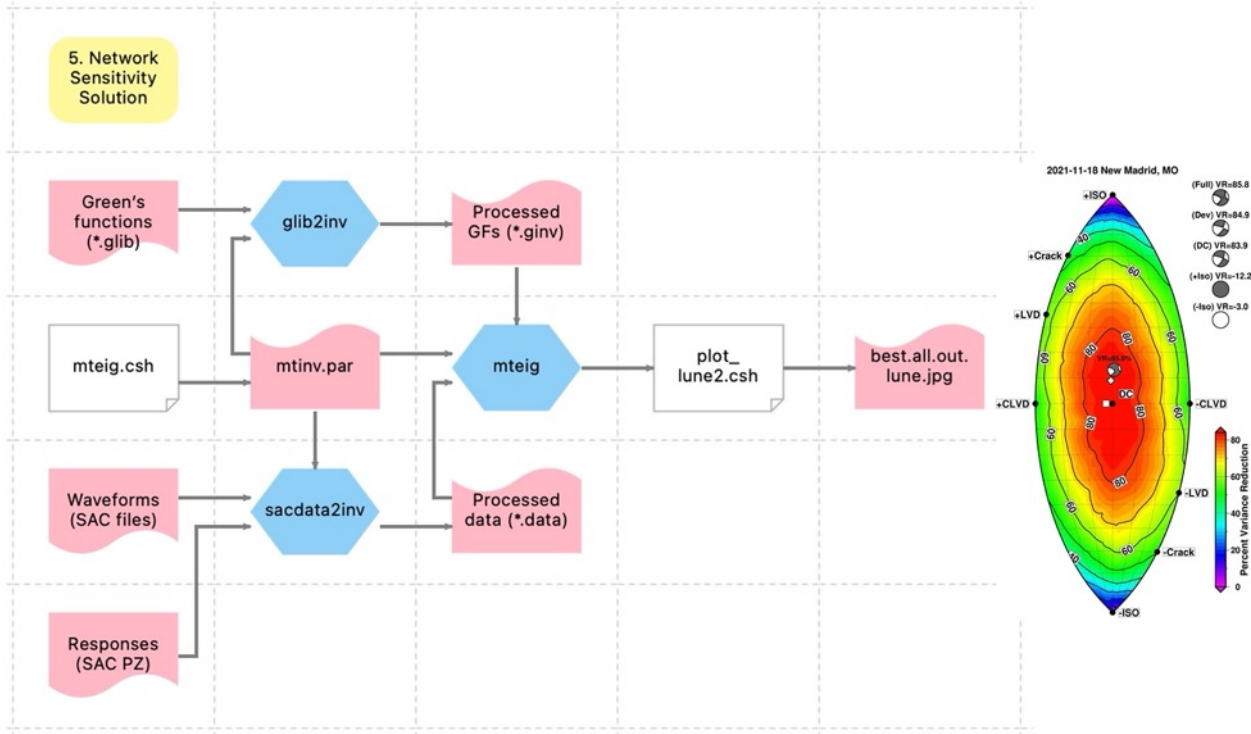


Figure 7. Stage-5 is a new stage added in MTINV version 4+ that computes the Network Sensitivity Solution (NSS). The stage flow is similar to `run.csh` in that it replaces `mtinv` with `mteig` with the same inputs (`mtinv.par`, `*.ginv`, and `*.data` input files). The program `mteig` assumes the origin-time and depth are fixed, optimally estimated from previous stages.

`mtinv` reads output from `sacdata2inv` and `glib2inv`, performs the inversion for each source depth and generates output. The input file (`glib2inv.par`), the same parameter input read by `sacdata2inv` and `glib2inv`, can be modified to remove certain stations completely or just turn stations off in the inversion yet still have a prediction made using the current solution. The forward calculation is still an option in `mtinv`. Just add any pure double couple solution in the EV line of the parameters file. Changing any parameters, for example, changing the filter bands requires re-running both `glib2inv` and `sacdata2inv`. The Green's functions do not need to be re-calculated in `makeglib`, only the two programs (`sacdata2inv` and `glib2inv`) are rerun to apply any changes. The origin time shift can be introduced in `mtinv` version 2 without having to rerun `glib2inv` and `sacdata2inv`. Version 2 also has the correct calculation for the deviatoric and full moment tensor including the isotropic component.

Additionally, a distance normalization option has been added to `mtinv` and a 1-degree of freedom (isotropic only) inversion has been added in version 2.

For estimating origin time and source depth a loop in the shell script provides (at best) one second increments in origin time without recalculation or reprocessing of the Green's functions. Only the data are shifted in time within `mtinv` without reprocessing the data windows therefore time shifts should be kept to within a few seconds because over-shifting may cause edge effects from windowing and Fourier-domain filtering.

This NSS approach (e.g., *Ford et al.*, 2010; *Tape & Tape*, 2017), with a spherical uniform random sampling in both eigenvalue and vectors on the fundamental lune, is different compared to other approaches using a uniform grid. The application used to compute the NSS is `mteig` and it has two main required parameters `nsim_eig` and `eigvec_fac`. We used a random sampling of `nsim_eig` number of eigenvalues from a uniform spherical distribution thereby avoiding oversampling near the poles. For every random point on the fundamental lune, we search `nsim_evec` number of eigenvectors by mapping a uniformly distributed random vector into a  $3 \times 3$  rotation matrix. The parameter `nsim_evec` is deprecated and replaced by `eigvec_fac`. The new parameter `eigvec_fac` replaces `nsim_evec`, instead of a constant number, we scale the number of eigenvectors by the distance from the origin (equator DC point).

Given a point on the fundamental lune represented by a set of three-eigenvalues, each randomly chosen three-eigenvectors are then paired to form a MT. Then the MT is used to forward calculate synthetics and a %VR is estimated with the data. We finally sort the resulting number of eigenvalue and eigenvectors sets of calculated %VR for the maximum value and this is contoured for the given point in fundamental lune. The main reason for using random sampling instead of uniform sampling NSS approach (e.g., *Ford et al.*, 2010) is that uniform sampling requires calculating %VR for  $10^{11}$  to  $10^8$  synthetic seismograms, using strike, dip, and rake formulation for each point on the source grid assuming fixed depth and  $M_0$ , rather than a more reasonable  $10^6$  calculations using random sampling in eigenvalues and eigenvector formulation approach.

We begin to identify some rough broad %VR contours on the fundamental lune with only `nsim_eig=2000` points and with higher values there are only small differences relative to using `nsim_eig=4000` or more points. The use of a minimum `eigvec_fac=17000` eigenvectors is required and particularly important for reaching a smooth %VR contour near the double-couple (DC) region on the fundamental lune. Tests using `nsim_eig=2000` and `eigvec_fac=17000` results in approximately 8-million forward calculations for each event NSS but with multithreading can be processed on a desktop computer within several minutes for four-station data set. A faster and more efficient alternative developed by *Nayak and Dreger* (2015) performs an iterative non-linear inversion of the eigenvectors for each point in source type space avoiding the issues of under- or over-sampling of the eigenvectors on the fundamental lune. Overall, the forward calculated grid-search results are like their method in test comparisons, despite small differences of the %VR contours near the DC point on the eigen-sphere lune.

## 1.4 Input file formats

Station database: The station database is a simple text file in free form column format. The first column is the 3-4 letter stations code, followed by the 1-2 letter network code, station latitude, longitude, elevations. The following are included in the file but not needed, a string

including the available components (e.g., “BHZ, BHN, BHE”), description of the station location in quotes, and the start and end dates. This file is automatically generated by RDSEED (see man RDSEED and `unpack.csh`). A file with most global stations is included in this toolkit package.

**Model database:** The format of the velocity mode database is also in simple free-form column format. The first column is the layer thickness, followed by the P-wave velocity in km/sec, the P-wave Q, S-wave velocity, S-wave Q, and density in grams/(cubic centimeters). Comments begin with the ‘#’ character in the first column.

**SAC binary files.** The SAC file format has stayed the same over the last 25 years. The first 632 bytes are the header values, a collection of floating, integer numbers and strings. This is followed by a series of data values in 4 byte floating type words. See the file `sac.h` in the include directory for the format. Programs that convert SAC binary files to text files are included in the misc directory.

## 1.5 Output file formats

1. Binary i/o files have a suffix which end in \*.data, \*.glib, \*.ginv (see include/mt.h and source codes for specifications).
2. SAC binary format (SEE ABOVE).
3. Postscript/PDF graphics. Files are named `plot_TXXX.X_sec_ZYYY.Ykm_p01.ps` where **XXX.X** is the origin time in seconds relative to the nearest minute, and **YYY.Y** is the source depth in km. These files show the predicted and observed waveform ground displacement fits in units of microns ( $1 \times 10^{-6}$  meters), the full, major and minor double couple solution and some information about the solution.
4. Text files:
  - 4.1. Email\_`TXXX.X_sec_ZYYY.Ykm`.txt files list the comments, input values and resulting moment tensors in ASCII text format. The values **XXX.X** is the origin time in seconds relative to the nearest minute, and **YYY.Y** is the source depth in km. The files are ready for email or linking/posting in a html web page. The plain text files include the input parameters, the moment tensor solution including the tensor, principal axes and strike, dip and rake, a graphical text form of the major double couple focal mechanism and information about the station and frequency bands.
  - 4.2. Free form text files in column form. Lists the earthquake longitude, latitude, best depth for that origin time, origin date, origin time, moment magnitude, scalar seismic moment in Dyne-cm, percent double couple, percent variance reduction, fitness 1, fitness function 2, raw amplitude misfit, the six moment tensor elements and the exponent of the seismic moment, strike, dip, rake for nodal plane 1, strike, dip, rake for nodal plane 2, T, P, and N-axis plunge and azimuth, number of stations used, and stations codes.
  - 4.3. Free form text files for input into GMT script `plotmech.csh`. The GMT script is automatically generated by `mtinv`, just run, and view the postscript or PDF output plot showing the variation of moment tensor as a function of source depth and origin time.

