AxiSEM Tutorial

Kasra Hosseini, Martin v. Driel, Simon Stähler, Lion Krischer, Tarje Nissen-Meyer

Fairbanks, Alaska, July 15th 2013

1 Overview

- 10min intro(what is AXISEM, ObsPy; what we want to do in the tutorial)
- 10min big run (maybe drop this, depending on progress next week)(2seconds, seismograms and not wavefields, 2 settings [different structures, sharpness of boundaries])
- 60min virtual box: data and AxiSEM
- 20min big run post processing
- installation of AxiSEM, obspy locally

2 Tutorial tasks

2.1 ObsPy

Some of the tools employed in this tutorial use ObsPy but the tutorial does not have a formal introduction to ObsPy due to temporal constraints. The VirtualBox image contains an extensive amount of training material for Python, ObsPy and some third party libraries intended to get you started. Furthermore one of the core developers of ObsPy is present and able to answer any questions you might have. You can find all material related to ObsPy at /home/Desktop/ObsPy.

2.2 Familiarity with AxiSEM

The goal of this initial task is to familiarize users with the basic principles behind AxiSEM, its input/output structures and peculiarities such as post-processing. An end-to-end approach (meshing to wavefield movie and seismograms) will be conducted for long-period settings within the virtual box.

2.3 Data versus modeling: Obspy, AxiSEM, SPECFEM

The main goal of this part of the tutorial is to use AxiSEM for realistic scenarios, compare the results with real seismograms and explore the effects of source parameters and backgound model on the waveforms. In particular, we will learn how to:
Load data with Obspy and plot seismogram cross sections;
plot data vs. AxiSEM and SPECFEM synthetics (at 20s);
compare data vs. AxiSEM synthetics for various frequencies and background models;
analyze the influence on different source parameters on waveforms.

3 Virtual Box content

The box is organized into 3 folders: AXISEM, OBSPY, EVENTS. The entire data-vs-modeling section is within EVENTS, including all data and pre-computed synthetics. AXISEM contains the source code and the precomputed run for the long-period section.

4 AxiSEM familiarity

<u>Low-frequency simulation at period 50/100 s (20min)</u>

1. change AxiSEM input parameters to run this scenario, submit job
2. check mesh, background model, source-receiver geometry (google earth)
3. post processing on 50s run: filter, sum, rotate, movie snapshots

5 Data and modeling

The main goal of this tutorial is to use AXISEM for real scenarios, compare the results with real seismograms and explore the effects of source parameters and backgound model on the waveforms. In case that you need more information, refer to Appendix-4 in which a complete example with all the commands and results is presented. For a brief walk-through, follow this:
Start from within the ~/Desktop/EVENTS/SCRIPTS directory:

1. Plot one of the events (listed in EVENTS directory, we choose EVENT-1 in this example):

   ```
   $ plot_station_event_distribution.py EVENTS/EVENT-1
   ```

   * For more information on the folder structure of your Virtual-box, refer to Appendix-2.

2. To get an overview on both real data and pre-simulated seismograms: (e.g. PREM_ANISO for 5 seconds dominant period)

   ```
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec
   ```

3. Cut the waveforms for Pdiff and PKiKP seismic phases and plot the seismograms: (compared with real data)

   ```
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec PKiKP
   ```

4. Change the filter in *plot_seismograms.py* script and compare the results with *SPECFEM3D*: (for changing the filter, open *plot_seismograms.py* and change values at top of the file)

   ```
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
   specfem3d
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec PKiKP
   specfem3d
   ```

5. Compare the results of two different background models (*PREM_ANISO_5sec* with *IASP91_5sec)* for Pdiff phase:

   ```
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
   EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/IASP91_5sec
   ```

6. Change the filter, as explained in step 4, and repeat step 5.

7. Compare the seismograms calculated for two different source parameters (PREM_ANISO_5sec and PREM_ANSIO_5sec_GCMT) for Pdiff phase: (For more information about the source parameters, refer to Appendix-1)

   ```
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
   EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec_GCMT
   ```

8. Change the filter, as explained in step 4, and repeat step 7.

9. Find the time shift between the synthetics and real data, shift the synthetics accordingly and plot the results:

   ```
   $ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
   shift_synthetics
   ```

10. Map the calculated time shifts in step 9 on the location of stations:

    ```
    $ plot_travel_time_map.py EVENTS/EVENT-1
    ```

APPENDIX-1: Events

Three events are selected for this tutorial (Figure-A1) with the following source characteristics:

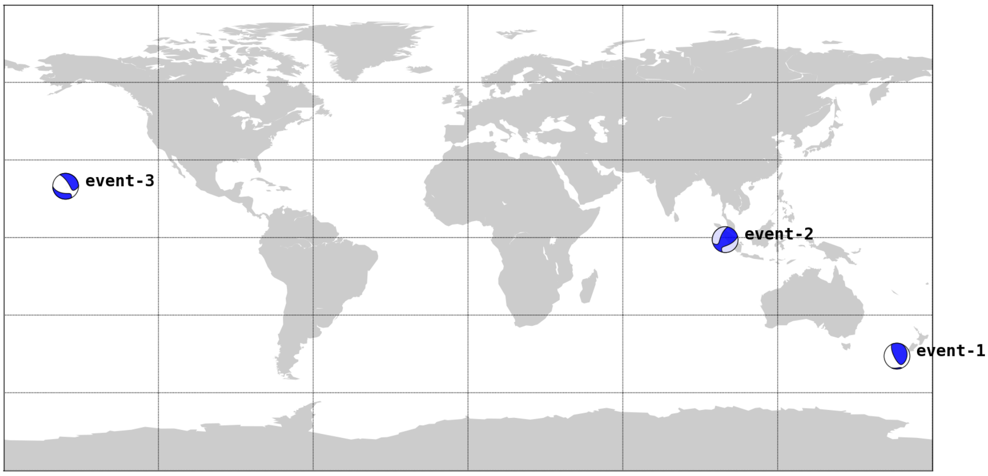

Figure A1: beach ball diagrams of event-1 to event-3 (based on GCMT catalog)

