

# hybridMT

Seismic moment tensor and refinement package for  
MATLAB/shell environment

User Manual



## Contents

|   |    |
|---|----|
| Introduction .....  | 5  |
| Authors.....  | 5  |
| Package download and update .....                               | 5  |
| Licensing information .....                                     | 5  |
| Referencing .....   | 5  |
| Acknowledgements .....  | 5  |
| Quick start guide .....   | 7  |
| Installation.....   | 7  |
| Getting help/documentation .....                                | 7  |
| Moment tensor inversion .....                                   | 9  |
| Getting parametric output data.....                             | 11 |
| Quality assessment.....   | 13 |
| Handling multiple events at once.....                           | 18 |
| Package overview.....   | 22 |
| Moment tensor inversion (focimt + focimt.m).....                | 22 |
| Moment tensor refinement (hybridmt.m).....                      | 23 |
| Input focimt ASCII file formats .....                           | 25 |
| RAW ASCII input file .....                                      | 25 |
| 1D velocity model ASCII input file .....                        | 26 |
| Input focimt cell array formats .....                           | 29 |
| Structure of input cell array in RAW format .....               | 29 |
| Structure of input cell array in 1D velocity model format ..... | 30 |
| Output focimt cell array formats .....                          | 33 |
| General seismic moment tensor solution data.....                | 33 |
| Seismic moment tensor solution data.....                        | 34 |
| Handling output cell array data from many solutions .....       | 37 |
| Using focimt command line application.....                      | 41 |
| Basic usage.....  | 41 |
| Moment tensor inversion parameters .....                        | 43 |
| Outputting inversion results to ASCII file .....                | 47 |
| Auxiliary options of focimt application .....                   | 48 |
| Command line options .....                                      | 50 |
| Theoretical background.....                                     | 53 |
| Moment tensor inversion .....                                   | 53 |
| Hybrid moment tensor refinement .....                           | 54 |



## Introduction

The seismic moment tensor is becoming a standard for description of earthquake kinematic source processes in the whole range of magnitudes. Seismic moment tensor inversion allows to estimate the fault plane parameters and the balance between volumetric and non- volumetric strain in the seismic source (Knopoff and Randall, 1970). The resolved seismic moment tensors are typically decomposed into volumetric and deviatoric components using various decomposition schemes allowing to understand the detailed physical kinematic processes occurring in the sources regardless of the type of seismicity as well as regardless of its size.

We provide the community with **hybridMT** package for performing the moment tensor inversion and refinement software in MATLAB/command shell environment. The command line application `focimt` and its MATLAB wrapper `focimt.m` allows to perform different types of seismic moment tensor inversions in time domain and provide the output parameters as well as generate the graphical representation of the calculated seismic moment tensors directly in MATLAB environment. The second routine `hybridmt.m` performs the refinement of seismic moment tensors of events forming a cluster. The package was thoroughly tested with synthetic datasets, acoustic emission data from laboratory experiments as well as induced seismicity data from a geothermal sites.

## Authors

---

Grzegorz Kwiatek [kwiatek@gfz-potsdam.de](mailto:kwiatek@gfz-potsdam.de)

Patricia Martinez-Garzon [patricia@gfz-potsdam.de](mailto:patricia@gfz-potsdam.de)

## Package download and update

---

The newest version of the **hybridMT** package together with installation instructions, quick start guide, documentation and examples are available at: <http://www.induced.pl/hybridmt> and <http://www.induced.pl/focimt>.

## Licensing information

---

**hybridMT** package is licensed under GNU General Public License (GPL) and freely available to all users.

## Referencing

---

When you find **hybridMT** package useful, please reference our work: Kwiatek, G., Martinez-Garzon, P. (2016). hybridMT – MATLAB package for seismic moment tensor inversion and refinement. *Seismological Research Letters* XX, pp. XXX-XXX, DOI: XXXX.

## Acknowledgements

---

fociMT uses portions of FORTRAN code by Pawel Wiejacz (Institute of Geophysics, Polish Academy of Sciences, Warsaw, Poland) related to the seismic moment tensor inversion. focimt binaries uses CAIRO library (<http://cairographics.org/>). The source code also contains routines from PSMECA program, which is a part of Generic Mapping Tools (GMT) software package available to download from <http://gmt.soest.hawaii.edu/>. The 1D velocity model ray-tracing routines were translated from FORTRAN from hypoDD v1.3 package by Felix Waldhauser, see <http://www.ldeo.columbia.edu/~felixw/hypoDD.html> for details.

The Authors wish to thank Pawel Wiejacz, *Bogusław Domański*, Marek Burdzy, Grzegorz Makowski, Marian Król, Wojciech Dębski, Ilona Kwiatkowska, Marcin Pussak, Anita Zych-Kotwicka, Artur Kotwicki, Łukasz Rudziński, Grażyna Maziarz-Wiorek, Nikos Melis, Beata Orlecka-Sikora, Bożena Hersztowska, Katrin Plenkers, Oliver Germer, Grzegorz Lizurek, Wojciech Białoń, Maria Kozłowska, Konstantinos Leptokarpoulos, Joanna Kocot, Jerzy Giza, Krystyna Stec for help, feedback and beta-testing of the software.

## References

Knopoff, L., and M. J. Randall (1970). The compensated linear-vector dipole. A possible mechanism for deep earthquakes, J. Geophys. Res. 75, 1957–1963.

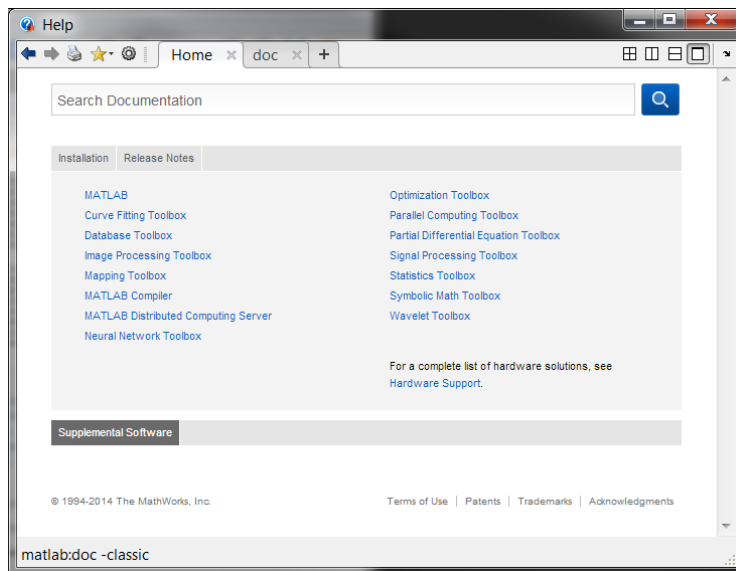
## Quick start guide

### Installation

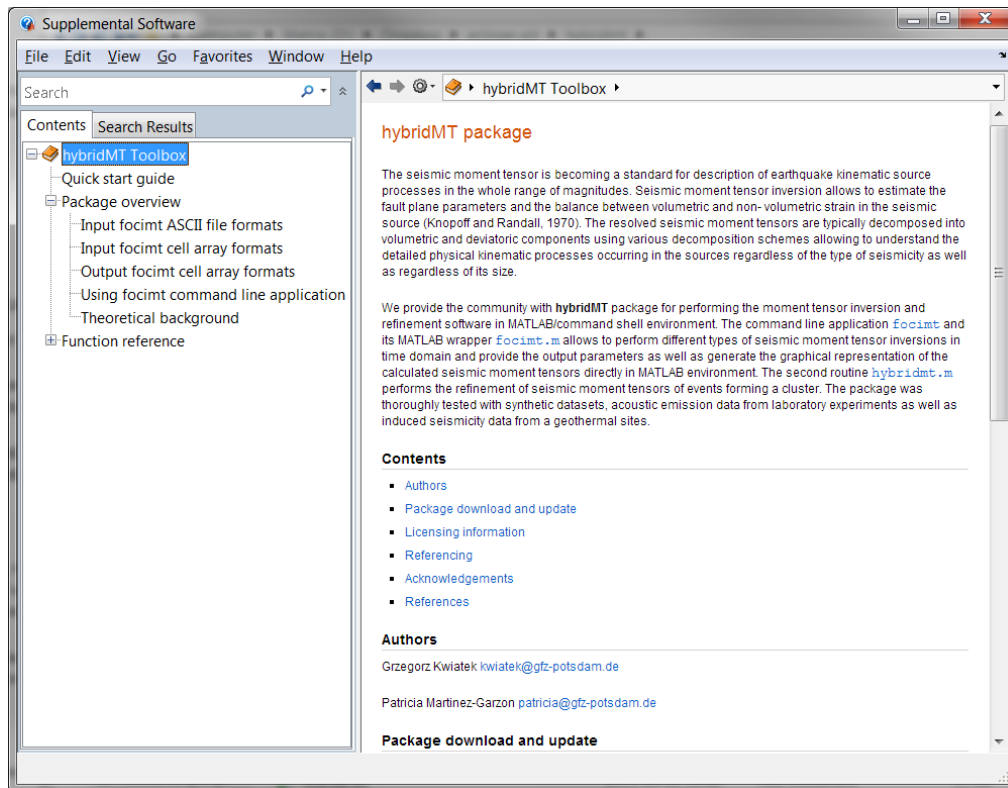
The downloaded package should be firstly unpacked to a separate directory. In the following, it is recommended to add the location of the directory with the content of **hybridMT** package to MATLAB path. One can either do it through MATLAB interface (Home>Set Path in R2014b) or by calling the script `hybridmt_install.m` located in the root folder of **hybridMT** package. The script will add the location of root folder to MATLAB search path and also update the help search database with the documentation of **hybridMT** package.

### Getting help/documentation

The package integrates with MATLAB help system and search database. Accessing **hybridMT** package help system may vary with different versions of MATLAB. In newest releases, the documentation is accessible by opening the MATLAB help system and clicking on *Supplemental Software* link located in the bottom part of the help homescreen:



resulting in opening the package help system:



Alternatively, the following command may be executed when current MATLAB folder is set to the directory where the package was installed:

```
doc -classic;
```

Older versions of MATLAB may integrate the **hybridMT** help with MATLAB documentation.

The help is also provided in a form of HTML files accompanying each function. For example, typing down:

```
help focimt;
```

```
FOCIMT Perform seismic full moment tensor inversion using fociMT software.
Use FOCIMT to perform the seismic moment tensor inversion using fociMT
software via MATLAB wrapper.

part of hybridMT package
<a href="matlab:open('html/doc_focimt.html')">Reference page for focimt</a>
```

provides the User with a short description of the routine. The User can follow the hyperlink that provides the extended documentation of the routine.



Finally, the **hybridMT** package provides a documentation in a form of PDF file located in the main directory.

## Moment tensor inversion

---

The seismic moment tensor inversion may be performed by a direct call to command line application [focimt](#) or by calling the [focimt.m](#) routine. The latter routine is in fact a MATLAB "wrapper" to [focimt](#) allowing to easily specify the inversion parameters as well as to gather the moment tensor output.

The input moment tensor inversion data must be provided in a form of ASCII file following one of two available input formats handled by the package functions described in the manual later on. Examples of ASCII input files are located in `/examples` subdirectory of **hybridMT** package installation folder.

In the following, we use sample file `example6a.txt` located in `/examples` folder that is formatted according to *1D velocity model ASCII format*. One can open the input file with any ASCII text editor:

```
71761660 19      4299668      514987      -2300      2700
AL4 Z P  -5.837934028e-009  4298878  514278  661
MCL Z P  -1.597535590e-008  4300663  515328  961
BUC Z P  -1.974913194e-010  4297161  514321  888
BRP Z P  -1.531125000e-008  4300690  517602  905
SQK Z P  -5.010868056e-009  4297196  516487  639
CLV Z P  -4.782118056e-009  4298866  518196  989
AL5 Z P   1.063269965e-008  4299029  511402  593
RGP Z P  -3.901814236e-009  4303292  516370  799
SB4 Z P  -1.578637153e-009  4295581  514832  327
HBW Z P   9.751050347e-009  4301085  510786  985
HVC Z P  -2.176428908e-009  4299334  519405  779
AL1 Z P   7.571961806e-009  4298831  510115  704
NEG Z P  -1.369678819e-009  4297989  519971  922
STY Z P   2.129661458e-009  4296077  519041 1112
AL6 Z P   1.136909722e-009  4294586  512040  749
ACR Z P  -7.998354869e-010  4298682  520805  803
DRK Z P   1.260164931e-009  4293306  517103  757
HER Z P   1.906137153e-009  4299553  507480  668
LCK Z P   2.227144097e-009  4296783  522420 1166
```

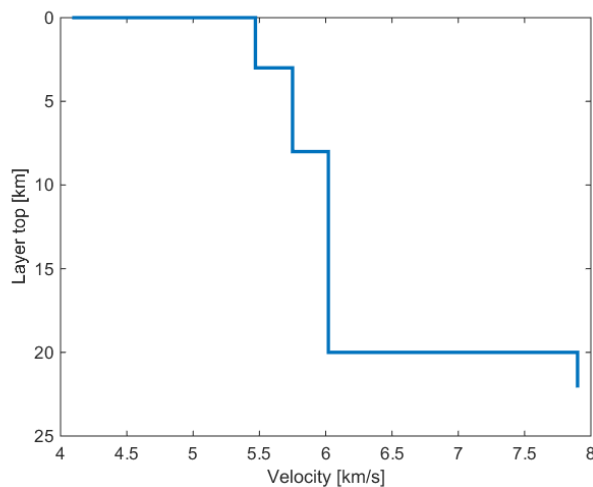
The first line of the input file is *Event Line* containing basic information about seismic event. In this particular case the 71761660 reflects event ID number followed by number of phases (26) and location of earthquake in local cartesian coordinate system (Northing = 4299447 meters, Easting = 515161 meters, Z = -2460 meters). The last value corresponds to the rock density in hypocentral area (2900kg/m<sup>3</sup>).