## 2. TUTORIAL EXAMPLES

1. Download `mtinv4_samples.tar.gz`
2. `> tar -zxvf mtinv4_samples.tar.gz`
3. `> cd ./mtinv4_sampledata/2021-11-18T025303_Missouri/`
4. `> unpack.csh 2021-11-18-new-madrid-missouri-region.tar`
  - a. `unpack.csh` will extract the SAC data files, SAC\_PZs pole-zero files
  - b. runs `sac2gmtmap` which then generates
    - i. `plotmap.csh`, `sac2gmtmap.jpg`, and `rdseed.stations`
  - c. `unpack.csh` will make a directories
    - i. `./Resp`, `./Data`, `./dev`
  - d. `unpack.csh` will make a `setupMT.csh` script in `./dev`
5. `> cd dev`
6. `> vi setupMT.csh`
  - a. Use a text editor to uncomment one of the 3 lines for the New Madrid, MO earthquake of 2021-11-18.
7. `> setupMT.csh`
  - a. Executing generates `makeplib.csh`
8. `> vi makeplib.csh`
  - a. Use a text editor to inspect the following in the file
    - i. In this case the model name is `cus` – or central US.
    - ii. In `par` file `cus.par`, make sure `stadb=` and `modeldb=` paths are set correctly.
    - iii. `zrange={start,increment,stop}` in `km`
    - iv. Any changes to epicenter location `evla` and `evlo` are placed here. Updates would require rerunning the script to recalculating the Green's functions.
    - v. For `mkgrnlib.par`, add or remove comments `"#"` from 1st column
    - vi. Version 2 includes the tool `makepar`. This tool will scan the directory for Green's function libraries and generates a generic `run.csh` script for the inversion. This application will not overwrite `run.csh` if one exists already.
    - vii. `date={origin-time}`
    - viii. confirm `DataDir=./Data` and `RespDir=./Resp`
    - ix. `maxsta=8` (only use max 8 stations initially in the inversion)
    - x. `maxdist=800 km` (only initially use stations within 800 km) all other stations will be initially used as predictions
    - xi. initially sets bandpass filter corners `lf=` and `hf=` for all stations, likely need individual changes by station later after inspection
  - b. The script then executes `mkgrnlib` and uses the parameter file to generate the Green's function libraries for a specific station code and network.
  - c. In version 3.0 the application `multithread_mkgrnlib` is provided to speed up the computation on multicore machines.
  - d. Create a `mkgrnlib.par` file that contains station code, net code, Green's function parameter file and `dt` (the sampling rate in seconds/sample).
  - e. In the command line set the parameter `executable_pathname` to your location of executable `mkgrnlib` (full pathname is necessary).
9. `> makeplib.csh`

- a. execute `makeglib.csh` script where `mkgrnlib` or `multithread_mkgrnlib` output goes to the screen or redirected to a file
  - b. `> cat multithread_mkgrnlib.out,`
    - i. Optionally Inspect the redirected output from `mkgrnlib`, the output is not in order since the process is run multithreaded.
  - c. `> ls -l *.glib`
    - i. Optionally show a listing of the computed Green's function libraries
10. `> vi run.csh`
- a. open, examine and modify the auto-generated `run.csh` script if needed.
  - b. The script is automatically set `DEGFREE=5` for solving for a deviatoric moment tensor which is the recommended setting for all tectonic sources. The advance user can use the setting `DEGFREE=1` that will only invert for pure isotropic sources or `DEGFREE=6` for full moment tensor (isotropic + deviatoric).
  - c. The script will generate a parameter file for 3 toolkit applications: `glib2inv`, `sacdata2inv`, and `mtinv`. The par file can have any name. See the man pages for the format.
  - d. The first line tagged CM is a comment for the event that get printed as a title for output files. The second line tagged OT is the origin time. Be careful with the format, the date and time are one string separated by commas.
  - e. The EV tagged line is only needed if the user would like to do a forward calculation assuming a pure double couple source.
  - f. The following lines are the station parameters.
  - g. Check and modify the `run.csh` file so that the system dependent plotting and graphics routines are available.
  - h. To run a forward MT calculation, in `run.csh`
    - i. Add strike, dip, rake, Mw to EV line in `mtinv.par`
    - ii. Comment `# mtinv` line in for loop
    - iii. Add the following line after `sacdata2inv`
      1. `"mtinv fwd mtdegfree=5 par=mtinv.par"`
11. `> run.csh`
- a. View plots
  - b. `> open plot*.jpg` and `results.5.jpg`
12. `> vi run.csh`
- a. Through iteration, edit bandpass filter corners, `ts0` values, etc...
13. `> run.csh`
- a. Each run of the script cleans `automt.txt` and then append to the file the fit at each origin-time shift and depth for `mtbestfit` to read.
  - b. The last line of the file runs `mtbestfit` to determine the best parameters to finalize the MT solution
14. `> run2.csh`
- a. Finalize the MT solution
  - b. `list_MTdb.csh` – lists all the MT solutions in database
15. `> mteig.csh`

**TABLE 1.**

./mtinv4\_sampledata/2021-11-18T025303\_Missouri/dev/setupMT.csh

```
#!/bin/csh
setupMT -z 16 -ev 36.9077 -90.543 -mod cus -ot 2021-11-18T02:53:04 \
-com "New Madrid, MO" -fil 0.075 0.150 ../Data/*Z?.SAC ../Data/*Z.SAC
```

**TABLE 2.**

./mtinv4\_sampledata/2021-11-18T025303\_Missouri/dev/makeglib.csh

```
#!/bin/csh

cat >! cus.par << EOF
velmod=cus
#zrange=3,3,33
zrange=1,1,25
evla=36.9077
evlo=-90.543
dt=0.15
nt=2048
fmax=0.4
t0=0.
redv=18.
damp=1.
kmax=20000
eps=0.0005
smin=0.0005
modeldb=${MTINV_PATH}/data/modeldb/
stadb=../Data/rdseed.stations
noverbose
nodump
EOF

cat >! mkgrnlib.par << EOF
### station-code network-code location-code mkgrnlib.parfile dt(sec/sample) ### comments
CGM3 NM "00" cus.par 0.050 ### R= 90 Az=061 NM.CGM3.00 nchan=1 nseg=1 (HHZ)
PENM NM "00" cus.par 0.050 ### R= 96 Az=122 NM.PENM.00 nchan=1 nseg=1 (HHZ)
HENM NM "00" cus.par 0.050 ### R= 98 Az=102 NM.HENM.00 nchan=1 nseg=1 (HHZ)
GNAR NM "00" cus.par 0.050 ### R= 115 Az=156 NM.GNAR.00 nchan=1 nseg=1 (HHZ)
CCM IU "00" cus.par 0.070 ### R= 142 Az=334 IU.CCM.00 nchan=1 nseg=1 (BHZ)
CCM IU "10" cus.par 0.070 ### R= 142 Az=334 IU.CCM.10 nchan=2 nseg=2 (BHZ HHZ)
SIUC NM "00" cus.par 0.070 ### R= 148 Az=052 NM.SIUC.00 nchan=1 nseg=1 (HHZ)
SLM NM "00" cus.par 0.080 ### R= 194 Az=008 NM.SLM.00 nchan=2 nseg=2 (BHZ HHZ)
WVT IU "10" cus.par 0.110 ### R= 257 Az=109 IU.WVT.10 nchan=2 nseg=2 (BHZ HHZ)
WVT IU "00" cus.par 0.110 ### R= 257 Az=109 IU.WVT.00 nchan=2 nseg=2 (BHZ HHZ)
EOF

multithread_mkgrnlib \
parfile=mkgrnlib.par \
executable_pathname=${MTINV_PATH}/bin/mkgrnlib > multithread_mkgrnlib.out

makepar com="New Madrid, MO" \
date="2021-11-18T02:53:04.00" \
DataDir=../Data \
RespDir=../Resp \
gmt5 nooracle nomysql sqlite\
maxsta=8 maxdist=800 \
lf=0.075 hf=0.15 \
minsnr=3.0 ctoi=0.85 maxshift=10 realtime no local *.glib
```



