SItomo – User Guide

Content

1. Requirements

SItomo currently works on Linux and Windows (64-bit). MATLAB 2020b or newer is needed. If you have the Parallel Processing Toolbox, the program will work considerably faster. It is recommended to run multiple independent iterations simultaneously or successively for the same data to allow a statistical analysis of the results (access to a cluster might be appropriate).

2. License

If the use or alteration of this program leads to a scientific publication, please consider citing the accompanying publications:

Link, Frederik & Long, Maureen D. (2024). *SItomo – A toolbox for Splitting Intensity Tomography and Application in the Eastern Alps.* Journal of Geodynamics, 159, 102018, https://doi.org/10.1016/j.jog.2024.102018.

3. Installation

To install the SItomo toolbox, simply mirror or download (and unpack) the github folder into your desired location (e.g., **~/programs/SItomo**). To check for the current version available on your computer, navigate to the program folder (e.g., **~/programs/SItomo**) and run the command start_SItomo including the option 'version':

```
>> start_SItomo('version')
Current version of SItomo: 1.0.0
```

To check if an update is available, you can use the option 'check', which checks for the most recent version on github:

```
>> start_SItomo('check')
Current version of SItomo: 1.0.0
Checking online for recent version. Please wait...
Release SItomo 1.0.0 seismolink/sitomo · GitHub
Current version is the most recent.
```

If a new version is available, the program will advice you to visit github to download the newest version and replace the old structure on your computer:

```
>> start_SItomo('check')
Current version of SItomo: 0.0.0
Checking online for recent version. Please wait...
Release SItomo 1.0.0 · seismolink/sitomo · GitHub
It is likely that a new version is available.
Visit https://github.com/obspy/obspy/ to download the newest version.
```

4. Introduction

SItomo is a program for the tomographic inversion of teleseismic shear-wave splitting measured by Splitting Intensity. If you are unfamiliar with shear-wave splitting and inversion procedures, please see the accompanying paper for further information (Link & Long, 2024). See Below for a workflow of the program SItomo.

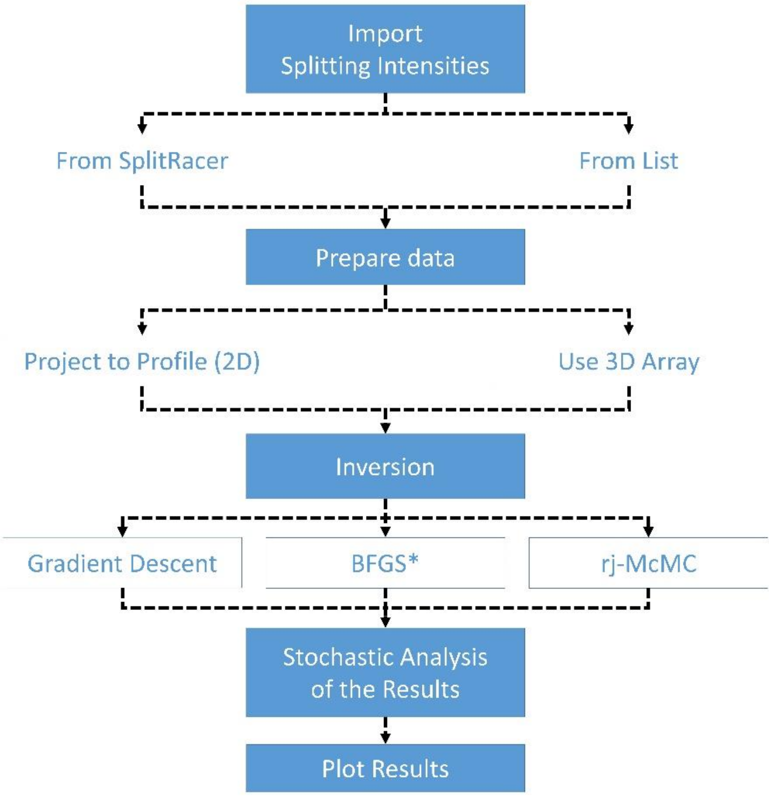

Figure 1: Workflow (*For the BFGS-Algorithm the Optimization Toolbox within Matlab is required)

The following chapters will walk you through every processing and analysis step and its available options.

5. Basic Usage

To start SItomo simply run the function `start_SItomo` including an option to indicate the step to execute in the MATLAB Command Window.

```
>> start_SItomo([option])
```

or

```
>> start_SItomo [option]
```

If you run this command without an additional option the Program provides you a list with all possible options and a short description.

