A Manual for ANSEICCA

July 6, 2023

Contents

Chapter 1

Getting Started

1.1 Installation

Download the program from this link:
`https://github.com/arjundatta23/cc_kern_inv`

Three things are needed to access full functionality:

1. Devito
2. ObsPy
3. SW1D_earthsr

Devito

To install the latest Devito release along with any additional dependencies, follow:

```
pip install devito
```

or you can install it via `Anaconda`

```
git clone https://github.com/devitocodes/devito.git
cd devito
conda env create -f environment-dev.yml
source activate devito
pip install -e .
```

ObsPy

ObsPy is an open-source project [1] dedicated to provide a Python framework for processing seismological data. To install this package run the following:

```
conda install -c conda-forge obspy
```

SW1D_earthsr

You'll also need the package *SW1D_earthsr* available at:
`https://github.com/arjundatta23/SWRT`

1.2 Set up (setting of paths)

You will need to modify the paths (in various programs/modules) specifying the locations of the following directories to facilitate import of various modules:

1. modules_common
2. SW1D_earthsr

Paths can be set using `sys.path.append()`.

Chapter 2

Running the code

To run the serial or parallel code, one simply needs to run the corresponding wrapper (specifying the correct number of processors in the parallel case, see below). Two controls are provided in the wrapper programs, via the following boolean flags:

1. *do_inv*: if *True*, the code executes to completion – as determined by the second flag described below – and produces an output file. If set to *False*, the code executes only up to the 'set up' stage; no forward or inverse modelling is performed, and no output file produced. The *False* setting is useful for plotting and visualization purposes.

2. *iter_only1*: if *False*, inversion proceeds until 'natural' termination. If set to *True*, the inversion is force-terminated after one iteration. The *True* setting is useful for testing and forward modelling purposes.

All other code settings, including parameter specifications, are provided in the *Par_file*, which is discussed in Chapter 3.

Parallel version of the code

Parallelization is achieved with mpi4py.

To run the parallel-version of the code, use the following:

```
mpirun -np <nrecs> python -m mpi4py anseicca_wrapper_parallel.py
```

Note that the number of processors np must be equal to the number of receivers <nrecs>.

Chapter 3

Input

The minimum input for the package consists of two files with hardwired names and locations:

1. *receivers.csv* – a text file containing the list of stations or receivers with their IDs and location in UTM coordinates. Stations need to be specified in a tab-separated list as in the following example.

No.	ID	Easting	Northing
1	1	401887.32	8490063.26
2	2	403102.96	8490621.37
3	3	403880.98	8490441.22

Figure 3.1: Example to illustrate the format of the receivers file

2. *Par_file* – a text file specifying the values of various simulation parameters. This file must be compatible with *modules_common/config_file.py*, which reads it using Python's *configparser* utility.

Both these files need to be present in the *INPUT* sub-directory within *anseicca*.

3.1 Description of the *Par_file*

- *epsilon*: a small-valued constant (added to internally calculated distances, used to prevent the Hankel function from blowing up at the origin.)
- *nrecs*: the number of receivers to be used, must be less than or equal to the total number of receivers listed in *receivers.csv*.
- *ext_data*: specifies external data mode (if *True*) or internal data mode (if *False*).

SIMULATION GEOMETRY

- *dx*: uniform grid spacing in x-y directions (in km).
- *zmax*: maximum depth to consider, when working with a depth-dependent Earth model (in km).
- *domain_size*: size of modelling domain - length of (side of) main box OR half-length of outer box (in km).
- *grid_origx*: x-coordinate of the origin of the modelling domain (km units). Chosen such that the origin lies roughly at the network centre.
- *grid_origy*: y-coordinate of the origin of the modelling domain (km units). Chosen such that the origin lies roughly at the network centre.

CROSS-CORRELATION MODELLING THEORY

- *src_dir*: orientation of ambient noise (point-force) sources.

- *cc_component_RTZ*: component(s) of Green tensor to work with.

