UNIVERSITÀ
DEGLI STUDI DI TRIESTE

dipartimento di matematica e geoscienze

The DMG Manuals

Indirect parameterization (INPAR) method for the retrieval of moment tensor

Mac OS Version 2010

Document name: DMGInpar.pdf
Document Version: 0.9.1
Date Issued: September 25, 2017 5:38 PM
Author: Enrico Brandmayr

University of Trieste
Department of Mathematic and Geosciences
Group of Seismology
via Weiss, 4
34128 Trieste, Italy

Table of Contents

Indirect parameterization (INPAR) method for the retrieval of Moment Tensor

Mac OS Version 2010

USER GUIDE

Theoretical background

INPAR (INdirect PARameterization), was developed at the Department of Geosciences, University of Trieste (Sileny et al., 1992; Sileny, 1998). The INPAR method for the inversion of moment tensor, that adopts a point-source approximation, is a sophisticated non-linear method using the dominating part of complete waveforms from events at regional distances (up to 2500 km), in this way maximizing the signal-to-noise ratio. It consists of two steps, the former linear, the latter non-linear.

The k-th component of displacement at the surface determined by equivalent forces in the point-source approximation is the convolution product of moment tensor and Green's function spatial derivatives (Aki and Richards, 1980):

$$u_k(t) = M_{ij}(t) * G_{ki,j}(t) \quad (1)$$

The Green's functions represent the response of the medium to singlet sources with Dirac-delta time dependence.

In the first step, Green's functions are extracted from signals to calculate the time derivatives of the six independent components of the moment tensor (hereafter MTRF) (Panza, 1985; Florsch et al., 1991; Panza et al., 2000), in this case with a time dependence given by Heaviside (step) function. Computing the MTRFs from Green's functions does not require an a priori source model. Since a bad location of the focus may strongly affect the results of the inversion, INPAR performs, whenever necessary, a re-location of the focus within a fixed volume around the available location. For intermediate coordinates within this volume the base functions, i.e. the medium response to a seismic source (modeled as an elementary single source with the time dependence given by Heaviside function), are computed interpolating the base functions calculated at the corners of the volume itself. From base functions, the synthetic seismograms are computed and are compared with observed ones: the L^2 norm of synthetic and observed seismograms is minimized to determine the new focus location.

In the second, non-linear step, the MTRFs are factorized in a time constant moment tensor mij and a corresponding source time function f(t):

$$aMij(t)/at=mij.f(t)$$

assuming the same time dependence for all moment tensor components, i.e. a rupture mechanism constant in time.

The “predicted” values of the MTRFs, mij·f(t), are matched to the “observed” MTRFs (obtained as the output of the first step of the procedure). Thus, by introducing MTRFs in the first step, the problem of matching the seismograms is transformed to the problem of matching the MTRFs. This allows the reduction of the equations system for the searched source parameters, since the MTRFs are always six, while seismograms can be more than six. In order to reduce the errors coming from the poor knowledge of the structural inhomogeneities of the epicenter-station profile, which can cause scattering, only the correlated parts of the MTRFs are taken into account (Kravanja et al., 1999).

Additional constraints, such as the positivity of the source time function and, whenever available, consistency with polarities, are posed. After several iterations, the focal mechanism is obtained and reduced to the best double couple in case of tectonic events. In this way the volume components, which are often artefacts of the inversion process, are discarded (Kravanja et al., 2000).

INPAR is particularly suitable for determining the foci of shallow events, due to the possibility to use waveforms of relatively short period (down to 10 s). Theoretically, the seismic moment tensor cannot be univocally determined from surface waves signals whose wavelength significantly exceeds the source depth (Bukchin and Mostinskii, 2007), since two components of the moment tensor, Mzx and Mzy, excite a Green’s function vanishing at the free surface. Because of this, the CMT and RCMT methods, which respectively use body and surface waves at periods greater than 35 s, usually fix shallow events at depths of 10 km the former and 15 km the latter.

Before starting

The INPAR inversion method consists of several programs and most of them are compiled using the Sac libraries since Sac allows direct Fortran reading/writing and plotting/processing facilities that will avoid the use of several accessory programs. Therefore the input data format is the international standard Sac format. If you get data from international networks, you'll probably get them directly in Sac format, or in formats for which the conversion to Sac is straightforward (SEED, AH, etc., the most common program for conversion of seed files is rdseed).

The files with response of each station should exist (#please include an example, because rdseed creates responses slightly different and relevantly unreadable for ker4#).

The structure files: at least 2 files (one for the origin and other for the station-to-epicenter paths, can be the same file but named differently). Structure calculation: calculation of str -> stp files. (#I should ajust the calculation of average model#)

N.B. The names in **bold** are fixed and should not be changed, other file names are user-dependant.