event-1

	GCMT	Inverted
Date-Time (UTC)	2009-07-15 09:22:49	-----
Location	OFF W. COAST OF S. ISLAND, N.Z.	-----
Latitude, Longitude	-45.850 °, 166.260 °	-----
Magnitude	7.8 MW	-----
Depth	23.5 km	24.0 km
Mrr, Mtt, Mpp	3.14e20, 0.759e20, -3.9e20	0.324e21, 0.303e20, -0.354e21
Mrt, Mrp, Mtp	2.59e20, -3.75e20, -0.028e20	0.161e21, -0.439e21, 0.455e20

event-2

	GCMT	Inverted
Date-Time (UTC)	2009-09-30 10:16:10	-----
Location	SOUTHERN SUMATRA, INDONESIA	-----
Latitude, Longitude	-0.790 °, 99.670 °	-----
Magnitude	7.6 MW	-----
Depth	77.8 km	82.0 km
Mrr, Mtt, Mpp	1.76e20, -0.765e20, -0.992e20	0.163e21, -0.148e20, -0.148e21
Mrt, Mrp, Mtp	0.658e20, -0.991e20, -1.94e20	0.349e20, -0.397e20, -0.157e21

event-3

	GCMT	Inverted
Date-Time (UTC)	2006-10-15 17:07:55	-----
Location	HAWAII	-----
Latitude, Longitude	19.830 °, -155.940 °	-----
Magnitude	6.7 MW	-----
Depth	48.0 km	28.0 km
Mrr, Mtt, Mpp	-0.816e19, 0.869e19, -0.053e19	-0.307e19, 0.356e19, -0.496e18
Mrt, Mrp, Mtp	0.063e19, -0.818e19, -0.853e19	-0.142e19,-0.355e19, -0.104e20

APPENDIX-2: Folder structure

Figure A2 shows how the events (and their meta-data), waveforms and scripts are organized in the Virtual-box:

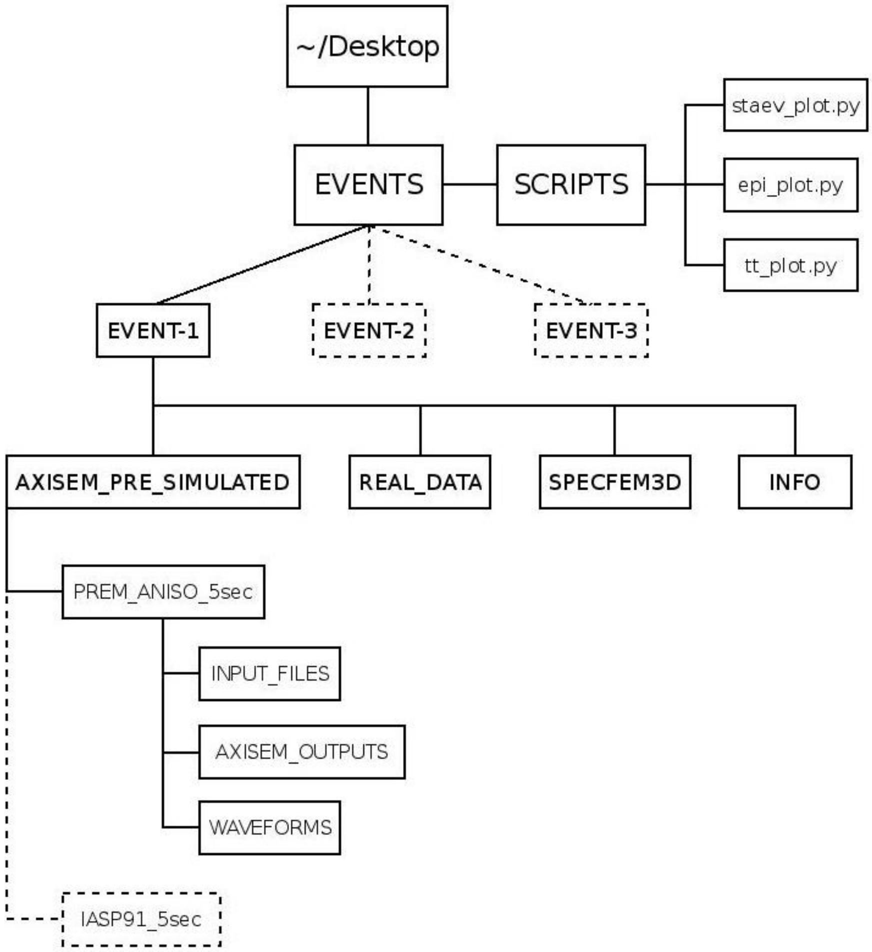

Figure A2: Folder structure

In *SCRIPTS* directory, there are three python scripts that we use here:

1. **plot_station_event_distribution.py**: maps event and stations of an event directory.
2. **plot_seismograms.py**: plotting tool for comparison purposes.
3. **plot_travel_time_map.py**: project the time shift derived by cross correlating the AXISEM waveforms and real data.

