Manual of ASWMS

Ge Jin[1] and James Gaherty[1]

[1]Lamont Doherty Earth Observatory, Columbia University

May 16, 2014

Contents

1 Introduction

This package is developed to measure the phase and amplitude of surface waves from the raw waveforms, and to generate phase velocity maps via Eikonal and Helmholtz equations. The detailed theoritical development of this package can be found in Jin and Gaherty, 2014.

From Section 2 to Section 5 we present you a quick work flow to set up this system on your computer, and adjust the parameters to fit your project. This setup should give a good initial result for most teleseismic studies with the array size ranging from 100 km to continual scale, and with the station spacing smaller than 100 km. You can refer to Section 7 if you want to make more customized adjustment for better result.

In this manual, the names of files and folders are written in blue and the names of variables are written in orange. All the adjustable parameters in this package are stored in the setup_parameters.m, and the input data should be put in the folder eventmat.

The ASWMS package is written in Matlab, and requires these toolboxes:

```
Curve Fitting Toolbox (for lsqcurvefit.m)
Signal Processing Toolbox  Statistics Toolbox
Mapping Toolbox
Statistics Toolbox
```

The Statistics Toolbox is not needed if users download the NaN suite and include all of its *.m files in the /matgsdf directory.

`http://www.mathworks.com/matlabcentral/fileexchange/6837-nan-suite/content/nansuite.`

2 Data Preparation

Users can either use existing SAC files and convert them to the required .mat format (Section 2.2) or use the data_download.m included in the ASWMS package which will download and format data using irisFetch.m (Section 2.3).

2.1 Data Structure

The input data are event-based matlab structure saved in mat files, which should be put in the folder eventmat. The filename for each event should follow the rule of YYYYMMDDhhmm_comp.mat, for example: 200901081921_LHZ.mat, and the structure name has to be event.

Here shows an example of this structure:

```
event =
        evla:  10.2152
        evlo:  -84.2159
       otime:  6.3399e+10
          id:  '200901081921'
    otimestr:  '08-Jan-2009 19:21:34'
     stadata:  [1x654 struct]
```

In which the field otime is the number of seconds to a certain time, which the actual time is not critical. We used 1970-01-01 in the sac version, and 0000-01-01 in the IRIS version, but it has to be consist in the whole project.

The field stadata contains the waveform and megadata from each stations, here is an example:

```
event.stadata(1) =

      stla:  32.8920
      stlo:  -116.4223
      stel:  1.8750
      dist:  4.1447e+03
     otime:  6.3399e+10
     delta:  1
      data:  [7200x1 double]
       cmp:  'LHZ'
      stnm:  'MONP2'
```

The stadata.otime has to have the same origin time as the event.otime, and has to have the unit in seconds. The stadata.dist has the unit of km, and delta is the sample interval in seconds.

Rather than writing your own code to transfer your data into the matlab structure, we provide two alternative methods.

2.2 Using SAC files

The script sac2eventmat.m can be used to transfer SAC files into the eventmat files.

The input SAC files have to be stored in the folder sacdata. Each event should have its own folder with the folder's name as YYYYMMDDhhmm. The file name should include the components name defined in setup_parameters.m and end with ".sac". Here shows an example:

```
>>ls sacdata
200801011855
200801051101
200801051144
200801070312
...
...
>>ls sacdata/200801011855
200801011855.TA.G09A.LHZ.sac
200801011855.TA.G10A.LHZ.sac
200801011855.TA.G11A.LHZ.sac
200801011855.TA.G14A.LHZ.sac
...
...
```

The origin time of each sac file has to be the origin time of the earthquake, while the waveform starting time can be different. Each sac file should have the following information in its header:

Event information: NZYEAR NZHOUR NZMIN NZSEC NZMSEC EVLA EVLO EVDP

station information: STLA STLO STEL KSTNM

data information: B DELTA KCMPNM

An ASCII file named eventlist with all the names of event folders should be put in the sacdata folder as well. An example of this file is:

```
>>cat sacdata/eventlist
200801011855
200801051101
200801051144
200801070312
...
...
```

You can easily generate this file by using the shell pipe function.

2.3 Using IRIS DMC Service

The ASWMS package can directly look up event information and download waveform data from IRIS DMC by using the matlab script data_download.m. This script utilizes the DMC's Matlab interface irisFetch.m. For more details of this service please visit IRIS website:

`http://www.iris.edu/dms/nodes/dmc/software/downloads/irisFetch.m`, or Google: irisFetch.