The *event line* is followed by limited (in this particular case 26) *phase lines* containing phase information. Each phase lines contains information on station name ('AL4 '), component and phase type ('Z' and 'P'), area below the first pulse of the phase onset (=spectral level, = proportional to seismic moment) together with sign information, and finally location of stations in Cartesian coordinate system.

The ASCII input file may contain multiple blocks *event line - phase\_lines* allowing performing the moment tensor inversion for many earthquakes at once. More details on the format of the input files is presented in section [Input focimt ASCII file formats](#) of the main manual.

This type of input file assumes the User provide the 1D velocity model to perform the 1D ray-tracing and calculate auxiliary parameters necessary for the moment tensor inversion such as the takeoff angles, incidence angles and distance along the ray path (the latter is necessary to calculate the seismic moment). We therefore prepare the velocity model first:

```
VELOCITYMODEL = [0.00 3.00 8.00 20.00 22.00; 4.10 5.47 5.75 6.02 7.90]';
stairs(VELOCITYMODEL(:,2),VELOCITYMODEL(:,1),'-','LineWidth',2);
ylabel('Layer top [km]'); xlabel('Velocity [km/s]');
set(gca,'YDir','reverse');
```



Having the input file and the 1D velocity model, the moment tensor inversion (using default inversion options) can be performed easily by executing the `focimt` routine:

```
Solution =
focimt('examples/example6a.txt','VelocityModel',VELOCITYMODEL,'BeachBallFormat','PNG');
```

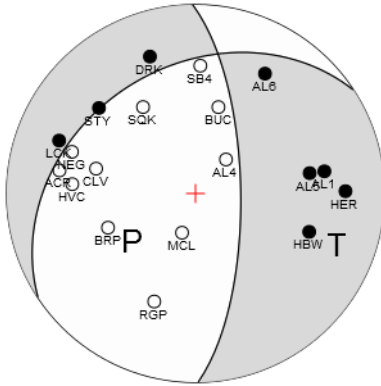
The velocity model for the moment tensor inversion with the ASCII file above was provided by auxiliary parameter *VelocityModel*. The additional parameter *BeachBallFormat* allows to create a graphical representation of the seismic moment tensor inversion in a form of beach ball plot. By default, three moment tensors are calculated:

- Unconstrained (full) moment tensor solution,
- Deviatoric moment tensor solution (no volumetric change in the source, i.e. ISO = 0, CLVD+DC), and
- Double-couple moment tensor solution (no volumetric change in the source, no linear dipole, ISO = 0, CLVD = 0, DC).

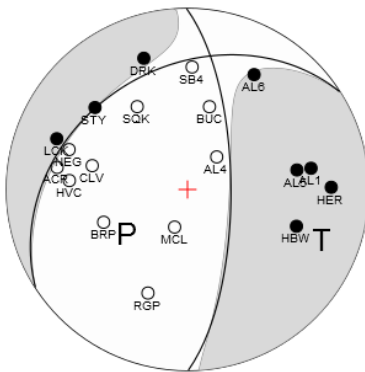
The detailed information on the inversion method used as well as the moment tensor decomposition scheme used in the package is presented in section [Theoretical background](#) of the main manual.

Setting the second auxiliary parameter *BeachBallFormat* to 'PNG' results in generation of three PNG images representing unconstrained, deviatoric and double-couple solutions. Other image formats such as PDF, SVG and PS are available by modification of *BeachBallFormat*.

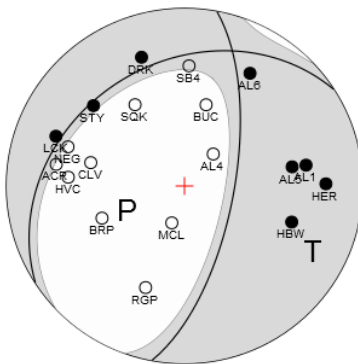
Double-couple solution (ISO = 0%, CLVD = 0%):



Deviatoric solution (ISO = 0%):



Full solution:



By default, the generated beach ball contains center cross, best double-couple nodal lines, station locations on the focal sphere, location of P and T axes and center cross. Each of these elements may be switch on or off by modification of *PlotCross*, *PlotDC*, *PlotStations* and *PlotAxes* properties. Also, the projection type (equal area/equal angle) as well as projection hemisphere (lower/upper) can be adjusted to User preferences by modification of *Projection* and *Hemisphere* properties. Finally, the size of output image(s) can be modified by *BeachBallSize* property.

## Getting parametric output data

The moment tensor solution is returned as cell array:

```
Solution
```

```
Solution =  
  
[1x1 struct]
```

Each cell array element corresponds to resulting moment tensor from a single earthquake. For example, to access the moment tensor solutions (full, deviatoric, and double-couple solutions) of the first earthquake we can use the indexing operator. Using the moment tensor solution from example above, this can be done by calling:

```
Solution{1}
```

```
ans =  
  
    event_id: '71761660'  
    n_trials: 1  
 calculation_dt: 7.3626e+05  
        full: [1x1 struct]  
 deviatoric: [1x1 struct]  
        dc: [1x1 struct]
```

Each element of the cell array is a structure array containing basic information on earthquake (`event_id`, `n_trials`, and `calculation_dt`) and three structures containing output parameters of calculated unconstrained full, deviatoric and double-couple constrained moment tensors (`full`, `deviatoric` and `dc` structures). For example, to access the output parameters of the full moment tensor solution of the first earthquake one may type:

```
Solution{1}.full
```

```
ans =  
  
    Type: {'N'}  
 STATION_ID: 0  
    ISO: 11.63  
 CLVD: 3.9735  
    DC: 84.397  
    M0: 2.253e+11  
    MT: 2.3707e+11  
 M0ERRMAX: 6.7724e+09  
    MW: 1.5385  
    P: [246.78 61.337]  
    T: [116.91 19.311]  
    B: [19.453 20.33]  
    F1: [10.517 67.255 -112.13]
```

```

F2: [236.97 31.319 -48.056]
Fault: {'NF'}
MXX: [1x6 double]
CXX: [1x6 double]
RMSError: 0.25025
Station: {1x19 cell}
UMEASURED: [1x19 double]
UTH: [1x19 double]

```

The elements of moment tensor solution structure array are generally self-explanatory (more details on meaning of all parameters is presented in section [Output focimt cell array formats](#) of the manual). For example, the orientation of the two fault planes is stored in variables F1 and F2 as [strike dip rake] triplets. Similarly, the orientation of P, T, B cardinal axes is formatted as [trend plunge]. The scalar and total seismic moment is stored in M0 and MT variables. The detailed overview of all output parameters is presented in section [Output focimt cell array formats](#) of the manual.

## Quality assessment

Each moment tensor solution structure array contains two direct parameters corresponding to the quality of the seismic moment tensor inverted:

- RMS error between observed and predicted amplitudes. The value is stored in RMSError variable.
- Maximum error of the seismic moment calculated as the square root of the maximum element of covariance matrix. The value is stored in M0ERRMAX variable.

For example, the deviatoric solution of the moment tensor solution calculated previously has RMS error and maximum error:

```
Solution{1}.deviatoric.RMSError
```

```
ans =
```

```
0.30186
```

```
and
```

```
Solution{1}.deviatoric.M0ERRMAX
```

```
ans =
```

```
6.4676e+09
```

The package allows to perform two other types of quality assessment of the seismic moment tensors:

- Station Jackknife test
- Bootstrap resampling

In *Station Jackknife Test* the seismic moment tensor inversion is firstly performed with all phases available. In the second step, the inversion is repeated with one phase removed from the input data. The second stage is repeated for every phase. The resulting moment tensor contains one solution with all phases used and  $n$  auxiliary solutions (where  $n$  is the number of phases). This procedure allows to detect potential bias introduced by a single phase (or station) to the moment tensor solution (for example bad gain or bad polarity of the station).

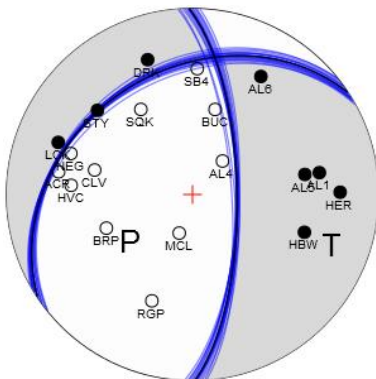
The *Station Jackknife test* may be toggle on by modification of the *Jackknife* property of `focimt` routine. In the following example we perform the station Jackknife test using the same input file as before:

```
JackknifeSol = focimt('examples/example6a.txt','VelocityModel',VELOCITYMODEL, ...
    'BeachBallFormat','PNG','Jackknife','on','ProjectDir','jtest');
```

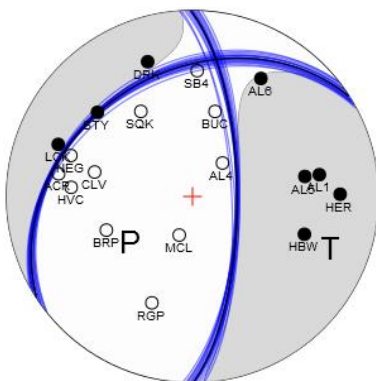
This time we stored the results in the project folder specified by *ProjectDir* parameter and we additionally turned on the *Jackknife station test*.

The generated graphical representation of the seismic moment tensors for full, deviatoric and double-couple moment tensors look as follows:

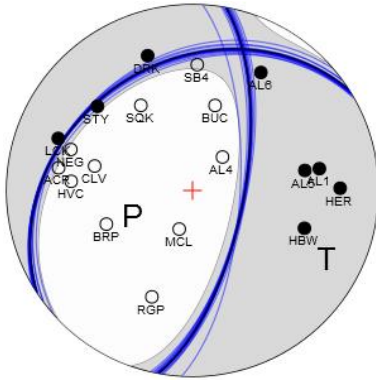
Double-couple solution (ISO = 0%, CLVD = 0%):



Deviatoric solution (ISO = 0%):



Full solution:



The tensor shading and black nodal lines corresponds to the moment tensor solution calculated with all phases (stations). The blue lines correspond to nodal lines calculated with *Jackknife* method. The color of the *Jackknife* lines depicts the sense of fault motion. Blue, green and red nodal lines correspond to the normal, strike-slip and thrust faulting. In the example above the moment tensor solution is very stable and removing of random one station in general does not affect the solution nearly at all.

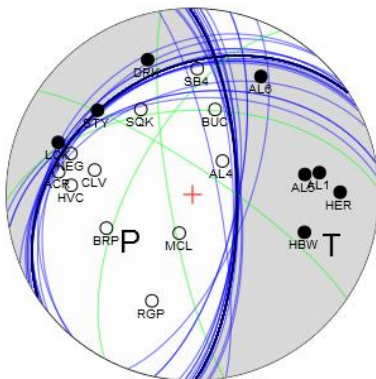
The second type of uncertainty assessment is *bootstrap resampling* of the original catalog. In bootstrap resampling, the original input phase data is used to generate  $n$  derived input phase data. For each bootstrap sample, up to  $m$  percent of randomly selected phases will have polarity reversed. The bootstrap resampling parameters is handled by *Bootstrap* option followed by a 1-by-2 elements vector where first number correspond to the number of bootstrap resamplings and second number specifies what is the percentage polarities to be reversed.