INVERSION-RELATED

- *gamma_inv*: damping parameter for regularized least squares inversion.
- *noise_amp_pcent*: A parameter relevant for synthetic test inversions only – determines the noise level in the synthetic data.
- *noise_band*: A parameter relevant for synthetic test inversions only – determines the frequency range of the noise.
- *outer_box_size_factor*: A parameter relevant for synthetic test inversions only – allows the true/test model to be larger than the modelling domain.

HOMOGENEOUS ACOUSTIC MODELLING PARAMETERS

- *density*: the density of the medium.
- *wavespeed*: the wavespeed in the medium.

SIGNAL CHARACTERISTICS

- *dt*: sampling interval of the signal.
- *num_samples*: number of samples in the signal.
- *pow_spec_type*: parameter that decides whether a narrow band (when the value is 1) or a broad band (when the value is 2) of power spectrum is defined.
- *freq_centre*: centre frequency of the signal.
- *freq_bw_frac*: width (sigma) for gaussian spectrum as a fraction of centre frequency.
- *freq_lb*: lower bound of the frequency for Tukey spectrum.
- *freq_ub*: upper bound of the frequency for Tukey spectrum.
- *tukey_alpha*: alpha parameter for Tukey window (tapered cosine window) in spectral domain from lower bound frequency to upper bound frequency. The alpha parameter is the ratio of cosine-tapered section length to the entire window length.

MODEL PARAMETERIZATION

- *src_mod_type_testdata*: parameter that defines the true source model. Can take one of the values among −1, 0, 1, 2, 3, for 'every grid point', 'multiple gaussians', 'ring of gaussians', 'radially gaussian rings', 'gaussian grid' respectively.
- *src_mod_type_inversion*: parameter that defines the source model for the inversion. Can take one of the values among −1, 0, 1, 2, 3, for 'every grid point', 'multiple gaussians', 'ring of gaussians', 'radially gaussian rings', 'gaussian grid' respectively.

FORWARD MODELLING

- *struc_lat_homo_testdata*: parameter that determines if the structure model pertaining to forward modelling of the true source model is homogenous (if True) or not (if False).
- *struc_lat_homo_inversion*: parameter that determines if the structure model pertaining to inversion is homogenous (if True) or not (if False).
- *modelling_level1_testdata*: parameter that determines if the forward modelling of the true source and structure model is to be carried out analytically (if 0) or numerically (if 1).
- *modelling_level1_inversion*: parameter that determines if the forward modelling of the inversion result is to be carried out analytically (if 0) or numerically (if 1).
- *modelling_level2_testdata*: parameter that determines if the forward modelling of the true source and structure model is to be carried out: for acoustic (if 0 when modelling_level1_testdata=0) or elastic (if 1 when modelling_level1_testdata=0); using frequency-domain solver (if 0 when modelling_level1_testdata=1) or using time-domain solver (if 1 when modelling_level1_testdata=1)

- *modelling_level2_inversion*: parameter that determines if the forward modelling of the inversion result is to be carried out: for acoustic (if 0 when modelling_level1_testdata=0) or elastic (if 1 when modelling_level1_testdata=0); using frequency-domain solver (if 0 when modelling_level1_testdata=1) or using time-domain solver (if 1 when modelling_level1_testdata=1)

SPECIFICATIONS FOR MODEL PARAMETERIZATION

For 'multiple gaussians' source model type

- *somod_mg_specs_r0_x*: x-coordinates of 2-D gaussian centres
- *somod_mg_specs_r0_y*: y-coordinates of 2-D gaussian centres
- *somod_mg_specs_w_x*: x-widths of 2-D gaussians (km)
- *somod_mg_specs_w_y*: y-widths of 2-D gaussians (km)
- *somod_mg_specs_mag*: magnitude of the gaussian

For 'ring of gaussians' source model type

- *somod_rg_specs_r*: radius of the ring (in km).
- *somod_rg_specs_w*: width (in km)
- *somod_rg_specs_as*: basis angular sampling (degrees)
- *somod_rg_specs_np*: number of perturbed segments
- *somod_rg_specs_t1*: segment start (degrees)
- *somod_rg_specs_t2*: segment end (degrees)
- *somod_rg_specs_pert*: segment perturbation (additive)