**TABLE 3.**

./mtinv4\_sampledata/2021-11-18T025303\_Missouri/dev/run.csh

```

#!/bin/csh
#### Realtime version - set in makepar ####
set DEGREE=5 # 1-isotropic_mt 5-deviatoric_mt 6-full_mt

cat >! mtinv.par << EOF
#### REGION COMMENT #####
CM New Madrid, MO
#### Date and Origin Time #####
OT 2021-11-18T02:53:04.00
#### Forward Calculations #####
## stk dip rak Mw evlo evla Z #####
EV -999.0 -999.0 -999.0 0.0 -90.543 36.9077 15.0
#####
# sta net loc model np pas lf hf nt dt tr tt v/d mulfac used(Y/N) ts0 weight ## #
CGM3 NM 00 cus 3 2 0.075 0.150 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=89.6 Az=61
PENM NM 00 cus 3 2 0.075 0.150 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=96.1 Az=122
HENM NM 00 cus 3 2 0.075 0.150 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=97.6 Az=102
GNAR NM 00 cus 3 2 0.075 0.150 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=114.8 Az=156
CCM IU 00 cus 3 2 0.075 0.150 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=141.8 Az=334
CCM IU 10 cus 3 2 0.075 0.150 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=141.8 Az=334
SLUC NM 00 cus 3 2 0.075 0.150 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=147.5 Az=52
SLM NM 00 cus 3 2 0.075 0.150 1024 0.08 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=193.9 Az=8
#####
##WVT IU 00 cus 3 2 0.075 0.150 1024 0.11 0.0 0.0 d 1.0 n +0.00 +1.0 Surf/Pnl ### R=257.2 Az=109
##WVT IU 10 cus 3 2 0.075 0.150 1024 0.11 0.0 0.0 d 1.0 n +0.00 +1.0 Surf/Pnl ### R=257.2 Az=109
EOF

### CLEAN UP ###
/bin/rm -f *.ginv *.data plot_T???.?sec_Z???.?km_.p??ps email_T???.?sec_Z???.?km_.txt *.?dat.xy *.?syn.xy
/bin/rm -f plot_T???.?sec_Z???.?km_.p??jpg plot_T???.?sec_Z???.?km_.p??pdf *.sql
/bin/rm -f results.?.??? plotmech.??? plotz.??? gmtmap.??? mtinv.out multithread_mkgrnlib.out snr.out var_red.out
/bin/rm -f automt.txt

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel

### PROCESS DATA ###
sacdata2inv par=mtinv.par path=../Data respsdir=../Resp noverbose nodumpsac parallel

# foreach ts0 ( -8 -7 -6 -5 -4 -3 -2 -1 -0.5 0 0.5 1 2 3 4 5 6 7 8 )
foreach ts0 ( -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 )
    mtinv AutoAuth ts0=${ts0} par=mtinv.par gmt5 mtdegfree=${DEGREE} use_snr minsnr=3 shift ctol=0.85 maxshift=10 >> mtinv.out
end ### loop over ts0

### CHECK ORIGIN TIME SHIFT ###
csh results.${DEGREE}.csh
### uncomment for review
# open results.?.jpg

### convert each postscript file to jpg
foreach ps ( plot_T???.?sec_Z???.?km_.p??ps )
    gmt psconvert -Z -Tj -E120 ${ps}
end
### uncomment
#open plot_T???.?sec_Z???.?km_.p??jpg

### MAKE DEPTH SENSITIVITY PLOT ###
# csh plotz.csh
# open plotz.jpg

### MAKE DEPTH / OT-SHIFT SENSITIVITY PLOT ###
# csh plotmech.csh
# open plotmech.jpg

# csh gmtmap.csh
#open gmtmap.jpg

mtbestfit gmt5 evid=-1 db pretty_plot noforce_best_vred mteig decimate_factor=2

```

**TABLE 4.**

**./mtinv4\_sampledata/2021-11-18T025303\_Missouri/dev/run2.csh**

```
#!/bin/csh
###
### Created by mtbestfit fitType=BEST VARIANCE-REDUCTION
### ot=-0 fsec=4 z=14 vred=86.4559 fit=81.7389 pdc=100 piso=0 pclvd=0
### fit_max=81.7389 fit_max_threshold=20 vred_diff=0 vred_diff_threshold=30
###
setenv MTINV_GMT_GRID_FILE /Users/ichinose1/Work/topogmt/etopo5.grd
setenv MTINV_GMT_INT_FILE /Users/ichinose1/Work/topogmt/etopo5.int
setenv MTINV_GMT_CPT_FILE /Users/ichinose1/Work/topogmt/etopo5.cpt
setenv MTINV_DATABASE_FILE /Users/ichinose1/Work/mtinv.v4.0.1/data/mt.db

set DEGREE=5 # 1-isotropic_mt 5-deviatoric_mt 6-full_mt

###
### Clean Up
###
/bin/rm -f email_T???.?sec_Z???.?km_.txt plot_T???.?sec_Z???.?km_.p???.jpg *.ps

### MAKE GMT PLOT WITH LOCATION/STATION/SOLUTION ###
###
mtinv \
    evid=-1 \
    ts0=0 \
    par=mtinv.par \
    gmt5 \
    mtdegfree=${DEGREE} \
    gmtmap \
    write_emails \
    dumpxy nodumpsac print_gmtwf_mt_info \
    sqlite \
    PltXcorLabel \
    use_snr \
    minsnr=3 \
    ctol=0.85 shift \
    maxshift=10 >> mtinv.out

gmt psconvert -Z -Tj -E300 plot_T???.?sec_Z???.?km_.p???.ps
csh gmtmap.csh
#open gmtmap.jpg

### dumpxy option for publication quality GMT plots
gmtwf.csh

sqlite3 ${MT_DATABASE_FILE} << EOF
.read insert.sql
.quit
EOF

# updatedb conflicts in unix/Linux, changed name to updateMTdb
updateMTdb

### mtbestfit : current MT_DATABASE_FILE=/Users/ichinose1/Work/mtinv.v4.0.1/data/mt.db
# list_MTdb.csh
print_MTdb.csh > db.txt
# remove_MTdb.csh
```

## Deviatoric Moment Tensor Solution 2021-11-18 New Madrid, Missouri earthquake

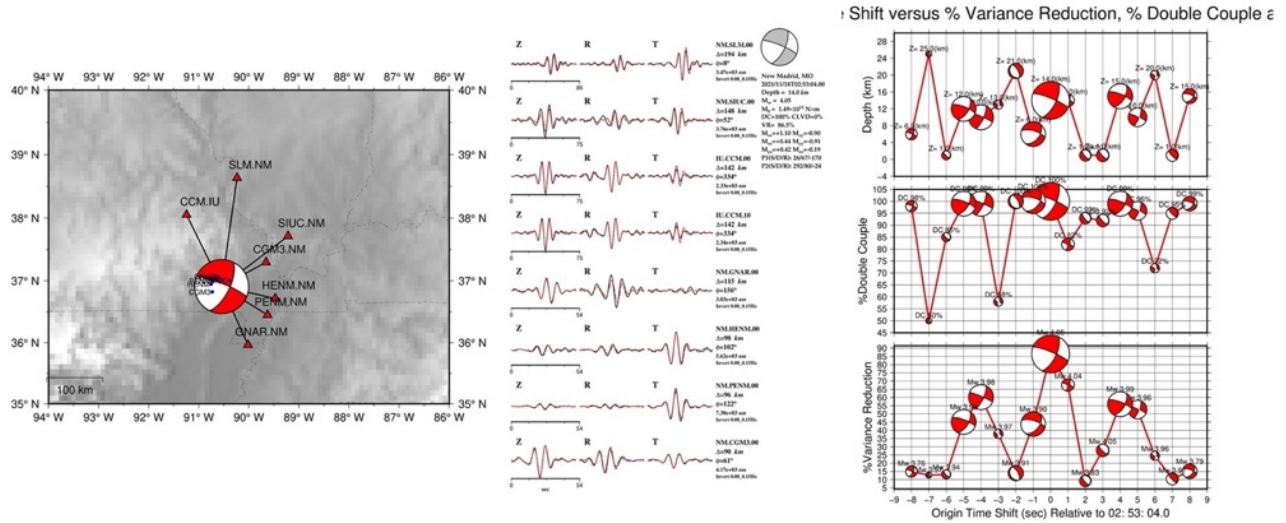


Figure 8. Deviatoric moment tensor (plots: gmtmap.jpg, gmtwf.000.jpg, and results.5.jpg). The plots were made from scripts run.csh (TABLE 3) and run2.csh (TABLE 4).

## Full Moment Tensor Solution 2021-11-18 New Madrid, Missouri earthquake

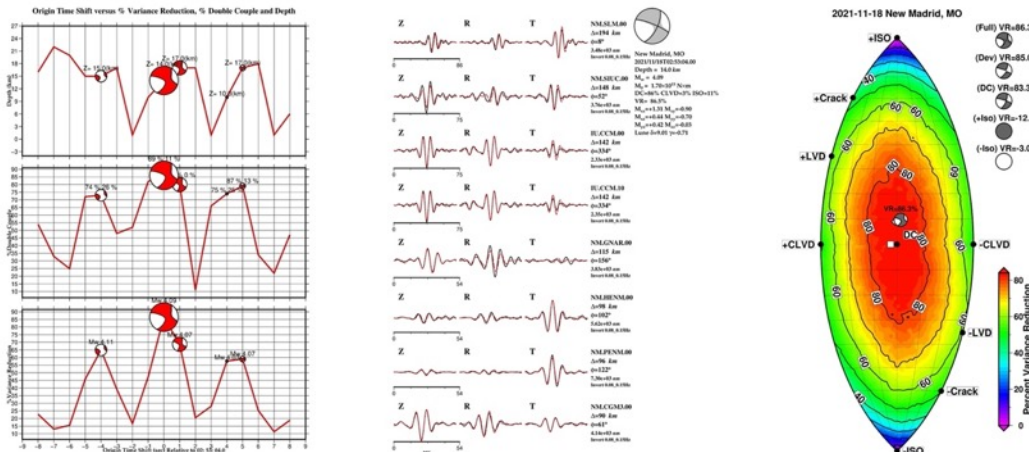


Figure 9. Full moment tensor (plots: gmtmap.jpg, gmtwf.000.jpg, and best.fit.all.1.tune.jpg). The plots were made from scripts run.csh (TABLE 3) and run2.csh (TABLE 4), and mteig.csh. (TABLE 5).