In the following example we generate and then perform the seismic moment tensor inversion of 100-elements bootstrap resample of the input dataset generated under the assumption that up to 1% of input phases may have wrongly picked polarity:

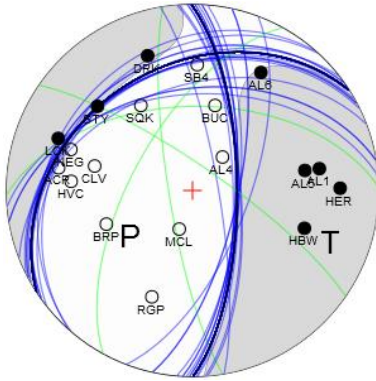
```
BootstrapSol = focimt('examples/example6a.txt', 'VelocityModel', VELOCITYMODEL, ...
    'BeachBallFormat', 'PNG', 'Bootstrap', [100 0.01], 'ProjectDir', 'btest_n01');
```

This results in the following distribution of uncertainties of the fault plane estimation:

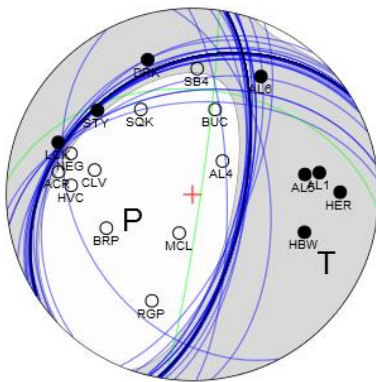
Double-couple solution (ISO = 0%, CLVD = 0%):



Deviatoric solution (ISO = 0%):



Full solution:



Both *Station Jackknife Test* and *Bootstrap resampling* generate a set of seismic moment tensor solutions. This set is obviously returned in output cell array. For example, while accessing the full moment tensor solution from bootstrap resampling method one can realize that the output parameters are now matrices with the number of rows corresponding to the total number of bootstrap resamplings plus one additional row corresponding to the original solution with original input data:

```
BootstrapSol{1}.full
```

```
ans =
```

```
Type: {101x1 cell}
STATION_ID: [101x1 int32]
ISO: [101x1 double]
CLVD: [101x1 double]
DC: [101x1 double]
M0: [101x1 double]
MT: [101x1 double]
M0ERRMAX: [101x1 double]
MW: [101x1 double]
P: [101x2 double]
T: [101x2 double]
```



```

        B: [101x2 double]
        F1: [101x3 double]
        F2: [101x3 double]
    Fault: {101x1 cell}
        MXX: [101x6 double]
        CXX: [101x6 double]
    RMSError: [101x1 double]
    Station: {1x19 cell}
    UMEASURED: [101x19 double]
    UTH: [101x19 double]

```

The type of solution can be easily recovered by inspecting the `Type` element of moment tensor solution structure:

```
[BootstrapSol{1}.full.Type{:}]
```

```
ans =
```

```

NB
BBBB

```

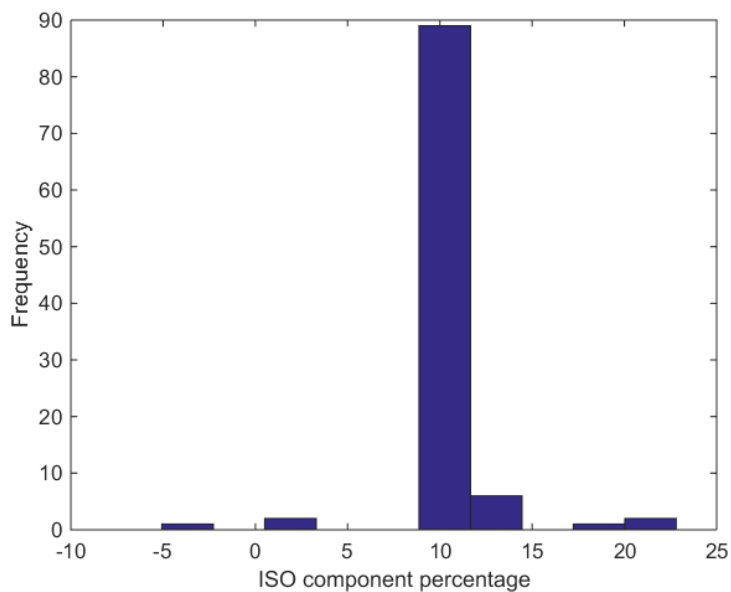
In the example above 'N' corresponds to [N]ormal (original) solution from input dataset provided and 'B' denotes bootstrap-resampled solution (J corresponds to Station Jackknife Test solution).

Having the bootstrap resampling one can investigate the uncertainties of various parameters. For example, to plot the distribution of deviatoric component of resampled dataset:

```

hist(BootstrapSol{1}.full.ISO);
xlabel('ISO component percentage');
ylabel('Frequency');

```

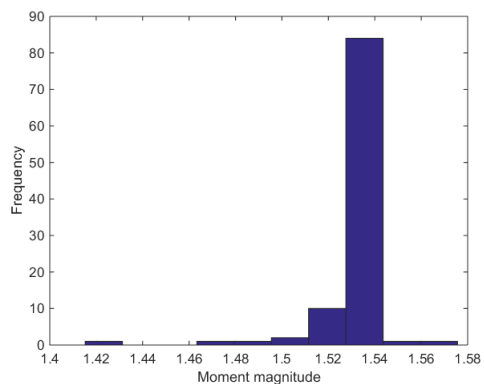


The following example plots the uncertainties of moment magnitude:

```
hist(BootstrapSol{1}.full.MW);

xlabel('Moment magnitude');

ylabel('Frequency');
```



## Handling multiple events at once

So far we discussed the moment tensor inversion of a single earthquake. However, the **hybridMT** package allows to process multiple events at once. As an example, let's perform the moment tensor inversion for 90 events using the sample data located in `/examples` directory and default processing parameters:

```
MultipleEventsSolution = focimt('./examples/example3.txt');
```

The resulting `MultipleEventsSolution` cell array contains 90 elements corresponding to 90 moment tensors calculated, one for each earthquake. Now, to access the results of a particular moment tensor inversion we have to use indexing operator. For example, to access the 5th solution, one should use the following code:

```
MultipleEventsSolution{5}.full
```

```
ans =
```

```
      Type: {'N'}
STATION_ID: 0
      ISO: -41.123
      CLVD: -31.527
      DC: 27.35
      M0: 5.6378e+14
      MT: 7.3059e+14
MOERRMAX: 2.4521e+13
      MW: 3.8041
      P: [329.94 58.774]
      T: [154.38 31.15]
      B: [63.19 1.9677]
      F1: [62.706 76.178 -92.026]
      F2: [251.13 13.967 -81.821]
      Fault: {'NF'}
      MXX: [1x6 double]
      CXX: [1x6 double]
RMSERROR: 0.33807
      Station: {1x14 cell}
UMEASURED: [1x14 double]
      UTH: [1x14 double]
```

that provide us with results of *full* moment tensor inversion for the 5th seismic event.

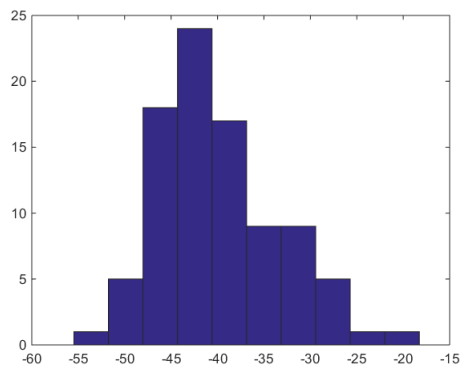
Consider a scenario when we want to extract values of *isotropic* component from *full moment tensor* for ALL earthquakes, for example in order to plot the statistics of *isotropic component* of the analyzed dataset. In other words, we would like to extract the value of `ISO` variable in full moment tensor solution structure array `full` for each element of output cell array. This is quite complicated in general, however the **hybridMT** package provides a convenient function `getsolution.m` to handle such case.

The following code will extract the amount of *isotropic* component of full moment tensors for all 90 events from the output cell array:

```
ISO = getsolution(MultipleEventsSolution, 'full', 'ISO');
```

This will result in a vector composed of 90 elements where each element corresponds to the amount of *isotropic* component from all 90 earthquakes for which the full seismic moment tensor was calculated. This can be used for plotting the statistics of various parameters for the inverted dataset, e.g.:

```
hist(ISO)
```



will display the distribution of *isotropic components* in the analyzed dataset.

In principle, more parameters can be extracted from the output cell array at once. For example, to extract the moment magnitude and orientation of P axes for the *deviatoric* moment tensor solutions of all analyzed earthquakes, the following code may be executed:

```
[MW,P] = getsolution(MultipleEventsSolution,'deviatoric','MW','P');
```

It is important to note that the order of input parameters ('MW','P' in the example above determines the order of output parameters located on the left side of the formula. Also, the names of input parameters to extract must correspond to the names of variables in the output structure array!

The function may be also used to extract general output data outside of 'full', 'deviatoric', and 'dc' structures inside of each cell array element. For example, the general data contains the ID number of events:

```
MultipleEventsSolution{1}
```

```
ans =
```

```
    event_id: '004-03781'  
    n_trials: 1  
 calculation_dt: 7.3626e+05  
        full: [1x1 struct]  
    deviatoric: [1x1 struct]  
         dc: [1x1 struct]
```

Finally, the routine works also for general output data. For example, to extract all ID numbers, the following code may be used:

```
ids = getsolution(MultipleEventsSolution, 'deviatoric', 'event_id');
```

In this case, the second parameter ( 'deviatoric' ) is of no use, but still it must be set up to either ' full ', 'deviatoric', or 'dc'.

## Package overview

**hybridMT** package is composed of two major parts: seismic moment tensor inversion application `focimt` accompanied by `focimt.m` MATLAB wrapper and `hybridmt.m` routine performing the refinement of seismic moment tensor using Hybrid Moment Tensor technique. These core functions are accompanied by various auxiliary functions, documentation and sample files.

### Moment tensor inversion (`focimt` + `focimt.m`)

---

The moment tensor inversion in **hybridMT** package may be performed either by using the command line tool `focimt`. Additionally, in MATLAB environment, the moment tensor inversion may be performed using `focimt.m` routine.

`focimt` is the stand-alone command line application performing seismic moment tensor inversion in time domain optimized for local-to-regional applications and seismic networks (Cartesian coordinate system is used). The core of the program is based on the formal description presented in Fitch et al., (1980) and its fortran implementation (Wiejacz, 1992). The precompiled binaries are available for both Windows/Linux platforms. More detailed description of algorithm used is provided in separate [section](#).

`focimt` command line tool is accompanied by MATLAB wrapper function `focimt.m` to handle the input and output parameters of `focimt` directly in the MATLAB environment. Numerous parameters can be specified to maintain input data, define the seismic moment tensor inversion and optimization parameters as well as to handle the properties of the graphical and text output by MATLAB properties.

#### Input data

The input data for the seismic moment tensor inversion, regardless of whether it is performed in a shell mode by using `focimt` application or whether the `focimt.m` function is used in MATLAB environment include information on amplitudes of the first P-wave motions and rise times, i.e. the program assumes that the area below the first P-wave ground displacement pulse is proportional to the seismic moment and this information is the main input parameters for the moment tensor inversion. In addition the User provides earthquake-station geometry data and rock medium parameters at the source. The input data is provided as ASCII file or MATLAB cell array.

Two different ASCII input data formats are available. In the first one, the geometry data (takeoff angles, angles of incidence, azimuths and distances between earthquake and sensors) is directly provided in the [ASCII input file](#). In the second input data format, earthquake hypocenter, station locations as well as 1D velocity model are used to perform the 1D ray-tracing resulting in geometrical parameters calculation.

#### Inversion

Using `focimt` or equivalently `focimt.m` is very easy. Having the input file, the moment tensor inversion may be started from directly from console window (or from MATLAB command prompt by starting the line with exclamation mark):

```
!focimt -i ./examples/example1.txt -t PNG -p SU -z 300 -o test
```

The moment tensor inversion results in generation of output ASCII file `test` containing moment tensor inversion results and graphical representation of the solution in a form of beach ball plot. The equivalent call from MATLAB command prompt using `focimt.m` routine would look as follows:

```
Solution =  
focimt('./examples/example1.txt', 'BeachBallFormat', 'PNG', 'Projection', 'schmidt', 'Hemisphere', 'upper', 'BeachBallSize', 300);
```

resulting in output parameters being stored in output cell array `Solution` than can be easily used in MATLAB environment.

### Output data

When in console mode, the output data of moment tensor inversion are exported to ASCII files formatted according to User preferences. If `focimt.m` function is used, the results are returned by the routine in a form of cell array. Apart from moment tensor solution parametric data, the application is also providing highly customizable graphical representation of the seismic moment tensor solution in a form of *beach ball plot*. The output files are either raster (PNG) or vector format (EPS, PDF, SVG).

## Moment tensor refinement (hybridmt.m)

---

Hybrid moment tensor technique was originally developed by Andersen (2001). This methodology overall aims at detecting and suppressing the influence of local path/site/sensor-related effects on estimation of seismic moment tensors resulting in their quality improvement. The methodology is especially effective in finding stations with inappropriate gain, bad site effects or high attenuation of incoming waves and provides means to account them.

The technique has been implemented in a form of MATLAB routine `hybridmt.m`. The routine calculates and refines the seismic moment tensors from a group of seismic events forming a cluster. The routine whole wraps data preparation, refinement and data output. Similarly to `focimt.m` function, numerous parameters can be specified to optimize data preparation, processing and output and adjust them to personal needs of the User. For the details on the algorithm, see the [methodology section](#). For the description of MATLAB implementation of hybrid moment tensor technique, see the documentation of `hybridmt.m` routine.

### Input data

The `hybridmt.m` routine takes essentially the same input data as `focimt.m` or `focimt` application, i.e. an [ASCII file](#) containing event/phase data in one of the two [input formats](#). However, it is now explicitly assumed all events in the input file provided by the User form a cluster with events located close by in comparison to the distances between the cluster center and sensors. This should be preserved in order to satisfy the invariance of Green's function at a particular station of all seismic events forming a cluster.