For 'radially gaussian rings' source model type

- *somod_rgr_specs_r0_x*: x-coordinates of ring centres
- *somod_rgr_specs_r0_y*: y-coordinates of ring centres
- *somod_rgr_specs_r*: ring radii (in km).
- *somod_rgr_specs_w*: ring widths (in km).
- *somod_rgr_specs_mag*: ring magnitudes

For 'gaussian grid' source model type

- *somod_gg_specs_w* : width (km)
- *somod_gg_specs_ls*: basis linear sampling (km)
- *somod_gg_specs_pos_x*: perturbed x-coordinates
- *somod_gg_specs_pos_y*: perturbed y-coordinates
- *somod_gg_specs_np*: number of perturbations
- *somod_gg_specs_x_l*: lower limit of gaussian array along the row
- *somod_gg_specs_x_u*: upper limit of gaussian array along the row
- *somod_gg_specs_y_l*: lower limit of gaussian array along the column
- *somod_gg_specs_y_u*: upper limit of gaussian array along the column
- *somod_gg_specs_mag*: magnitude

3.2 Data input

- *Format of the data*: The data must be in SAC format or .npz format.
- *SAC headers necessary*: Following SAC headers need to be filled-

NPTS number of data points

B begin time

E end time

LEVEN evenly sampled time series

DELTA time increment

KSTNM station name

KEVNM event ID

DIST source receiver distance in km

AZ azimuth

BAZ back azimuth

Chapter 4

Output and Graphics

When do_inv is set to True, the output of the program is a pickle file, which can be read using the *view_output_anseicca.py* script. The user is presented with the following options:

```
What do you want to do ?
Choose from the following options
                  1 - View models
                  2 - View kernels
                  3 - View inversion progress
                  4 - View individual waveforms
                  5 - Write SAC output (all waveforms)
                  6 - Miscellaneous
                  7 - Exit
Enter your choice (number) here:
```

Figure 4.1: Options presented to the user

Below we show the figures corresponding to Example 1 (E1) provided with the code. This example is the same as Figure 2(e or f) from [2], but without noise added to the synthetic data, for the sake of reproducibility.