The following parameters in the setup_parameters.m should be adjusted if you want to use this service.

```
% parameters for data downloading (if using IRIS DMC)
parameters.start_time = '2009-01-07 00:00:00';
parameters.end_time = '2009-06-08 00:00:00';
parameters.is_use_timestamp = 1;
parameters.network = '_US-ALL';
parameters.minMw = 6;
parameters.maxdepth = 50;
parameters.datalength = 7200;  % in second
parameters.resample_delta = 1; % in second
```

If the parameters.end_time is empty (''), then the end time of the data fetching will be 4 days before the current date. If the is_use_timestamp is 1, then after each successful run of the data_download.m, it will save the end time of this run into a mat file named tempstamp.mat. So the next time data_download.m is triggered, it will start from where the last run left off. This option is useful for setting up a self-updated system.

The station network can be '*' to fetch all the channels available in the region.

In the case of LHZ component is not available, the user can download the BHZ component and resample the data with the sample interval defined by parameters.resample_delta.

The downloaded data are stored in the folder datacache first before they are further processed. It skips the files already existing in this folder to avoid repeat download and save running time (actually, most of the scripts in this program have this feature). Then the station responses are removed, and the data are organized and transformed into the event matlab structure in the folder eventmat.

3 Parameter Adjustment

All the adjustable parameters are defined in the different sections in the setup_parameters.m, and most of them are under the hood and do not need to be changed for most of the teleseismic based projects.

Here is a list of the key parameters that should be changed for each project, with their initial values chosen for the USArray project:

```
parameters.proj_name = 'USArrayExample';
parameters.component = 'LHZ';   % determined by filenames
parameters.lalim=[25 50];
parameters.lolim=[-125 -65];
parameters.gridsize=0.3;   % in degrees
```

where the proj_name can be any string. The component should be included in the sac filenames (if you are using sac as input); it is also shown in the names of the output files.

It is recommanded to be “LHZ” for teleseismic Rayleigh-wave projects. The lalim, lolim and gridsize define the bounds of the tomorgraphy grid and output files and figures. The final output grid will be defined as:

```
xnode = lalim(1):gridsize:lalim(2);
ynode = lolim(1):gridsize:lolim(2);
[xi yi] = ndgrid(xnode,ynode);
```

Another group of parameters that frequently need to be adjusted are the periods and the smoothing weight in each period (smweight_array). The periods are defined as:

```
parameters.periods = [20 25 32 40 50 60 80 100];  % in seconds
```

which are the central frequencies of the narrow-band filters. The width of the narrow-band filters around 10% of the central frequency and are defined by min_width and max_width.

And the smoothing weight is defined for each period as:

```
parameters.smweight_array = 3*[0.4 0.3 0.2 0.2 0.2 0.5 1 2];
```

It should have the same length as parameters.periods, with smaller value for the frequency bands with higher SNR or short wavelengths. More details on the smoothing parameters can be found in the Section 7. The example presented here shows a good ratio for the periods listed above, and you may only need to change the constant (3 in this case) for your project.

For most projects, changing these parameters should be able to provide you a good initial result. We will discuss other adjustable parameters in the Section 7.

4 Run the Scripts

After the data have been prepared and parameters adjusted, you can start to run this program and get an initial result. The main sequence of the commands is listed in main_driver.m. You can go ahead and run it after you modify it by choosing one of the data preparation methods.

Here we briefly list these commands and describe their function and input/output. We will discuss them in details in the Section 7.

1. sac2eventmat.m OR data_download.m
   choose one of your data preparation method, or build your own.
   Input: sac files/NaN Output: eventmat/*.mat

2. cleanup_events.m
   This script finds and deletes the events in the eventmat folder that are close in time whose surface waves may interfere with each other at the array location.
   Input: eventmat/*.mat Output: NaN

3. run_autowinpick.m
   This script will define the window function to isolate the energy of surface waves. It

creates a new field winpara in the event structure, and also generates a ASCII file in the folder winpara. If you want the window to be recalculated, you need to delete the files in winpara.
Input: eventmat/*.mat Output: winpara/* eventmat/*.mat

4. gsdfmain.m
   This script measures the phase delay between nearby stations using cross-correlation. It is the core part of the entire program. For each event, it generates an structure named eventcs and saves it to the folder CSmeasure.
   Input: eventmat/*.mat Output: CSmeasure/*.mat

5. eikonal_eq.m
   This script performs the tomography inversion via Eikonal equation, based on the cross-correlation measurements. It generates a structure named eventphv, which contains the apparent phase velocity maps for each event, and save it to the folder eikonal.
   Input: CSmeasure/*.mat Output: eikonal/*.mat