### Refinement and output

The input data is analyzed before the Hybrid Moment Tensor refinement, and the trial moment tensor inversion is performed. The statistical information on input dataset is presented to the User with information and hints regarding on how to (initially) optimize the input data. Program detects stations with insufficient number of amplitude readings, possible existence of wrong polarities etc. The User may account for problems before actual refinement starts.

In the following, the Hybrid Moment Tensor refinement is performed. The process is iterative is the User may observe the progress of refinement and trace the improvement in quality of moment tensors. The procedure finishes with a set of refined seismic moment tensors and User-defined graphical output that allows to detect potential problems with stations/site effects/coupling or attenuation (per station). The provided output may be used to repeat the moment tensor with corrected station characteristics resulting in further improvement of seismic moment tensors.

## **References**

Andersen, L. M. (2001). A relative moment tensor inversion technique applied to seismicity induced by mining, Univ. of the Witwatersrand, Johannesburg.

Fitch, T. J., D. W. McCowan, and M. W. Shields (1980). Estimation of seismic moment tensor from teleseismic body wave data with application to intraplate and mantle earthquakes, J. Geophys. Res. 85, 3817–3828.

Wiejacz, P. (1992). Badanie mechanizmów wstrząsów górniczych przy wykorzystaniu tensora momentu sejsmicznego, Institute of Geophysics, Polish Academy of Sciences.



## Input focimt ASCII file formats

focimt application, [focimt.m](#) as well as [hybridmt.m](#) routines are capable to read two types of ASCII input files:

- **RAW input file** contains all input data necessary to perform the seismic moment tensor inversion in time domain.
- **1D velocity model input file** uses 1D velocity model and station/event locations provided to perform the 1D ray-tracing and to calculate the auxiliary data for the seismic moment tensor inversion (takeoff and incidence angles as well as distances along the ray paths).

Both input file formats starts with a *header line* containing information about event that is followed by *phase lines* containing phase data. User may specify multiple header-phase blocks to process or refine multiple events at once. The ASCII input file formats are described in details in the following sections. The input files can be read by auxiliary functions [readraw.m](#) and [readvel1d.m](#).

### RAW ASCII input file

Raw input file is composed of a *header line* containing basic information about the seismic event. This line is followed by a limited number of *phase lines* containing phase information necessary for the seismic moment tensor inversion. The input text file may contain multiple blocks header-phases if the user wants to process more than one earthquake at once. The latter case is actually required once User aims to refine seismic moment tensors by means of hybrid moment tensor technique and use the [hybridmt.m](#) routine. ASCII input file in RAW format can be read using auxiliary [readraw.m](#) routine and converted to [moment tensor inversion input cell array](#) used across the whole package, if necessary.

The RAW ASCII input file generally looks as follows:

```
fileid1 n
station_1 component_1 phase_1 omega_1 azm_1 aoi_1 tko_1 v_1 r_1 rho_1
station_2 component_2 phase_2 omega_2 azm_2 aoi_2 tko_2 v_2 r_2 rho_2
...
station_n component_n phase_n omega_n azm_n aoi_n tko_n v_n r_n rho_n
fileid2 n
station_1 component_1 phase_1 omega_1 azm_1 aoi_1 tko_1 v_1 r_1 rho_1
station_2 component_2 phase_2 omega_2 azm_2 aoi_2 tko_2 v_2 r_2 rho_2
...
station_n component_n phase_n omega_n azm_n aoi_n tko_n v_n r_n rho_n
...
```

The header line contains only two parameters:

|        |         |  |
|--------|---------|--|
| fileid | string  | Event id string. The parameter contains a *UNIQUE* alphanumeric string to identify the seismic event. As the string may be used (depending on parameters passed to [hybridmt] or [focimt] programs) to create a directory where event-specific results will be stored, it is necessary to use only characters that can be used to create files/directories under specific file system. |
| n      | integer | Number of following lines containing phase information data.   |

The header line is followed by *n* lines containing phase information data:

|         |        |  |
|---------|--------|--|
| station | string | Station name. As the station is used in graphical representation of seismic moment tensor solution, short station names are recommended. |
|---------|--------|--|

|           |        |   |
|-----------|--------|---|
| component | string | Station component name. Currently only vertical components are supported, and the column is basically ignored by either <code>[fociMT]</code> or <code>[hybridMT]</code> routines. However, future version of <code>[fociMT]</code> and <code>[hybridMT]</code> will use different phase. |
| phase     | string | Phase identifier. Currently only P phases are supported, therefore only 'P' value is accepted by application. The remaining values are ignored.   |
| omega     | float  | Area below the first P-wave ground velocity pulse together with sign information. This value is proportional to the spectral level in Fourier domain, and, overall, proportional to the seismic moment.   |
| azm       | float  | Azimuth of the vector pointing from earthquake to the station. The value is measured in degrees, positive from north towards east.  |
| aoi       | float  | Angle of incidence to the sensor measured in degrees from down direction (towards center of Earth). The seismic ray pointing directly up, perpendicular towards the surface has 0 degrees incidence angle.  |
| tko       | float  | Takeoff angle measured in degrees from down (towards center of Earth) direction. The seismic ray leaving the source purely up (towards earth surface) has takeoff angle of 180 degrees. The ray leaving earthquake source towards center of Earth has a takeoff angle of 0 degrees.       |
| vel       | float  | Phase velocity in the source area in [m/s]. This value is used to calculate seismic moment from omega parameter.  |
| r         | float  | Distance along the ray from the source to the station [m]. This value is used to calculate seismic moment from omega parameter.   |
| rho       | float  | Rock density in the source area. This value is used to calculate seismic moment from omega parameter.   |

Here is example of input in RAW ASCII format:

```
testmech 20
W1 Z P 8.1579314473706e-007 20.0 160.0 20.0 5000 3000.00 2900
W2 Z P -6.8673365691379e-007 80.0 160.0 20.0 5000 3000.00 2900
W3 Z P -8.7312547285917e-007 140.0 160.0 20.0 5000 3000.00 2900
W4 Z P -2.2167789675607e-007 200.0 160.0 20.0 5000 3000.00 2900
5 Z P 1.1069303997884e-007 260.0 160.0 20.0 5000 3000.00 2900
6 Z P 9.617722655266e-007 320.0 160.0 20.0 5000 3000.00 2900
7 Z P 5.816403152331e-007 20.0 30.0 150.0 5000 3000.00 2900
8 Z P 5.9484552418295e-007 80.0 30.0 150.0 5000 3000.00 2900
A9 Z P -7.0384473729451e-007 140.0 30.0 150.0 5000 3000.00 2900
C10 Z P -7.0656157910976e-007 200.0 30.0 150.0 5000 3000.00 2900
C11 Z P 1.5849903479442e-006 260.0 30.0 150.0 5000 3000.00 2900
C12 Z P 1.5745019808096e-006 320.0 30.0 150.0 5000 3000.00 2900
C13 Z P 1.9286895552236e-007 20.0 130.0 50.0 5000 3000.00 2900
14 Z P -2.7763323964771e-006 80.0 130.0 50.0 5000 3000.00 2900
D17 Z P -1.9406208284731e-006 260.0 130.0 50.0 5000 3000.00 2900
D18 Z P -2.565372376189e-007 320.0 130.0 50.0 5000 3000.00 2900
21 Z P 5.1376195957182e-007 140.0 80.0 100.0 5000 3000.00 2900
22 Z P 1.6575456762096e-007 200.0 80.0 100.0 5000 3000.00 2900
23 Z P 1.1523550907002e-006 260.0 80.0 100.0 5000 3000.00 2900
24 Z P 6.9418020992634e-007 320.0 80.0 100.0 5000 3000.00 2900
```

The examples of reading RAW input files for the purpose of the seismic moment tensor inversion with `focimt` application are shown in [Using focimt command line application](#) section.

## 1D velocity model ASCII input file

Similarly to RAW input file format, the 1D velocity model input file also starts with header line describing basic information on seismic event and it is followed by a limited number of lines describing phase information. The main difference between raw input format and

1D velocity model format is that in the latter case User do not need to specify all the geometrical parameters for the moment tensor parameters such as azimuth, takeoff angles, distances along the seismic rays or angles of incidence. Instead, these parameters are calculated using 1D velocity model provided. ASCII file in 1D velocity model format can be read using auxillary routine `readvel1d.m`.

The 1D velocity model ASCII input file syntax looks as follows:

```
fileid N e_northing e_easting e_z density
station component phase omega s_easting s_northing s_z
station component phase omega s_easting s_northing s_z
...
station component phase omega s_easting s_northing s_z
fileid N e_northing e_easting e_z density
station component phase omega s_easting s_northing s_z
station component phase omega s_easting s_northing s_z
...
station component phase omega s_easting s_northing s_z
...
```

The header line contains following parameters:

|            |         |  |
|------------|---------|--|
| fileid     | string  | Event id string. The parameter contains "UNIQUE" alphanumeric string to identify the seismic event. As the string may be used (depending on parameters passed to  hybridmt  or  focimt  programs) to create a directory where event-specific results will be stored, it is necessary to use only characters that can be used to create files/directories under specific file system. |
| N          | integer | Number of following lines containing phase information data.   |
| e_northing | float   | Northing component of earthquake hypocenter location in local cartesian coordinate system [m]. The value must increase towards north.  |
| e_easting  | float   | Easting component of earthquake hypocenter location in local cartesian coordinate system [m]. The value must increase towards east   |
| e_depth    | float   | Depth component of earthquake hypocenter location in local cartesian coordinate system [m]. The value must increase towards up (i.e. if mean sea level is 0 in local cartesian coordinate system chosen, the negative values correspond to the hypocenter below the mean sea level)  |
| density    | float   | Density in the seismic source area [kg/m3].  |

The header line is followed by  $n$  lines containing phase information data:

|            |        |   |
|------------|--------|---|
| station    | string | Station name. As the station is used in graphical representation of seismic moment tensor solution, short station names are advisable.  |
| component  | string | Station component name. Currently only vertical components are supported, and the column is basically ignored by either  fociMT  or  hybridmt  routines. However, future version of  fociMT  and  hybridMT  will use different phase. |
| phase      | string | Phase identifier. Currently only P phases are supported, therefore only 'P' value is accepted by application. The remaining values are ignored.   |
| omega      | float  | Area below the first P-wave ground velocity pulse together with sign information. This value is proportional to the spectral level in Fourier domain, and, overall, proportional to the seismic moment.                               |
| s_northing | float  | Northing component of station coordinate in local cartesian coordinate system [m]. The station coordinate must be in the same system as earthquake northing coordinate provided in first line   |
| s_easting  | float  | Easting component of station coordinate in local cartesian coordinate system [m]. The station coordinate must be in the same system as earthquake northing coordinate provided in first line  |

|         |       |  |
|---------|-------|--|
| s_depth | float | Depth component of station coordinate in local cartesian coordinate system [m]. The station coordinate must be in the same system as earthquake northing coordinate provided in first line |
|---------|-------|--|

Here is example of input file for [hybridmt.m](#), [focimt.m](#) as well as for [focimt application](#) in 1D velocity model format:

```

134_044_+163 25 0 0 -1500 2700
S01 Z P -3.4116906326339e-007 -10000 -10000 0
S02 Z P -3.7814556483238e-007 -10000 -5000 0
S03 Z P -5.0939122438081e-007 -10000 0 0
S04 Z P -2.7462338649334e-007 -10000 5000 0
S05 Z P 3.2166275730773e-007 -10000 10000 0
S06 Z P -1.2841376369079e-007 -5000 -10000 0
S07 Z P -6.7801192106985e-007 -5000 -5000 0
S08 Z P -2.0756006966737e-006 -5000 0 0
S09 Z P -4.2127683545031e-007 -5000 5000 0
S10 Z P -4.1841523522169e-008 -5000 10000 0
S11 Z P 3.905562102742e-008 0 -10000 0
S12 Z P 2.0061444414936e-007 0 -5000 0
S13 Z P -5.7225282386188e-006 0 0 0
S14 Z P -1.6899600936722e-007 0 5000 0
S15 Z P -1.3448018845833e-008 0 10000 0
S16 Z P 3.8527723071411e-008 5000 -10000 0
S17 Z P -4.5116743747675e-008 5000 -5000 0
S18 Z P -1.0622893299273e-006 5000 0 0
S19 Z P -5.0292277936872e-007 5000 5000 0
S20 Z P -1.1584426977891e-007 5000 10000 0
S21 Z P -1.0162790980247e-006 10000 -10000 0
S22 Z P -1.6473470781745e-007 10000 -5000 0
S23 Z P -3.6544904262446e-007 10000 0 0
S24 Z P -3.0215676249735e-007 10000 5000 0
S25 Z P -9.6393340287165e-007 10000 10000 0

```

The examples of reading 1D velocity model input files for the purpose of the seismic moment tensor inversion with `focimt` application are shown in section [Using focimt command line application](#).

## Input focimt cell array formats

The input cell array format is used across the whole package to provide all input data necessary for the seismic moment tensor inversion or seismic moment tensor refinement procedure. The input cell array is typically generated internally by `focimt.m` by calling either `readraw.m` or `readvel1d.m`. These routines read input ASCII file (see `/examples` directory) in **RAW** or **1D velocity model formats**, determine its format and create the input cell array. The cell array is then used in the seismic moment tensor inversion or moment tensor refinement.

