A Matlab GUI for use with ISOLA Fortran codes

User's Guide

by
Efthimios Sokos 1) and Jiri Zahradnik 2)

1) University of Patras, Seismological Laboratory
2) Charles University in Prague, Faculty of Mathematics and Physics

esokos@upatras.gr , jz@karel.troja.mff.cuni.cz

Ver. 2.5
June 2006

Contents

GUI Purpose

The ISOLA GUI has been created using the Matlab GUIDE tool. Its purpose is to help the user of ISOLA-FORTRAN part with data preprocessing, Green function preparation, waveform inversion and display of the results. In addition to the main tools a few utilities are also present that help the user to inspect the quality of the data using filters, convert to displacement, etc. In utilities there are also two other options that could help: Backup files and Create folder structure. The Backup files option saves in a user specified folder a number of files that are critical for running the inversion; in this way the user can keep track of various attempts during an experiment. The Create folder structure option creates the folders needed for ISOLA if these aren't present. Finally the import SAC and GCF (Guralp Compressed Format) options help converting SAC and GCF binary files in ISOLA ascii format (4 columns of ascii data per station, i.e.,Time, NS, EW and Z component).

Requirements

In order to use ISOLA GUI, you need to have the following software installed

- Matlab 6.5 or 7.0 (the software has not been tested using other versions of Matlab)
- Generic Mapping Tools (GMT, Wessel and Smith, 1991, 1998) (the software was tested using version 4.0 and 4.1, other versions should work also)
- M_Map, this is a freeware Matlab based package, it is used in map creation within Matlab, you can download from http://www2.ocgy.ubc.ca/~rich/map.html and you have to add it in the Matlab path.
- Ghostscript and GsView for Postscript files display http://www.cs.wisc.edu/~ghost/, and you have to add it in the system path.

.

Installation

Installation is easy; just unzip the zipped distribution file in a folder. This will create the necessary folders and extract the files needed for running the Matlab GUI. After this operation you should get a folder structure like the following:

\isola

   \gmtfiles

   \green

   \invert

   \pzfiles

   \polarity

A typical location would be the Matlab folder e.g. **c:\matlab\isola**.

Method

ISOLA is based on multiple point-source representation and iterative deconvolution method, similar to Kikuchi and Kanamori (1991) for teleseismic records, but here the full wavefield is considered, and Green's functions are calculated by the discrete wavenumber method of Bouchon (1981). Thus the method is applicable for regional and local events. Instrumentally corrected band-pass filtered velocity records are used. The code transforms velocity into displacement, inverts the displacement, and provides synthetic displacement. Seismogram $s(t)$ is approximated by a combination of elementary seismograms $e_i(t)$, $i=1,2,...6$, i.e. $s(t)= sum(a_i . e_i(t))$. The elementary seismograms correspond to basic focal mechanisms as follows:

```
i   strike  dip   rake
---------------------
1   0       90    0
2   270     90    -90
3   0       90    90
4   90      45    90
5   0       45    90
6   isotropic
```


Seismograms (3-component), from a set of stations, represent the data, d, while the coefficients a_i are the parameters to be found, m. The linear inverse problem d= G m is solved by the least-square method: m = (GT G)-1 GT d, where GT is G transposed and ()-1 is the inverse of the so-called system matrix (GT G). Eigenvalues of the system matrix (GT G) are calculated to understand how well-posed (or ill-posed) the inverse problem is. Ratio of min/max eigenvalue is reported. The lower the value, the worse the results are (although, formally, the quality of the fit may be good, e.g. variance reduction can be large). Question how the individual stations contribute to the well-posed inversion has no simple answer. It is to be carefully analyzed case by case.

Optionally, the weighted inversion is performed, with weights prescribed by user, different weights for different stations and components. As a rule, up-weighting records with small amplitudes, and down-weighting large amplitudes, is recommended. Another strategy is to up-weight stations whose seismograms appear to be the most sensitive with respect to small modifications of the problem.

The coefficients a_i are related to moment tensor Mpq (where p,q refer to geographic coordinates x (>0, N), y (>0, E), z (>0, up)):

Mxx=-a4+a6
Myy=-a5+a6
Mzz=a4+a5+a6
Mxy=Myx=a1
Mxz=Mzx=a2
Myz=Mzy=-a3

Moment tensor components in spherical coordinates r, t(=theta), p(=phi), used in Harvard catalogue, or in GMT, are given by:

Mtt=Mxx
Mpp=Myy
Mrr=Mzz
Mtp=-Mxy
Mrt=Mxz
Mrp=-Myz

Eigenvectors of the moment tensor provide strike, dip and rake. Eigenvalues provide scalar moment Mo and decomposition of the moment tensor into three parts: double-couple (DC), compensated linear vector dipole (CLVD), and voluminial (VOL).

Possible moment tensor (MT) inversion modes are as follows:
-full MT inversion: all six basic focal mechanisms, 1,2,...6; DC+CLVD+VOL
-deviatoric MT inversion: basic focal mechanisms 1,2,...5; DC+CLVD; VOL%=0;
-DC-constrained MT inversion: DC only; VOL%=CLVD%=0
-known and fixed DC moment tensor (only position and time is searched)

Opposed to strike, dip, rake and Mo, the DC% is strongly unstable. It closely relates with the fact that a large deviation from pure DC has, as a rule, a very small effect upon the seismograms. Indeed, it can be demonstrated by many examples in which synthetics for, say DC%=50 and DC%=100, cannot be distinguished by eye from each other, and quantitative measure of their fit with real data is nearly identical (e.g. variance reduction, introduced below, differs by 1%). Therefore, we can say that the DC% value provided by the routine moment tensor inversion is rarely of any physical meaning. The same applies for the volume percentage VOL%.

Optionally, the inversion can be constrained through the Lagrange multipliers to give a nearly 100% DC solution. The (non-linear) condition that the moment tensor must have zero-valued determinant is solved iteratively. It slows down the calculation and may create problems. There is no special advantage in this option, because the strike, dip and rake of the DC-constrained solution is usually very near to that of the (unconstrained) deviatoric MT inversion.

Thus the most recommended inversion mode is deviatoric inversion (VOL%=0) in which, however, the resulting DC% is considered with great caution.

It is assumed in the above explanation that the source position and time are known. This code allows their optimization through a grid search over set of trial source positions and time shifts, pre-defined by the user. The search seeks the trial source position and time for which the residual error of the least square method is minimized. It is equivalent to maximize correlation between the observed and synthetic seismograms. The concept of optimizing position and time is applicable to each point element (the so-called subevent) of the generally more complex source model. If the whole earthquake is represented (in the low-frequency range) by a single source, the optimum solution provided by the grid search corresponds to the *centroid* (the centroid position, centroid time and the centroid MT).

User can check the correlation values as a 2D function of the trial source position and time. The code has an interactive part in which user can use automatic search of the optimum correlation, or can make his preferred choice. The preferred choice may be the source position or time with a somewhat lower correlation, close to formal maximum, in which however some other conditions (constraints) are better satisfied. In this way user can apply, for example, his knowledge about the first-motion polarities, geologically deduced fault dip, various assumptions about rupture propagation, etc. The 2D correlation may also help in identifying poor resolution of some parameters, for example the down-dip source position.

Narrow band-pass inversions around period T can often provide two correlation maxima, in time t and t+T/2, characterized by rake and rake+180° respectively. This is just the case where even a single guaranteed first motion polarity may help to decide between proper time, t or t+T/2.

The code works iteratively. A point source contribution is called subevent. Using data code searches for subevent 1. Once subevent 1 is found, its contribution is subtracted from the data to get the so-called residual data. In the next step the residual data are processed in the same way as the original data during the first step. Subevent 2 is found, etc. Original data minus the last residual seismogram provide the resulting synthetic seismogram.