6. stack_phv.m
   This script stacks all the events in the folder eikonal, output a structure avgphv containing stacked apparent phase velocity map, and saves it to the file eikonal_stack_LHZ.mat. It would be the final result if the amplitude correction is not applied.
   Input: eikonal/*.mat Output: eikonal_stack_LHZ.mat

7. helmholtz_eq.m
   This script applies the amplitude correction on the apparent phase-velocity results via Helmholtz equation. It generates a structure named helmholtz, which is stored in the folder helmholtz.
   Input: eikonal/*.mat CSmeasure/*.mat eikonal_stack_LHZ.mat Output: helmholtz/*.mat

8. stack_helm.m
   This script stacks the corrected phase velocity maps from each event and generates the final result, as a structure named avgphv, and save it to the file helmholtz_stack_LHZ.mat.
   Input: helmholtz/*.mat Output: helmholtz_stack_LHZ.mat

5 Reviewing the Result

5.1 Output Format

The final result is stored in helmholtz_stack_LHZ.mat. If helmoltz_eq.m was not run/commented out in main_driver.m, then the final results are in eikonal_stack_LHZ.mat. However, the structures avgphv in these two files have the same structure, only the one in eikonal_stack_LHZ.mat does not contain the correction related fields. Here we only present the structure in the helmholtz_stack_LHZ.mat. Each of the binary *.mat output files used for figures are also given as ASCII *.xyz files.

Here shows an example of avgphv:

```
avgphv(4) =

          sumV: [84x201 double]
      sumV_cor: [84x201 double]
     sumweight: [84x201 double]
        GV_std: [84x201 double]
    GV_cor_std: [84x201 double]
      eventnum: [84x201 double]
            xi: [84x201 double]
            yi: [84x201 double]
         xnode: [1x84 double]
         ynode: [1x201 double]
        period: 40
        GV_cor: [84x201 double]
            GV: [84x201 double]
```

where the important fields are:

xi Latitude of the grid.

yi Longitude of the grid.

GV Phase velocity before amplitude correction.

GV_cor Phase velocity after amplitude correction.

eventnum Number of events each grid stacked.

5.2 Result Visualization

There are map plotting scripts contained in the stack_phv.m and stack_helm.m, which can be actived by changing the variable isfigure to 1 at the beginning of these two files. You can also refer them to plot your own figures. Here shows an simple example of plotting the result:

```
load helmholtz_stack_LHZ.mat
ip = 4;  % plot the 40s result
figure(88)
clf
ax = worldmap(lalim,lolim)
surfacem(avgphv(ip).xi,avgphv(ip).yi,avgphv(ip).GV_cor);
% set the color scale
cmap = colormap('jet');
cmap = flipud(cmap);
colormap(cmap);
% set the color range
r = 0.1;  % 20 % peak to peak
meanphv = nanmean(avgphv(ip).GV_cor(:));
```

```
caxis([meanphv*(1-r) meanphv*(1+r)]);
colorbar
```

.mat output files are also available as ASCII .xyz files for users choosing to plot results themselves.

Another package is available to generate a good summary of the result and database in the format of HTML. This package, named GSDF-Report, can be downloaded here:

```
https://github.com/jinwar/gsdf_report
```

After setting up the path of the ASWMS package in the setup_parameters.m, user can run the main_driver.m script to generate the HTML files in the folder htmls.

5.3 Adjust Parameters and Re-run the result

Depends on your result, you may need to adjust the parameters like cross-correlation distance (parameters.maxstadist), smoothing weight (parameters.smweight_array) and others. However, the package is written in a way that it will pick up where was left since last run and continue with it. In order to generate new results, you may need to delete all the files in the winpara (for run_autowinpick.m), CSmeasure (for gsdfmain.m), eikonal (for eikonal_eq.m), helmholtz (for helmholtz_eq.m) folders, depending on from which step you want to redo. The csh script cleandata.csh in the package can help you quickly reset your project folder.

Because the data_download.m takes a long time to recover, it is suggested to have it commented out once the data download is finished.

6 The Limitation of ASWMS

6.1 Array Geometry

The major limitation of this package is on the array geometry. In general, it should be applied on a 2D near-even distributed array with average station spacing smaller than the wavelength of surface in the interested highest frequency.

It also requests the cross-correlation paths to have ray crossing. The user can adjust the cross-correlation distance to increase ray crossing. However, if the cross-correlation distance is more than 3-4 wavelength, it is likely to generate cycle-skipping problem at high frequencies.

Figure 1 demonstrates these different situations.