**TABLE 5.**

./mtinv4\_sampledata/2021-11-18T025303\_Missouri/dev/mteig.csh

```
#!/bin/csh
#####
### this is an autogenerated script for mteig ###
### mteig compute %VR NSS on the lune following Ford et al., 2012 ###
#####
###
### Created by mtbestfit fitType=BEST VARIANCE-REDUCTION
### ot=-0 fsec=4 z=14 vred=86.5162 fit=86.5162 pdc=86 piso=11 pclvd=3
### fit_max=86.5162 fit_max_threshold=20 vred_diff=0 vred_diff_threshold=30
###

#set DEGREE=6 ### mteig assumes degree-of-freedom is 6 for full-MT to compute %VR on lune

cat >! mtinv.par << EOF
#### REGION COMMENT #####
CM New Madrid, MO
#### Date and Origin Time #####
OT 2021-11-18T02:53:04.000
#### Forward Calculations #####
## stk dip rak Mw evlo evla Z #####
EV 26 66 -169 4.09 -90.543 36.9077 14
#####
# sta net loc model np pas lf hf nt dt tr tt v/d mulfac used(Y/N) ts0 weight ### #
CGM3 NM 00 cus 3 2 0.075 0.150 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl # R=89.5843 Az=60.8059
PENM NM 00 cus 3 2 0.075 0.150 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl # R=96.0943 Az=121.678
HENM NM 00 cus 3 2 0.075 0.150 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl # R=97.6226 Az=102.281
GNAR NM 00 cus 3 2 0.075 0.150 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl # R=114.776 Az=155.709
CCM IU 00 cus 3 2 0.075 0.150 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl # R=141.762 Az=334.341
CCM IU 10 cus 3 2 0.075 0.150 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl # R=141.762 Az=334.341
SIUC NM 00 cus 3 2 0.075 0.150 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl # R=147.527 Az=52.165
SLM NM 00 cus 3 2 0.075 0.150 512 0.16 0 0 d 1 y +0.000 1 Surf/Pnl # R=193.921 Az=7.88835
### NOTE! stas not used, are commented out and not loaded by mteig ###
### because there is no prediction only millions of forward calcs ###
EOF

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel

### PROCESS DATA ###
sacdata2inv par=mtinv.par path=./Data respdir=./Resp noverbose nodumpsac parallel

time mteig par=mtinv.par nthreads=8 \
    nsim_eig=2000 nsim_evec=4000 eigvec_fac=17000 \
    Mo=1.698473e+22 fixz=14 \
    color doplt seed=1 parallel \
    title="2021-11-18 New Madrid, MO" Add_user_Eig e0=+1.69847 e1=+0.171992 e2=-1.29014
```

## Number of Eigenvectors – affects smoothness of contours

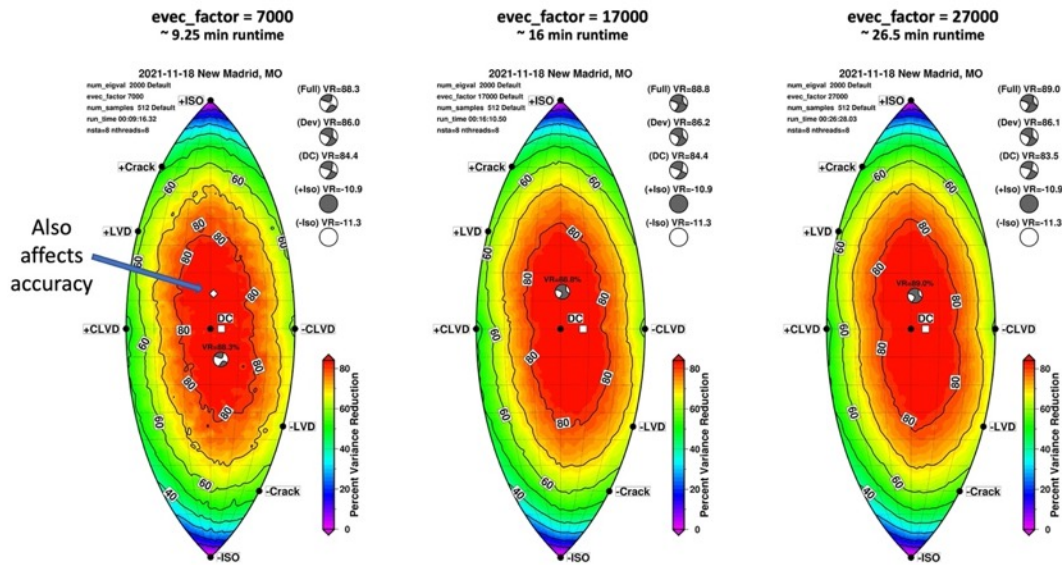


Figure 10. NSS (plots `best.fit.all.1une.jpg`) created using script `mteig.csh` (TABLE 5). The NSS plots are from `evec_factor=7000`, 17000 (default), and 27000. With `evec_factor=7000`, the contours are not smooth and the NSS best fit solution is different from the full-MT solution (see arrow).

## Number of Samples – affects runtime

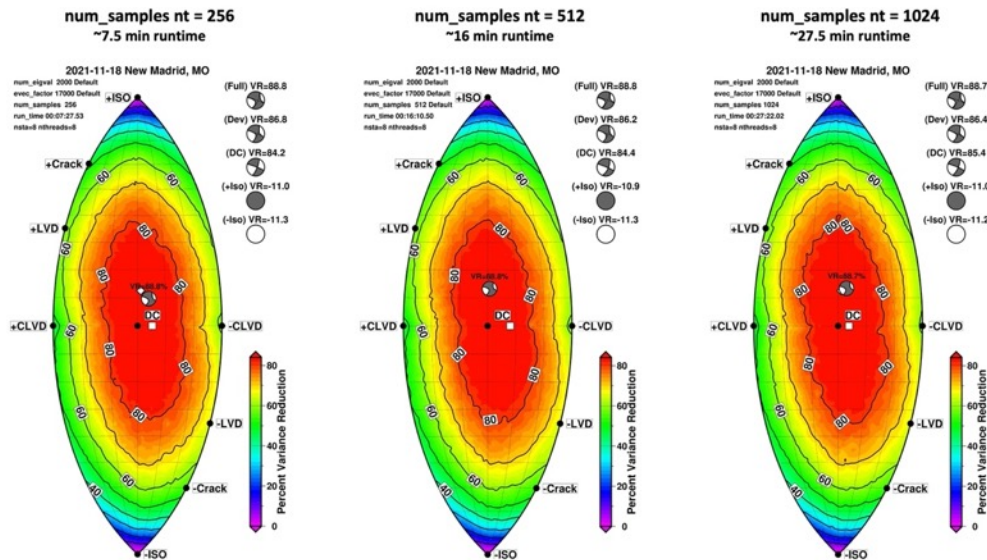


Figure 11. NSS (plots `best.fit.all.1une.jpg`) created using script `mteig.csh` (TABLE 5). The NSS plots use `nt=256`, 512 (default), 1024. There are not many differences between the three settings.



## Number of Eigenvalues – affects smoothness of contours

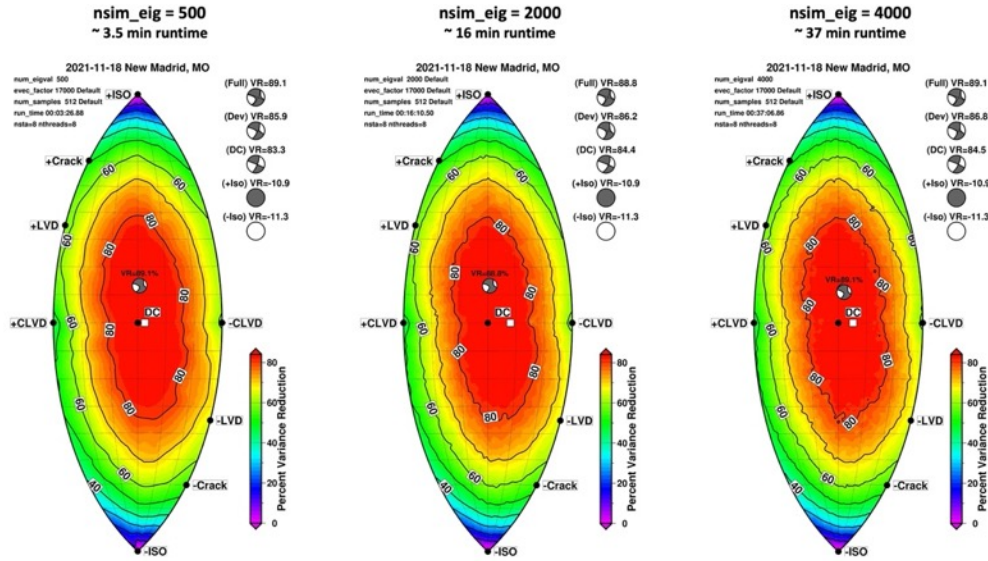


Figure 12. NSS (plots best.fit.all.lune.jpg) created using script mteig.csh (TABLE 5). The NSS plots use  $nsim\_eig = 500$ , 2000 (default), 4000. There are not many differences between the three settings but there is a significant speed up in computation time when using  $nsim\_eig = 500$  and 2000.