Output:

```
SItomo is a program for inverting Splitting Intensity measurements from
teleseismic shear-wave splitting analysis for 2D and 3D structure.
------------------------------------------------------------------------
Basic Usage of SItomo:
start_SItomo(['option'])

The following options are available:
version         - Print the current version of the program.
check           - Check if there is a newer version available.
synthetics      - Produce synthetic data (experimental).
import2D        - import SI-data to constuct 2D Profile.
import3D        - import SI-data as 3D array.
invert2Dgd      - invert Splitting Intensities with Gradient Descent in 2D.
invert2Dbfgs    - invert Splitting Intensities with BFGS Algorithm in 2D.
invert2Dmcmc    - invert Splitting Intensities with rj-McMC Algorithm in 2D.
invert3Dgd      - invert Splitting Intensities with Gradient Descent in 3D.
invert3Dbfgs    - invert Splitting Intensities with BFGS Algorithm in 3D.
invert3Dmcmc    - invert Splitting Intensities with rj-McMC Algorithm in 3D.
plot2D          - Process and plot results from 2D inversion.
plot3D          - Process and plot results from 2D inversion.
remote          - Run inversion with predefined settings.
```

6. Project Preparation and Data Import

First prepare a work folder, where the imported data, processed products and results will be stored (e.g., **~/project_name/**).

A data set is required to be inverted for an anisotropic structure. A new data set can be imported from SplitRacer or can be prepared manually as a list. In the latter case, it is recommended to store the list in the work folder (or a project folder that contains the work folder).

If you already have a data set make sure to prepare it for usage in SItomo:

- Make a work folder for your project called **~/{project_name}/**
- Produce a .csv file with | separated columns containing the following information:

STAT	NET	LAT [°]	LON [°]	SI	ERR	T [s]	DIST [°]	DEPTH [km]	BAZ [°]
D001	ZS	45.89	10.22	0.83	0.34	11.2	110.1	10	94.3

If you wish to import the data, simply add the option `'import'` to the function call:

```
>> start_SItomo('import')
```

You will be asked for the path to the previously created work folder (e.g., ~/project_name/):

```
Define working directory for SI tomography: ~/project_name/
```

You can decide, if you wish to import data from a previous analysis with SplitRacer or if you would like to import from a data list as described above. If you want to import from a SplitRacer structure enter 1 or 0 if you prepared a list:

```
Import from SplitRacer? 1=yes/0=no :
```

The import will create a folder **input** in your work folder (e.g., **~/project_name/input/**) and a matlab structure containing the relevant data for the information named **SIdata.mat**.

6.1 Import from SplitRacer

Let's first assume you did your previous analysis with SplitRacer and wish to import the results for further analysis in the tomographic inversion SplitRacer structure.

```
Import from SplitRacer? 1=yes/0=no : 1
```

In that case you have to provide the path to the SplitRacer work folder, where the results can be found (e.g., ~/SRwork/):

```
Define working directory from SplitRacer:  ~/SRwork/
```

It is possible to include multiple sets of work folders from SplitRacer if the splitting analysis was done in separated steps. Usually only one work folder is used. Select 0 in that case:

```
Add another folder structure? 1=yes,0=no :    0
```

All folders are scanned for available results and for each work folder the user can decide, which settings should be used for the tomography:

```
The following settings are available.
1 ~/SRwork/results/Filter_4-50s_SNR_2.5_tw_50/saved_settings.txt
2 ~/SRwork/results/Filter_4-25s_SNR_1.8_tw_50/saved_settings.txt
Choose a setting file for analysis: 1
```

SItomo will prompt the absolute path of the settings file that has been selected:

```
Chosen setting is: ~/SRwork/results/Filter_4-50s_SNR_2.5_tw_50/saved_settings.txt
```

The user can decide (similar to an analysis within SplitRacer) to only use good and null measurements or all measurements:

```
Use all (1) or only null and good measurements (0)?
```

Before the import starts the user can decide to re-evaluate the splitting intensities with the method introduced in the accompanying paper of SItomo (Link & Long 2024), or use the splitting intensities already calculated in SplitRacer. While it is much faster to simply import the previous calculations, there was no information evaluated and stored in the previous analysis of SplitRacer and the period is set to 10s, which is a rough average for core-mantle converted phases. If the re-evaluation is selected, the splitting intensities and periods are evaluated from the pre-cut traces in the SplitRacer structure:

```
Re-evaluate (1) or read pre-calculated splitting intensities (0)? 1
```

Now a data structure is prepared from the SplitRacer structure. The program prompts the station names, from which it is currently reading the data. If the re-evaluation was selected, the program prompts also messages, when it starts and finishes an evaluation:

```
Doing D001 ...
Starting analysis
Analysis done
```

6.2 Import from a manually created list

If you prepared your data as a list for the import you have to select 0 in the previous input:

```
Import from SplitRacer? 1=yes/0=no : 0
```

In that case you have to provide the absolute path to the prepared list (e.g., ~/SRwork/input/project.csv):

```
Provide absolute path to SI file:  ~/SRwork/input/project.csv
```

The input structure is created from the file with no further prompts.

7. Prepare data for Inversion

The user can decide to prepare the data for a 2D or 3D inversion depending on the shape of the network and appropriate configuration. If the network is a profile a 2D inversion is suitable, for an even coverage of an area a 3D inversion might be appropriate. Be aware, that for the tomographic inversion generally a dense station spacing is required between 5 and 15 km.