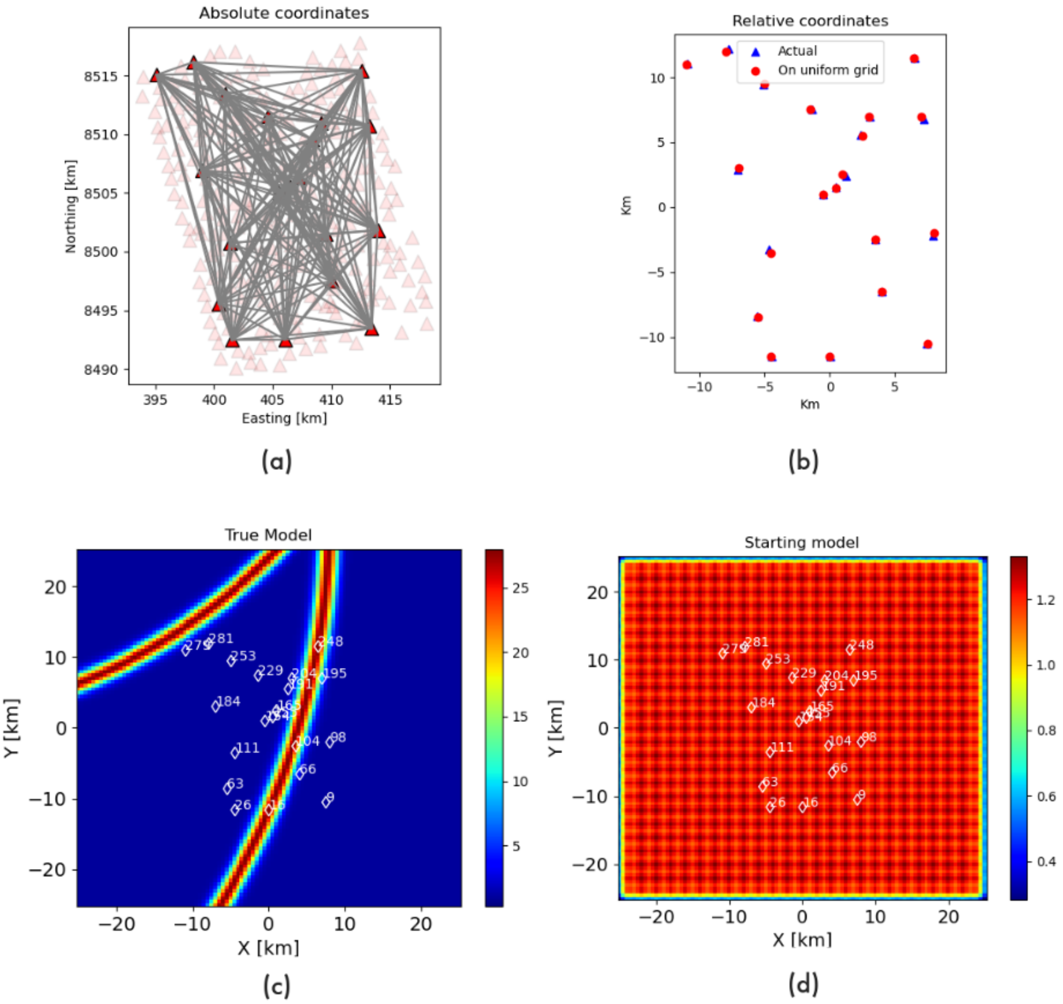

Figure 4.2: (a) Absolute coordinates of stations and the interconnecting paths (b) Relative coordinates (c) True model (d) Starting model

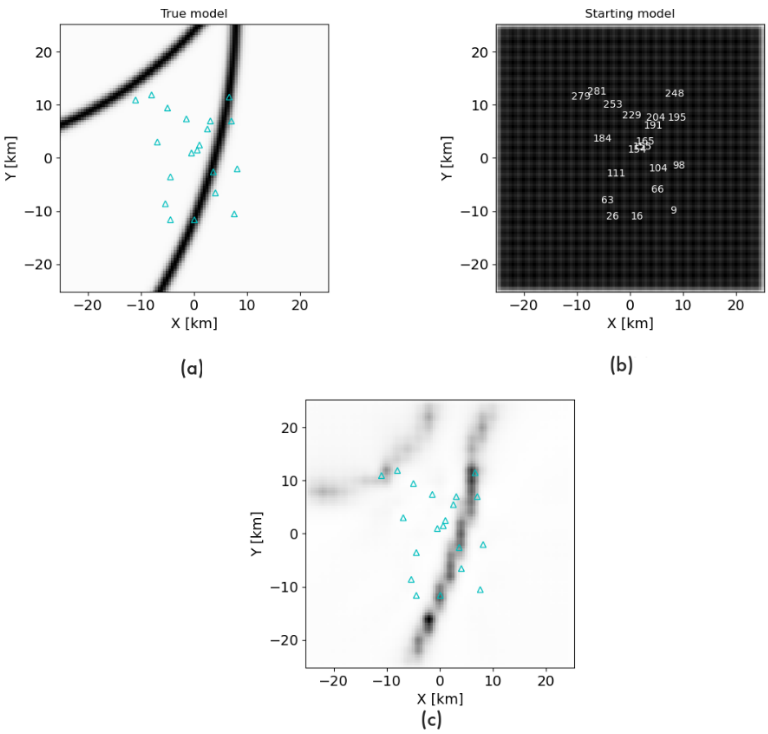

Figure 4.3: Figures generated with option 1 - View models (a) True model (b) Starting model (c) Inversion result