Optionally, the search of spatial and temporal distribution of subevents can be performed with fixed (100% DC) moment tensor. The user prescribes the known (or assumed) strike, dip and rake, and these values are kept fixed for all subevents. It can be recommended when the unconstrained moment tensors vary chaotically from one subevent to the other.

When the solution is represented by several subevents of different focal mechanism, the resulting scalar moment cannot be obtained as the sum of

the subevent scalar moments. That is why we also provide an output file where, subsequently, the subevent contributions are summed up as tensors and the scalar moment of the resulting (cumulative) tensor is shown.

Denoting data as **d**, and synthetics as **s**, the fit between them is measured by the so-called variance reduction varred= $1 - |\mathbf{d-s}|^2/|\mathbf{d}|^2$, where $|.|$ denotes the L2 norm, e.g., $|\mathbf{d}|^2 = suma(\mathbf{d_i}^2)$. The norm is calculated as a single number summing up over all time series, all components, and all stations. The optimum value of varred is 1. Wrong synthetics may give even varred < 0 (no physical meaning).

The variance reduction is determined repeatedly, "when building up" the synthetics, i.e. after adding every subevent. In other words, when s1, s2,... are synthetics corresponding to individual subevents, then first we calculate varred from **d** and **s**=**s1**, then from **d** and **s**=**s1**+**s2**, etc. As such, varred is a measure of the convergence of the iterative deconvolution process. Slow increase of varred with adding subevents indicates that additional subevents do not further improve the fit.

At this point it is useful to clarify how the variance reduction relates with the correlation. The relation is simple when all stations and all components are used, and all weights equal 1, and when we calculate only the first subevent. Take the observed seismogram as data **d**, and the first subevent as synthetics, **s**=**s1**. It is an internal property of the least-square minimization of $|\mathbf{d-s}|$, during the corresponding matrix operations, G, GT G, etc., that we automatically get the value of $\mathbf{c}^2$, where **c** is the correlation between **d** and **s**, without numerically calculating the correlation integral. (Speaking more precisely, we get sqrt($\mathbf{c}^2$)=abs(**c**), i.e. not the correlation **c**, but only its *absolute value*. Anyway, hereafter, we simply say only "correlation"). At the same time, it can be shown that in $\mathbf{c}^2 = 1 - |\mathbf{d-s}|^2/|\mathbf{d}|^2 = $ varred.

When we calculate already the second subevent, **s2**, the least-square method minimizes |**r**-**s2**|, where **r** is the residual seismogram defined as difference between the original data and the first subevent, **d**-**s1**. Therefore, from this least square run, we get automatically, and we report on the code output, (the absolute value of) the correlation between **r** and **s2**. At the same time, after adding the second subevent, the code calculates varred using **d** and **s**=**s1**+**s2**, i.e. varred= 1- $|$**d**-(**s1**+**s2**)$|^2/|$**d**$|^2$. Obviously, the varred value reported after the second subevent is not the same as the (squared) correlation between **r** and **s2**. Moreover, this varred is also not the same as the (squared) correlation between **d** and **s**=**s1**+**s2**, because we did not find the synthetic **s**=**s1**+**s2** by a single least-square minimization of $|$**d**-(**s1**+**s2**)$|$.

When non-unit weights are used, and/or some stations are deselected from a particular inversion run, the relation between correlation and variance reduction gets even more complicated.

Running the GUI

In order to run the code, within the Matlab command window, change directory (using the **cd** command) to the **\isola** folder and type **isola**. This will run the main GUI file and you should get a form like the following (Fig. 1).
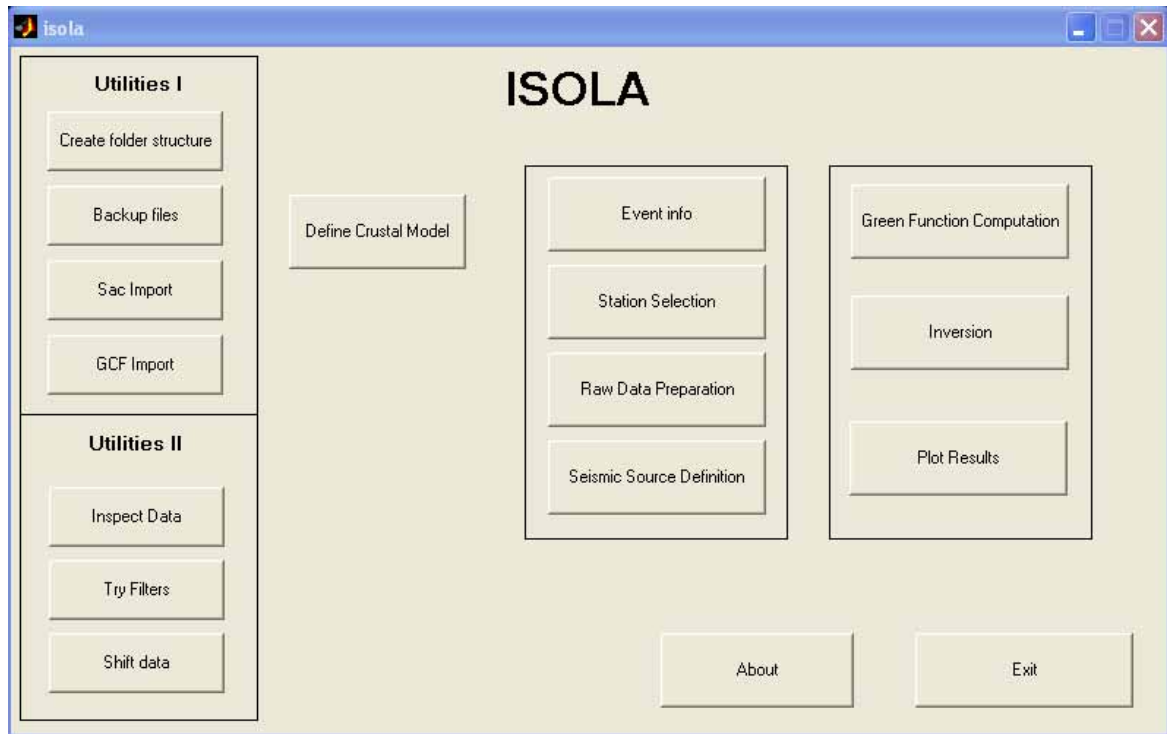
Figure 1. Main form of the ISOLA Matlab GUI.

Crustal Model

The first step in the inversion procedure is the definition of the crustal model. This is done by pressing the **Define Crustal Model** button in the main form, which will launch the form shown in Fig. 2. There is a fixed option for a maximum of 15 layers in the crustal model. For each layer, the user should give the depth of the layer top, the Vp, Vs velocities in km/sec, density in g/cm^3, Qp, Qs and, optionally, a title. In order to allow the program to understand the last layer, the user should type 999 in the last row as shown in Fig. 2.

There are buttons for saving the data (Save) in a text format file suitable for ISOLA Fortran, as well as reading it back in the form (Read). The latter is useful for editing a previously saved model. The plotting option (Plot) generates graphs of Vp and Vs versus depth. Automatic calculation of Vs

according to a specified Vp/Vs value is also possible, as well as the calculation of density according to formula density (g/ cm^3) =1.7+0.2*Vp (km/s).