7.1 Preparation for 2D-Inversion

For a 2D-inversion the data can be prepared using the option `'prep2D'`. In the following, the user will be asked to define a profile on which the data will be projected for further analysis.

```
 >>start_SItomo('prep2D')
```

As for every step, the user is asked to provide the location of the work folder (e.g., ~/project_name):

```
Define working directory for SI tomography: ~/project_name/
```

The user is asked to provide the maximum distance a station may be away from the final selected 2D profile. A value up to 0.6 degrees is recommended.

```
 Maximum distance of station to profile in degree (e.g., 0.6): 0.6
```

This will produce a station map with an event density map in the background. This graphic is supposed to support the user for his decision on the trend of the profile. In this graphic, the user will select by point and click the start and end location of the profile.

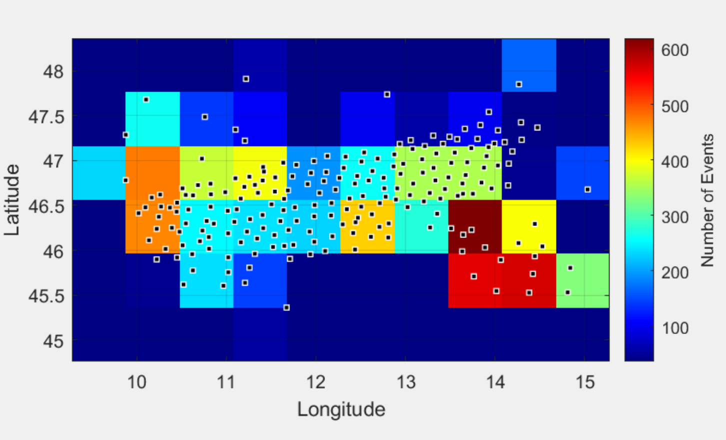

Figure 2: Station distribution and event density

The user is asked to provide the start and end points of the profile.

```
 Define start point (s), end point (e) or finish selection (y):
```

First the user has to decide if he wants to define the start or the end point by entering s for the start point or e for the end point of the virtual profile in the matlab terminal. Only if the start and end point are selected, the selection can be finished by entering y:

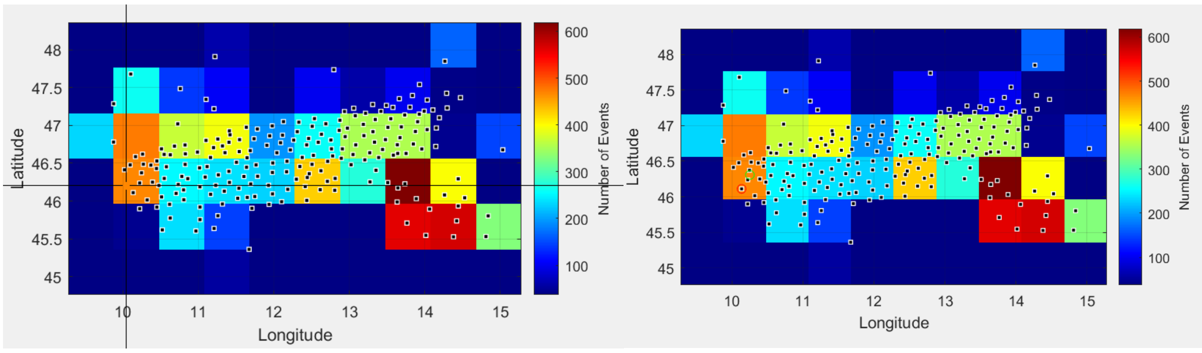

Figure 3: Selection of a start point

Entering s will produce a cross hair in the graphic and by clicking, the closest station to the position is marked by a red circle. From a harmonic mean weighted by the neighboring station distances within the maximum distance defined previously an optimum start point is calculated and marked as green circle.

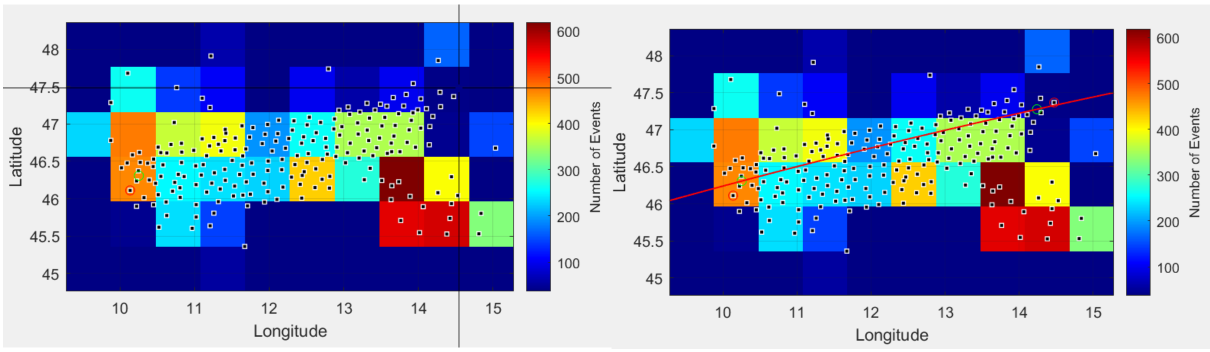

Figure 4: Selection of an end point

Equivalent to the selection of the start point, entering e will produce a cross hair in the graphic and by clicking the closest station to the position is marked by a red circle. From a harmonic mean weighted by the neighboring station distances within the maximum distance defined previously an optimum end point is calculated and marked as green circle. If both, an end and a start point are selected, the virtual profile is calculated and plotted. The user has now the possibility to alter the start and end points by entering s or e and selecting a different location.