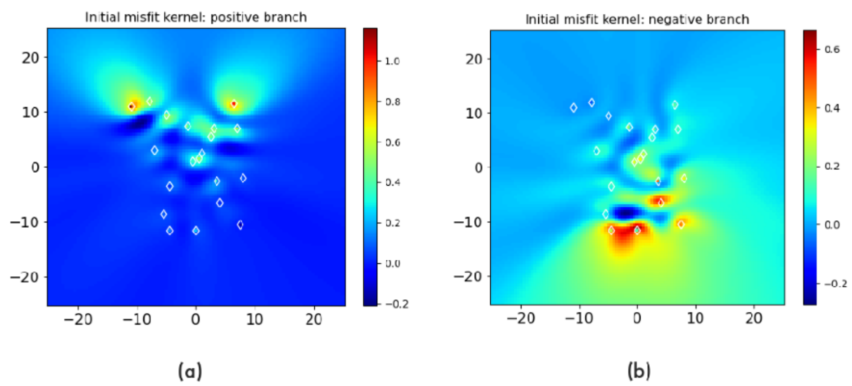

Figure 4.4: Figures generated with option 2 - View kernels (a) Initial misfit kernel with positive branch (b) Initial misfit kernel with negative branch

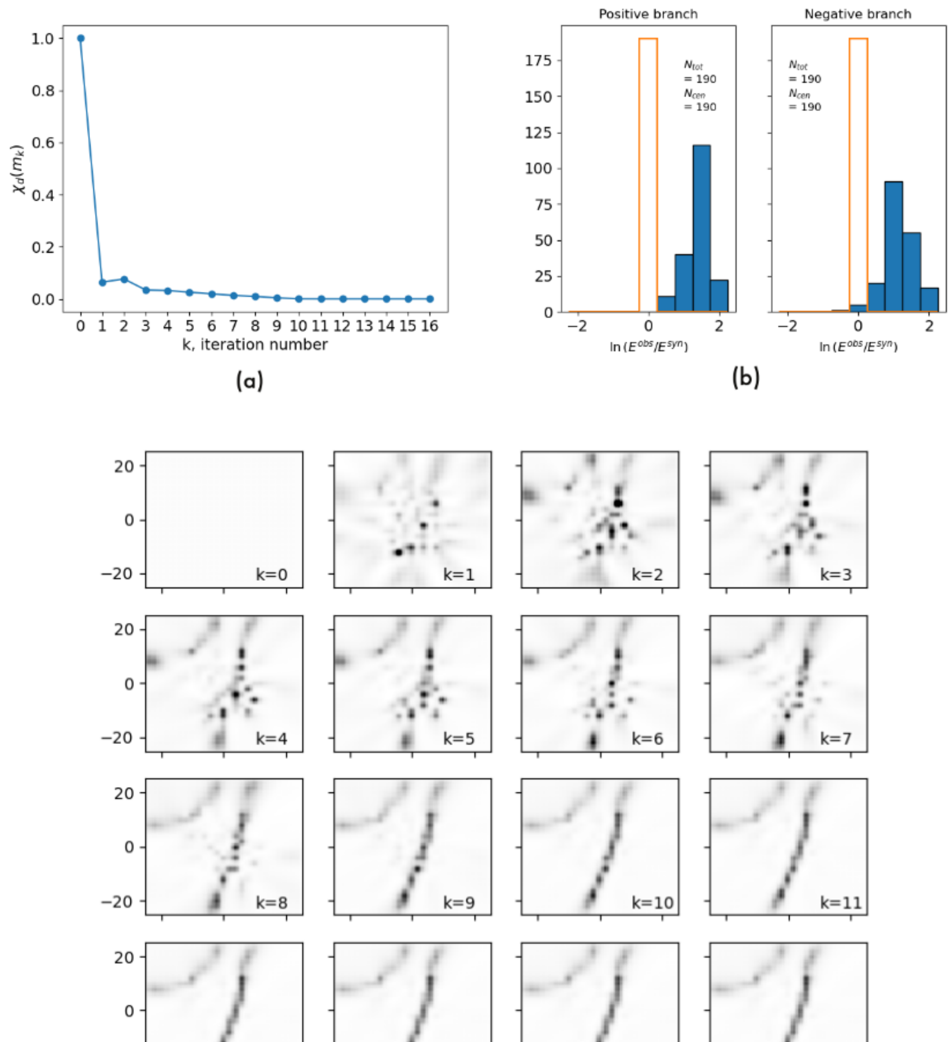

Figure 4.5: Figures generated with option 3 - View inversion progress (a) Decrease of misfit with each iteration (b) Histogram of misfit versus number of receiver pairs (c) Evolution of the predicted model with number of iterations

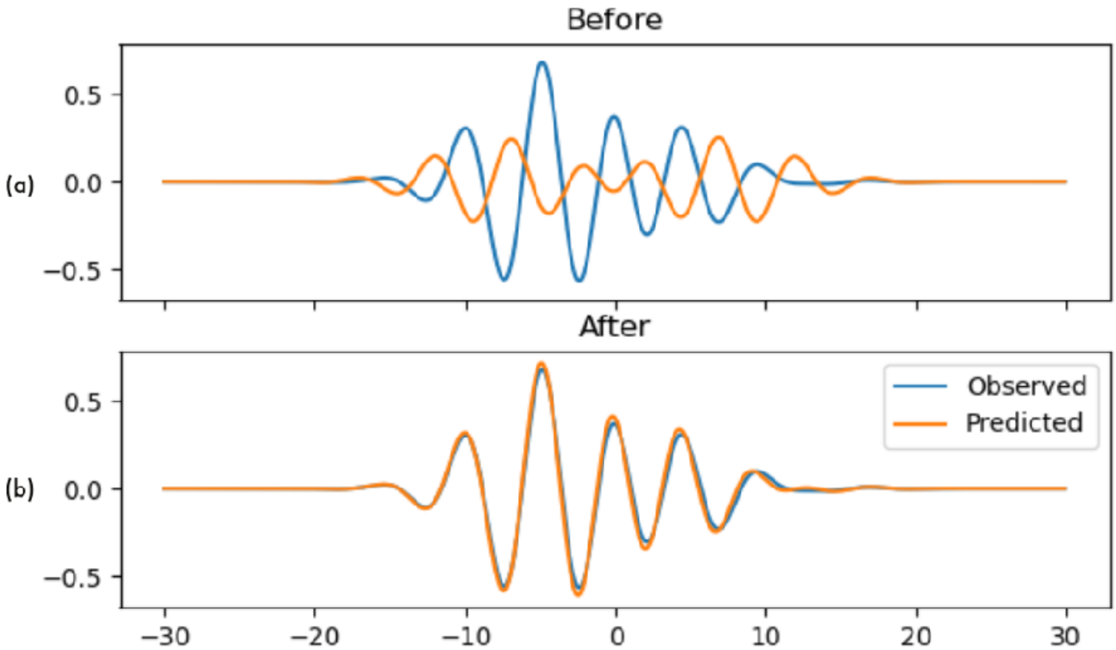

Figure 4.6: Figures generated with option 4 - View individual waveforms; Predicted waveform overlayed on Observed waveform (a) Before inversion (b) After inversion

Bibliography

[1] Moritz Beyreuther et al. "ObsPy: A Python Toolbox for Seismology". In: *Seismological Research Letters* 81 (June 2010), pp. 530–533. ISSN: 08950695. DOI: 10.1785/gssrl.81.3.530. URL: https://pubs.geoscienceworld.org/ssa/srl/article/81/3/530/143693/ObsPy-A-Python-Toolbox-for-Seismology.

[2] Arjun Datta, Bharath Shekar, and Pushp L Kumar. "Acoustic full waveform inversion for 2-D ambient noise source imaging". In: *Geophysical Journal International* 234 (3 Apr. 2023), pp. 1628–1639. ISSN: 0956-540X. DOI: 10.1093/GJI/GGAD158. URL: https://academic.oup.com/gji/article/234/3/1628/7117958.