1 The first step

1.1 Data processing and computaion of the base functions

Program

ker4_is

Input

ker4.inp (arg)

Output

CHECK_ME, ker.tmp, picker, kernels, corrected Sac files

The programme **ker4_is** performs the data processing and computes the base functions (kernels).

Data processing steps

- data alignment to the event origin time (in order to allow the source time function to start before the theoretical origin time, the origin time is shifted by a quantity related to the error in the determination of the origin time of data (at least two sampling steps, but it could be defined as the maximum shift between origin times estimated by different agencies);

- correction for instrumental magnification (but not instrumental response) in order to preserve the amplitude radiation pattern;

- filtering. Filtering may consist of a gaussian (non-dephasing) low-pass filter and/or a common variable time-frequency filter. Usually the filter should be applied to both data and synthetics (option 2), or not applied at all (option 0). If the data have already been filtered, only the synthetics should be low-pass filtered (option 1 of the gaussian filter). Since variable time-frequency filtering is very time consuming, it is possible to apply such filter only to the data (option 1 of the variable filter: in this case the filtering window should correspond to the synthetic structural one in order to perform coherent filtering of data and synthetics). Different filters can be used for the various stations (for example the variable filters may be defined from the **FTAN** diagram of each observed seismogram). Processed data are saved in Sac format with the extension **.data.**

The base functions are computed for a set of values of the hypocentral coordinates lying between two extremes, namely for a set of points in a 3D grid, and for two extreme structural models. The spectra of the structural models must have been prepared in advance with *ray* or *lov*. The instrumental response is applied to the kernels. One Sac file is created for each base function. The name of each base function is defines as follows:

STA.C.nnnn.m

where STA is the station name, C is the component (Z=vertical, R,=radial, T=transverse, N=North-South, E=East-West,), nnnn is a set of four numbers which correspond respectively to the depth index on the grid (0 to 9), the latitude index (0 to 9), the longitude index (0 to 9) and the Mij index (1 to 6). The last number refers to the structural model used to compute the base function. All the base functions are computed with a scalar moment of 10e15Nm if no correction is applied.

It is possible to use up to two structural models for each source-receiver path (this is convenient at a large regional scale where path are inhomogeneous) and only one model for the source region (in order to better constrain the inversion for the source parameters); it is also possible to use different models for source region and source-receiver paths (best for short distances).

If only one structural model is used for source region or for source-receiver path, or both, just fill the field for second structure spectrum in *ker4.inp* with the same file name of the first structure. Source structure may be eventually the same of some of the source-receiver path structures, but in this case the structure should be copied to different files, with different names, one for source region and one for source-receiver path.

In general, the source region spectrum cannot have a number of computed modes higher than any of the source-receiver path spectra, otherwise **ker4_is** will abort execution and an error message will be written in **CHECK-ME.** In this case the source spectrum should be recomputed with *ray* and/or *lov*, with the number of modes set to the lowest one between the source-receiver path spectra.

Source-receiver distances should not exceed 2500 Km, otherwise the computation of the excitation term is affected by the Earth's sphericity.

The base functions corresponding to intermediate values of hypocentral coordinates and structures are computed with a linear interpolation of the base functions evaluated at the grid defined by the assumed set of coordinates and structures. This is a strongly non-linear interpolation in terms of structure. Therefore, you should take care to specify relatively homogeneous structures. You may for example just change the Q factor. Or use a common homogeneous structures as first choice and individual structures as second limit.

No other inputs are needed (except structure spectra, data and poles/zeros files).

Run ker4_is.

ker4_is needs one *input file*. By default of argument, it will search for the *ker.inp* input file. But you may specify an argument afterwards in order to differentiate several input files. For example you can run **ker4_is** myarg, which will read the *ker.inp* myarg input file.

The program will write the processed data in the work directory and the base functions in the specified directory. Information, warnings and errors are reported in the **CHECK-ME** file. **CHECK-ME** file records all warning and error messages that might occur during the program execution. It is recommended to have a look to it afterwards. If a major error happens, the program will stop with a short error message returned.

It will also keep in memory basic information in a file named **ker.tmp** and produce a Sac macro command named **picker,** ready to use to cut afterwards data and kernels.

First, program **ker4_is** will read your data and align them on the Event Origin time (as kernels will be). A gaussian filter may be applied either to the kernels only (do it only if data have been pre-filtered !) or to both data and kernels equally. A variable time-frequency filter may be applied to data only (since it is highly time-consuming for numerous kernels; but in this case use the structural path phase velocities as limits for coherency), or to both data and kernels equally. Filters may differ for each station. The processed data are rewritten in Sac format, named Station.Component.data.