Pressing Save the crustal model file that was created is saved under the **GREEN** and **POLARITY** folders.
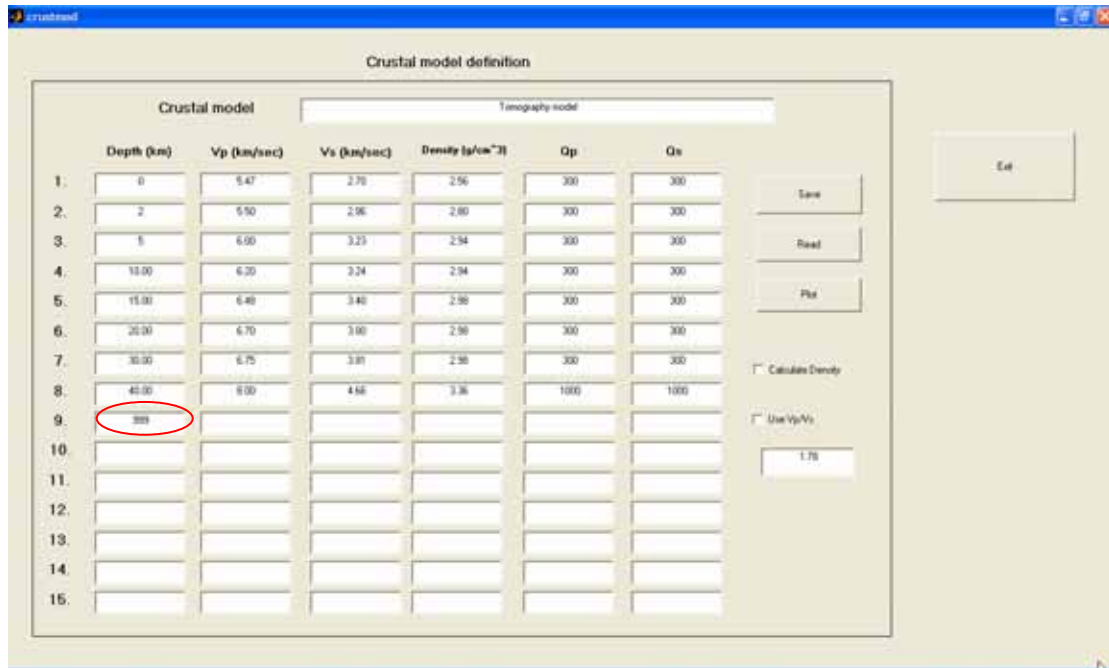

Figure 2. The Crustal model definition form.

The native format of the crustal model file in ISOLA Fortran is the following:

```
Crustal model                         Tomography model
number of layers
   8
Parameters of the layers
depth of layer top(km)    Vp(km/s)    Vs(km/s)    Rho(g/cm**3)    Qp      Qs
        0.0               5.47        2.700       2.560          300     300
        2.0               5.50        2.860       2.800          300     300
        5.0               6.00        3.230       2.940          300     300
       10.0               6.20        3.240       2.940          300     300
       15.0               6.48        3.400       2.980          300     300
       20.0               6.70        3.800       2.980          300     300
       30.0               6.75        3.810       2.980          300     300
       40.0               8.00        4.660       3.360          1000    1000
******************************************************************************
```

After finishing with the crustal model it is time for the user to give some information related to the event he wants to analyze; this is done by pressing **Event Info** in the main form (Fig. 3)

Event Info

Figure 3. The Event Info form.

In this form the user provides information **that will be used in all the subsequent steps** of the inversion, like the event Lat, Lon, Depth, Magnitude, Date (the last two are included just for information), Origin Time of the event, Start Time of the data file (seismograms), Sampling Frequency, and, finally, the Time Length of the data that will be used for the inversion. The time length should satisfy some specific conditions. For user's convenience, there are 3 predefined time length intervals that should cover most applications at local and regional distances. After pressing the update button the GUI writes or updates the appropriate files e.g. **event.isl,**

rawinfo.isl, duration.isl; those are ISOLA (*.isl) related text files that hold the above information and are stored in **ISOLA** folder.

Next step is the selection of the stations that will be used for the inversion (Fig. 4).
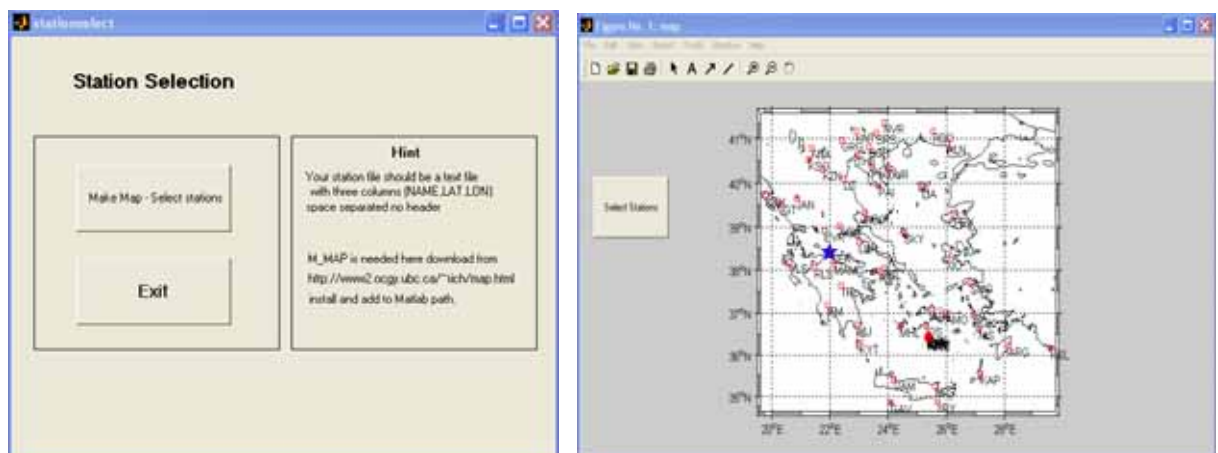
Station Selection

Figure 4. The Station Selection form.

Options in this form are simple; the user should press **Make Map - Select stations** in order to select his (previously prepared) text file containing the geographical coordinates of his network. The name and location of the file is arbitrary, the user will be guided to browse. The file should be an ascii file with three columns, **Station Name, Latitude, Longitude,** separated by spaces. The Station Name should have 3 characters. If it is longer a warning is issued and the station name is reduced to the first three characters. Pressing Make Map – Select stations (and assuming that M_MAP is properly installed) a map of the station distribution is created, and the user can select the stations he wants to include in the inversion. Selection of the stations is done using the left mouse button; the last station should be picked using the right mouse button. After pressing the right mouse button a warning window comes up with the total number of the selected stations. *Later during the inversion stage, if necessary, the user can again "deselect" some stations, i.e.*

to remove them from the inversion process, but once the station was selected the corresponding seismograms must be provided.

After finishing the selection of the stations the user should just press Exit on the Station Selection form. Doing this the GUI will create files like **stations.isl, station.dat, source.dat** (in **GREEN** folder) and **allstat.dat** in **INVERT** folder.

Raw Data Preparation

Next step is the **Raw Data Preparation** (Fig. 5), maybe the most important option of the GUI, since it processes the seismograms (doing the instrument correction, origin time alignment, resampling) and produces data files ready for the inversion. The data, i.e. the uncorrected seismograms, must be available in 4-column text files (with the *strictly required* column order, i.e., Time in seconds , NS, EW, Z in counts, separated with spaces), one file per station. The filenames are not completely arbitrary; for example, for the LTK station the filename must start with the letters LTK, where LTK is the station name used in the Station Selection. The remaining part of the filename (including the extension) is already arbitrary. The location of the file is also arbitrary as well as the extension and the part of the filename after the first three characters, the user will be guided to browse.

The user should make available also the files containing the parameters that will be used for the instrument correction. An example of such a "pole and zero file" follows. A few pole and zero files for several instruments are included with the software distribution also.