### Structure of input cell array in RAW format

---

Input cell array contains a limited number of elements which are structure arrays. Each element of the cell array correspond to a single seismic event and describes the input data necessary for the seismic moment tensor inversion.

To see the structure of the input cell array in RAW format we can try to read sample input ASCII file in RAW format:

```
Input = readraw('./examples/example2_raw.txt');
```

The only input parameter of `readraw.m` routine corresponds to the path to ASCII input file in RAW format. Typing down:

```
Input
```

```
Input =  
  
    [1x1 struct]    [1x1 struct]
```

we see that the input cell array contains two elements. Each element of the `Input` cell array correspond to complete input data from a single earthquake. To see the input data for the first earthquake we may wish to type the following:

```
Input{1}
```

```
ans =  
  
    event_id: '344_014_+165'  
    n_phases: 25  
    format: 'raw'  
matrixmode: 1  
    Station: {25x1 cell}  
    Component: {25x1 cell}  
    Phase: {25x1 cell}  
    OMEGA: [25x1 double]  
    AZIMUTH: [25x1 double]
```

```
AOI: [25x1 double]
TAKEOFF: [25x1 double]
V: [25x1 double]
R: [25x1 double]
DENSITY: [25x1 double]
```

The meaning of the fields in the structure array describing a single seismic event is as follows:

**event\_id**(string)

Event id string. This parameter contains a **UNIQUE** alphanumeric string to identify the seismic event. As the string may be used (depending on parameters passed to [hybridmt.m](#) or [focimt.m](#) routines) to create a directory where event-specific results will be stored, it is necessary to only use characters that can be used to create files/directories under specific file system.

**n\_phases**(integer)

Number of phases (phase lines) following the header line.

**format**(string 'raw' | 'velld')

String containing information about the input file format.

**matrixmode**(integer)

[\_variable used internally by the package\_] Determines whether the phase data in the input cell array is stored either as matrices and cell arrays of strings or single structure array containing multiple fields.

**e\_northing, e\_easting, e\_z**(floating point)

Coordinates of seismic event in the local Cartesian coordinate system used. The northing coordinate is positive towards north, the easting coordinate is positive towards east and finally the depth coordinate is positive towards up (away from Earth center). The values must be provided in meters.

**density**

Rock density in [kg/m<sup>3</sup>] in the source area. The value is used for calculating the seismic moment from recorded ground displacement amplitudes.

The following fields contains input data read and they are described in [fociMT input file formats](#) page.

## Structure of input cell array in 1D velocity model format

---

The structure of `focimt` input cell array in [1D velocity model format](#) is very similar to the RAW format. As an example, calling:

```
Input = readvelld('./examples/example2_1d.txt');
```

Resulting in loading of ASCII file into Input cell array:

```
Input
```

```
Input =
```

```
    [1x1 struct]    [1x1 struct]
```

Similarly to [readraw.m](#) routine, the input data for a single earthquake are accessible by indexing operator:

```
Input{1}
```

```
ans =
```

```
    event_id: '344_014_+165'  
      format: 'velld'  
    n_phases: 25  
  e_northing: 0  
    e_easting: 0  
         e_z: -1500  
    density: 2700  
matrixmode: 1  
    Station: {25x1 cell}  
  Component: {25x1 cell}  
      Phase: {25x1 cell}  
      OMEGA: [25x1 double]  
 S_NORTHING: [25x1 double]  
 S_EASTING:  [25x1 double]  
       S_Z:  [25x1 double]
```

The meaning of the fields in the structure array is as follows:

**event\_id** (character array)

Event id string. This parameter contains a **UNIQUE** alphanumeric string to identify the seismic event. As the string may be used (depending on parameters passed to `hybridmt` or `focimt` programs) to create a directory where event-specific results will be stored, it is necessary to only use characters that can be used to create files/directories under specific file system.

**n\_phases** (integer)

Number of phases (phase lines) following.

**format** ('raw' | 'velld')

String containing information about the input file format.

**matrixmode** (integer)

Determines whether phase data in the input cell array is stored either as matrices and cell arrays of strings or single structure array containing multiple fields.

The following fields contains input data read and they are described in [fociMT input file formats](#) page.



## Output focimt cell array formats

The results of seismic moment tensor inversion performed using `focimt.m` routine are by default returned in a cell array. The cell array is composed of  $n$  elements, each containing seismic moment tensor solution from a single earthquake. The order of seismic moment tensor solutions in the output cell array corresponds to the order of input cell array data, or accordingly to the order of event/phase data in the input ASCII files.

The package provides `getsolution.m` routine that can efficiently extract information from moment tensor inversion output cell array. See *Examples* section for details.

Consider the output of the seismic moment tensor inversion using `focimt`, returned in the following example:

```
[Output, Input] = focimt('./examples/example2_raw.txt');  
Output
```

```
Output =  
  
    [1x1 struct]    [1x1 struct]
```

This particular dataset contained event and phase data from two earthquakes. To access the seismic moment tensor solution from a particular earthquake, one have to specify explicitly the element of `Output` cell array:

```
Output{1}
```

```
ans =  
  
    event_id: '344_014_+165'  
    n_trials: 1  
 calculation_dt: 7.3626e+05  
         full: [1x1 struct]  
    deviatoric: [1x1 struct]  
         dc: [1x1 struct]
```

Each cell array element contains basic information on earthquake as well as three MATLAB structure arrays `full`, `deviatoric` and `dc` containing seismic moment tensor inversion solutions for unconstrained, deviatoric and double-couple seismic moment tensors.

### General seismic moment tensor solution data

---

The basic data holds the general information on processed earthquake:

## event\_id

Contains the event ID number specified in the input file or input cell array.

## n\_trials

This field contains the number of calculations of the seismic moment tensor. Unless *Jackknife test* or *Bootstrap resampling* is used while running the seismic moment tensor inversion using `focimt.m` routine, the number of solutions is always 1, i.e.

`n_trials=1`. If *Jackknife* parameter is 'on', three structures `full`, `deviatoric` and `dc` will contain various fields containing seismic moment tensor output parameters in a form of cell arrays (or matrices) with  $m+1$  elements (or rows) where  $m$  is the number of stations. First elements (or rows) of fields will correspond to the regular moment tensor solution using all stations available and the following elements (or rows) will correspond to the *Jackknife* solutions, every other with different station removed.

Similarly, if *Bootstrap resampling* procedure is enabled, the output cell array will contain various field where each field will have  $m+1$  elements. Again, the first element corresponds to the *regular* seismic moment tensor inversion result, where all stations were used and the following elements corresponds to bootstrap-resampled data.

## calculation\_dt

Contains time of calculation in MATLAB datetime format.

## Seismic moment tensor solution data

---

The last three structure arrays, **full**, **deviatoric** and **dc** contain the actual seismic moment tensor inversion results for unconstrained full moment tensor (structure `full`), constrained deviatoric or equivalently trace-null moment tensor (structure `deviatoric`) and constrained double-couple moment tensor (structure `dc`). The format of all structures is the same and can be easily visualized by accessing the particular element of the structure array. For example, to access the full (unconstrained) moment tensor inversion results:

```
Output{1}.full
```

```
ans =
```

```
      Type: {'N'}
STATION_ID: 0
      ISO: 27.13
    CLVD: 10.836
       DC: 62.034
      M0: 6.2537e+13
      MT: 6.91e+13
MOERRMAX: 1.1698e+10
      MW: 3.1674
       P: [188.83 34.198]
       T: [341.17 52.505]
       B: [89.43 13.513]
```

```

F1: [87.146 80.585 76.298]
F2: [323.28 16.574 145.01]
Fault: {'TF'}
MXX: [1x6 double]
CXX: [1x6 double]
RMSError: 0.0013436
Station: {1x25 cell}
UMEASURED: [1x25 double]
UTH: [1x25 double]

```

The meaning of the fields is as follows:

**Type** (string or cell array of strings)

Contains information on the type of seismic moment tensor inversion, only meaningful while *Jackknife test* or *Bootstrap resampling* is performed. 'N' character denotes the regular solution calculated with all stations available. The 'J' character denotes seismic moment tensor solution calculated with Jackknife approach. In this case the information on which station was removed is specified in STATION\_ID variable. Finally, the 'B' character denotes the bootstrap-resampled dataset.

**ISO** (scalar or vector)

Contains the amount of isotropic component in the calculated seismic moment tensors in percents.

**CLVD** (scalar or vector)

Contains amount of CLVD (compensated linear vector dipole) component in the calculated seismic moment tensor. The value is specified in percents.

**DC** (scalar or vector)

Contains amount of double-couple components in the calculate seismic moment tensor. The value is provided in percents.

**M0** (scalar or vector)

Contains the scalar seismic moment in Newton-Meters [Nm].

**MT** (scalar or vector)

Contains the information on total seismic moment in Newton-Meters [Nm]. The value is calculated according to the following formula:

$$M_T = \sqrt{\sum (m_{ii}^2)}$$

where  $m_{ii}$  are eigenvalues of the full seismic moment tensor. The total seismic moment is equal to the seismic moment  $M_0$  for double-couple constrained moment tensor solution.

**MOERRMAX** (scalar or vector)

Hold information on maximum error of the seismic moment tensor. The value is extracted from covariance matrix:

$$M_0^{errmax} = \max\{\sqrt{C_{ii}}\}$$

The maximum error is therefore the square root of the maximum diagonal elements of the covariance matrix.

**MW** (scalar or vector)

Moment magnitude calculated from the total seismic moment using Hanks and Kanamori (1979) formula:

$$M_W = 0.66 \log_{10} M_0 - 6.03$$

**P, T, B** (matrix)

These three matrices contain information on trend and plunge of P, T, and B axes. The trend is provided in degrees from north towards east. The plunge is presented in degrees from horizontal towards bottom (plunge equal to 0 corresponds to horizontal (parallel to earth surface) directio while plunge equal to 90 corresponds to pure vertical directio (towards center of Earth).

**F1, F2** (matrix)

These two matrices contain fault parameters. First, second and third element (or column when either *Jackknife test* or *Bootstrap resampling* is turned on) corresponds to strike, dip and rake. All values are provided in degrees and follows the seismological convention. The strike (first column) is measured from north to east. The dip (second column) is measured from horizontal to vertical and rake (third column) is positive/negative for thrust/normal faulting.

**Fault** (string or cell array of strings)

Provides information on the *fault type*. This value is calculated by comparing the orientation of *P*, *T* and *B* axes with respect to global coordinate system (i.e. which one is the most "vertical" axis, plungin with the highest angle). 'NF', 'SS', 'TF' denotes *normal fault*, *strike slipe fault* and *thrust fault*, respectively.

**MXX** (matrix)

Contains information on moment tensor components in AKI's convention. The following elements (or columents, if jackknife mode is used) correspond to  $M_{11}$ ,  $M_{12}$ ,  $M_{13}$ ,  $M_{22}$ ,  $M_{23}$ , and  $M_{33}$ . The values are specified in [Nm].

**CXX** (matrix)

Contains diagonal elements of the covariance matrix with following elements (or columns if jackknife method is used) corresponding to  $C_{11}$ ,  $C_{22}$ ,  $C_{33}$ ,  $C_{44}$ ,  $C_{55}$ , and  $C_{66}$ . The covariance matrix is calculated only if *L2* norm is used.

**RMSERROR** (scalar or vector)

Contain RMS error of the seismic moment tensor solution. The RMS error is calculated following the formula:

$$RMS = \sqrt{\frac{\sum_{i=1}^N (U_i^{measured} - U_i^{th})^2}{\sum_{i=1}^N (U_i^{measured})^2}}$$

The RMS error is calculated from measured and predicted seismic moments on all  $N$  stations (the measured and predicted seismic moments are provided in UMEASURED and UTH matrices).

**Station** (cell array of strings)

Contains the list of stations used in seismic moment tensor inversion. The order of stations in the cell array reflects the order of data in UMEASURED and UTH matrices.

**UMEASURED, UTH** (matrix)

These two matrices contain measured and theoretical moments recorded or predicted, respectively, on a particular station specified by Station cell array. The measured amplitudes correspond to the input data.