Then **ker4_is** will produce kernels in (binary) SAC format. This will make a big number of files (1 file per kernel), but the global storage is 3 to 5 times less voluminous than the previous storage in IGG-Ker0048 format. And kernels will be ready to plot and of direct access. The sac file name of each kernel is made of the station name and the component, followed by the depth, latitude, longitude and Mij indices (0 to max.9 for the grid, 1 to 6 for Mij), and by the structure number (1,2)

Example: BAD.Z.3001.2 is the kernel for station BAD vertical, fourth depth, first latitude and longitude, Mxx, and for the second structural model.

Usually your data are sampled finer than you really need (usually about 0.05 s), due to the fact that the period range used in the inversion usually starts at 10 seconds. The sampling step should be therefore chosen accordingly to the filtering frequency, following the Nyquist theorem.

Kernels will be automatically computed at the same sampling step as the data one. For the further inversion, **it is important that all data have the same sampling step**. If this is not the case, **ker4_is** will abort execution and an error message will be written in **CHECK-ME**. In this case, you have to resample the data using Sac command "interpolate delta *new_step*" and then run again **ker4_is**.

Following an example of the file ker4.inp is given, explaining in particular the parameters to be specified for the different types of instrument responses. The input file is read in free format, the only limitation being that filenames should not exceed 45 characters.

If you don't have any input file as model, or if you have a doubt, you can just run **ker** and answer to the questions; the program will prepare the input file for you, ready to use next time.

Example of input file ker4.inp

```
/scratch/kernels                Output directory for kernels (to be    created  by user)
/scratch/friul7s.spe                    [Source Structure spectrum 1]
/scratch/friul7s.spe                    [Source Structure spectrum 2]
                                                Source Structure Love spe 1
                                                Source Structure Love spe 2
0 15  6                     FirstMode, LastMode, Interp.Factor
46.3590  46.3590  0                     MIN.,MAX.Source LATITUDE, Nb.Interv
```

```
13.1252  13.1252  0                    MIN.,MAX.Source LONGITUDE,Nb.Interv
 7.   7                          MIN.,MAX.Source DEPTH(km),Nb.Interv

1 20 0.7                              Event Origin time: hour min sec
0.1                                   Time Shift in the origin time
5                                     Number of stations
STA1 Z                                STATION name, Component(s)
35. 45. 5.                         Station LatN dg, LongE dg, Depth km
/scratch/friul7a.spe               [Path Structure spectrum 1]
/scratch/friul7a.spe                [Path Structure spectrum 2]
0 0 0                              Instr.,GaussFilt.,VarFilt. options
1000.                              NoInst.: Mag.in Data_unit/m(/s)(/s)
0. 0. 0.                              GauFil:[CutFr,PeakFr/CutFr,AmpACut]
                                   VarFil:[file with Velocity limits]
sta1.sac                              Data file name (sac)
CAE  N E Z                         STATION name, Component(s)
46.0067 12.4367 0.                    Station LatN dg, LongE dg, Depth km
/Spectra/friul7w.spe                  Path Structure Rayleigh s 1
/Spectra/friul7w.spe                  Path Structure Rayleigh s 2
/Spectra/friul7w.lspe                 Path Structure Love spe 1
/Spectra/friul7w.lspe                 Path Structure Love spe 2
3 2 0                              Instr.,GaussFilt.,VarFilt. options
28700. 1.04 0.266                  Geophone (N):Magnific.,Period,Damping
26500. 1.02 0.28                   Geophone (E):Magnific.,Period,Damping
28700. 1.05 0.26                   Geophone (Z):Magnific.,Period,Damping
3. 0.9 0.1                         GauFil:[CutFr,PeakFr/CutFr,AmpACut]
                                      VarFil:[file with Velocity limits]
880202102341.016CAE_N.sac             Data file name (N) (sac)
880202102341.017CAE_E.sac             Data file name (E) (sac)
880202102341.015CAE_Z.sac             Data file name (Z) (sac)

STA2  R     ..... STATION name, Component(s)
45. 45. 0.                         Station LatN dg, LongE dg, Depth km
/scratch/friul7w.spe                  [Path Structure spectrum 1]
/scratch/friul7a.spe               [Path Structure spectrum 2]
1  2  0                                Instr.,GaussFilt.,VarFilt. options
4660.0   0.85251D+00 0.9691               Mec.Instr:Magnific.,Period,Damping
10.  .9  .1                           GauFil:[CutFr,PeakFr/CutFr,AmpACut]
                                         VarFil:[file with Velocity limits]
```

```
sta2.R.sac                          Data file name (sac)
STA3  Z R   ........................... STATION name, Component(s)
45. 60. 0.                          Station LatN dg, LongE dg, Depth km
                                    [Path Structure spectrum 1]
/scratch/friul7a.spe                    [Path Structure spectrum 2]
2 1 0                               Instr.,GaussFilt.,VarFilt. options
5250. .8 .9 .9 .7 .9                    ElMag.:P.Per,G.Per,P.Dam,G.Dam,Coup
GauFil:[CutFr,PeakFr/CutFr,AmpACut]
                                          VarFil:[file with Velocity limits]
STA3.Z.sac                          Data file name (sac)
STA3.R.sac                          Data file name (sac)
TS  T                               STATION name, Component(s)
39. -130. 0.                        Station LatN dg, LongE dg, Depth km
/scratch/friul7wLOV.spe             [Path Structure spectrum 1]
                                    [Path Structure spectrum 2]
3 2 0                               Instr.,GaussFilt.,VarFilt. options
0.49600D+04 0.7874 0.11228E+01          Geophone:Magnific.,Period,Damping
10 .8 .1                            GauFil:[CutFr,PeakFr/CutFr,AmpACut]
TS.T.vlim                           VarFil:[file with Velocity limits]
TS.T.sac                            Data file name (sac)
UDI Z R                             STATION name, Component(s)
37.3 -146.3 0.                          Station LatN dg, LongE dg, Depth km
[Path Structure spectrum 1]
[Path Structure spectrum 2]
4 2 1                               Instr.,GaussFilt.,VarFilt. options
../PZfiles/UDI.pz                   Poles/Zeros File
8. .95 .05                          GauFil:[CutFr,PeakFr/CutFr,AmpACut]
UDI.vlim                            VarFil:[file with Velocity limits]
UDI.Z.sac                           Data file name (sac)
UDI.R.sac                           Data file name (sac)
```