A0	Comment
133310	A0 value
count-->m/sec	Comment
3.33e-10	conversion counts to m/sec
zeroes (Re and Im in rad/sec)	Comment

```
3                                     Number of zeroes
0.0 0.0                               Zero
0.0 0.0                               Zero
51.5 0.0                              Zero
poles (Re and Im in rad/sec)          Comment
5                                     Number of Poles
-272    218                           Pole
-272   -218                           Pole
56.5    0                             Pole
-0.1111  0.1111                       Pole
-0.1111 -0.1111                       Pole
```

It is *strictly required* that the file describes the instrument transfer function for input *velocity*. The filename must be a combination of the Station Name and the "BH*.pz" string; for example, for the station named LTK and the EW component, it must be LTKBHE.pz, thus three pole and zero files per station, are needed. The location of the "pz-files" is not arbitrary, the user must place them in the **PZFILES** folder.

Caution must be devoted to the appropriate units. ISOLA needs the poles and zeros in rad/sec, and the constant of the transfer function A0 (guaranteeing that the "plateau"of the amplitude response equals to 1) should be consistent with that. In case that the user has the zeros and poles in Hz, they all must be multiplied by 2π, and, correspondingly, A0 must be multiplied by $(2\pi)^{NP-NZ}$, where NP and NZ is the number of the poles and zeros, respectively.
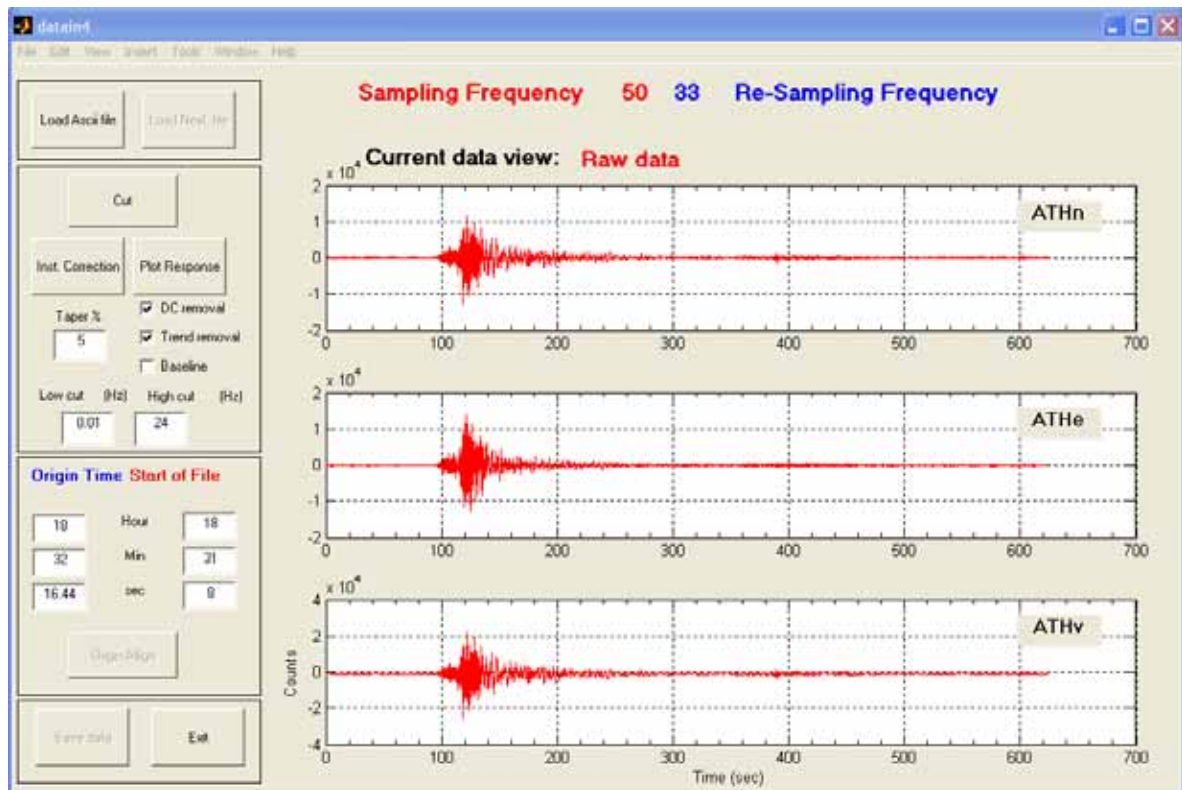
Figure 5. The Data Preparation form.

Pressing **Load Ascii file**, a standard open file dialog appears and the user selects a data file. When the file is read, the sampling frequency is displayed (in red) at the top of the form (Fig. 5), side by side with the resampling frequency of the data in blue. The resampling is made automatically, based on specific methodical requirements and the data input from the previous steps.

After data loading the Cut and Instrument correction buttons are enabled. If not all of the data is needed using the **Cut** button the last part of it can be deleted. By pressing the **Inst. Correction** button the program searches in **PZFILES** folder for the appropriate pole and zero file and then performs the instrument correction. During this operation a filter can be applied (Butterworth, 2nd order, zero-phase) along with the DC and trend removal, all using the built-in Matlab functions. The baseline option is an alternative of the DC removal, in which the baseline is determined from the beginning of the seismogram only.

Next step is the time alignment of the seismogram, i.e., irrespectively of its Start Time, since now it will begin at the earthquake Origin Time; this is done by pressing the **Origin Align** button which becomes active after the instrument correction.

Finally the user should press the **Save Data** button in order to save the files in **INVERT** folder. The files are saved as text files (4 columns, Time in sec, NS, EW, Z in m/sec), one per a station, always containing just 8192 points of the appropriately resampled data. The GUI automatically creates the proper filename for the corrected resampled data, which for station LTK would be LTKraw.dat. This filename *must not* be changed by the user.

Trial Source Definition

The final data that the user should prepare is the trial source location and this is done by pressing the **Seismic Source Definition** button in the ISOLA main form. The user has to select **Single Source or Multiple Source** definition.

In the case of single source, we can vary the depth only, thus the options in the form are Starting depth, Depth step and No of Sources, the explanation of which is given in Fig. 6. In this case, all the trial positions are below the epicenter. By pressing **Exit** the program writes the appropriate files in **GREEN** and **GMTFILES** folders. These files are **src*.dat** files in **GREEN** and **sources.gmt** in **GMTFILES**. The **sources.gmt** file is formatted for GMT (psxy) plotting.

It is assumed that the Single Source option will be always used to retrieve (at each trial depth) just a single subevent, that one which predominates in the low-frequency inversion. The grid search over the depth then serves to

provide the optimum source position characterized by the best fit between the waveform data and synthetics. Thus we get a first approximation of the centroid. By "the first approximation" we mean that (opposed to assumption of this option) the centroid is not necessarily below the epicenter.
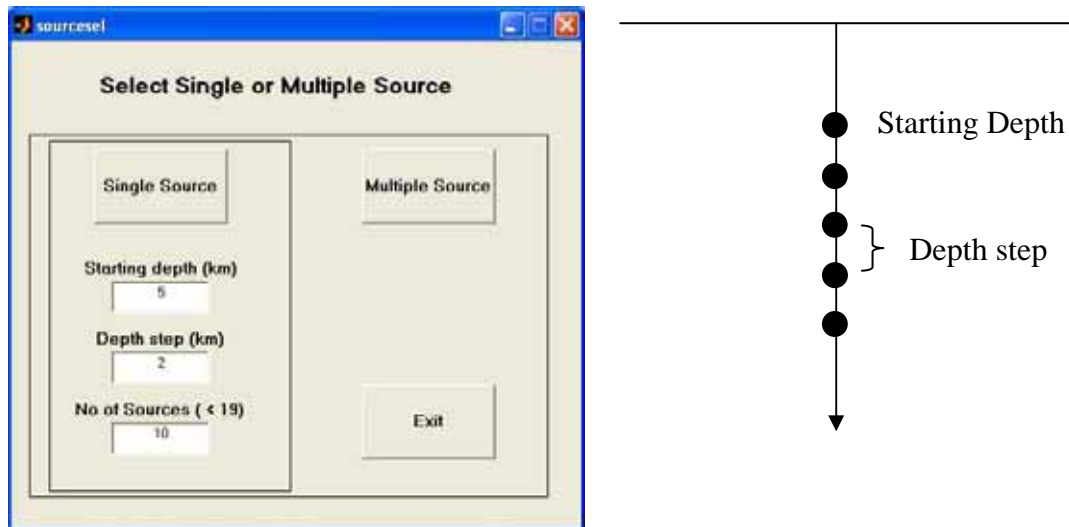

Figure 6. The Single Source definition form.