In *EVENTS* directory, there are three events, each with the following sub-directories:

1. **AXISEM_PRE_SIMULATED**: contains seismograms simulated by AXISEM with the required input files to re-produce them.
2. **REAL_DATA**: seismograms retrieved from *IRIS.* (refer to APPENDIX-5)
3. **SPECFEM3D**: waveforms simulated by SPECFEM3D for comparison purposes (downloaded from *IRIS*, refer to APPENDIX-5)
4. **INFO**: information about the event and stations: event_1.xml and STATIONS

APPENDIX-3: A Quick Guide to PyAxi

PyAxi is a Python script developed as an interface for AXISEM. All the options available in AXISEM are included in only one input file (inpython.cfg). By running the script, all the necessary steps (MESHER, SOLVER and Post-Processing) will be done automatically.

All you should do to run PyAxi for an input file (inpython.cfg) and station list (STATIONS) is:

```
$ python PyAxi <inpython.cfg> <STATIONS>
```

and the rest should be done automatically. inpython.cfg is a conguration file that contains all the AXISEM options. To change the input file, open inpython.cfg with an editor. However, you could find some already prepared inpython.cfg files for the events in Virtual-box. (refer to APPENDIX-2 for more information, Figure A2 *INPUT_FILES*) Therefore, to run the AXISEM for the provided events (EVENT-1 and IASP91-5sec as an example), it is enough to replace:

```
<inpython.cfg >:
~/Desktop/EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/IASP91_5sec/INPUT_FILES/inpython.cfg

<STATIONS>:
~/Desktop/EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/IASP91_5sec/INPUT_FILES/STATIONS
```

APPENDIX-4: Run AXISEM for EVENT-1

In this appendix, we follow the steps in the main tutorial and show the commands and outcomes for each step. For this reason, we focus on one of the events, EVENT-1.

Start from within the ~/Desktop/EVENTS/SCRIPTS directory:
1. Plot one of the events (listed in EVENTS directory, we choose EVENT-1 in this example):

```
$ plot_station_event_distribution.py EVENTS/EVENT-1
```

* For more information on the folder structure of your Virtual-Box, refer to Appendix-2.

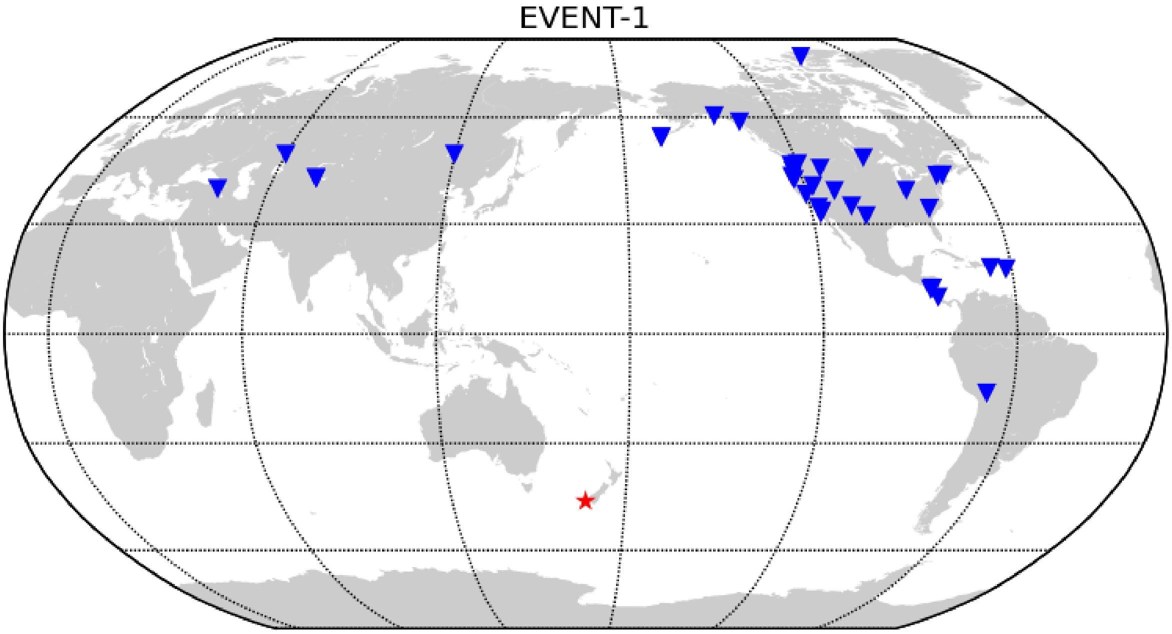

Figure A2: Event-station configuration for EVENT-1

2. To get an overview on both real data and pre-simulated seismograms: (e.g. PREM_ANISO for 5 seconds dominant period) [Figure A3]

```
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec
```

3.Cut the waveforms for Pdiff and PKiKP seismic phases and plot the seismograms: (compared with real data)

```
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec PKiKP
```

4. Change the filter in *plot_seismograms.py* script and compare the results with *SPECFEM3D*: (for changing the filter, open *plot_seismograms.py* and change values at top of the file)
For this example, ????? we change hfreq (high frequency) in line 34 to 0.05Hz (20sec) and *lfreq* to 0.01Hz: (Figure A5)

```
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff specfem3d
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec PKiKP specfem3d
```

5. Compare the results of two different background models (*PREM_ANISO_5sec* with *IASP91_5sec)* for Pdiff phase: (Figure A6).

```
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/IASP91_5sec
```