_____ end of input file_____

This file is read in free format, but filenames should not exceed 45 characters.

You can use a maximum of 26 stations with 3 (N, E, Z) components, 2 Rayleigh components OR 1 Love component. [Lines into brackets may be left blank if the corresponding option is not used]. The input is made of three parts: General options, Source parameters, Stations.

The file is read in free format. Its structure is the following one:

the OUTPUT DIRECTORY where to put the (numerous) kernels (1 file per kernel).

the file containing the FIRST STRUCTURE SPECTRUM for the SOURCE region (it is compulsory to use at least one source structure. Even if it is the same of one of the source-receiver path structures, the name of the file has to be different, otherwise the program will stop.).

the file containing the SECOND STRUCTURE SPECTRUM for the SOURCE region.

the file containing the FIRST STRUCTURE LOVE SPECTRUM (IF NEEDED) for the SOURCE region.

the file containing the SECOND STRUCTURE LOVE SPECTRUM (IF NEEDED) for the SOURCE region.

the FIRST and LAST MODES to be used (fundamental=1, all= 0 0, up to 15th mode= 0 15); the INTERPOLATION FACTOR (0: no interpolation, >0: increases the number of time points).

the MINIMUM and MAXIMUM SOURCE LATITUDES (North>0,dg) of the grid, and the NUMBER OF SUBDIVISIONS of this interval (maximum 9).

the MINIMUM and MAXIMUM SOURCE LONGITUDES (East>0,dg) of the grid, and the NUMBER OF SUBDIVISIONS of this interval (maximum 9).

the MINIMUM and MAXIMUM SOURCE DEPTHS (km) of the grid, and the NUMBER OF SUBDIVISIONS of this interval (maximum 9).

the event ORIGIN TIME in HOURS, MIN., SEC. (if possible, as in data header, otherwise the earliest from bulletins).

the NUMBER OF STATIONS

Time shift in the origin time : this value is related to the error of the origin time in the data, or, if such information is not available can be considered at least as twice the original sampling step.