If the user wishes to use the Multiple Source inversion a new form appears (Fig. 7). This inversion has three options: sources along the strike of the fault, sources along the dip, or sources on a plane. The user has to select the fault strike and dip, the number of the trial sources he wants to try, their spacing, and the first source position. After pressing the **Calculate** button the program writes the source files in **GREEN** folder and produces a map with the trial source distribution.

The Multiple Source option has two different applications. The first one is to prepare trial positions for the grid search of the centroid, inspecting usually positions in the vicinity of the optimum depth found with the Single Source inversion. In this application we do design a Multiple Source stencil, but make inversion in such low frequencies that just a single subevent represents the whole source. The second possible application of the Multiple Source option is to design the trial source positions for inversion of the earthquake into a complex source model, consisting of more than a single subevent. Obviously,

as a rule, the latter is achieved by the inversion including higher frequencies and using a denser grid of the trial positions than of the first application.

This manual describes the code options. It cannot describe all possible strategies how to use the code. For example, the user may use the Multiple Source option in the beginning of his study to find the centroid (the first application), which might need trial positions relatively far from the hypocenter, and with a relatively large grid spacing. Then he may wish to repeat the same procedure with finer grid spacing to find the exact position of the centroid. Finally, he can try the application of the second type, i.e. to resolve several subevents. The latter may often fail, due to lacking data, or simply because the earthquake has no clear complexity, then his result remains to be just the precise centroid solution (with no additional details). This discussion explains why it is nearly impossible to give an "always working hint" of where and how densely the trial source positions should be designed.
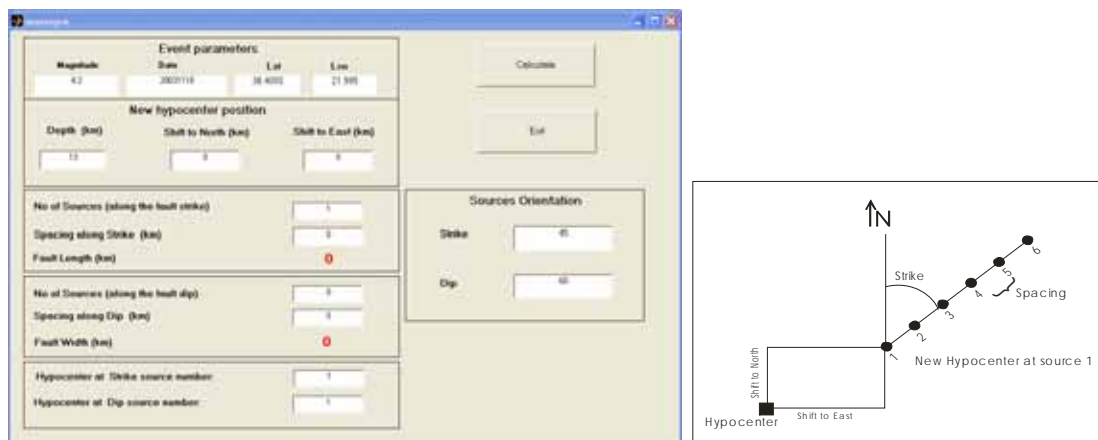
Figure 7. The Multiple Source definition form.

Green Function Computation

After finishing with the trial source definition the inversion can start and the first step is to prepare the Green functions. This is done by pressing **Green**

Function Computation in the ISOLA main form. The **greenpre.m** is running (Fig. 8) and, for simplicity, the user has only one option to fill in, this is the maximum frequency of the Green functions to be computed (fmax). (Obviously, no higher frequency than this one can be requested later in the inversion.) Pressing **Run** the necessary files are created in **GREEN** folder and the executable Fortran codes are called from Matlab: **gr_xyz.exe** and **elemse.exe**, for the Green function and elementary seismograms, respectively. The execution is monitored through the Command Window. The user should watch the execution of the Fortran programs and press OK in the "Copy files in invert folder" dialog, when the procedure ends. After that, copying starts and a message is issued when the files are copied in **INVERT** folder and the user can proceed to the inversion.

For simplicity, this version of the code does not specify any particular slip-rate time function. It means that the assumed slip rate is that given by the Green function calculation. By definition, the Green function is response of the medium to the impulsive (Dirac delta function) source, but note that we work in a limited frequency band. Therefore, working with the band-limited delta function, the actual slip-rate function applied in the code has duration approximately equal to dur=1/fmax, where fmax is the maximum frequency considered in the inversion. The only correct application of this code is for *subevents* whose true durations are shorter than dur.

Figure 8. The Green function computation form.

Inversion

The last step starts by pressing **Inversion** in the ISOLA main form; it runs **invert.m** that produces the following form (Fig. 9).

Figure 9. The Inversion form.

Before running the inversion, the use has to select his options regarding the **Filter**, **Type of Inversion**, **Number of Subevents** to be retrieved, and parameters of the **Time Search** (temporal grid search of the subevents). The same filter is automatically used for the data and synthetics.

Then the user may press the **Compute weights** button which automatically calculates weights to be applied in the inversion. (If not pressing this button, unit weights will be used.) The automatic weights are inversely proportional to peak values of the individual components in the used frequency band. *In fact what are computed by the **Compute weights** function are just the peak*

values for the traces and latter isola.exe takes the 1/peak value as the weight.
Finally, pressing **Run,** the main executable Fortran code **isola.exe** is running in the Command window (Fig. 10). When the code makes a pause it prompts the user to accept the automatic selection of the trial source position (that of maximum correlation), or to select some other user's preferred trial source position. To simplify the decision, the user can plot a correlation diagram using the **Plot Correlation** option (Fig. 11); GMT and Gsview is needed. In this 2D diagram the correlation value and focal mechanism are plotted against the trial source position and time shift for the subevent under study.

Pressing the **Deselect Stations** button, the **allstat.dat** file opens using **notepad,** and the user can remove a station from the inversion by editing the second column of the file (1=use the station, 0=not use it). The columns 3, 4, 5 contain the above mentioned peak displacement values for the NS, EW, Z component, respectively. The user can manually edit them, thus making his preferred specification of weights. *Note that the larger the number in allstat.dat, the lower is the weight.*

See also the Method chapter to learn more about the weights and correlation.