6. Change the filter, as explained in step 4, and repeat step 5. Here, we increase the *hfreq* to 0.2Hz and *lfreq* to 0.05Hz (Figure A7).
7. Compare the seismograms calculated for two different source parameters (PREM_ANISO_5sec and PREM_ANSIO_5sec_GCMT) for Pdiff phase: (For more information about the source parameters, refer to Appendix-1) [Figure A8]

```
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff
EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec_GCMT
```

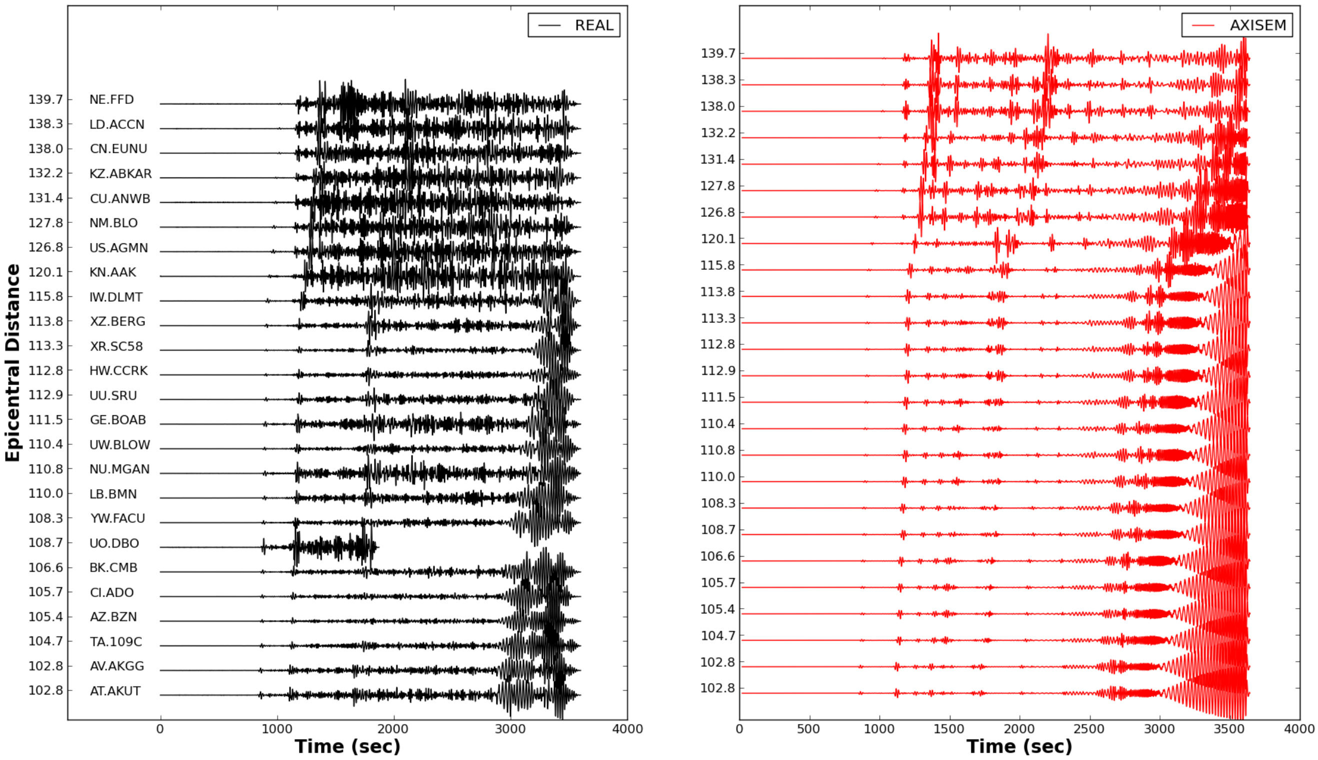

Figure A3:Real and AXISEM waveforms for EVENT-1

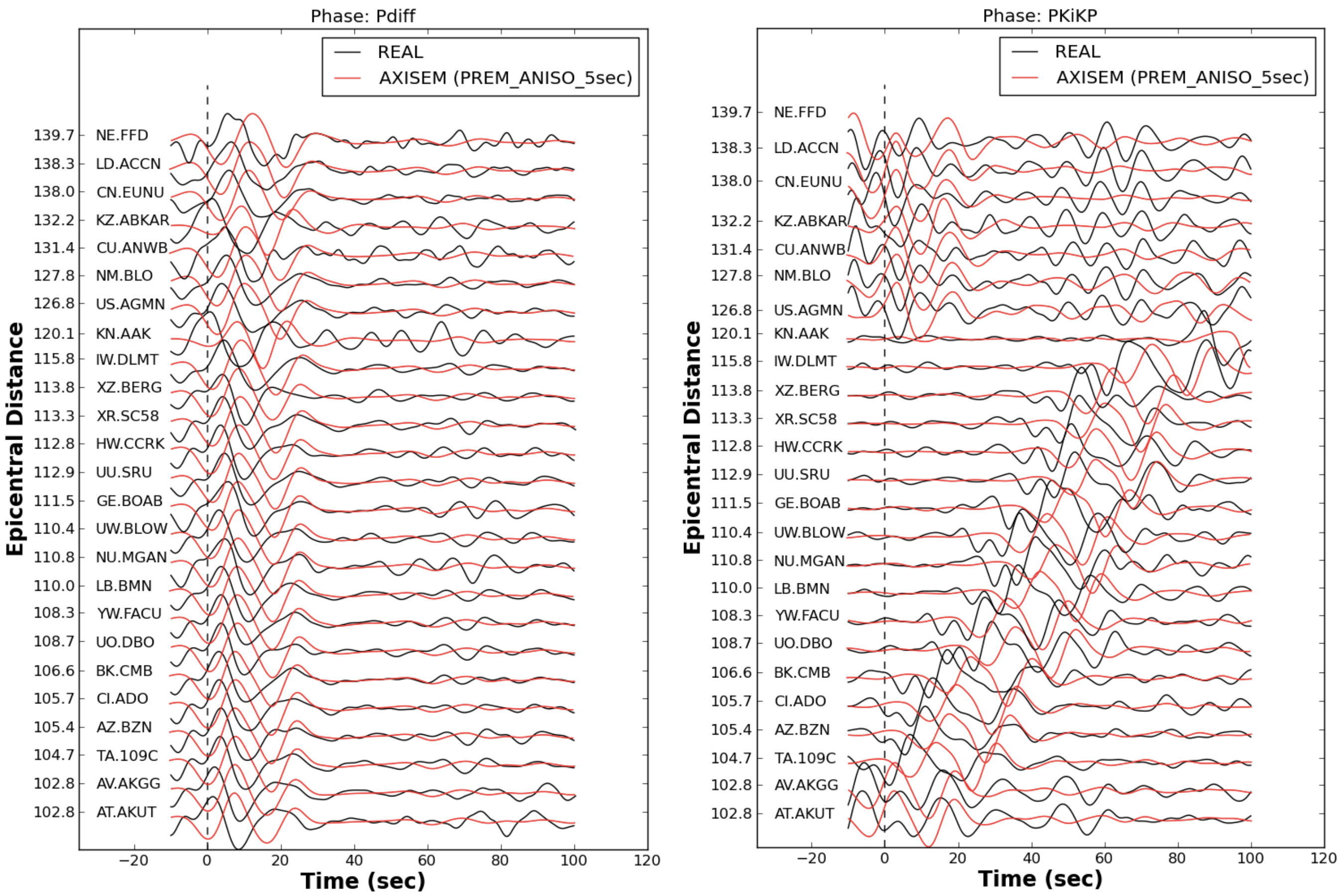

Figure A4: Comparison between real and synthetic waveforms for Pdiff and PKiKP phases.

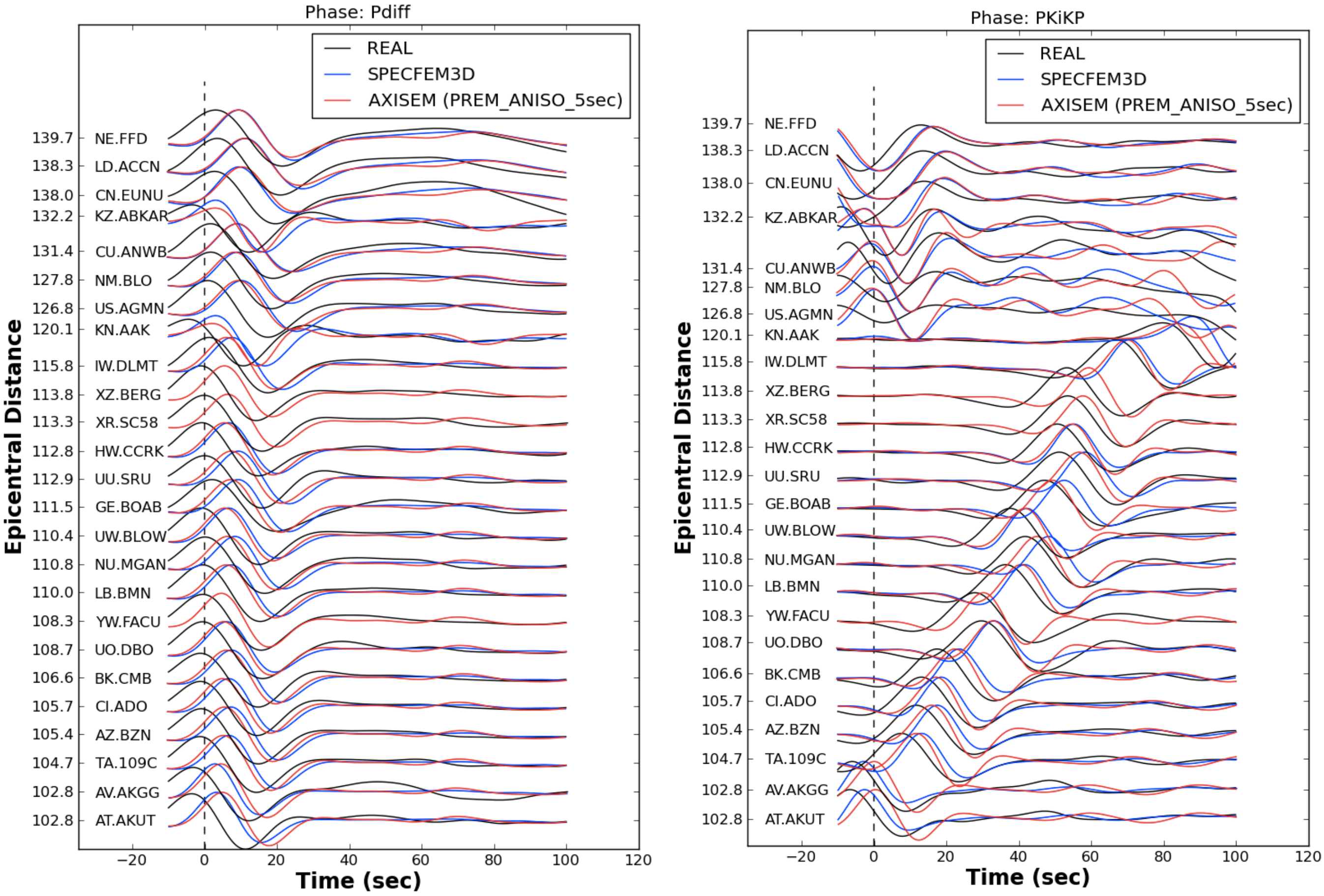

Figure A5: Comparison between real, AXISEM and SPECFEM3D waveforms for Pdiff and PKiKP phases.