## Deviatoric Moment Tensor Solution

Kambrata-2, Kyrgyzstan 2009-12-22 blast filled dam

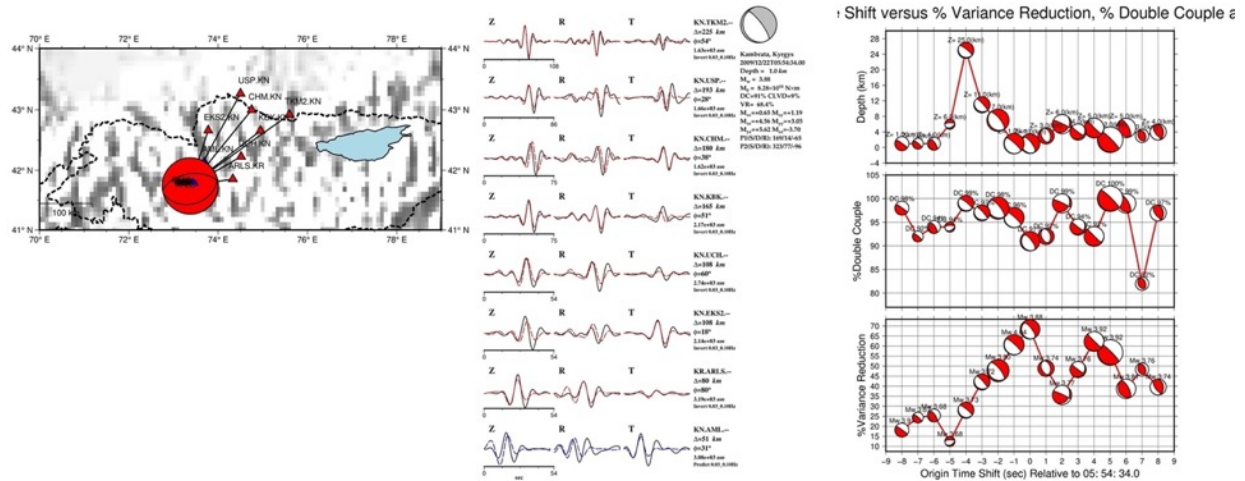


Figure 13. Deviatoric moment tensor (plots: gmtmap.jpg, gmtwf.000.jpg, and results.5.jpg). The plots were made from scripts run.csh (TABLE 6) and run2.csh.

## Full Moment Tensor Solution

Kambrata-2, Kyrgyzstan 2009-12-22 blast filled dam

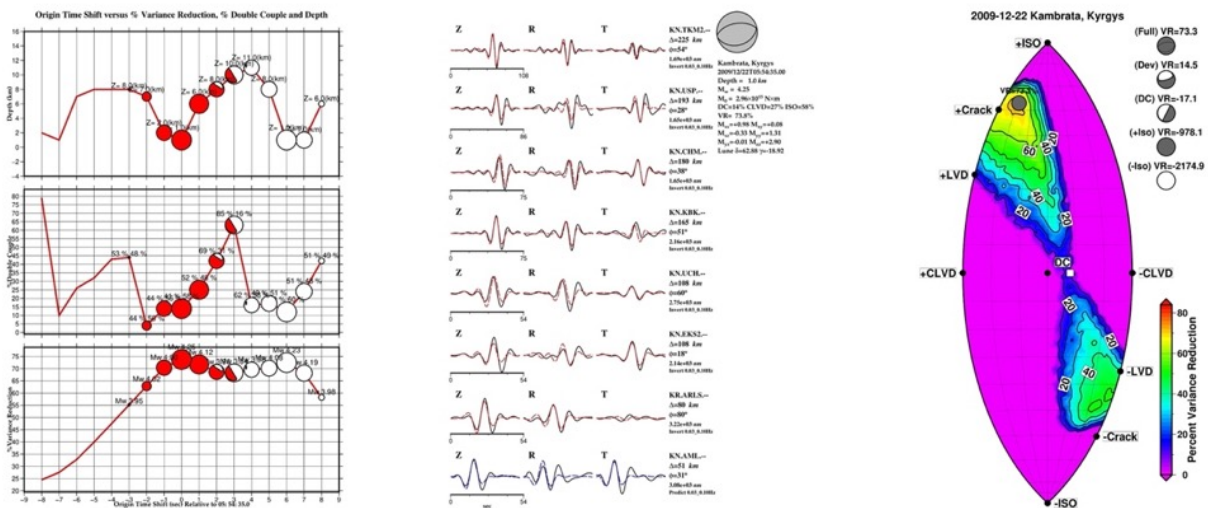


Figure 14. Full moment tensor (plots: gmtmap.jpg, gmtwf.000.jpg, and best.fit.all.1une.jpg). The plots were made from scripts run2.csh and mteig.csh. (TABLE 7).



**TABLE 6.****./mtinv4\_sampledata/2009-12-22T055435\_kyrgystan/full/run.csh**

```

#!/bin/csh
#### Realtime version - set in makepar ###
set DEGREE=6 # 1-isotropic_mt 5-deviatoric_mt 6-full_mt

cat >! mtinv.par << EOF
##### REGION COMMENT #####
CM Kambrata, Kyrgyz
#### Date and Origin Time #####
OT 2009-12-22T05:54:34.00
#### Forward Calculations #####
## stk dip rak Mw evlo evla Z #####
EV -999.0 -999.0 -999.0 0.0 73.377 41.7392 15.0
#####
# sta net loc model np pas lf hf nt dt tr tt v/d mulfac used(Y/N) ts0 weight ### #
AML KN "" wus 3 2 0.033 0.100 1024 0.05 0.0 0.0 d 1.0 n +0.00 +1.0 Surf/Pnl ### R=50.8 Az=31
ARLS KR "" wus 3 2 0.033 0.100 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=79.9 Az=80
EKS2 KN "" wus 3 2 0.033 0.100 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=107.6 Az=18
UCH KN "" wus 3 2 0.033 0.100 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=108.4 Az=60
KBK KN "" wus 3 2 0.033 0.100 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=164.6 Az=51
CHM KN "" wus 3 2 0.033 0.100 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=179.7 Az=38
USP KN "" wus 3 2 0.033 0.100 1024 0.08 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=193.1 Az=28
TKM2 KN "" wus 3 2 0.033 0.100 1024 0.10 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=224.7 Az=53
#####
#ULHL KN "" wus 3 2 0.033 0.100 1024 0.11 0.0 0.0 d 1.0 n +0.00 +1.0 Surf/Pnl ### R=243.2 Az=76
EOF

### CLEAN UP ###
/bin/rm -f *.ginv *.data plot_T???.?sec_Z???.?km_.p???.ps email_T???.?sec_Z???.?km_.txt *?.dat.xy *?.syn.xy
/bin/rm -f plot_T???.?sec_Z???.?km_.p???.jpg plot_T???.?sec_Z???.?km_.p???.pdf *.sql
/bin/rm -f results.?.??? plotmech.??? plotz.??? gmtmap.??? mtinv.out multithread_mkgrnlib.out snr.out var_red.out
/bin/rm -f automt.txt

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel

### PROCESS DATA ###
sacdata2inv par=mtinv.par path=../Data respdir=../Resp noverbose nodumpsac parallel

# foreach ts0 ( -8 -7 -6 -5 -4 -3 -2 -1 -0.5 0 0.5 1 2 3 4 5 6 7 8 )
foreach ts0 ( -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 )
  mtinv AutoAuth ts0=${ts0} par=mtinv.par gmt5 mtdegfree=${DEGREE} use_snr minsnr=3 shift ctol=0.85 maxshift=10 >> mtinv.out
end ### loop over ts0

### CHECK ORIGIN TIME SHIFT ###
csh results.${DEGREE}.csh
### uncomment for review
# open results.?.jpg

### convert each postscript file to jpg
foreach ps ( plot_T???.?sec_Z???.?km_.p???.ps )
  gmt psconvert -Z -Tj -E120 ${ps}
end
### uncomment
#open plot_T???.?sec_Z???.?km_.p???.jpg

### MAKE DEPTH SENSITIVITY PLOT ###
# csh plotz.csh
# open plotz.jpg

### MAKE DEPTH / OT-SHIFT SENSITIVITY PLOT ###
# csh plotmech.csh
# open plotmech.jpg

#csh gmtmap.csh
#open gmtmap.jpg

mtbestfit gmt5 evid=-1 db pretty_plot noforce_best_vred mteig decimate_factor=2

```

TABLE 7.