For each station, 9 or 10 lines specifying:

the STATION NAME (max.A4) and the LIST OF COMPONENTS to compute (N for North, E for East, Z for vertical, R Z for both, or T for transverse). Separate Love & Rayleigh ! (components North, East cannot be computed: it's better to invert R and T).

the STATION LATITUDE (North>0 degrees),LONGITUDE (East>0 degrees), DEPTH (km).

the file containing the FIRST STRUCTURE SPECTRUM for this source-station PATH.

the file containing the SECOND STRUCTURE SPECTRUM for this source-station PATH.

the INSTRUMENT OPTION (0: no instrument applied, ground displacement computed in meters, -1: derivator to ground velocity in m/s applied, -2: derivator to ground acceleration in m/s/s applied, 1: mechanical instrument applied, 2: electromagnetic instrument applied, 3: geophone applied, 4: instrument res-ponse in poles/zeros applied);

the lowpass GAUSSIAN FILTER OPTION (0:no filter applied, 1: only kernels to be filtered, 2: both data and kernels to filter);

the time-frequency VARIABLE FILTER OPTION (0: no, 1: synthetic limits to be applied to data only, 2: observed limits to apply to both data & kernels). one line specifying the appropriate instrument characteristics: for -2,-1,0: the UNIT CORRECTION (in data_unit / meter(/s)(/s), by default 1); for 1 or 3: MAGNIFICATION, NATURAL PERIOD, DAMPING (not the damping ratio !);for 2 MAGNIFICATION, Pendulum&Galva-PERIODS,Pendulum&GalvaDAMPINGS, COUPLING fac; for 4: the name of the POLES/ZEROS FILE (which should contain the unit).

the CUTOFF FREQUENCY (Hz),

the PEAKfreq./CUTOFFfreq. RATIO (0. to 1.), the AMPLITUDE AT CUTOFF (0. to 1.) of the lowpass gaussian filter (the line may be left blank when no gaussian filter is used).

the FILE containing the VELOCITY limits of the variable time-frequency filter to be applied (the line may be left blank when no variable filter is used).

the file containing the DATA, in Sac format (use igg2sac to convert IGG data).

(the file containing the DATA for the second Rayleigh component if both used).

* (next station(s))

See the above file as example. The various descriptions are given for each of

the five possible instrument types.

Other inputs

input structure spectrums are specified in the input file and should have been prepared by ray program for Rayleigh or lov for Love. Read the proper manual for detailed instructions. You can calculate the average source-station model using the 1°x1° database available at DST

eventual input instrument responses in poles&zeros are specified in the input file and should be in SAC-compatible format, with the unit specified in du/m, du/m/s, du/ms-1, du/m/s/s, du/ms-2 or du/m/s**2.

the original data files in Sac format.

==========

Example of output file ker.tmp

```
     1.500     1.500     0.000   0  DepMin,DepMax,dDep,Nintv
    40.825    40.825     0.000   0  LatMin,LatMax,dLat,Nintv
    14.421    14.421     0.000   0  LonMin,LonMax,dLon,Nintv
scr4dt/
SGV   N    1.067  213.616         Station,Comp., Dist,Az.
SGV.N.data
SGV   E    1.067  213.616         Station,Comp., Dist,Az.
SGV.E.data
SGV   Z    1.067  213.616         Station,Comp., Dist,Az.
SGV.Z.data
  3   1   2                       Nsignals, Nstat., Nstruc.
 10                               Triangle half-width / dt
     0.100                             Time shift of the origin time (s)
```

The width of the Triangle is computed as

2*sampling of data*DTAU

(where dtau is the “Triangle half-with/dt”). **To change the width of the triangle change DTAU!**

==========

1.2 Cutting data and base functions

Once **ker4_is** has been executed without errors, run **Sac** and use autogenerated Sac macro command **picker** to visualize together the data and extreme base functions (extreme points of the grid, for the two structures, only the M11 component plotted) and defined the limits of the window to invert (window may be different for each station, depending on their epicentral distance, noise...):

SAC> **m picker**

A graphic window should appear, with the data on top, the extreme base functions for the first and second structure below, and a cursor. The limits of the window are set automatically in order to see the useful part of the signal, whatever the distance: the first point correspond to the initial time, the last one to a group velocity of 1 Km/s.

Position vertically the cursor on the data signal to define the limit of the noise window (this is needed to compute the noise variance) and press **J.** You may also position the cursor horizontally to pick at the same time your estimation of the noise level, to compare it to the computed value, but anyhow only the computed value will be considered in the inversion.

Position vertically the cursor to define the beginning of the window to invert and press **P**. Position vertically the cursor to define the end of the window to invert and press and press **F** (a FINI bar appears) Then press **N** to pass to the next station. For each plot, a Sac Graphic File (**f001.sgf, f002.sgf, ..**.) is created.

You should keep in memory, during the picking the maximum time shift (pick to pick) observed between the base functions of the two structures, since this will be required afterwards, in order to interpolate the base function between the structural nodes. Alternatively, you can review the plots converted using the Sac command sgftops f00#.sgf f00#.ps).