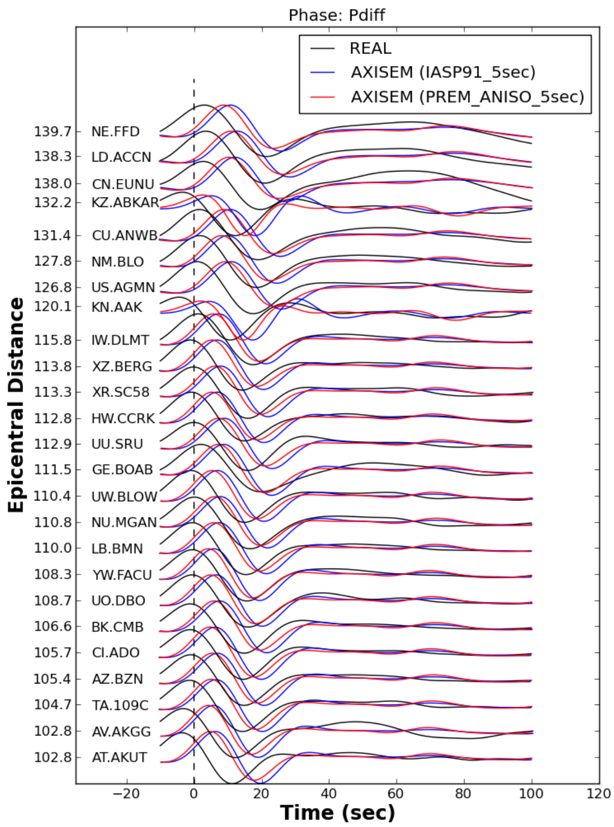

Figure A6: Comparison between real and AXISEM waveforms for two different background models (Pdiff).

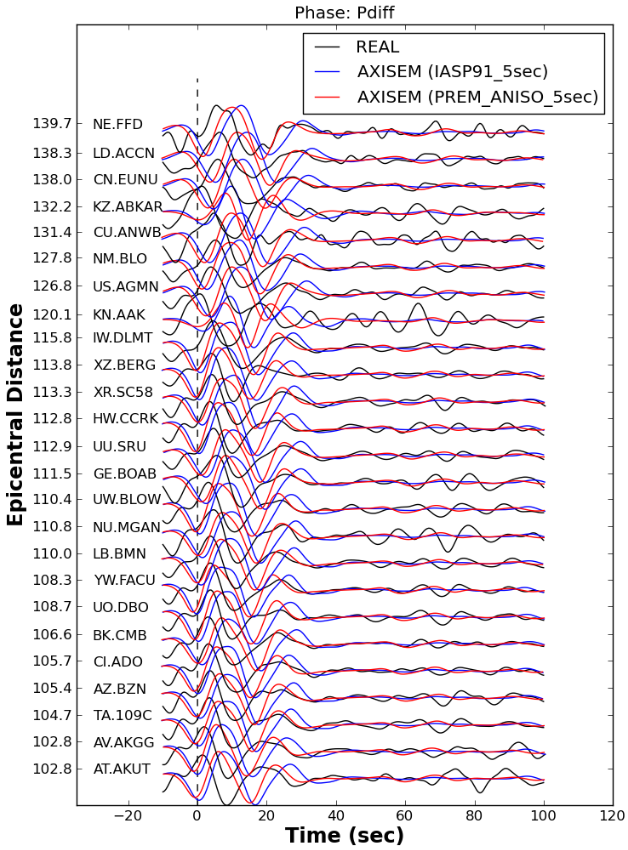

Figure A7: Comparison between real and AXISEM waveforms for two different background models (Pdiff).

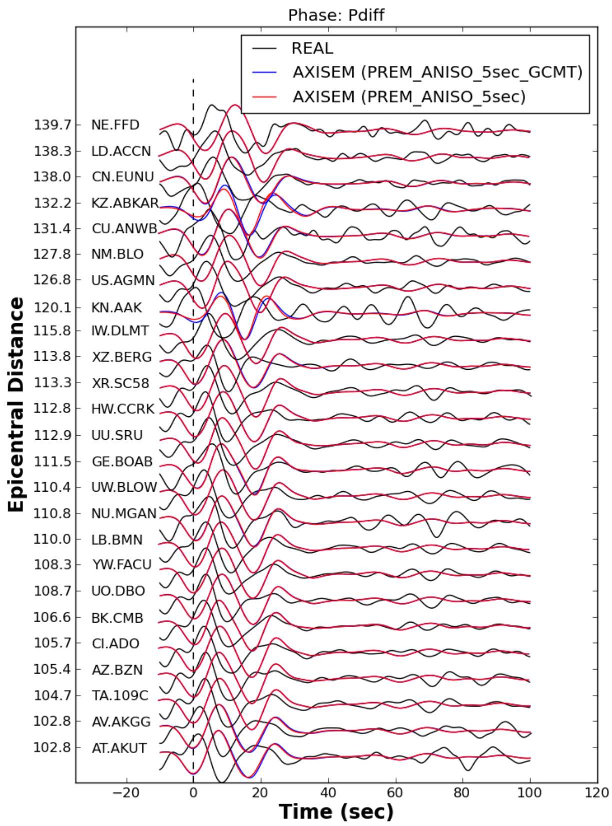

Figure A8: Comparison between real and AXISEM waveforms for two different source parameters (Pdiff).