Example of the allstat.dat file.

```
GUR   1 6.057019e-005 2.130505e-004 3.082369e-004
LTK   1 6.179501e-005 2.366762e-004 2.285469e-004
DID   1 1.139388e-004 1.451932e-004 1.765126e-004
PYL   1 1.145303e-004 1.020548e-004 1.030644e-004
```

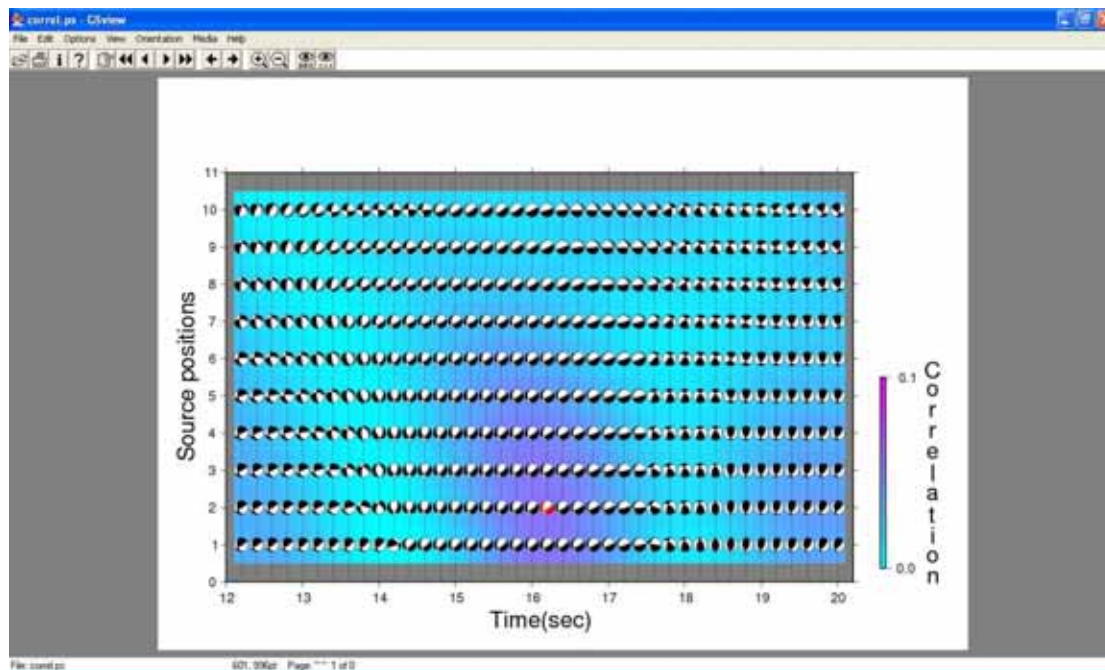**Figure 10.** Example of the inversion run.

Figure 11. Example of the correlation plot.

Plotting

The inversion code **isola.exe** creates a number of output files that can be plotted after the inversion run. This is done by pressing **Plot Results** button of the main ISOLA form. The **plotres.m** is called and produces the form of Fig. 12. There are options for plotting Synthetic versus Real data, the MT results, as well as for comparing the solution with first motion polarities.

The MT (moment tensor) plotting is done using GMT, and the user has the option for a single source summary plot (Fig. 13) a plot of Correlation versus Source number and DC% (Fig. 14) or multiple source results plot in map view, pressing Plot Sources. In all the plots (except the one in the single source summary) the fault plane solutions are displayed by beach balls without showing the non-DC component, because, as explained above, the non-DC part is usually unstable.

Note that the single-source plotting option applies also for the inversion which resulted in several subevents. The code prompts the user to specify the subevent he wishes to plot in Fig. 13 and Fig. 14.

The variance reduction given in the Single Source plot in Fig. 13 then refers to the situation after adding the chosen subevent (for example, when plotting subevent 2, the varred value applies for the fit between data and the synthetics composed of subevent 1 and subevent 2). The fit is calculated only from stations used in the inversion (the deselected stations are not considered).
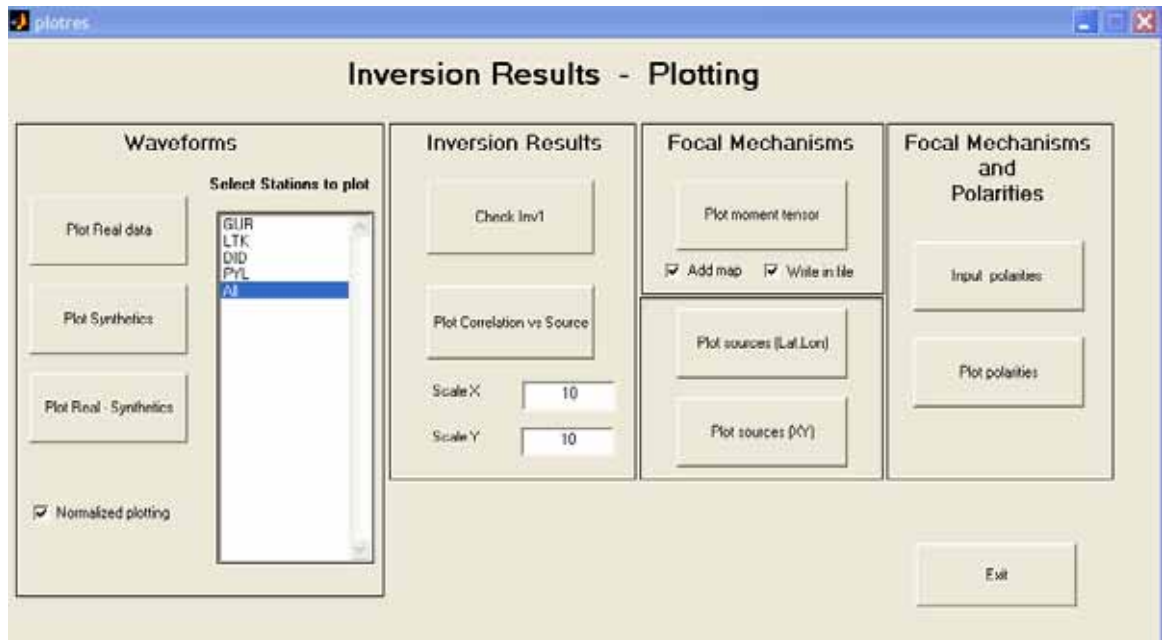
Figure 12. The Plotting form

Pressing the **Plot Sources** button the **plsource.m** opens. This creates the form of Fig. 15. The user has a lot of options in order to create a map with the results (Fig. 16). Most of the options are self explanatory, but some knowledge of GMT is needed. The most critical are the Beach ball shift and Beach ball X shift to prevent overlapping of the beach balls. The plot formally includes all the calculated subevents. Lot of care is needed to find out which of them are physically justified. This complicated physical problem is not solved here in the manual. See, for example, Zahradnik et al. (2005).
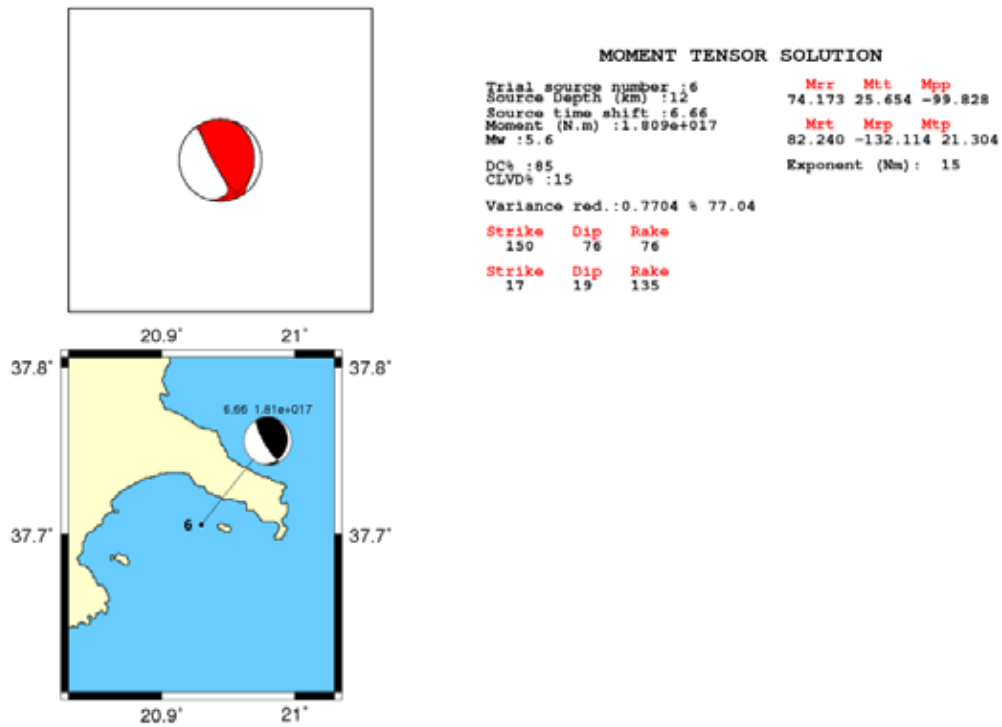
MOMENT TENSOR SOLUTION