At the end, exit from Sac. The picked values have been written in the **pick.dat** file: each line contains the filename on which the picking has been made, the pick identification (NL: noise level, IP: beginning, FINI: end) and the pick time. Other values are not useful (pick amplitude, etc....). N.B. if you want to redo the picking, you must delete the previous **pick.dat** file before; otherwise new picks are written after the old ones.

If you do not have Sac, you may create the **pick.dat** file manually, in free format, respecting the previous requests. See also the example below:

```
BAD.Z.data              NL                      1.274
BAD.Z.data              IPUO                    4.885
```

```
BAD.Z.data              FINI                    8.955
LSR.Z.data              NL                      2.814
LSR.Z.data              IPUO                    9.928
LSR.Z.data              FINI                    15.53
PLR.Z.data              NL                      1.631
PLR.Z.data              IPUO                    5.609
PLR.Z.data              FINI                    11.21
CSZ.Z.data              NL                      3.69
CSZ.Z.data              IPUO                    12.26
CSZ.Z.data              FINI                    17.15
CAE.Z.data              NL                      8.198
CAE.Z.data              IPUO                    20.32
CAE.Z.data              FINI                    26.37
UDI.Z.data              NL                      4.446
UDI.Z.data              IPUO                    10.2
UDI.Z.data              FINI                    15.9
RCL.Z.data              NL                      5.098
RCL.Z.data              IPUO                    12.75
RCL.Z.data              FINI                    18.19
```

Then run **cutker4_is** to cut simultaneously the data and all the corresponding basefunctions to the limits defined previously (**cutker4** reads **ker.tmp** and **pick.dat**). The cut is not sharp, but it is made by a hanging window (cosine taper) starting 8 point before and finishing 8 points after the selected window.

The truncated signals are written in the files **data.cut, base1.cut, base2.cut**, in the ASCII format and the order needed by the inversion program.

The program **cutker4** computes also the noise standard error σ to be used in the inversion, on the noise window defined by picking (see Paola Campus's thesis for formula) #ma per sigma mi pare che deve esere dato anche l'amplituda di rumore e di segnale). Usually this computed value appears to be lower than a visual estimation of the noise level. It will be used in the inversion as the absolute data standard error. It is relative value, associated to the damping chosen, it will give the resulting relative error on the model.

Finally, to make the next step (the inversion) easier to run, the inversion input files may be created automatically by the program **cutker4**. The unknown parameters, which are the contribution of the user's feeling to the inversion are asked:

- the number of triangles wished to modelize the source time function: it depends mainly on the source duration you want, that should be adapted to the earthquake size (indicative values: M=7: 15-24s; M=6: 5-8s; M=5: 2-3s; M=4: <1s). The width of the triangles is set automatically, so that it corresponds to the lowest period of the filtered signals (it is possible but not really useful to try to solve finer triangles). If you want to reduce the number of parameters of the inversion, you may use less but wider triangles, but the triangles half-width should always be a multiple of the sampling step.

- the observer shift between the base functions of structures 1 and 2 (in seconds): this is used as starting point to compute the shift of the kernels between the grid nodes, in order to interpolate the kinematics during relocation and structural optimization.

- the damping factor: it helps to condition better the problem (the conditionings limited to 1/damping) by eliminating the lowest eigenvalues of the system. But by increasing the damping, you limit the number of parameters that can be solved and give more importance to the initial model. Therefore you should use a small damping factor only if your data are of good quality and if you need to solve a big number of parameters.

If not, you should increase the damping, but the solution will be less precise. The recent indication of the eigenvalues of the problem should help you to define a compromise between the number of parameters that can be solved and the damping. Typical damping values range between 10^{-4} and 10^{-2}.

-the number of iteration wished: a minimum of 300 iterations is often required to obtain convergence of the inversion.

You may always adapt these input files afterwards, considering the format described in the next section.

1.3 Inversion

Program

invjan4_is

Input

***.cut** files, inversion input files

Output

moment.out, minimizz.out, brminim.out

If you did not make an automatic generation of the input files, or if you want to modify them, here is a description of the different input files needed. They are all formatted on the command file specified in invjan.cntl (for example invjan.auto). The structure of the command file is as follows:

1	`invauto.in1`	(input file containing the inversion options)
2	**`moment.out`**	(output file containing the moment rate functions)
3	**`data.cut`**	(input file containing the truncated data)
5	`invauto.in5`	(input file containing the iteration parameters)
6	`minimizz.out`	(output containing the inversion-minimization results)
7	**`brminim.out`**	(output containing only the iterative values of the model)
11,12,17,18	`tmp...`	(output temporary files, used during inversion)
19	**`base1.cut`**	(input file containing the truncated kernels for the structure 1)