8. Change the filter, as explained in step 4, and repeat step 7. ????? Here, we decrease the *hfreq* to 0.05Hz and *lfreq* to 0.01Hz. (Figure A9)
9. Find the time shift between the synthetics and real data, shift the synthetics accordingly and plot the results: (Figure A10)

```
$ plot_seismograms.py EVENTS/EVENT-1/AXISEM_PRE_SIMULATED/PREM_ANISO_5sec Pdiff shift_synt
```

10. Map the calculated time shifts in step 9 on the station locations: (note that it always plots the results of the latest comparison, i.e. Pdiff here)

```
$ plot_travel_time_map.py EVENTS/EVENT-1
```

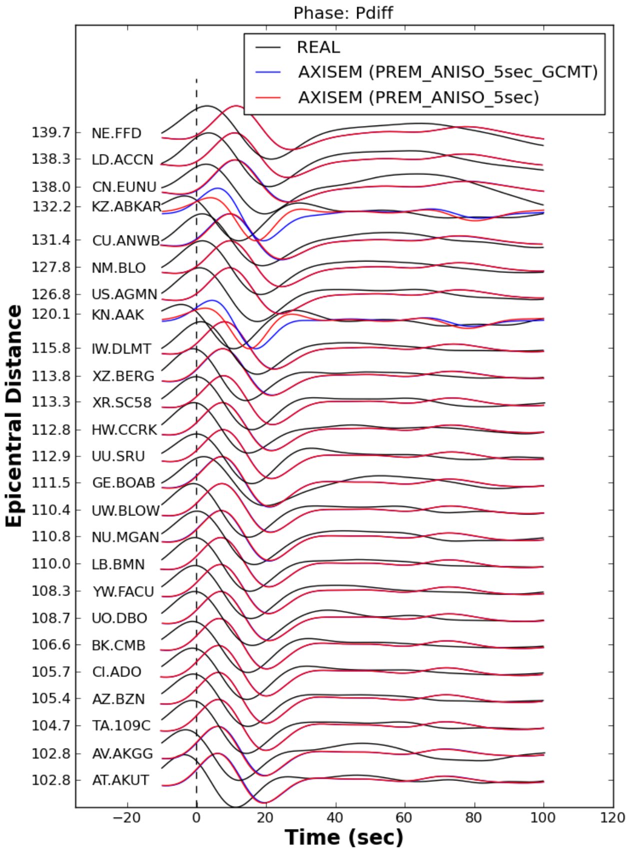

Figure A9: Comparison between real and AXISEM waveforms for two different source parameters (Pdiff).

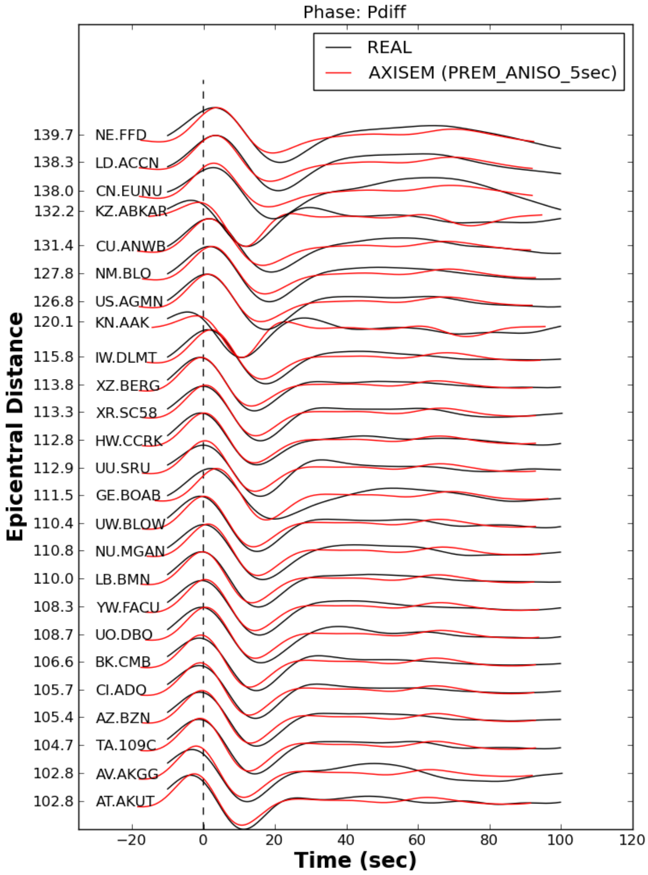

Figure A10: Comparison between real and AXISEM waveforms for Pdiff. AXISEM waveforms are shifted in order to gain the maximum cross correlation coefficient.