./mtinv4\_sampledata/2009-12-22T055435\_kyrgystan/full/mteig.csh

```
#!/bin/csh
#####
### this is an autogenerated script for mteig ###
### mteig compute %VR NSS on the lune following Ford et al., 2012 ###
#####
###
### Created by mtbestfit fitType=BEST VARIANCE-REDUCTION
### ot=1 fsec=35 z=1 vred=74.2228 fit=74.2228 pdc=13 piso=58 pclvd=29
### fit_max=74.2228 fit_max_threshold=20 vred_diff=0 vred_diff_threshold=30
###

#set DEGREE=6 ### mteig assumes degree-of-freedom is 6 for full-MT to compute %VR on lune

cat >! mtinv.par << EOF
#### REGION COMMENT #####
CM Kambrata, Kyrgyz
#### Date and Origin Time #####
OT 2009-12-22T05:54:35.000
#### Forward Calculations #####
## stk dip rak Mw evlo evla Z #####
EV 75 37 87 4.27 73.377 41.7392 1
#####
# sta net loc model np pas lf hf nt dt tr tt v/d mulfac used(Y/N) ts0 weight ## #
#AML KN "" wus 3 2 0.033 0.100 512 0.1 0 0 d 1 n +0.000 1 Surf/Pnl ## R=50.8 Az=31
ARLS KR "" wus 3 2 0.033 0.100 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl ## R=79.9 Az=80
EKS2 KN "" wus 3 2 0.033 0.100 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl ## R=107.6 Az=18
UCH KN "" wus 3 2 0.033 0.100 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl ## R=108.4 Az=60
KBK KN "" wus 3 2 0.033 0.100 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl ## R=164.6 Az=51
CHM KN "" wus 3 2 0.033 0.100 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl ## R=179.7 Az=38
USP KN "" wus 3 2 0.033 0.100 512 0.16 0 0 d 1 y +0.000 1 Surf/Pnl ## R=193.1 Az=28
TKM2 KN "" wus 3 2 0.033 0.100 512 0.2 0 0 d 1 y +0.000 1 Surf/Pnl ## R=224.7 Az=53
### NOTE! stas not used, are commented out and not loaded by mteig ###
### because there is no prediction only millions of forward calcs ###
EOF

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel

### PROCESS DATA ###
sacdata2inv par=mtinv.par path=../Data respdir=../Resp noverbose nodumpsac parallel

time mteig par=mtinv.par nthreads=8 \
    nsim_eig=2000 nsim_evec=4000 eigvec_fac=17000 \
    Mo=3.143999e+22 fixz=1 \
    color doplt seed=1 parallel \
    title="2009-12-22 Kambrata, Kyrgyz" Add_user_Eig e0=+3.144 e1=+1.37961 e2=+0.968433
```

## Number of Eigenvectors – affects smoothness of contours

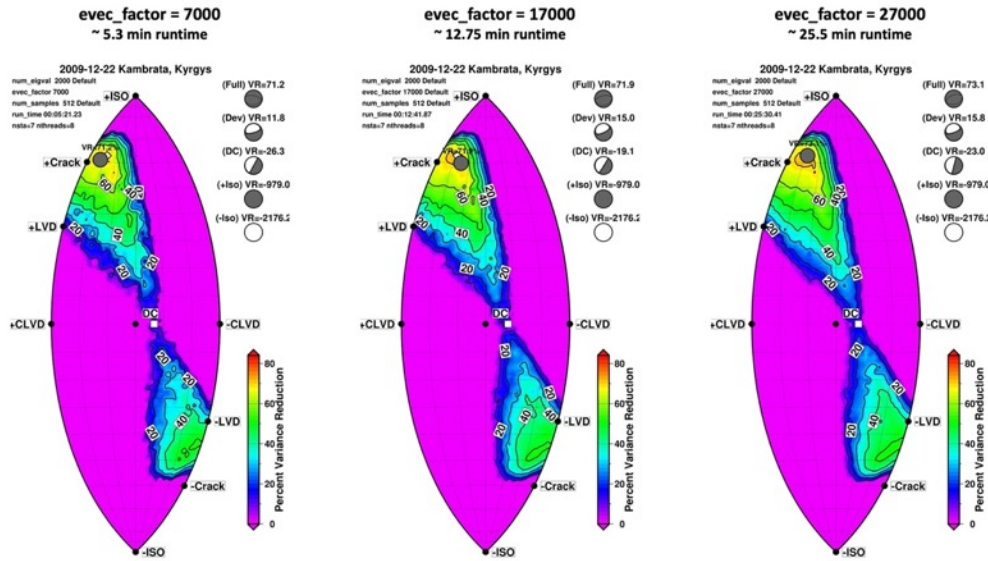


Figure 15. NSS (plots `best.fit.all.1une.jpg`) created using script `mteig.csh` (TABLE 7). The NSS plots are from `evec_factor=7000`, 17000 (default), and 27000. With `evec_factor=7000`, the contours are not smooth.

## Number of Samples – affects runtime

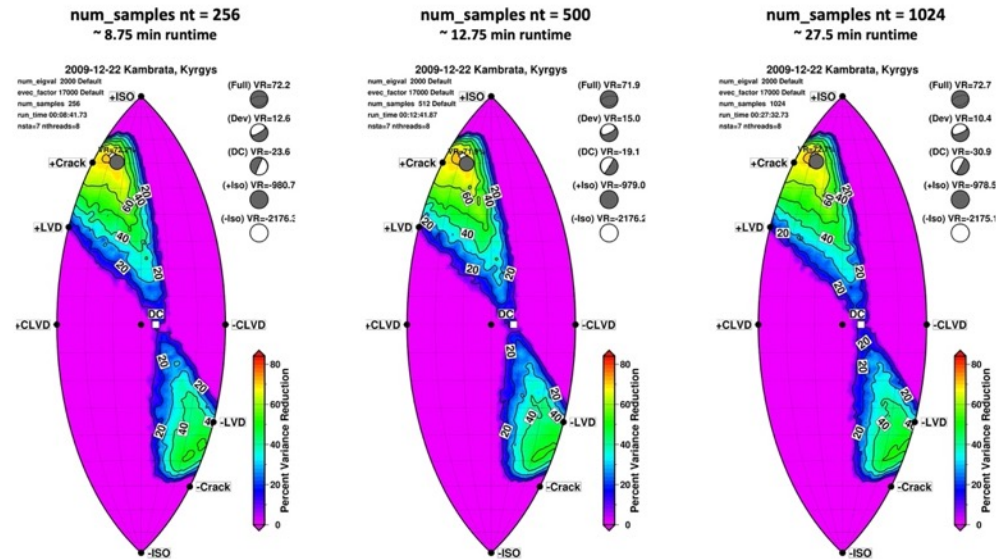


Figure 16. NSS (plots `best.fit.all.1une.jpg`) created using script `mteig.csh` (TABLE 7). The NSS plots use `nt=256`, 512 (default), 1024. There are not many differences between the three settings.

## Number of Eigenvalues – affects smoothness of contours

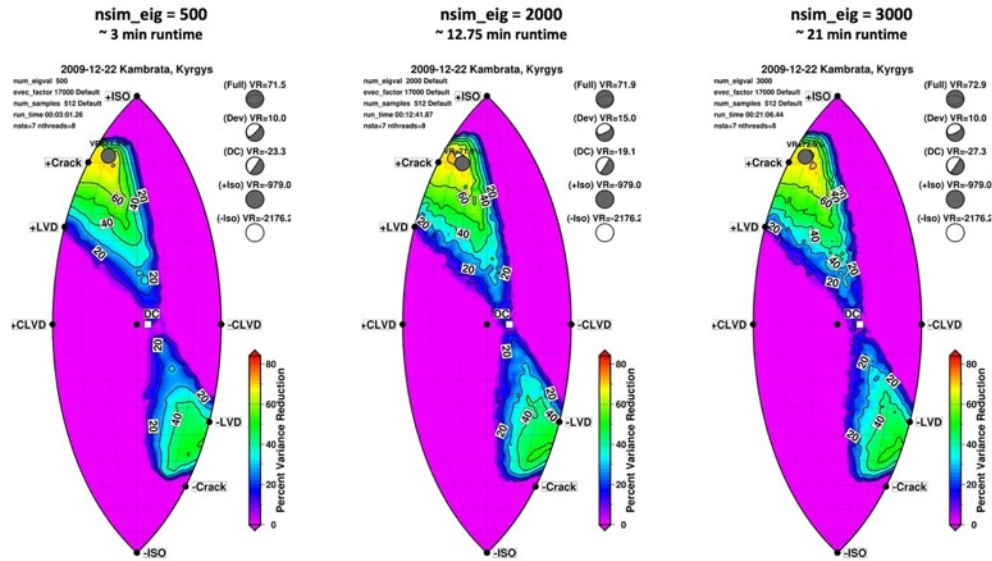


Figure 17. NSS (plots best\_fit.all lune.jpg) created using script mteig.csh (TABLE 5). The NSS plots use  $nsim\_eig = 500, 2000$  (default),  $4000$ . There are not many differences between the three settings but there is a significant speed up in computation time when using  $nsim\_eig = 500$  and  $2000$ .