6.2 Incorrect Instrument Response

Many arraies contain multiple types of sensors, and their instrument response stored in the IRIS DMC may not be correct. This may generate problems in the amplitude correction step

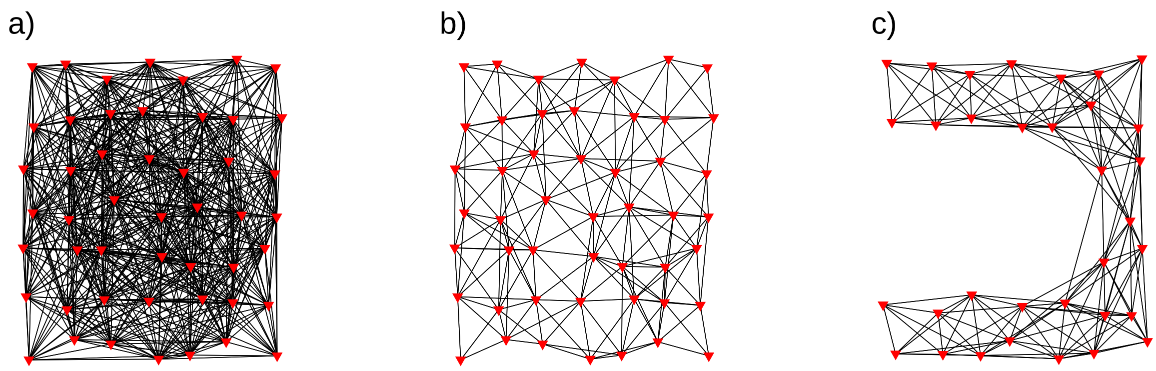

Figure 1: *Different array geometries and cross-correlation distance a) The ideal array geometry and ray density for ASWMS. b) The cross-correlation measurements are too sparse. parameters.maxstadist should be increased. c) Array geometry is not ideal for ASWMS.*

and bias the final helmholtz result. In this case, the amplitude correction is not suggested and the Eikonal result eikonal_stack_LHZ.mat is suggested to be final result.

7 Technical Details

In this section we will discuss the technical details in each component of the program, and provide the users more options to customize the package to fit their own projects.

7.1 run_autowinpick.m

The script run_autowinpick.m is used to define the time window to isolate the surface-wave energy. Figure 2 shows you an example of this function.

The notion of this window function selection is based on the estimation of the frequency-dependent group delays of surface waves within a defined group-velocity range. The location of the window function is linearly dependent on the epicenteral distance as:

$$T_1 = \frac{L}{v_1} + t1$$

$$T_2 = \frac{L}{v_2} + t2$$

where T_1 and T_2 are the beginning and ending time of the window function W_S, L is the epicentral distance, and v_1, v_2, t_1, t_2 are the parameters estimated by the linear regression.

The following parameters control the selection of this window function:

```
parameters.min_groupv = 2;
parameters.max_groupv = 5;
```