## Handling output cell array data from many solutions

---

As an example, let's perform the moment tensor inversion for 90 events using the sample data located in /examples directory:

```
Solution = focimt('./examples/example3.txt');
```

The resulting Solution cell array contains 90 elements corresponding to 90 moment tensors calculated, one for each earthquake. Now, to access the results of a particular moment tensor inversion we have to use indexing operator, e.g.:

```
Solution{5}.full
```

```
ans =
```

```
    Type: {'N'}  
STATION_ID: 0  
    ISO: -41.123  
    CLVD: -31.527  
    DC: 27.35  
    M0: 5.6378e+14  
    MT: 7.3059e+14  
MOERRMAX: 2.4521e+13  
    MW: 3.8041  
    P: [329.94 58.774]  
    T: [154.38 31.15]
```

```
B: [63.19 1.9677]
F1: [62.706 76.178 -92.026]
F2: [251.13 13.967 -81.821]
Fault: {'NF'}
MXX: [1x6 double]
CXX: [1x6 double]
RMSError: 0.33807
Station: {1x14 cell}
UMEASURED: [1x14 double]
UTH: [1x14 double]
```

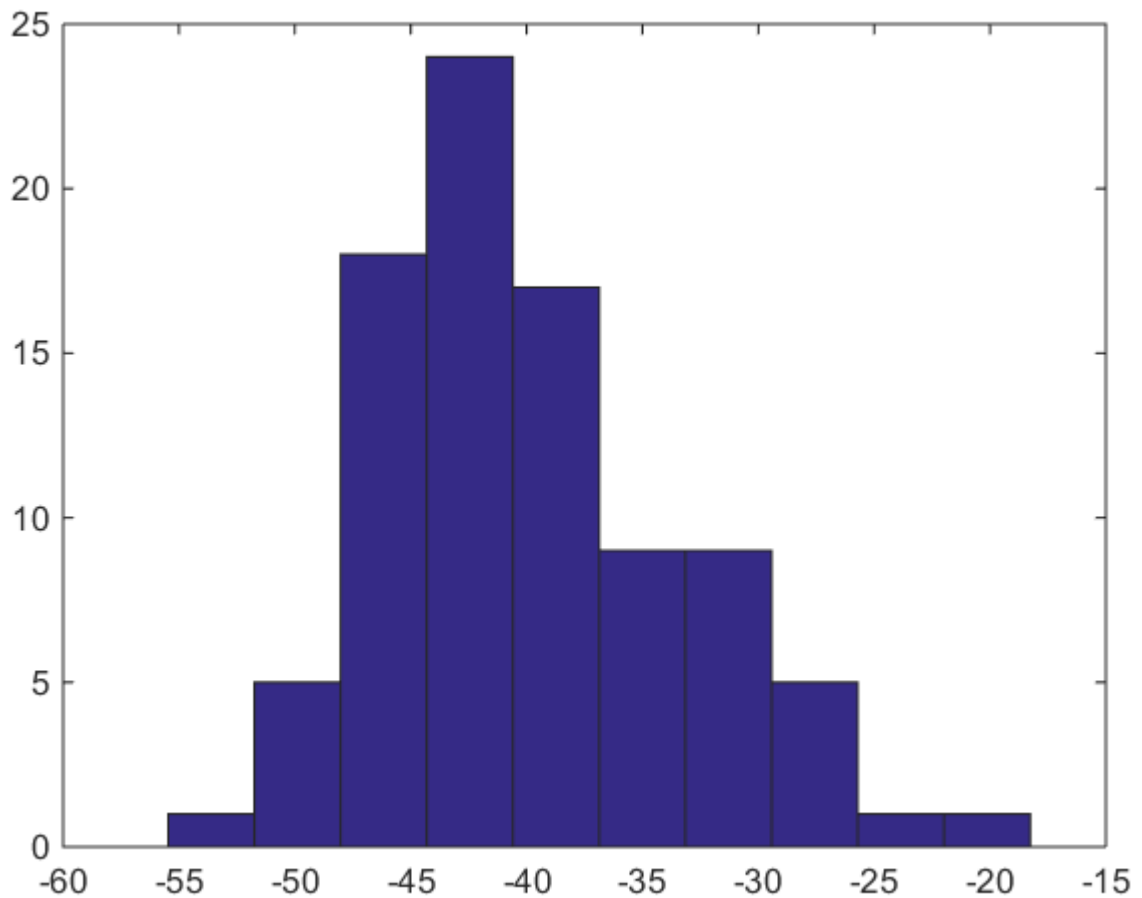
that provide us with results of *full* moment tensor inversion for the 5th seismic event.

Consider a scenario when we want to extract values of isotropic component from full moment tensor for all earthquakes, for example in order to plot statistics of *isotropic component* distribution in the analyzed dataset. This can be easily achieved by using the [getsolution.m](#) routine:

```
ISO = getsolution(Solution, 'full', 'ISO');
```

This will result in a vector composed of 90 elements where each element corresponds to the amount of *isotropic* component from all 90 earthquakes for which the full seismic moment tensor was calculated. This can be used for plotting the statistics of various parameters for the inverted dataset, e.g.:

```
hist(ISO)
```



will display the distribution of *isotropic components* in the analyzed dataset.

In principle, more parameters can be extracted from the output cell array at once. For example, to extract the moment magnitude and orientation of P axes for the *deviatoric* moment tensor solutions of all analyzed earthquakes, the following code may be executed:

```
[MW,P] = getsolution(Solution,'deviatoric','MW','P');
```

It is important to note that the order of input parameters ('MW','P' in the example above determines the order of output parameters located on the left side of the formula. Also, the names of input parameters to extract must correspond to the names of variables in the output structure array!

The function may be also used to extract general output data outside of 'full', 'deviatoric', and 'dc' structures inside of each cell array element. For example, the general data contains the ID number of events:

```
Solution{1}
```

ans =

```
    event_id: '004-03781'  
    n_trials: 1  
calculation_dt: 7.3626e+05  
    full: [1x1 struct]  
    deviatoric: [1x1 struct]  
    dc: [1x1 struct]
```

To extract all ID numbers, the following code may be used:

```
ids = getsolution(Solution, 'deviatoric', 'event_id');
```

In this case, the second parameter ('deviatoric') is of no use, but still it must be set up to either 'full', 'deviatoric', or 'dc'.



## Using focimt command line application

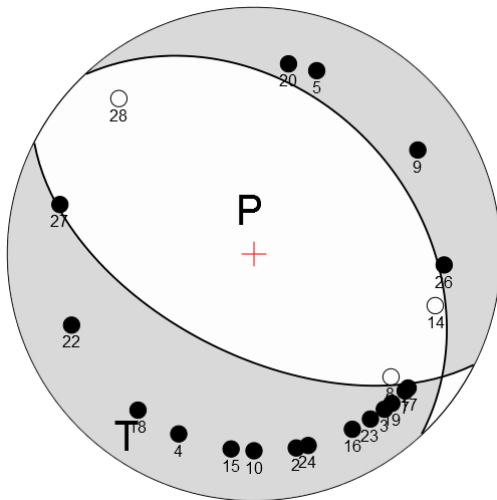
`focimt` application can be either used directly in shell mode or via `focimt.m` routine, depending on whether the User is outside or inside of MATLAB environment. It is also worth to note that MATLAB environment allows to call batch programs directly from command window by preceding the application name ('focimt') with exclamation mark. In this section we will use the latter option to present the features of `focimt` application and most important command line options. The full list of available command line options is presented later in this section. User can invoke the full list of command line options directly from MATLAB environment by typing `focimt -h`. Note that NOT all options of `focimt` are available within `focimt.m` routine.

### Basic usage

To perform the seismic moment tensor inversion in time domain using amplitude data stored in RAW ASCII format in `example1.txt` file stored in `/examples` directory and output the graphical representation of the solution to PNG and PDF formats:

```
!focimt -i ./examples/example1.txt -t PNG,PDF -o test01a
```

This will perform the constrained double-couple moment tensor inversion and the result will be stored with in files with `test01a` prefix. The call should result in creation of two files (beach balls) in PDF and PNG formats:

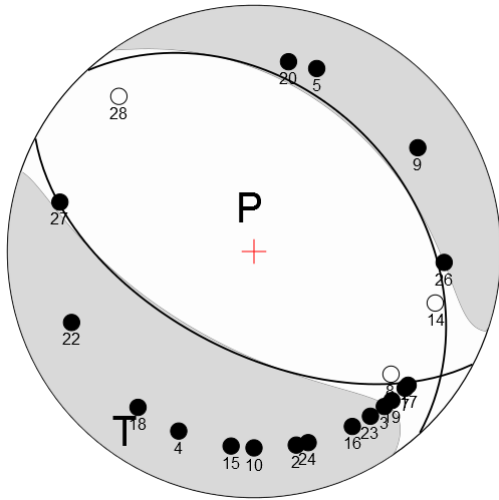
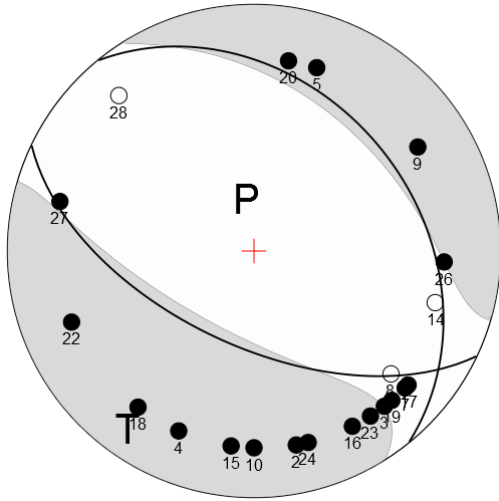


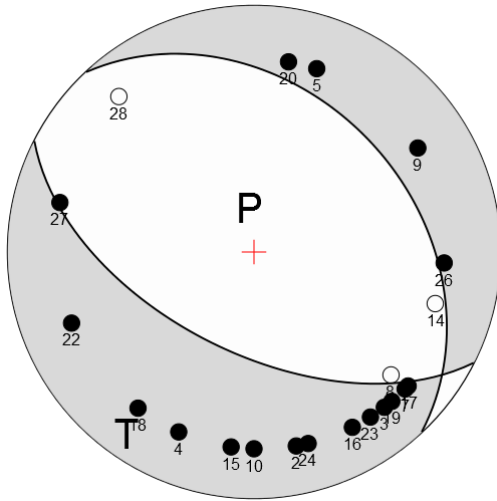
The `focimt` application is capable to generate four types of output files: one raster format PNG and three vector formats SVG, PS and PDF. Note the vector formats are suitable for high-quality publishing.

By default, only double-couple constrained moment tensor is calculated. To calculate full, deviatoric and double-couple constrained moment tensors at once for a particular event or events, option `-s FTD` must be used, where `[F]`, `[T]`, and `[D]` corresponds to full, deviatoric and double-couple solutions, respectively, as in the example below:

```
!focimt -i ./examples/example1.txt -t PNG -s FTD -o test01b
```

The successful execution results in generation of 3 PNG files with filename suffixes `-full`, `-deviatoric`, and `-dc` and corresponding to the [F]ull moment tensor solution (ISO+CLVD+DC), deviatoric or [T]race-null constrained solution (ISO=0,CLVD+DC) and [D]ouble-couple constrained solution (ISO=0,CLVD=0,DC).

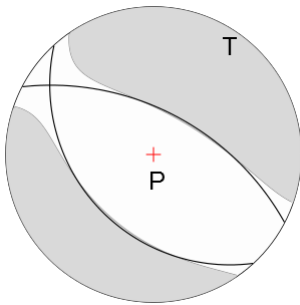




The graphical representation of seismic moment tensor solution and image size can be adjusted with `-b` and `-z` options. Also, the stereographic projection type can be modified using `-p` option. For example, to calculate the deviatoric solution only and create an upper-hemisphere, equal angle projection of the moment tensor solution without center cross and station polarity markers and with size of 300 pixels, one can call:

```
!focimt -i ./examples/example1.txt -t PNG -s T -p WU -b ACD -z 300 -o test01c
```

which results in the following picture:



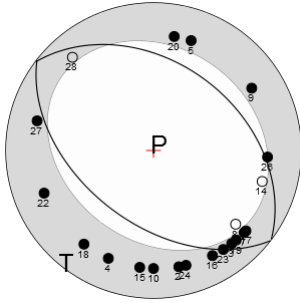
## Moment tensor inversion parameters

By default, the moment tensor inversion is performed using  $L2$  norm. This sometimes may NOT be an optimal case, especially in situations where one expects to have input data with large outliers. In this case, the program allows to perform the inversion with  $L1$  norm, which is less sensitive to large errors in the input data. However, this results in radically increased computation time.