By entering y the selection is finished and the virtual profile is used for further preparation for the tomographic inversion.

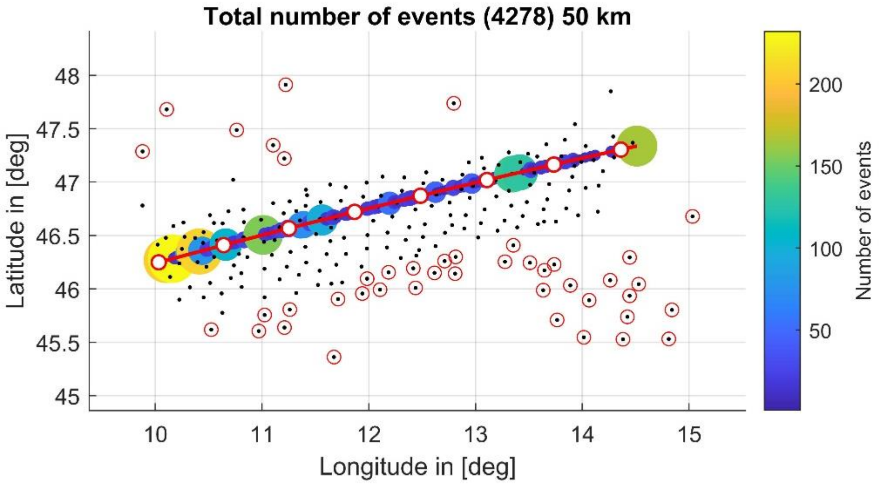

Figure 5: Virtual profile (red line with white dots) showing station used for the inversion (black dots without red circle), discarded stations (black dots with red circles) and number of projected events onto the profile in increments of 10 km. The white dots on the profile indicate the profile coordinate from start point to end point in increments denoted in the title (50 km in this example). The total number of events that is projected onto the virtual profile based on the selected stations is also indicated in the title.

Based on the selected profile a graphic is produced showing the final profile and stations used for the tomographic analysis and stored as StationMap.jpg in a newly created graphics folder within the work folder (e.g., ~/project_name/graphics/StationMap.jpg).

The virtual stations are prepared for the inversion as indicated by the prompt of the program and ray parameters are calculated for each event.

```
Preparing virtual station 1 of 162
Preparing virtual station 2 of 162
Preparing virtual station 3 of 162
```

…

```
Preparing virtual station 162 of 162
>>
```

A structure is created in the input folder (e.g., ~/project_name/input/input2D.mat), which contains the prepared data for the 2D inversion.

7.2 Preparation for 3D-Inversion

For a 3D-inversion the data can be prepared using the option `'prep3D'`. In the following, the user will be asked to define the corners for the area considered for further analysis.

```
 >>start_SItomo('prep3D')
```

As for every step, the user is asked to provide the location of the work folder (e.g., ~/project_name):

```
Define working directory for SI tomography: ~/project_name/
```

A station map is produced with an event density map in the background. This graphic is supposed to support the user for his decision on the area selected for the inversion. In this graphic, the user will select by point and click the upper left and lower right corner for a box which defines the stations used in the 3D inversion.

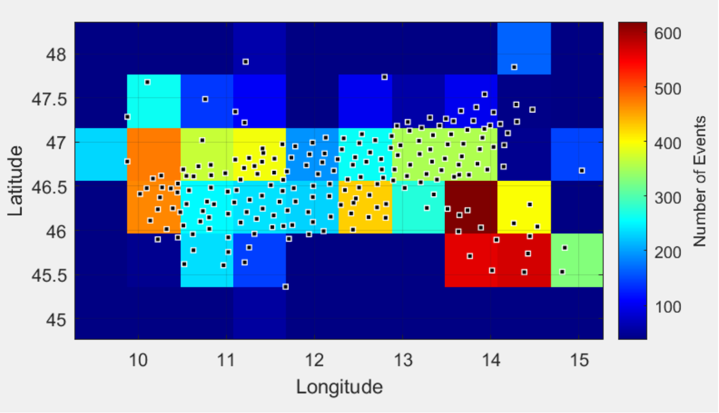

Figure 6: Station distribution and event density

The user is asked to provide the upper left and lower right corner points for the box.

```
 Define upper left corner (u), lower left corner (l) or finish selection (y):
```