## Deviatoric Moment Tensor Solution

### Franklin Mine, Crab Orchard, TN 2021-08-13 pillar collapse

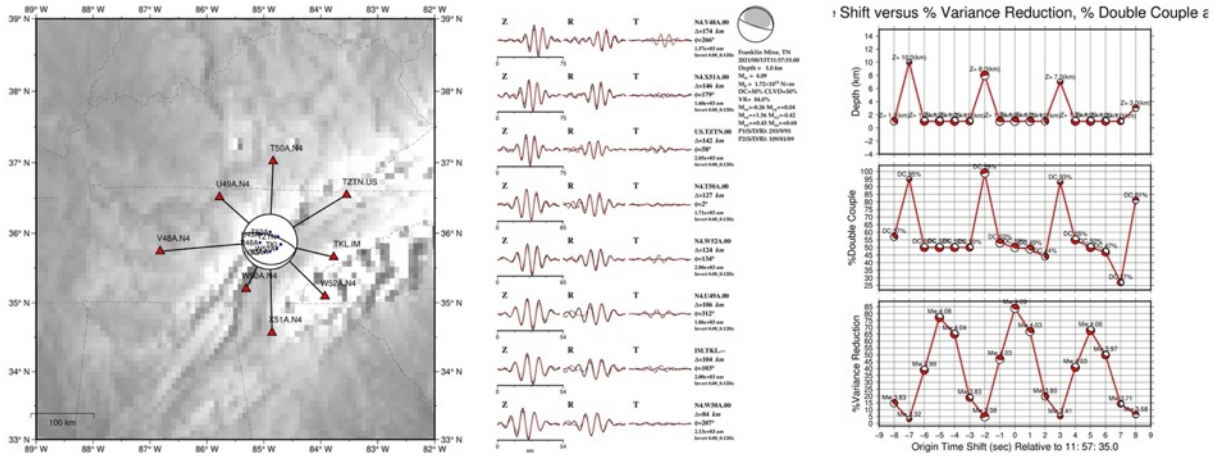


Figure 18. Deviatoric moment tensor of a collapse mechanism (plots: `gmtmap.jpg`, `gmtwf.000.jpg`, and `results.5.jpg`). The plots were made from scripts `run.csh` (TABLE 8) and `run2.csh`.

## Full Moment Tensor Solution

### Franklin Mine, Crab Orchard, TN 2021-08-13 pillar collapse

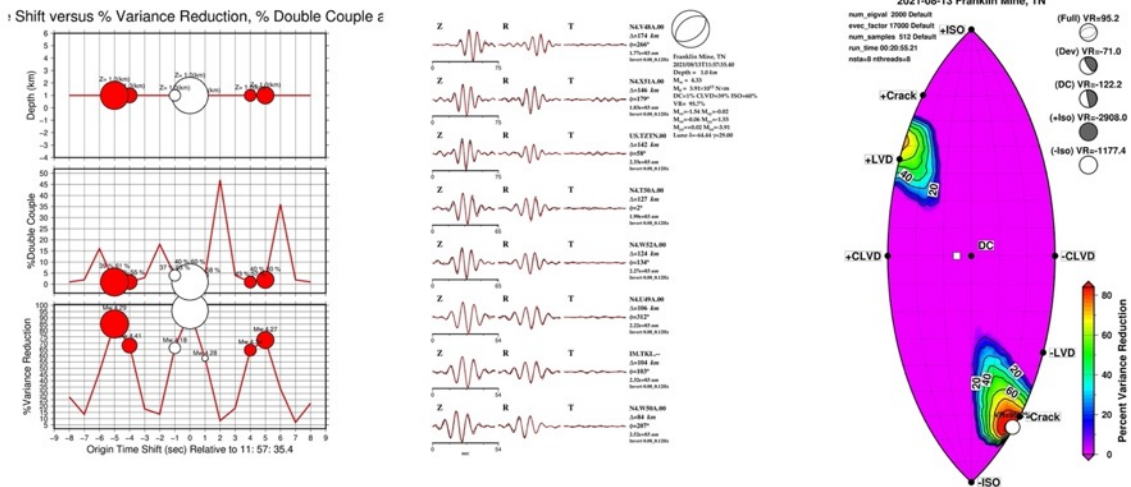


Figure 19. Full moment tensor (plots: `gmtmap.jpg`, `gmtwf.000.jpg`, and `best.fit.all.1une.jpg`) of a collapse mechanism. The plots were made from scripts `run2.csh` and `mteig.csh`. (TABLE 9).

**TABLE 8.**

./mtinv4\_sampledata/2021-08-13T115717\_Tennessee/full/run.csh

```
#!/bin/csh
#### Realtime version - set in makepar ####
set DEGREE=6 # 1-isotropic_mt 5-deviatoric_mt 6-full_mt

cat >! mtinv.par << EOF
#### REGION COMMENT #####
CM Franklin Mine, TN
#### Date and Origin Time #####
OT 2021-08-13T11:57:35.00
#### Forward Calculations #####
## stk dip rak Mw evlo evla Z #####
EV -999.0 -999.0 -999.0 0.0 -84.898 35.8767 15.0
#####
# sta net loc model np pas lf hf nt dt tr tt v/d mulfac used(Y/N) ts0 weight ## #
W50A N4 00 cus 3 2 0.075 0.120 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=84.0 Az=207
TKL IM "" cus 3 2 0.075 0.120 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=104.2 Az=103
U49A N4 00 cus 3 2 0.075 0.120 1024 0.05 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=106.0 Az=312
W52A N4 00 cus 3 2 0.075 0.120 1024 0.06 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=123.6 Az=134
T50A N4 00 cus 3 2 0.075 0.120 1024 0.06 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=127.2 Az=2
TZTN US 00 cus 3 2 0.075 0.120 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=141.8 Az=58
X51A N4 00 cus 3 2 0.075 0.120 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=145.7 Az=179
V48A N4 00 cus 3 2 0.075 0.120 1024 0.07 0.0 0.0 d 1.0 y +0.00 +1.0 Surf/Pnl ### R=174.0 Az=266
#####
#V53A N4 00 cus 3 2 0.075 0.120 1024 0.08 0.0 0.0 d 1.0 n +0.00 +1.0 Surf/Pnl ### R=189.4 Az=96
EOF

### CLEAN UP ###
/bin/rm -f *.ginv *.data plot_T???.?sec_Z???.?km_.p??ps email_T???.?sec_Z???.?km_.txt *?.dat.xy *?.syn.xy
/bin/rm -f plot_T???.?sec_Z???.?km_.p??jpg plot_T???.?sec_Z???.?km_.p??pdf *.sql
/bin/rm -f results.?.??? plotmech.??? plotz.??? gmtmap.??? mtinv.out multithread_mkgrnlib.out snr.out var_red.out
/bin/rm -f automt.txt

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel

### PROCESS DATA ###
sacdata2inv par=mtinv.par path=./Data respdir=./Resp noverbose nodumpsac parallel

# foreach ts0 ( -8 -7 -6 -5 -4 -3 -2 -1 -0.5 0 0.5 1 2 3 4 5 6 7 8 )
# foreach ts0 ( -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 )
#   mtinv AutoAuth ts0=${ts0} par=mtinv.par gmt5 mtdegree=${DEGREE} use_snr minsnr=3 shift ctol=0.85 maxshift=10 >> mtinv.out
# end ### loop over ts0

### CHECK ORIGIN TIME SHIFT ###
csh results.${DEGREE}.csh
### uncomment for review
# open results.?.jpg

### convert each postscript file to jpg
foreach ps ( plot_T???.?sec_Z???.?km_.p??ps )
  gmt psconvert -Z -Tj -E120 ${ps}
end
### uncomment
#open plot_T???.?sec_Z???.?km_.p??jpg

### MAKE DEPTH SENSITIVITY PLOT ###
# csh plotz.csh
# open plotz.jpg

### MAKE DEPTH / OT-SHIFT SENSITIVITY PLOT ###
# csh plotmech.csh
# open plotmech.jpg

# csh gmtmap.csh
# open gmtmap.jpg

mtbestfit gmt5 evid=-1 db pretty_plot noforce_best_vred mteig decimate_factor=2
```



**TABLE 9.**

./mtinv4\_sampledata/2021-08-13T115717\_Tennessee/full/mteig.csh

```
#!/bin/csh
#####
### this is an autogenerated script for mteig ###
### mteig compute %VR NSS on the lune following Ford et al., 2012 ###
#####
###
### Created by mtbestfit fitType=BEST VARIANCE-REDUCTION
### ot=-0 fsec=35 z=2 vred=90.4828 fit=90.4828 pdc=22 piso=53 pclvd=25
### fit_max=90.4828 fit_max_threshold=20 vred_diff=0 vred_diff_threshold=30
###

#set DEGREE=6 ### mteig assumes degree-of-freedom is 6 for full-MT to compute %VR on lune

cat >! mtinv.par << EOF
#### REGION COMMENT #####
CM Franklin Mine, TN
#### Date and Origin Time #####
OT 2021-08-13T11:57:35.000
#### Forward Calculations #####
## stk dip rak Mw evlo evla Z #####
EV 104 19 -93 4.23 -84.898 35.8767 2
#####
# sta net loc model np pas lf hf nt dt tr tt v/d mulfac used(Y/N) ts0 weight ## #
W50A N4 00 cus 3 2 0.075 0.120 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl ## R=84.0 Az=207
TKL IM "" cus 3 2 0.075 0.120 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl ## R=104.2 Az=103
U49A N4 00 cus 3 2 0.075 0.120 512 0.1 0 0 d 1 y +0.000 1 Surf/Pnl ## R=106.0 Az=312
W52A N4 00 cus 3 2 0.075 0.120 512 0.12 0 0 d 1 y +0.000 1 Surf/Pnl ## R=123.6 Az=134
T50A N4 00 cus 3 2 0.075 0.120 512 0.12 0 0 d 1 y +0.000 1 Surf/Pnl ## R=127.2 Az=2
TZTN US 00 cus 3 2 0.075 0.120 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl ## R=141.8 Az=58
X51A N4 00 cus 3 2 0.075 0.120 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl ## R=145.7 Az=179
V48A N4 00 cus 3 2 0.075 0.120 512 0.14 0 0 d 1 y +0.000 1 Surf/Pnl ## R=174.0 Az=266
### NOTE! stas not used, are commented out and not loaded by mteig ###
### because there is no prediction only millions of forward calcs ###
EOF

### PROCESS GREENS FUNCTIONS ###
glib2inv par=mtinv.par noverbose parallel

### PROCESS DATA ###
sacdata2inv par=mtinv.par path=./Data respdir=./Resp noverbose nodumpsac parallel

time mteig par=mtinv.par nthreads=8 \
    nsim_eig=2000 nsim_evec=4000 eigvec_fac=17000 \
    Mo=2.773636e+22 fixz=2 \
    color doplt seed=1 parallel \
    title="2021-08-13 Franklin Mine, TN" Add_user_Eig e0=-0.52422 e1=-1.12737 e2=-2.77364
```