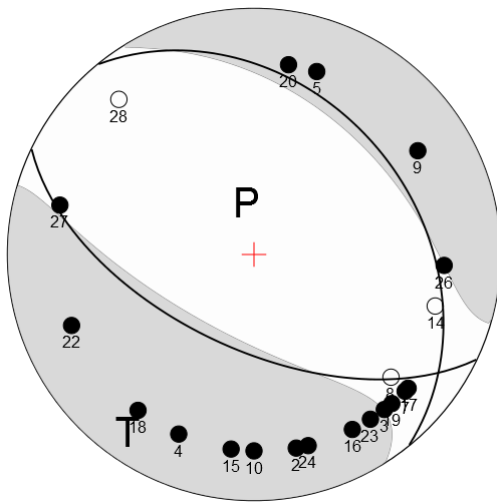
In example below we perform the full moment tensor inversion using  $L1$  norm and export the result to PNG file. This can be achieved by option `-n`:

```
!focimt -i ./examples/example1.txt -t PNG -z 300 -s F -n L1 -o test01d
```

The resulting graphical representation of the full moment tensor looks as follows:



and it is significantly different from full moment tensor solution calculated earlier using the *default* L2 norm:



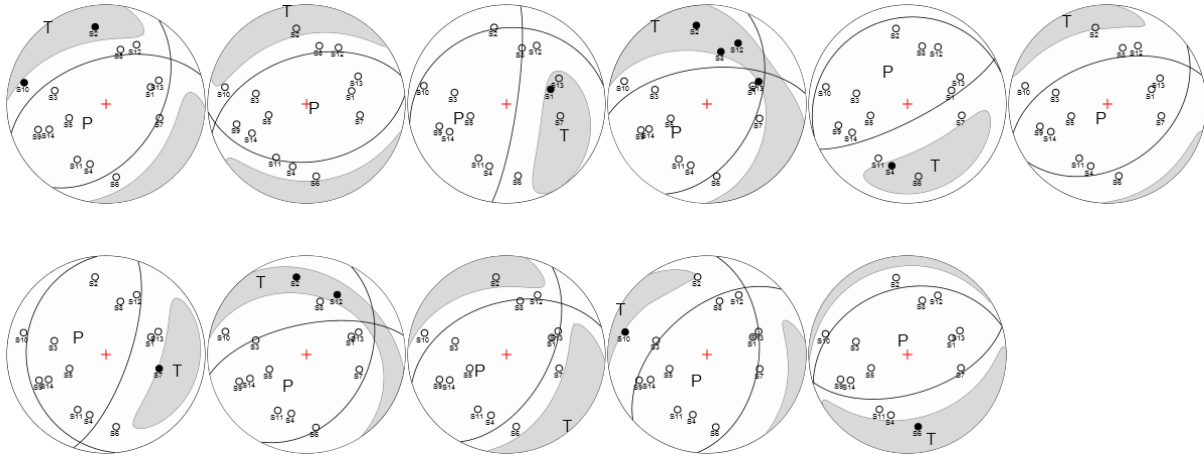
### Handling more input data at once

The moment tensor inversion can be performed for multiple files at once. The input ASCII file in either RAW or 1D velocity model formats may contain multiple blocks header event data - phase event data. This greatly speeds up the processing of input data for the purpose of moment tensor inversion.

Consider example of moment tensor inversion of acoustic emission input data for deviatoric seismic moment tensor. The input file `example3.txt` contains 50 events forming a tight cluster. In this example we calculate their full seismic moment tensors and export the output to PNG files. This can be achieved by issuing the following command:

```
!focimt -o test03a -t PNG -s F -z 200 -i ./examples/example3.txt
```

This will result in 50 PNG files being generated in the current directory. Here is example of 10 of them:



### Handling 1D velocity model input files

By default, the program expects data in [RAW ASCII format](#). However, the user is capable to provide input data for the moment tensor inversion in [1D velocity model format](#) as well. However, this type of data must be provided together with input 1D velocity model. It is important to mention that the stations should be located above the sea level for the ray-tracing routine to work correctly.

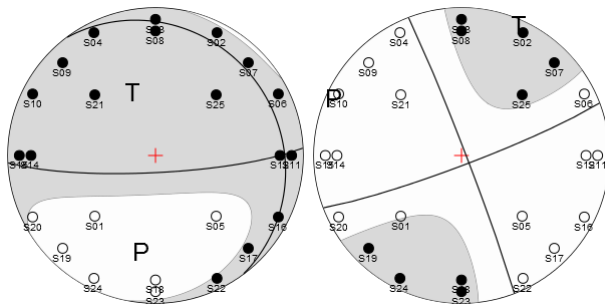
The 1D velocity model is provided using `-m` option followed by the path to ASCII file containing the model. The sample 1D velocity model file is located in `./examples` directory.

The `./examples` directory contains two files `example2_raw.txt` and `example2_1d.txt` with input data for the seismic moment tensor inversion. Both files have in fact the same input data, however the first one is in the RAW format and the second one is provided in 1D velocity model format. We can test now whether they both result in the same moment tensor solutions:

First, calculate the full moment tensor solutions of two earthquakes using the RAW input file:

```
!focint -o test02a_raw -t PNG -s F -z 300 -i ./examples/example2_raw.txt
```

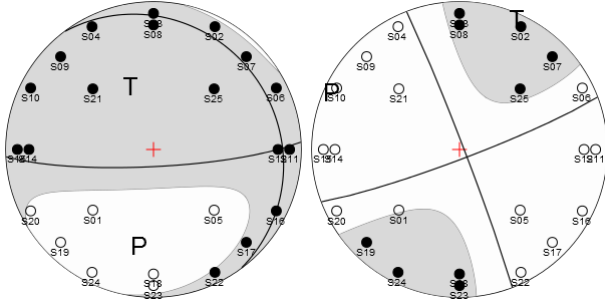
This results in the following graphical representation of the moment tensor solution:



Secondly, we calculate the same moment tensor solutions using the 1D velocity model input file. In this case we have to provide the additional 1D velocity model that will be used by ray-tracing routine:

```
!focimt -o test02a_1d -t PNG -s F -z 300 -i ./examples/example2_1d.txt -m ./examples/vmodel.txt
```

This results in the following beach balls for two earthquakes:



As one can see, both solutions are identical regardless of input file format chosen. To create input data in for the examples above, the auxiliary function `genmt_vel1d.m` from the package has been used. The detailed description of both input file formats is presented [here](#).

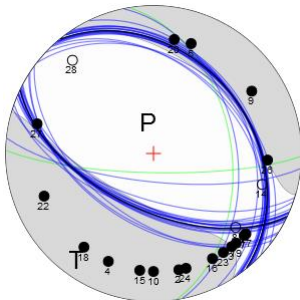
### Station Jackknife test

The station *Jackknife test* can be performed by adding the `-j` option to the command line. When this option is specified, the moment tensor inversion is firstly performed as usual with all amplitude information provided by the User. In the following, a series of  $n$  moment tensor inversions will be *additionally* performed, when  $n$  is a number of phase data for a particular earthquake, but now all but one station will be removed from input data. This tests aims to visualize the potential problem if one input station has a strong influence on moment tensor solution.

Again, we used the example ASCII file above to test the stability of full moment tensor inversion:

```
!focimt -i ./examples/example1.txt -t PNG -s F -j -z 300 -o test01e
```

The graphical representation of moment tensor inversion while using a *Jackknife test* looks as follows:



The shading corresponds to the original full moment tensor solution where all stations were used. The additional blue, green and red lines (the latter is not seen in the case above) corresponds to nodal planes of *Jackknife* solutions with the color corresponding to the sense of motion of the fault planes. The blue, green and red color of nodal planes corresponds to normal, strike-slip and thrust fault.

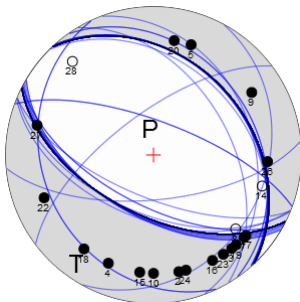
## Bootstrap resampling

The bootstrap resampling of the original dataset may be performed using `-rp` option followed by the number of bootstrap samples. Again, similarly to the *Jackknife test*, a series of  $n$  additional moment tensor inversions will be performed in addition to the regular inversion with an actual set of stations provided in the input file. For each elements of bootstrap sample specified percentage of randomly selected polarities will be reversed.

For example, to perform the 100 bootstrap resampling of the original dataset assuming that 1% of input phase data has reverted polarities, the following code should be executed:

```
!focimt -i ./examples/example1.txt -t PNG -s D -rp 100/0.01 -z 300 -o test01f
```

This result in following double-couple constrained moment tensor:



## Outputting inversion results to ASCII file

`focimt` is capable to export output moment tensor data to a customizable ASCII output text files. The option governing text output is `-d` followed by selected UPPERCASE or lowercase characters and eventually an asterisk `*` sign. Each character in the argument value string determines different parameter or set of parameters to export and the order of characters determine the order of columns in output ASCII file. Finally, if User specifies the character(s) in lowercase, they will be additionally formatted in eye-friendly manner.

The available characters in the argument value string and corresponding parameters that are exportes is as follows:

| character | Description  |
|-----------|--|
| M         | Moment tensor components in Aki's convention: M11,M12,M13,M22,M23,M33. The moment tensor components are in [Nm] units.   |
| C         | Moment tensor components in CMT conventions: M33,M11,M22,M13,-M23,-M12. The moment tensor components are in [Nm] units.  |
| F         | Fault plane solutions in format: STRIKEA/DIPA/RAKEA/STRIKEB/DIPB/RAKEB. All values are provided in degrees and follows the typical seismological convention.   |
| D         | Decomposition of the seismic moment tensor into Isotropic (ISO), Compensated linear vector dipole (CLVD) and double couple (DC) in format: ISO/CLVD/DC. The numbers are provided in percents and calculated according to Jost and Herrmann approach. |
| Y         | Decomposition of the seismic moment tensor into Isotropic (ISO), Compensated linear vector dipole (CLVD) and double couple (DC) in format: ISO/CLVD/DC. The numbers are provided in percents and calculated according to Vavrycuk approach.          |

|   |  |
|---|--|
| A | P/T/B Axes orientations in format: PTREND/PPLUNGE/TTREND/TPLUNGE/BTREND/BPLUNGE. All values are in degrees and trends ranging 0-360 degree and plunges ranging 0-90 degrees (0 - horizontal, 90 - vertical). |
| W | Seismic moment, total seismic moment, maximum error of the seismic moment tensor estimate and the moment magnitude calculated using Hanks & Kanamori formula. The first three values are in [Nm] units.      |
| Q | Quality index (used internally)  |
| T | Fault type. 'SS','NF' or 'TF' will be exported depending whether the faulting style is strike-slip, normal or thrust, respectively.  |
| U | Vector of synthetic displacements calculated (the number of exported numbers correspond to the number of amplitudes in the input file).  |
| E | RMS Error calculated between theoretical and measured amplitudes scaled to measured values, see page for details.  |
| V | Diagonal elements of the MT covariance matrix in the following order: C11, C22, C33, C44, C55, C66   |
| * | Export new line character.   |

As an example, specifying '-d FAD' option exports firstly fault plane solutions, then P, T and B axes directions and finally the moment tensor decomposition using ISO/CLVD/DC convention. The output ASCII file will have the following structure:

```
STRIKEA/DIPA/RAKEA/STRIKEB/DIPB/RAKEB/PTREND/PPLUNGE/TTREND/TPLUNGE/BTREND/BPLUNGE/ISO/CLVD/DBCP
```

Consider an example of seismic moment tensor inversion of two events. We would like now to produce two types of moment tensor solutions: full and deviatoric (option `-s FD`) and export the fault plane data (strike, dip and rakes) as well as moment tensor decompositions (iso/clvd/dc) into the text file. This can be achieved by calling:

```
!focimt -o test_export01a -i ./examples/example2_raw.txt -t SVG -s FD -d FD
```

Six files will be generated. Four of them will be graphical representations of moment tensor solutions for two events in SVG format and two types of moment tensors requested. The remaining two ascii files will contain the ASCII output, one for deviatoric and one for full moment tensor solutions.

## Auxiliary options of focimt application

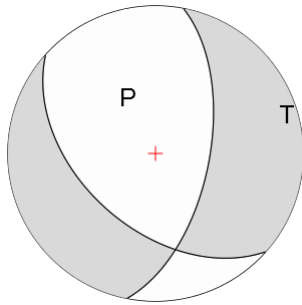
`focimt` provides a simple way to generate a double-couple graphical representation of the seismic moment tensor solution in a form of *beach ball* directly from fault plane solution parameters and (optionally) station and polarity information. This can be achieved by `-f` option:

In the easiest form, the strike, dip and rake parameters is required to plot a simple beach ball picture. To plot the fault plane solution of the fault with strike/dip/rake = 11/60/-50[deg] and save it to a PNG file with size of 300 pixels:

```
!focimt -o fault01a -f 11/60/-50 -z 300 -t PNG
```

with the resulting beach ball

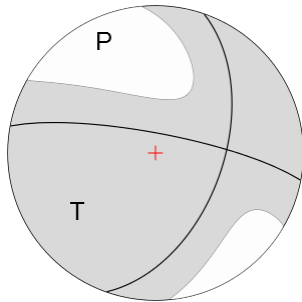




Note the output file name does not have any prefixes or suffixes as in the case of moment tensor inversion from the ASCII input file.

It is also possible to provide more fault plane parameters and produce a plot with multiple nodal lines, similar to what is plotted when station *Jackknife test* or *Bootstrap resampling* option is used. In this case, the first fault plane nodal line is painted with black color and the remaining nodal lines are painted according to the sense of motion. To achieve such effects, option `-fj` is used and the data from following nodal lines are separated with ':' character as in the example below:

```
!focimt -o fault01b -fj 11/60/-50:15/65/-42:17/55/-55 -z 300 -t PNG
```

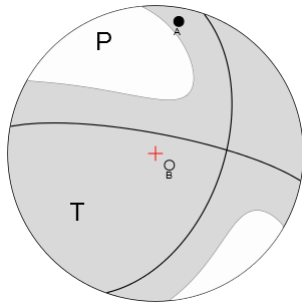


The example above draws three nodal lines in total with strike/dip/rakes 11/60/-50 (main nodal line), 15/65/-42 and 17/55/-55 (auxiliary nodal lines).