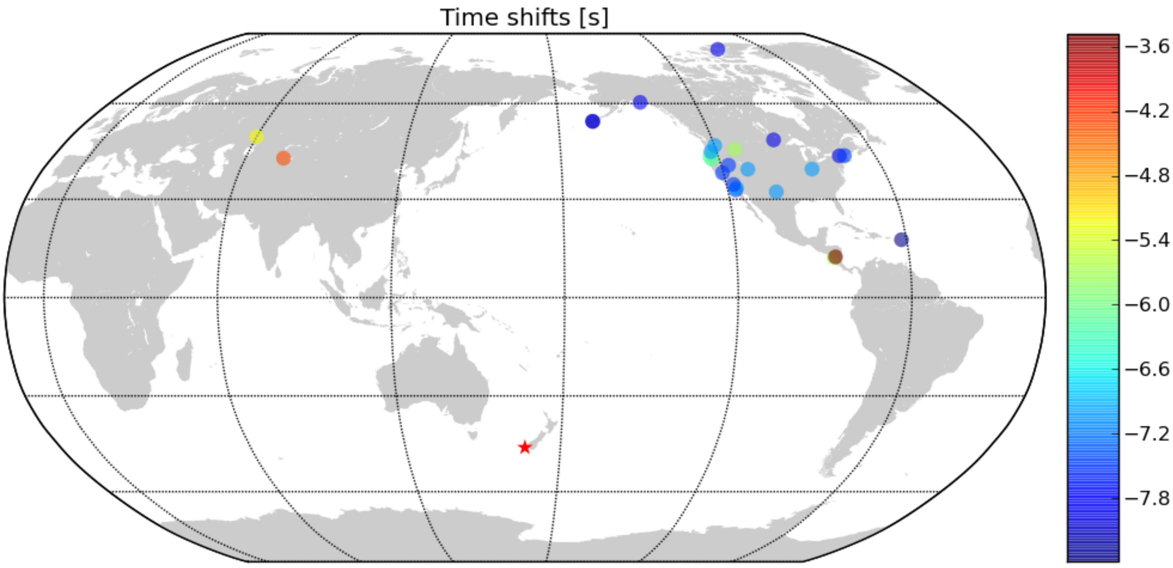

Figure A11: Time shift calculated by cross correlation between real and AXISEM waveforns. Blue color indicates that Pdiff in real data arrived sooner than that in the synthetic one.

APPENDIX-5: Retrieving real data and SPECFEM3D seismograms automatically

obspyDMT (ObsPy Data Management Tool) is a command line tool for retrieving, processing and management of massive seismological data in a fully automatic way which could be run in serial or in parallel.

This tool is developed to mainly address the following tasks automatically:
1. Retrieval of waveforms (MSEED or SAC), response files and metadata from IRIS and ORFEUS (via ArcLink) archives. This could be done in *serial* or in *parallel* for single or large requests.
2. Supports event-based and continuous requests.
3. Extracting the information of all the events via user-defined options (time span, magnitude, depth and event location) from IRIS and EMSC (European Mediterranean Seismological Centre).
4. Updating the existing archives (waveforms, response files and metadata).
5. Processing the data in *serial* or in *parallel* (e.g. *Tapering, removing the trend of the time series, filtering and Instrument correction*).
6. Management of large seismic datasets.
7. Plotting tools (events and/or station locations, Ray coverage (event-station pair), epicentral-distance plots for all archived waveforms and seismicity maps).

Here, we use obspyDMT to retrieve both real data and SPECFEM3D seismograms. For more information about this tool please refer to the following webpage:
https://github.com/kasra-hosseini/obspyDMT

obspyDMT is installed on your virtual machine. By running the following commands, the real data used in this tutorial can be retrieved automatically:

Event-1:

```
./obspyDMT.py --datapath EVENT-1_real --min_date 2009-07-15 --max_date 2009-07-16
--min_mag 7.0 --min_depth 20 --list_stas ~/Desktop/EVENTS/EVENT-1/INFO/STATIONS
--offset 3600 --req_parallel --arc N
```

Event-2:

```
./obspyDMT.py --datapath EVENT-2_real --min_date 2009-09-30 --max_date 2009-10-01
--min_mag 7.0 --min_depth 70 --list_stas ~/Desktop/EVENTS/EVENT-1/INFO/STATIONS
--offset 3600 --req_parallel --arc N
```

Event-3:

```
./obspyDMT.py --datapath EVENT-3_real --min_date 2006-10-15 --max_date 2006-10-16
--min_mag 6.0 --min_depth 20 --list_stas ~/Desktop/EVENTS/EVENT-1/INFO/STATIONS
--offset 3600 --req_parallel --arc N
```

Moreover, the SPECFEM3D seismograms can be also retrieved in the same manner:

Event-1:

```
./obspyDMT.py --datapath EVENT-1 --min_date 2009-07-15 --max_date 2009-07-16
--min_mag 7.0 --min_depth 20 --list_stas ~/Desktop/EVENTS/EVENT-1/INFO/STATIONS
--specfem3D --offset 3600 --req_parallel --arc N
```

Event-2:

```
./obspyDMT.py --datapath EVENT-2 --min_date 2009-09-30 --max_date 2009-10-01
--min_mag 7.0 --min_depth 70 --list_stas ~/Desktop/EVENTS/EVENT-1/INFO/STATIONS
--specfem3D --offset 3600 --req_parallel --arc N
```

Event-3:

```
./obspyDMT.py --datapath EVENT-3 --min_date 2006-10-15 --max_date 2006-10-16
--min_mag 6.0 --min_depth 20 --list_stas ~/Desktop/EVENTS/EVENT-1/INFO/STATIONS
--specfem3D --offset 3600 --req_parallel --arc N
```