First the user has to decide if he wants to define the upper left or the lower right corner by entering u for the upper left corner or l for the lower right corner of the area in the matlab terminal. Only if the corners are selected, the selection can be finished by entering y:

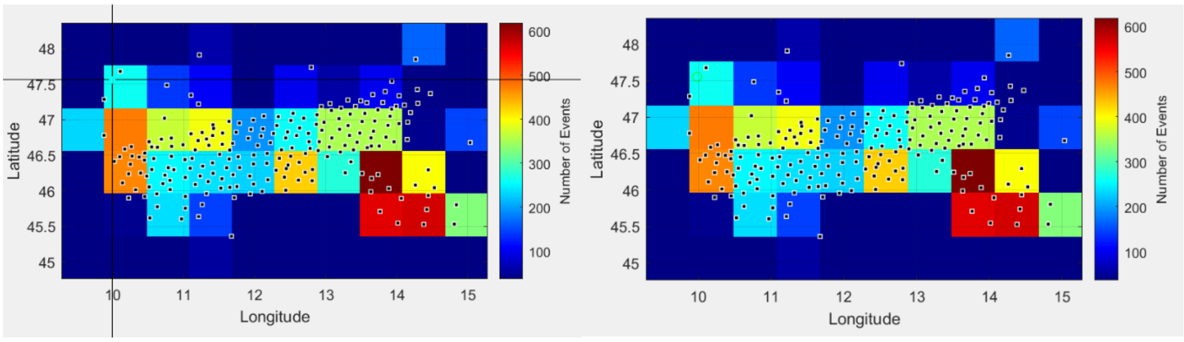

Figure 7: Selection of the upper left corner

Entering u will produce a cross hair in the graphic and by clicking, the position is marked by a green circle.

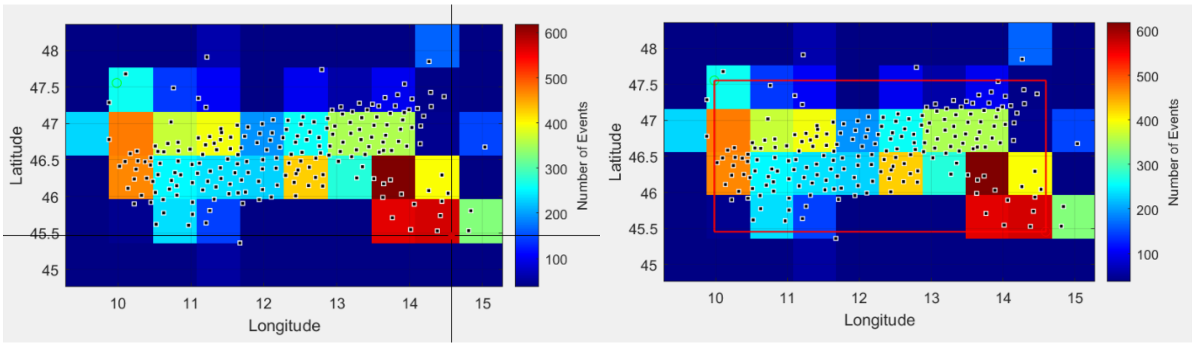

Figure 8: Selection of the lower right corner

Equivalent to the selection of the upper left corner, entering l will produce a cross hair in the graphic and by clicking the position is marked by a red circle. If corners are selected, the corresponding box plotted. The user has now the possibility to alter the corner points by entering u or l and selecting a different position.

By entering y the selection is finished and the box is used for further preparation of the data for the tomographic inversion.

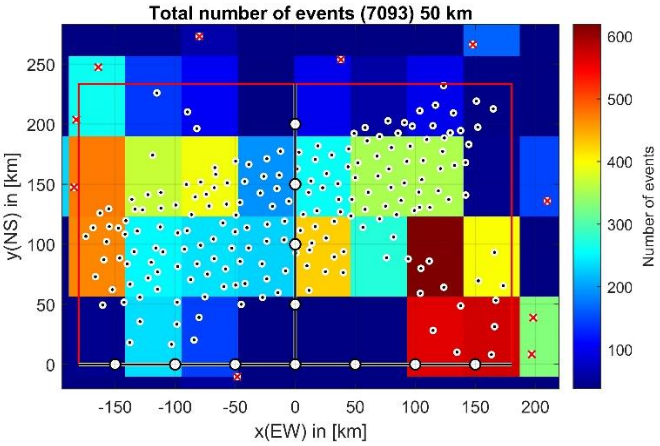

Figure 9: Area (red outline) and map showing stations used for the inversion (black dots without red circle) and discarded stations (black dots with red crosses). The background color shows the event density map as earlier. The station map is converted to coordinates in km. The black lines with white dots indicate the coordinate referred to in the inversion where the dots mark increments denoted in the title (50 km in this example). The total number of events used in the inversion is based on the selected stations, which is also indicated in the title.