## Number of Eigenvectors – affects smoothness of contours

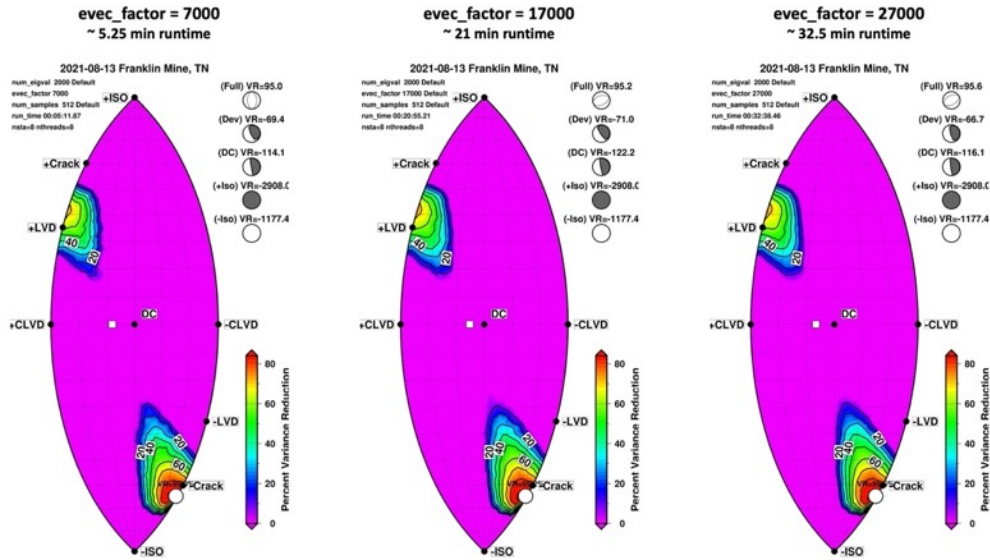


Figure 20. NSS (plots `best.fit.all.lune.jpg`) created using script `mteig.csh` (TABLE 9). The NSS plots are from `evect_factor=7000`, `17000` (default), and `27000`. With `evect_factor=7000`, the contours are not smooth.

## Number of Samples – affects runtime

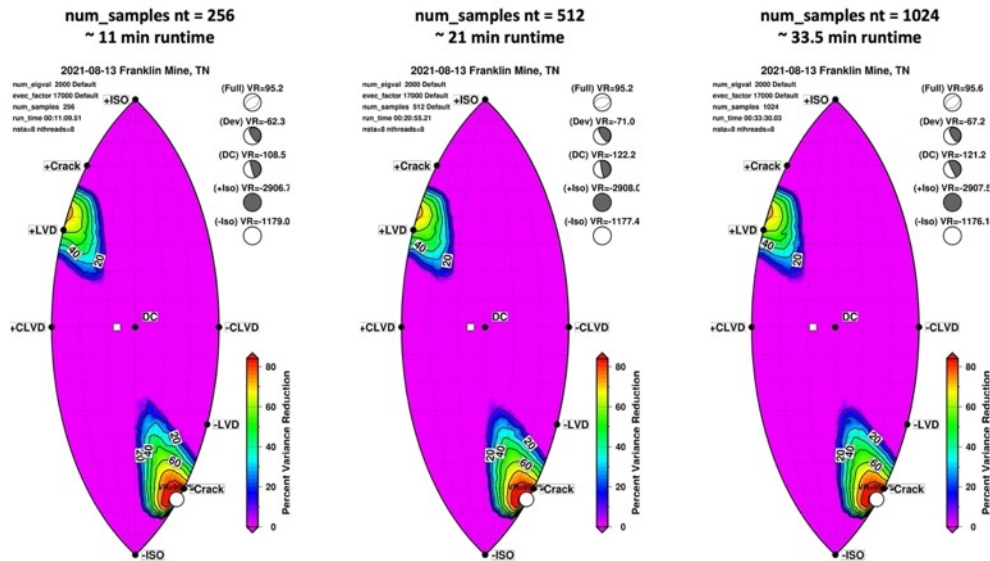


Figure 21. NSS (plots `best.fit.all.lune.jpg`) created using script `mteig.csh` (TABLE 9). The NSS plots use `nt=256`, `512` (default), `1024`. There are not many differences between the three settings.

## Number of Eigenvalues – affects smoothness of contours

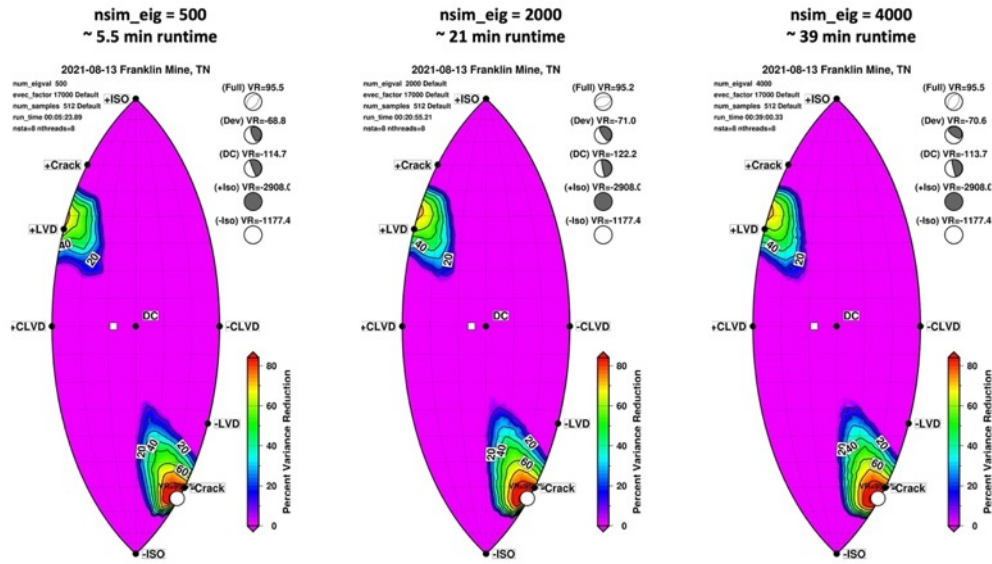


Figure 22. NSS (plots `best.fit.all.lune.jpg`) created using script `mteig.csh` (TABLE 9). The NSS plots use  $nsim\_eig = 500, 2000$  (default),  $4000$ . There are not many differences between the three settings but there is a significant speed up in computation time when using  $nsim\_eig = 500$  and  $2000$ .

### 3. INDEX OF MAN PAGES

1. **glib2inv** (processes Green's functions from library to format for MT inversion)
2. **glib2inv\_join** (joins two Green's function files to create a composite source)
3. **grn2Mxy** (converts fundamental faulting Green's functions to MT format)
4. **grnlib2sac** (creates synthetics from Green's functions from library to SAC)
5. **list\_MTdb.csh** – (listing catalog of MT solutions loaded into sqlite3 database)
6. **makepar** (creates run.csh, run2.csh scripts)
7. **mkgrnlib** (creates Green's function libraries)
8. **mtbestfit** (in realtime mode, mtinv creates shared output file, auto creates run2.csh and mteig.csh scripts)
9. **mteig** (computes the full-MT Network Sensitivity Solution %VR as function lune latitude and longitude in source-type space)
10. **mtinv** (does the MT inversion, reads glib2inv and sacdata2inv input files specified from mtinv.par)
11. **mtscreen.py** (python script, explosion screening hypersphere based on the Fisher von Misses distribution)
12. **multithread\_mkgrnlib** (computes Green's function libraries in parallel using Unix fork)
13. **pltmod** (converts Earth velocity mod files into simple GMT format for plotting)
14. **print\_MTdb.csh** (prints all MT database tables for last event or specific mtid)
15. **remove\_MTdb.csh** (drops 1 row in database table or whole catalog)
16. **renamesac** (renames SAC files)
17. **sac2gmtmap** (makes a GMT map of stations based on SAC files and creates a rdseed.station file)
18. **sac2xy** (converts SAC file to text file with time and amplitude for GMT plotting)
19. **sacdata2inv** (processes SAC files for inversion, filtering, windowing, instrument correction, etc...)
20. **sacmerge** (merges multiple SAC file segments into 1 continuous file, gaps filled zeros)
21. **setupMT** (auto generate makeglib.csh script to run mkgrnlib and make Green's functions)
22. **unpack.csh**
23. **whatshere** (utility tool to inspect all input files)