Trial source number :6
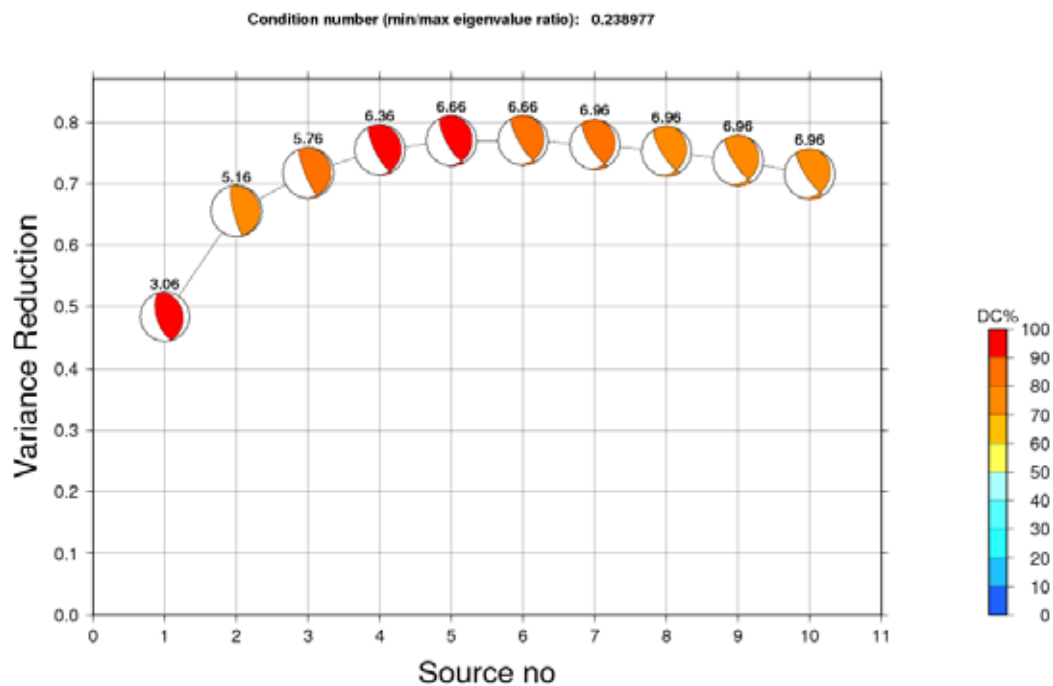Source Depth (km) :12
Source time shift :6.66
Moment (N.m) :1.809e+017
Mw :5.6

DC% :85
CLVD% :15

Variance red.:0.7704 % 77.04

Strike	Dip	Rake
150	76	76

Strike	Dip	Rake
17	19	135

Mrr	Mtt	Mpp
74.173	25.654	−99.828

Mrt	Mrp	Mtp
82.240	−132.114	21.304

Exponent (Nm): 15

Figure 13. The Single source summary plot

Figure 14. The Correlation vs Source number plot. By the source number we mean the sequential number of the trial source position. The values above the beach balls give the time shift.
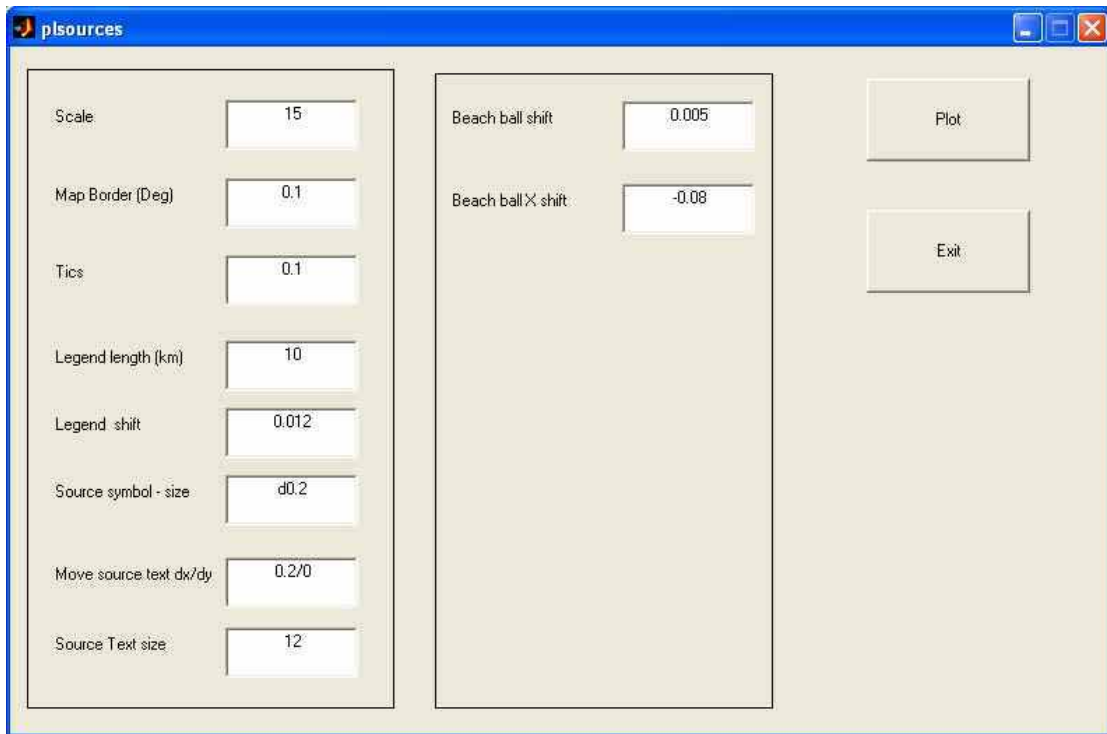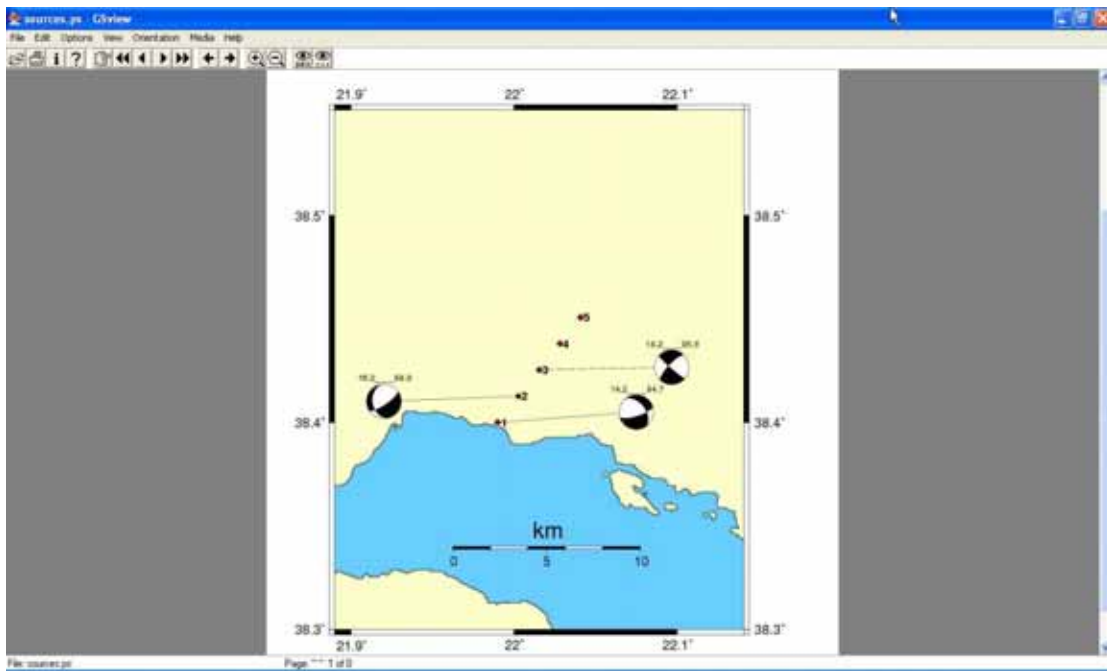
Figure 15. The multiple source plotting form .