In addition, the stations can be plotted as an overlay in any of the above two examples by using option `-fs` followed by ':'-separated station information. The station information is provided in azimuth/takeoff/polarity/name format. Consider we want to add two stations to the example above:

```
!focimt -o fault01c -fj 11/60/-50:15/65/-42:17/55/-55 -fs 10/80/1/A:130/10/-1/B -z 300 -t PNG
```

resulting in the following beach ball:



The azimuth of station is measured from north positive towards east. The station takeoff is measured from vertical direction.

## Command line options

The full list of command line options of `focimt` can be displayed in console window by calling:

```
!focimt -h
```

Usage: `focimt [options [args]]`

```
--help, -h          Displays this information
--input,            -i <args> Full path to the input file
--output,           -o <args> Output file name (without extension).
```

If specified, the output solution data in ASCII format will be exported to a single file. Otherwise, 'fileid' field from input file will be used instead and moment tensor solution data will be exported to multiple files.

```
--solution,        -s <args> Output solution type.
```

Arguments: [F][T][D] for the (F)ull, (T)race-null and (D)ouble-couple solutions. Defines which moment tensor inversion will be performed. The default option is '-s D'. Combine three options to export desired moment tensor solutions, e.g. '-s DFT' will produce all three solutions at once.

```
--type, -t <args> Output file type.
```

Arguments: [NONE][PNG][SVG][PS][PDF] for different output file types. Produce graphical representation of the moment tensor solution in a form of the beach ball. More than one output file format can be specified. The default value is '-t PNG'.

```
--norm, -n <args> Norm type.
```

Arguments: [L1|L2] for L1 and L2 norm, respectively. Defines norm used in seismic moment tensor inversion. The default option is '-n L2' (faster). When Jackknife method is used the option is ignored and L2 norm is always used.

--projection, -p <args> Projection type.

Arguments: [W|S][U|L]: Defines projection for the graphical representation of the seismic moment tensor. Choose either (W)ulff projection or (S)chmidt projection. Then select (U)pper hemisphere or (L)ower hemisphere projection. The default option is '-p SL' (Schmidt projection, Lower hemisphere).

--ball, -b <args> The details of the beach ball picture

Arguments: [S][A][C][D]: Defines features of the graphical representation of seismic moment tensor. Plot (S)tations, (A)xes, (C)enter cross, best (D)ouble-couple lines. The default option is '-b SADC' (all features are displayed on the beach ball).

--dump, -d <args> Output data format and order.

Arguments: [M][C][F][D][A][W][Q][T][U][\*].

[M]: Moment tensor components in Aki's convention: M11,M12,M13,M22,M23,M33.

The moment tensor components are in [Nm]

[C]: Moment tensor components in CMT conventions: M33,M11,M22,M13,-M23,-M12

The moment tensor components are in [Nm]

[F]: Fault plane solutions in format: STRIKEA/DIPA/RAKEA/STRIKEB/DIPB/RAKEB (all values are in degrees)

[D]: Decomposition of the moment tensor into Isotropic, Compensated linear vector dipole and double-couple in format: ISO/CLVD/DBCP. The numbers are provided in percents and calculated according to Jost and Herrmann (1989) approach.

[Y]: Decomposition of the moment tensor into Isotropic, Compensated linear vector dipole and double-couple in format: ISO/CLVD/DBCP. The numbers are provided in percents and calculated according to Vavrycuk (2001) approach.

[A]: P/T/B Axes orientations in format:

PTREND/PPLUNGE/TTREND/TPLUNGE/BTREND/BPLUNGE

All values are in degrees.

[W]: Seismic moment, total seismic moment, maximum error of the seismic moment tensor estimate and the moment magnitude calculated using Hanks & Kanamori formula. The first three values are in [Nm]

[Q]: Quality index

[T]: Fault type. 'SS','NF' or 'TF' will be exported depending whether the faulting style is strike-slip, normal or thrust, respectively.

[U]: Vector of synthetic displacements calculated (the number of exported numbers correspond to the number of amplitudes in the input file).

[E]: Scaled RMS Error calculated between theoretical and measured seismic moments.

[V]: Diagonal elements of the MT covariance matrix in the following order: C11, C22, C33, C44, C55, C66

[\*]: Export new line character

NOTE #1:

The order of arguments determine to order of output. For example -d FAD exports firstly fault plane solutions, then P, T and B axes directions and

finally the moment tensor decomposition into ISO/CLVD/DBCP. The output file will have the following structure:  
STRIKEA/DIPA/RAKEA/STRIKEB/DIPB/RAKEB/PTREND/PPLUNGE/TTREND/TPLUNGE/BTREND  
/BPLUNGE/ISO/CLVD/DBCP

NOTE #2:

Use lowercase arguments in order to export data in eye-friendly format.

--model,            -m <args> Velocity model file (with extension)

Velocity model in HYP071 format. Imposes different ASCII input file format.

--jackknife,        -j            Performs station Jackknife test.

--amplitude,        -a <args> Perform amplitude test.

Arguments: x[/y] where x is a floating-point positive number that describes the level of noise applied to each amplitude:  $A + x \cdot A \cdot N(0,1)/3$  where N is a normal distribution with mean 0 and std 1. The default value of x is 1 (i.e. amplitude vary by a max. factor of ~2). Optional parameter /y is a number of samples (default value is 100).

--drawfault,        -f <args> Generate picture with fault plane solution

Arguments: strike/dip/rake

--drawfaults,       -fj <args> Generate picture with jackknife solutions

Arguments: strike/dip/rake[:s1/d1/r1][:s2/d2/r2]...

--drawstations,    -fs <args> Generate picture with station and polarity data

Arguments: azimuth/takeoff/polarity/name[:a2/t2/p2/n2][:a3/t3/p3/n3]...

--size, -z <args> Beach ball file size

Size of the beach ball figure in pixels.

--resampling,       -rp <args> Perform bootstrap resampling of the input dataset

Arguments: x/y where x is the maximum number of polarity mismatches and y is the number of bootstrap resamplings performed.

--version,           -v            Display fociMT version info

## Theoretical background

### Moment tensor inversion

---

The moment tensor inversion in `focimt` is a well-known procedure that relies on optimizing the following inverse problem:

$$U = GM$$

where  $G$  is the  $n \times 6$  matrix containing Green's function derivatives,  $U$  is  $n \times 1$  matrix of ground displacements observed,  $M$  is  $1 \times 6$  matrix containing six independent moment tensor components, and  $n$  is the number of ground displacement observations. As the above system of equations is overdetermined (it has more observations than unknowns), it is typically solved using the least-squares approach (L2 norm) with the cost function being the sum of squares of residuals. In addition, `focimt` allows to use the absolute (L1) norm which is less sensitive to larger errors at the cost of significantly increased computation time.

Regardless of the norm used to optimize equation, the moment tensor inversion is performed systematically in three different ways assuming unconstrained, as well as constrained deviatoric and double-couple moment tensors. The deviatoric moment tensor solution is obtained by assuming no volumetric change in the resolved moment tensor (i.e. trace of the resolved moment tensor is equal to zero) during linear inversion. The double-couple constrained moment tensor is resolved by further imposing the determinant of the seismic moment tensor to be zero as well. The latter constraint makes the moment tensor inversion scheme non-linear and the Lagrange multipliers method (Oncescu, 1986) is used to determine the double-couple constrained moment tensor.

The resulting seismic moment tensors are decomposed into isotropic (ISO), compensated linear vector dipole (CLVD) and double-couple (DC) parts following the default decomposition scheme of Knopoff and Randall (1970) with percentage of decomposed tensor elements calculated by either Knopoff and Randall (1970) or Vavryčuk (2001). The isotropic part describes changes in the volume in the seismic source region. The deviatoric part (CLVD+DC) is used to estimate the orientation of tensional, compressional and null axis directions, fault plane orientation as well as slip direction. The uncertainties are provided as RMS errors between recorded and modelled ground displacements and covariance matrix. The fault type is categorized into either strike slip, normal or thrust faulting depending on the relative orientation of cardinal axes of the resolved moment tensor.

### Uncertainty assessment

Uncertainties of the estimated moment tensors can be estimated through the normalized root-mean-square (RMS) error between theoretical and estimated amplitudes (Stierle et al., 2014a, 2014b) following the formula:

$$RMS = \sqrt{\frac{\sum_{i=1}^N (U_i^{measured} - U_i^{th})^2}{\sum_{i=1}^N (U_i^{measured})^2}}$$

The application is capable to produce a bootstrap sample of seismic moment tensor solutions by randomly sampling input amplitude data. Having the bootstrap sample of solutions, it is possible to calculate the probability distribution functions of seismic moment tensor components and other derived parameters.

## Hybrid moment tensor refinement

---

Hybrid moment tensor (HMT) technique was originally developed by Andersen (2001). This methodology overall aims at decreasing the influence of local path, site and sensor effects on estimation of seismic moment tensors. Frequently, the knowledge on waveform propagation between the source and receiver is very poor, however the frequency and amplitude content of body waves excited from the source is affected by attenuation, scattering and local site effects resulting in generally unknown Green's function that can be hardly modelled with typically insufficient information on geological medium. In addition, misinformation on sensor characteristics (e.g. bad gain, reversed polarity) contributes directly to the actually measured ground displacement values leading to the additional bias of resolved seismic moment tensors.

The Hybrid Moment Tensor (HMT) concept is in principle based on the idea of empirical Green's function technique used in the seismic source tomography for extraction of the relative source time function or kinematic source inversion. In this method the small seismic event is used as an approximation of the Green's function and it is used to deconvolve the propagation, site and sensor effects from the recorded seismogram of the investigated earthquake. However, the eGf candidate and investigated earthquake must share certain source properties (Kwiatek, 2008) in order to not introduce further errors into the deconvolution procedure.

The HMT technique is performed for a selected cluster of seismic events located close enough in comparison to the source-receiver distances. This means the earthquakes forming a cluster should share similar travel paths of body waves from seismic sources to sensor and we may assume the invariance of Green's function between events forming a cluster and a particular station.

The HMT technique is an iterative procedure. In the first step, the seismic moment tensor inversion is performed for all seismic event forming the cluster using the provided input data. In the following, the resulting seismic moment tensors (i.e. the components of seismic moment tensor) are used to predict the amplitudes (moments) at each sensors for all events. The ratios:

$$r_{ij} = \frac{u_{ij}^{th}}{u_{ij}^{obs}}$$

are formed for the synthetic and observed amplitudes (seismic moments) for each station  $i$  and event  $j$ . Then, the median ratios  $r_i$  are calculated for each sensor separately. The median ratio at station  $i$  is used to update the input ground displacement amplitudes using the following scheme:

$$u_{ij}^{*,obs} = u_{ij}^{obs} + w_i u_{ij}^{obs} (r_i - 1)$$

where  $w_i$  is the weighting factor for each station. The updated ground displacement data are used to calculate a new set of seismic moment tensors. The procedure is repeated until the ratio correction factor becomes insignificant (Andersen, 2001):

$$|r_i - 1| < \epsilon$$

### Applicability

The HMT technique is a useful tool to improve the quality of seismic moment tensor solutions coming from earthquakes forming a tight cluster. The algorithm can detect the inappropriate gain/coupling of certain station(s) and provide the station corrections factors. However, the algorithm is NOT capable to detect the inappropriate station polarity.

### References

Andersen, L. M. (2001). A relative moment tensor inversion technique applied to seismicity induced by mining, Univ. of the Witwatersrand, Johannesburg.

Fitch, T. J., D. W. McCowan, and M. W. Shields (1980). Estimation of seismic moment tensor from teleseismic body wave data with application to intraplate and mantle earthquakes, *J. Geophys. Res.* 85, 3817–3828.

Knopoff, L., and M. J. Randall (1970). The compensated linear-vector dipole. A possible mechanism for deep earthquakes, *J. Geophys. Res.* 75, 1957–1963.

Oncescu, M. C. (1986). Relative seismic moment tensor determination for Vrancea intermediate depth earthquakes, *Pure Appl. Geophys.* 124, 931–940.

Stierle, E., V. Vavryčuk, J. Šílený, and M. Bohnhoff (2014). Resolution of non-double-couple components in the seismic moment tensor using regional networks—I: a synthetic case study, *Geophys. J. Int.* doi 10.1093/gji/ggt502. [online] Available from: <http://gji.oxfordjournals.org/content/early/2014/01/13/gji.ggt502.abstract>

Stierle, E., M. Bohnhoff, and V. Vavryčuk (2014). Resolution of non-double-couple components in the seismic moment tensor using regional networks—II: application to aftershocks of the 1999 Mw 7.4 Izmit earthquake, *Geophys. J. Int.* doi 10.1093/gji/ggt503. [online] Available from: <http://gji.oxfordjournals.org/content/early/2014/01/13/gji.ggt503.abstract>

Vavryčuk, V. (2001). Inversion for parameters of tensile earthquakes, *J. Geophys. Res.* 106, no. B8, 16339–16355, doi 10.1029/2001JB000372.

Wiejacz, P. (1992). Badanie mechanizmów wstrząsów górniczych przy wykorzystaniu tensora momentu sejsmicznego, Institute of Geophysics, Polish Academy of Sciences.