Based on the selected box a graphic is produced showing the final selected stations used in the tomographic analysis referenced to x and y coordinates in km and stored as StationMap3D.jpg in a newly created graphics folder within the work folder (e.g., ~/project_name/graphics/StationMap3D.jpg).

The stations are prepared for the inversion as indicated by the prompt of the program and ray parameters are calculated for each event.

```
Preparing virtual station 1 of 199
Preparing virtual station 2 of 199
Preparing virtual station 3 of 199
```

…

```
Preparing virtual station 199 of 199
>>
```

A structure is created in the input folder (e.g., ~/project_name/input/input3D.mat), which contains the prepared data for the 3D inversion.

8. Inversion

The inversion can be performed with three different algorithm, which are described in detail in the accompanying publication (Link & Long 2024). The user can select a simple gradient descent method (GD in the following), a BFGS algorithm and a reversible jump Markov chain Monte Carlo algorithm (rj-McMC in the following). The latter can be used with gradient information or without, while the use of the gradient increases the speed of convergence in the inversion procedure.

8.1 Gradient Descent

The GD-algorithm is the most simple inversion procedure, which iteratively improves the model by calculating gradients, changing the model where the gradients are large and estimating the new misfit. Add the option ‘invert2Dgd’ or ‘invert3Dgd’ to the function call (in the following, a 2D example is shown, while the 3D inversion is equivalent and differences are explained, where they occur):

```
>> start_SItomo('invert2Dgd')
```

First the user is asked to provide an ID for the current inversion to allow for multiple runs of the same inversion with different starting models. All results and figures are stored with a filename that ends with the corresponding string (e.g., ~/project_name/results/resulta.mat for the results file if the string a was provided for the ID).

```
Define string ID for current inversion: a
```

As for every step, the user is asked to provide the location of the work folder (e.g., ~/project_name):

```
 Define working directory for SI tomography: ~/project_name/
```

The parameters for the inversion can be set in two ways, either by following the input requests for each parameter (see in the following with exemplary input):

```
 Search for dipping symmetry axis? 1=yes,0=no :  0
 Consider non-vertical incidence? 1=yes,0=no :   1
 Downweight stations with large amount of data? 1=yes,0=no :  0
 Define minimum depth in [km]:  40
 Define maximum depth in [km]:  350
 Define model dimension in x:  50
 Define model dimension in z:  25
 Define free parameter dimension in x:  25
 Define free parameter dimension in z:  10
 Define width of boundary region in [km] (e.g., 100):  100
 Define initial step size for anisotropy (e.g., 0.25):  0.25
 Define initial step size for fast axis in [deg] (e.g., 30):  30
 Define number of statistical iterations:  10
 Define maximum number of events per statistical iteration: 500
 Define number of iterations in inversion:  50
 Define number of consecutive iterations in x/phi:  4
 Use initial model (requires init.mat in input folder)? 1=yes,0=no :  0
 Show intermediate results? 1=yes,0=no :  1
 Print intermediate results? 1=yes,0=no :  1
 Define damping parameter: 0.05
 Define magnitude of proximity penalty (exmpl: 0.5): 1
```

Or by preparing a settings file named settings2D.txt for a 2D inversion or settings3D.txt for a 3D inversion and store it in a folder with the name results in the work folder (e.g., ~/project_name/results/settings2D.txt). There are exemplary settings files for each inversion procedure within the defaults folder in the program structure (e.g., settings2Dgd.txt, settings3Dgd.txt, …). Please make sure to maintain the structure within the files and only change the parameters and rename it to settings2D.txt or settings3D.txt in the work folder.

Side note: The 3D settings do not contain the parameter nny but the parameter iy, instead. nny corresponds to a downsampling of the y direction in the 2D inversion, where the model parameters are constant. In the manual input, this parameter is per default set to 3.

The user can decide to invert for the dip of the symmetry axis (thetaflag; this is only recommended for areas with a good data coverage). The non-vertical incidence can be considered (vflag; recommended). If the data set is produced from permanent stations with large amount of data and temporary stations with a sparse coverage, the user might want to consider a weighting according to the amount of data per station to avoid artifacts introduced by the dominating permanent stations (wflag). With the depth (z1; z2), the user defines the depth range, in which the inversion searches for the anisotropic structure leading to the observed splitting intensities. With the model dimensions (nx; nz; {ny for 3D}), the user defines the resolution of the underlying grid on which the splitting intensities and gradients are calculated. With the free parameters (ix; iz; {iy for 3D}), the number of free parameters is defined (be sure to select a lower number than for the model dimensions). The boundary region must be defined to account for the width of the sensitivity kernels at the edges of the model space (edgeadd). The step sizes in each model parameter defines how wide the initial steps will be in each parameter at the maximum gradient in the model space (xstepini; phistepini; {thetastepini if searching for dip}; the step sizes are downscaled when getting closer to a local minimum to avoid overshooting). The user can decide to start several statistical runs, consecutively (N1). The events used in each run will be selected randomly from the total amount of data, where the maximum number of the events used in the inversion can be defined (nevmax). The number of steps in each inversion must be provided (N2) and the number of steps after which the inversion alternates between strength of anisotropy and fast axis (n1) {and the number of steps after which the dip of the symmetry is considered (n2)}. An initial model can be considered (initflag; the structure must be identical to a the results structure, finmod.mat, provided by plot2D or plot3D, but must be stored in the input folder and re-named as init.mat). The user can decide to show intermediate results after each step of the inversion (showflag) and print the corresponding figure after the final step of the inversion (printflag). The damping (alpha) is used to enforce a smooth result. The proximity penalty (dec; between 0 and 1) penalizes the model volumes closer to the stations, which have stronger sensitivity than model elements at larger distance, while the kernels don't overlap as good as for the deeper parts which might result in an over-estimated anisotropy at shallow depth if no penalty (dec=1) is chosen.