Figure 16. Plot of the multiple source inversion results.

Full numerical results of the inversion are available in the files **inv1.dat, inv2.dat, inv2c.dat, inv3.dat, inv4.dat**, located in the folder **INVERT.** For a detailed explanation, the readme files of the ISOLA Fortran guide should be consulted (see below). Most often, simpler output (that provided in the Plotting form) is sufficient.

Finally the user has the option to plot the MT inversion results along with selected polarities (Fig. 17) in order to check validity of the inversion. A few files need to be updated for this option. Namely the **station.dat** file has to be manually edited to assign the observed polarities per stations. This can be done within the GUI by pressing **Input Polarities** after which the station.dat file opens in an editor. The other two files that need to be created are **onemech.dat** and **moremech.dat** that the user has to create in the **POLARITY** folder. The onemech.dat contains data to be shown in red (e.g. the solution from ISOLA), while moremech.dat can include, for comparison, several other solutions (e.g. those from several agencies). The polarity plotting is still under heavy development. Present version needs some knowledge of the ISOLA Fortran manual. In particular, the mentioned files should share the same format as that of the output file INV2.DAT.
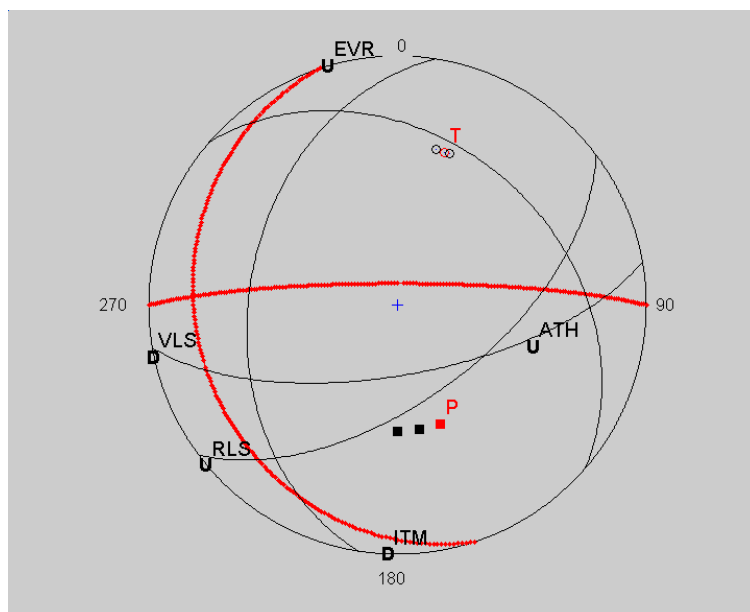
Figure 17. Plot of polarities with inversion results.

Moremech.dat example (<u>a</u> <u>space</u>-sepated text file; only columns 4,5,6 = strike,dip,rake are important here)

```
4  6.66 .185761E+18   151.  76. 77. 13. 19. 131. 251. 61. 44. 32.   88.9   .7718E+00
1  3.96 .560486E+17   201. 50. 144. 316. 63. 46. 76. 83. 175. 39.   68.4   .8426E+00
```

Utilities I and II

Besides the inversion related tools, a number of options is provided for assisting the user in converting his binary data to ISOLA ascii format and inspecting the data quality. Tools are available for importing data in two formats: SAC and GCF (Guralp Compressed Format). The tools are quite easy to handle; just read the files, *component by component*, and then save the file (Fig. 18). If the three components at a station don't share the same starting time, or don't have the same length, options for their trimming are provided.
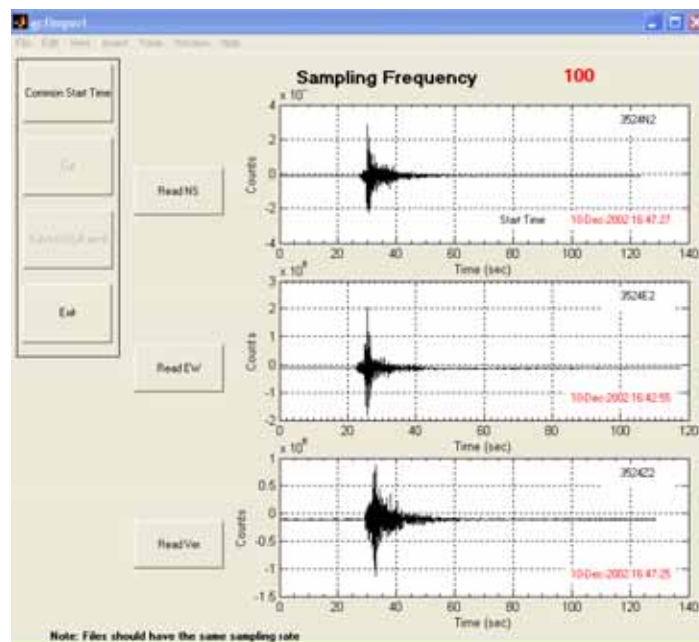

Figure 18. The GCF file import form.

The tools for importing data are based on free Matlab routines from the Web (see acknowledgment part of the manual). The user can use them as an example to easily add his preferred format conversion to the present GUI.

After converting the data to ISOLA format the user has the option to inspect the data quality either using the Inspect Data tool or the Try Filters tool. Data can be filtered in various frequency bands in order to define the proper band for inversion (i.e. the band with a good signal-to-noise ratio). Data can be

also inspected as the displacement. It is extremely important to inspect the displacement without filtration and instrument correction since it allows detection of various disturbances that are often *hidden* in the velocity records and/or in the displacement band-passed records (Zahradnik and Plesinger, 2005).

Finally the data can be shifted using the Shift data option. This option may be used (with great caution) when, for example, we have an indication that the employed crustal model is not appropriate at some station as regards the arrival time.

Running the Fortran codes without Matlab

ISOLA can be used also without the Matlab GUI. In case like that the user should study the readme files that are provided with the Fortran files distribution at http://???

For compactness of this Guide all the readme files of the ISOLA Fortran version are copied below. Users of the Matlab GUI can skip this part.

```
Instruction:
    Go trough directories and follow readme files there.
Start in DATA, continue with SOURSTAT, GREEN, INVERT,
POLARITY (and, optionally, also SYNT). Use sample (input
and output) data files to learn the package. The sample
files do not contain *.exe files because of their length
and because users should be able to compile all *.for
codes themselves. In case of problems, ask the author to
send the *.exe files. The example is highly simplified to
allow quick test run. For more details, see the
application referenced below.
```

Acknowledgments

Special thanks go to Petra Adamova for extensive testing of ISOLA GUI. The following m-files are part of other programs that we found useful and added to the GUI.

- rsac.m, lh.m by Michael Thorne,

- readgcffile.m by M. McGowan, Guralp Systems Ltd.,
- Bandpass.m Coral distribution.