20	**base2.cut**	(input file containing the truncated kernels for the structure 2)
21, 22, 23, 24, 25, 30	tmp...	(output temporary files, used during inversion)

The two extra input files, apart from the truncated data and kernels, are the unit 1 and 5 files;

Format of the **invauto.in1** input file:

1st line: number of stations, number of components.

following lines (one per station): flag (0;no, 1:yes) to use or not this station

next line: number of triangles to parameterize the source time function (depending on the maximum source duration you want)

next line: DTAU= half-width of the triangles in number of times the sampling rate (by default corresponds to 1/[2*sampling step*cutoff frequency] so that the width equals the lowest period), repeated for each Mij component (six times)

next line: number of points of the largest window, repeated two times

next line: number of points of the next power of two

next line: sampling step, maximum time shift between the extreme base functions for structures 1 and 2 (to perform intra-nodes interpolation), dt used for computing the width of triangle (=2*dt*DTAU), time shift

next line: damping ratio of the inversion, standard noise error σ (as defined by **cutker_is**)

next line: five flags (format 4A1: T: true, F: false) for: relative normalization of data and kernels (unit correction only), greater weight to the lowest signal, greater weight to the coda, inversion without isotropic component, save or not eigenfunctions

next line: minimum depth, maximum depth, sampling depth (as the initial grid, 3F10.3)

next line: extent of the north-south grid, sampling distance on the NS grid (idem)

next line: extent of the east-west grid, sampling distance on the EW grid(idem).

Example:

```
  3    1
1
1
1
7
2     2     2     2     2     2
293   293
512
.0500    -5.0    5.0     0.0500     0.100
0.1E-03  0.767E-06
TFFFF
```

```
10.000      20.000      2.000
51.160      51.160      0.000
5.820       5.820      0.000
```

* Format of the **invauto.in5** input file:

Some of those information should be redundant with the previous grid parameters.

1st line: title
2nd line: Dstart, Dstep, Dmin, Dmax where D is the depth
3rd line: NSstart, NSstep, NSmin, NSmax where NS is the latitude or NS shift.
4th line: EWstart, EWstep, EWmin, EWmax where EW is the longitude or EW shift
5th line: Ystart, Ystep, Ymin, Ymax where Y is the structure parameter (1-2)
following lines: Wstart, Wstep, Wmin, Wmaxwhere W is the weight of this station (0-1)
(Better if fixed. If the station is previously
indicated unused, weights should be 0).
one blank line:
next line: "SET ERR" and error parameter
next line: "SEEK" and number of iterations of the Monte-Carlo method.
next line: "MINIMIZE" and same number of the previous one.
next line: "CALL FCN" and parameter fixed to 3.
next line: "EXIT".

Example:

```
autogenerated input for invjan
          1DEPTH          3.502     0.556     1.000     6.004
         2LATITUDE       43.040     0.000    43.040    43.040
         3LONGITUDE      12.944     0.000    12.944    12.944
         4STRUCTURE       0.500     0.500     0.000     1.000
         5WEIGTHST1       1.000     0.000     1.000     1.000
         6WEIGTHST2       1.000     0.000     1.000     1.000
         7WEIGTHST3       1.000     0.000     1.000     1.000
         8WEIGTHST4       1.000     0.000     1.000     1.000
         9WEIGTHST5       1.000     0.000     1.000     1.000
        10WEIGTHST6       1.000     0.000     1.000     1.000
        11WEIGTHST7       1.000     0.000     1.000     1.000
        12WEIGTHST8       1.000     0.000     1.000     1.000
        13WEIGTHST9       1.000     0.000     1.000     1.000
        14WEIGTHST*       1.000     0.000     1.000     1.000
```

```
SET ERR   0.100
SEEK      400.
MINIMIZE  400.
CALL FCN  3.000
EXIT
```

N.B. The values of the grid indicated in those two files are purely indicative, since the inversion only thinks in terms of "nodes": just the number, the order and the relative position of the nodes must correspond to those for which the kernels have been computed. For example, you can say that the structural parameter will vary between 0 and 1, or between 1 and 2. You can also specify the surface grid in terms of absolute latitude/longitudes or in terms of relative NS/EW shifts in km around an initial epicenter (for the plane case). The values of the model will be returned in the same system of coordinates.

2 Inversion for the average mechanism and time function

The name of the PostScript files generated by the two following steps is argument**.ps** if you specified an argument, otherwise the files will be named **0.ps** by default.

2.1 Plot of the results of the first step

The inversion has several outputs to which is interesting to have a look.

The **minimizz.out** file contains all the information on the various step of the inversion. In particular, you may look inside if and how the inversion converged and note which are the final values of the parameters (longitude, latitude and depth) with their errors and correlations.