The program will first estimate a coverage for the model area by computing sensitivity kernels for a homogeneous anisotropic model for all events. One graphic is produced (e.g., project_name_sensitivity.jpg) with a simple stack of these kernels in the plane of the profile for a 2D analysis (and in a 2D plane crosscutting the middle of the y-direction along the x-axis for a 3D analysis). A second graphic is produced (e.g., project_name_nn.jpg) by setting the individual kernels to the value 1 at volumes where they exceed 10% of their maximum absolute value and 0 for the remaining volumes. These normalized kernels are stacked to estimate from how many earthquakes each model volume is sampled. To ensure equal sampling across the free parameter space, an averaging scheme is used, by combining neighboring cells until their sum of earthquake samplings reach the maximum sampling of an individual cell. The corresponding cell sizes are shown in a third graphic (e.g., project_name_cells.jpg).

If the settings file was not prepared prior to starting the inversion, the input of the parameters will create the results folder as well as the corresponding settings file. If the process is interrupted and has to be started again, this input file can be used instead of typing in all parameters again:

```
Settings file exists. Use existing (0) or define new settings (1)? :  0
```

During the inversion procedure, the program will provide information on the current step it is processing:

```
161.7212
Step 1 of 20 done. (round 1)
Testing phi
160.4169
```

The number above the current step shows the misfit of the model before the model was improved. The following line shows the current step as well as the number of steps in the current inversion. In parenthesis, the current round of the successive inversions with different starting models is shown. Beneath indicates which variable is changed in this step. All variables are changed alternating between fast axis direction (phi), strength of anisotropy (x) and dip (theta). The line beneath the fast axis direction shows the new misfit after changing the model.

After all inversions are finished the results are stored in a matlab file within the results folder with the name result{stringID}.mat (for 2D) and result3D{stringID}.mat (for 3D). This file contains a model structure that contains all inversion parameters and a result structure containing the results from each inversion.

8.2 BFGS-Algorithm

The BFGS-algorithm involves estimating the Hessian matrix iteratively during the inversion. Therefore, the sensitivity for the model parameters is considered. This option can only be used if optimization toolbox of matlab is installed and the function fminunc can be used. Otherwise, the use of the gradient descent or rj-McMC algorithm are recommended. Add the option 'invert2Dbfgs' or 'invert3Dbfgs' to the function call (as before, a 2D example is shown, while the 3D inversion is equivalent and differences are explained, where they occur):

```
>> start_SItomo('invert2Dbfgs')
```

The following requests and inputs are equivalent to the gradient descent method. Therefore, the parameters for the inversion can be set in two ways, either by following the input requests for each parameter:

```
Search for dipping symmetry axis? 1=yes,0=no :  0
Consider non-vertical incidence? 1=yes,0=no :   1
Downweight stations with large amount of data? 1=yes,0=no :  0
Define minimum depth in [km]:  40
Define maximum depth in [km]:  350
Define model dimension in x:  50
Define model dimension in z:  25
Define free parameter dimension in x:  25
Define free parameter dimension in z:  10
Define width of boundary region in [km] (e.g., 100):  100
Define number of statistical iterations:  10
Define maximum number of events per statistical iteration: 500
Define number of iterations in inversion:  50
Define number of pre-cycles to prepare initial guess (e.g., 3):  3
Define maximum number of iterations in pre-cycles (e.g., 10):  10
Use initial model (requires init.mat in input folder)? 1=yes,0=no :  0
Show intermediate results? 1=yes,0=no :  1
Print intermediate results? 1=yes,0=no :  1
Define damping parameter: 0.05
Define magnitude of proximity penalty (exmpl: 0.5): 1
```

Or by preparing a settings file named settings2D.txt for a 2D inversion or settings3D.txt for a 3D inversion and store it in a folder with the name results in the work folder (e.g., ~/project_name/results/settings2D.txt). There are exemplary settings files for each inversion procedure within the defaults folder in the program structure (e.g., settings2Dbfgs.txt, settings3Dbfgs.txt, …). Please make sure to maintain the structure within the files and only change the parameters and rename it to settings2D.txt or settings3D.txt in the work folder.

In the following, only the parameters different to the gradient descent method are explained. The step sizes in the BFGS-algorithm are estimated automatically. The BFGS-algorithm uses a defined number of pre-cycles (n1) with a fixed number of iterations (n2) to approach a local minimum prior to the free inversion, to get a better estimate of the Hessian matrix that might be obscured otherwise.

The sensitivity and effective cell sizes are estimated as for the gradient descend method (involving the production of corresponding graphics).

If the settings file was not prepared prior to starting the inversion, the input of the parameters will create the results folder as well as the corresponding settings file. If the process is interrupted and has to be started again, this input file can be used instead of typing in all parameters again.

During the inversion procedure, the program will provide information on the current step it is processing.

After all inversions are finished the results are stored in a matlab file within the results folder with the name result.mat (for 2D) and result3D.mat (for 3D). This file contains a model structure that contains all inversion parameters and a result structure containing the results from each inversion.

8.3 Rj-McMC Algorithm

The rj-McMC-algorithm is a gradient informed stochastic inversion scheme where the number of free parameters is a variable itself. Add the option ‘invert2Dmcmc’ or ‘invert3Dmcmc’ to the function call (in the following, a 2D example is shown, while the 3D inversion is equivalent and differences are explained, where they occur):

```
>> start_SItomo('invert2Dmcmc')
```

The following requests and inputs are equivalent to the gradient descent method. Therefore, the parameters for the inversion can be set in two ways, either by following the input requests for each parameter:

```
Search for dipping symmetry axis? 1=yes,0=no :  0
Consider non-vertical incidence? 1=yes,0=no :   1
Define minimum depth in [km]:  40
Define maximum depth in [km]:  350
Define model dimension in x:  50
Define model dimension in z:  20
Define width of boundary region in [km] (e.g., 100):  100
Define initial step size for anisotropy (e.g., 0.25):  0.25
Define initial step size for fast axis in [deg] (e.g., 30):  30
Define number of initial voronoi cells:  40
Define maximum number of events in the inversion: 500
Define number of iterations in inversion:  5000
Define after how many steps the reference should be re-calculated:  1000
Define minimum number of voronoi cells:  30
Use initial model (requires init.mat in input folder)? 1=yes,0=no :  0
Show intermediate results? 1=yes,0=no :  1
Print intermediate results? 1=yes,0=no :  1
Define damping parameter: 0.05
Define distance weighting exponent: 1
```

Or by preparing a settings file named settings2D.txt for a 2D inversion or settings3D.txt for a 3D inversion and store it in a folder with the name results in the work folder (e.g., ~/project_name/results/settings2D.txt). There are exemplary settings files for each inversion procedure within the defaults folder in the program structure (e.g., settings2Dmcmc.txt, settings3Dmcmc.txt, …). Please make sure to maintain the structure within the files and only change the parameters and rename it to settings2D.txt or settings3D.txt in the work folder.

In the following, only the parameters different to the gradient descent method are explained. The step sizes in the rj-McMC algorithm define the maximum step size in strength, fast axis and dip, while the actual width of each step is scaled with a random parameter. Each call of the rj-McMC algorithm only allows for the run of a single chain. Multiple calls are recommended to allow for a statistical search for the global minimum in the parameter space. However, the initial number of Voronoi cells must be defined (N1), the number of iterations (N2; this parameter is much larger than in the gradient descent and BFGS algorithms) and the minimum number of Voronoi cells (n2; the death of Voronoi cells is skipped if this number is reached). To accelerate the forward calculation, the model and splitting intensities are updated corresponding to a reference. This reference can be forced to be re-evaluated after a certain number of steps (n1) to avoid numerical instabilities.

The rj-McMC algorithm does not involve a sensitivity and effective cell-size estimates.

If the settings file was not prepared prior to starting the inversion, the input of the parameters will create the results folder as well as the corresponding settings file. If the process is interrupted and has to be started again, this input file can be used instead of typing in all parameters again.

During the inversion procedure, the program will provide information on the current step it is processing.

After all inversions are finished the results are stored in a matlab file within the results folder with the name result.mat (for 2D) and result3D.mat (for 3D). This file contains a model structure that contains all inversion parameters and a result structure containing the results from each inversion.

9. Stochastic evaluation and Presentation

The final models are evaluated using the command ‘plot2D’ or ‘plot3D’ depending on the configuration of the inversion.

```
>> start_SItomo('plot2D')
```

As for every step, the user is asked to provide the location of the work folder (e.g., ~/project_name):

```
 Define working directory for SI tomography: ~/project_name/
```

SItomo will automatically evaluate the results from all statistical iterations stored in the work folder and provide an estimate and error for the model parameters (stored in the structure finmod.mat/finmod3D.mat). Figures are produced equivalent to the results shown in the accompanying publication (in 2D for the full model and a 2D through the center of the y-direction, the full model can be extracted within matlab from the result structure finmod3D.mat as finmod.x, finmod.phi and finmod.theta for strength, fast axis and dip).