The **brminim.out** file report the successive values of X and Y during the exploration of the space of parameters. May be interesting to plot, from this file, the behaviour of the cost function (FCN) during the exploration.

The final output of the inversion is the **moment.out** file containing the final moment rate function and the recomputed synthetics. You will plot data and recalculated synthetics together, as well as the full and deviatoric moment rate functions, running the program:

plodatmij_is [argument]

Just answer to the questions. Most of time, you just have to press return if you want to choose the default answer indicated into brackets.

You will see on the left part the comparison between the original data (solid lines) and the final synthetics (dotted lines). The normalised correlation between the data and the recalculated synthetics

is indicated for each station and over all (-1: data and synthetics opposed, 0: non correlated, 1: perfect fit), in order to "quantify" the visual appreciation you might have of the fit.

On the right part, you will see the final moment rate functions (full tensor in solid lines, deviatoric tensor in dotted lines). This will help you to define which part of the moment rate functions you want to invert now (by default: the complete moment rates).

2.2 Factorisation

This is the second, iterative linear, inversion, to factorize the selected part of the moment rate functions M'ij(t) in average source mechanism Mij and source time function m'(t).

For this step you need the **moment.out** output of the main inversion and a **polarities** file containing some first motion polarities. If you don't have any polarities, you can run the inversion anyway, but you might get an indetermination on the solution: usually two almost opposite solutions are available, since you might take either the positive part of the source time function with the corresponding mechanism, or the negative part with an almost opposite mechanism. If at least one good polarity is available, it might be enough to choose the right sign. To create the **polarities** file run **pol_is.**

The **polarities** file is made of one line specifying the number of polarities, plus of one line per polarity, giving the azimuth (degrees clockwise from north) and incidence (degrees from the downwards vertical) of the ray leaving the source and its P wave polarity (on the scale of your choice: -10 to 10, -4 to 4, -1 to 1,....0). You can give different qualities to the different polarities in this way. That file is read in free format.

Example of file **polarities**:

```
1  polarities. Azimut dg, Incidence dg from vertical, Polarity (free scale)
     234.38     44.54        1.00
```

N.B.: If the instrument is dephasing at P-wave frequencies, the polarities should be picked on the signals corrected from the instrument, not yet filtered and sampled at the original sampling step. This can be done for example with Sac: for mechanical instruments or geophones, use "trans from general n 2 damping ??? free period ???" ; for poles/zeros files, use "trans from s polezero filename".

If you have not clear polarities, you can only choose the solution corresponding to the best RMS of the inversion, that could be misleading, (the L_2 norm of the difference between the original moment rates M'ij(t) and the factored ones Mij.m'(t)), knowing that the opposite solution might have a similar RMS too.

Then run **plojoi5_is** (argument) to perform the joint inversion of the moment rate functions.

Answer to the questions, just pressing Return to choose the default answers into brackets[].

One inversion is performed for each of the total, deviatoric and isotropic moment rates, and starting from the two main picks (usually one positive, one negative) of the two-side source time function (from -1 to 1). The source time function is searched iteratively, and for each trial, a simple analytical pure least-square give the best corresponding moment tensor. You can follow the progression of the iterative process on the screen, the source time function being printed for each improvement of the

RMS (in coded values for each point: 0,1,...9,*). At end, by default, you are requested to choose among the 2 best solutions. By default, the program will choose the two solutions giving the best correlation with the polarities, or the best RMS if no polarities are available.

If you want all the 10 best solutions to be written, you should run **plojoi5_is** with the proper option argument.

2.3 Looking in detail to the moment rate functions

If you want to see the evolution with time of the source mechanism, you can run the previous **plojoi5_is** inversion on a selected window of the moment rate functions (the first pick for example). In order to do it run: **plojoi5_is begin_time end_time**

INPAR short programs summary list

Programs are in **/XDST/bin** and must be run on **is01** server.

Run:	Input(s):	Output(s):
>**ker4_is** (arg)	**ker.inp** (arg) data, structures	**ker.tmp, picker**, data.*data*, kernels
>SAC> **m picker** (**J:** noise window), **P: begin**, **F: end**, **Q: exit**	data, kernels	plot-data+extreme-kernels.sgf cut limits in **pick.dat**
> **cutker4_is**	**ker.tmp, pick.dat**	**data.cut, base1.cut, base2.cut** (+inversion input files)
>**invjan4_is**	***.cut** files + inversion input file	**moment.out, minimizz.out,** **brminim.out**
> plodatmij_is	**moment.out, data.cut**	(0).ps
> plojoi5_is	**moment.out**	(0).ps, **moment.fac**