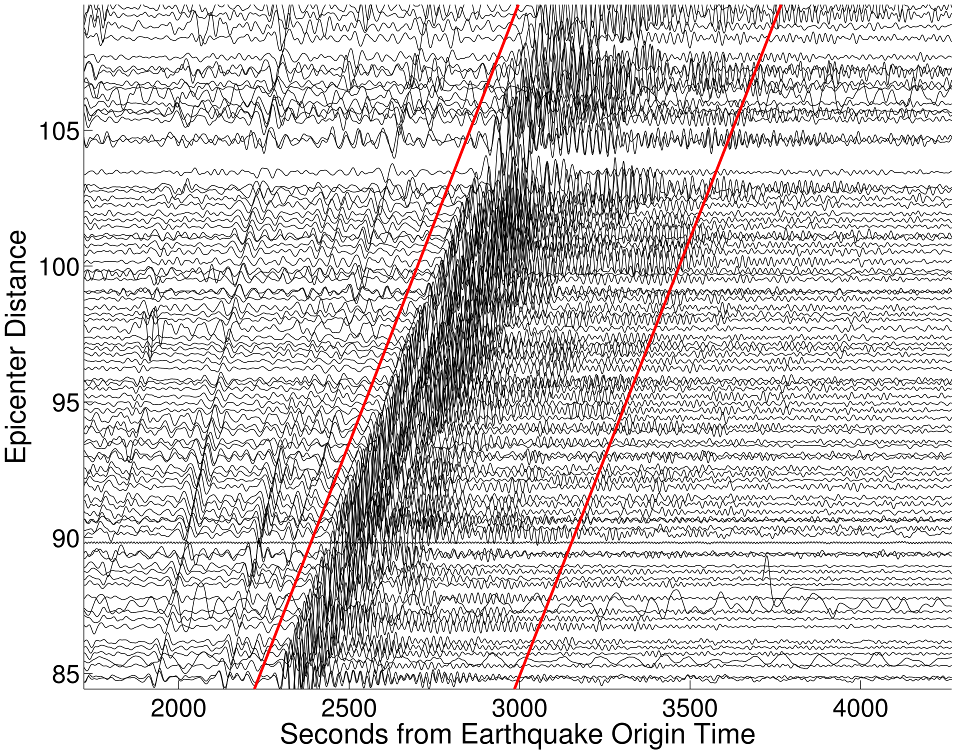

Figure 2: *Window function W_S (red line) to isolate the Rayleigh wave energy*

```
parameters.cent_freq = 0.025;
parameters.largest_epidist_range = 3000;
parameters.cycle_before = 2;
parameters.cycle_after = 5;
parameters.min_dist_tol = deg2km(20);
parameters.max_dist_tol = deg2km(160);
```

min_groupv : the minimum group velocity to search for.

max_groupv : the maximum group velocity to search for.

cent_freq : the frequency band that has the best signal to noise ratio. It should in the range of frequencies defined by parameters.periods.

largest_epidist_range : in most regional studies, you don't need to worry about this parameter. For some large arrays (e.g., USArray), the stations span a wide range of epicenter distance so the linear relation between group delay and epicentral distances break down. This parameter defines the largest epicentral distance range that can be processed. If it is exceeded, the program with select the epicentral distance range with the most stations and mark everything outside as bad stations.

cycle_before : the window function should include 1-2 cycles before the group delay (peak of the envelop function).

cycle_after : the number of cycles that the window function should include after the group delay.

min_dist_tol : minimum distance between the earthquake and the center of the array. Should be large enough to allow the separation between the body wave and surface wave.

max_dist_tol : maximum distance between the earthquake and the center of the array. Should be small enough to avoid the interference between R1 and R2.

Tips: in the winpara, there is a ASCII file generated for each event. It contains 4 numbers, which are the v_1, t_1, v_2, t_2 as in the equation. Something is wrong if most of your events have v_1 and v_2 being end-member values defined by the min_groupv and max_groupv.

7.2 gsdfmain.m

gsdfmain.m drives the cross-correlation measurements between the nearby stations, which is the core part of the whole package.

Figure 3 shows the waveforms from two nearby stations (90 km apart). After isolate the surface-wave energy of station 2 using window function W_S, the cross-correlation $C(t)$ is calculated between S_1 and $W_S S_2$. Then the $C(t)$ is narrow-band filtered and fitted to get the phase delay between these two stations (Figure 4).

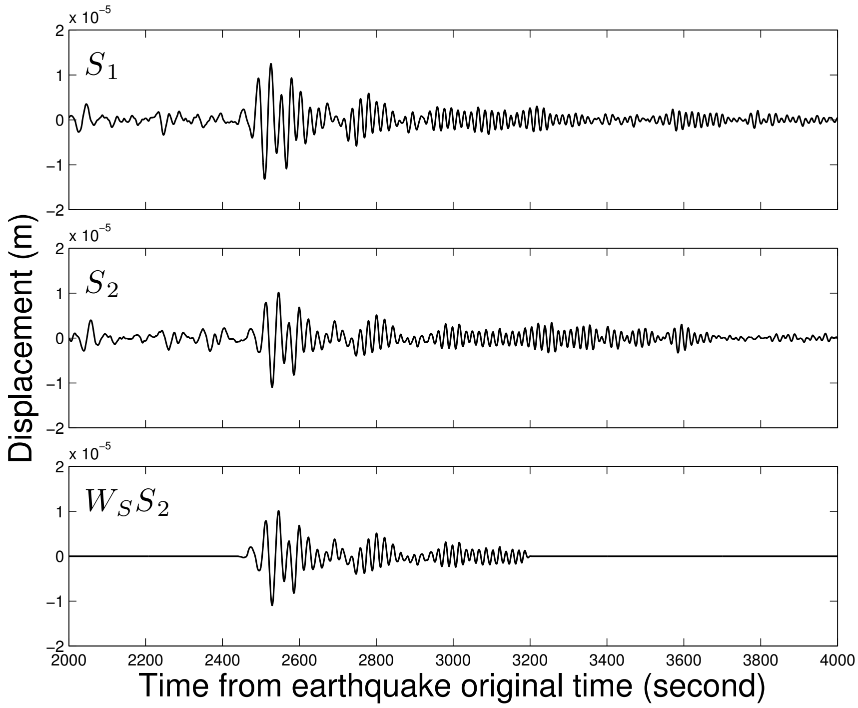

Figure 3: *Waveforms from two nearby stations*

The parameters that are adjustable in this section are:

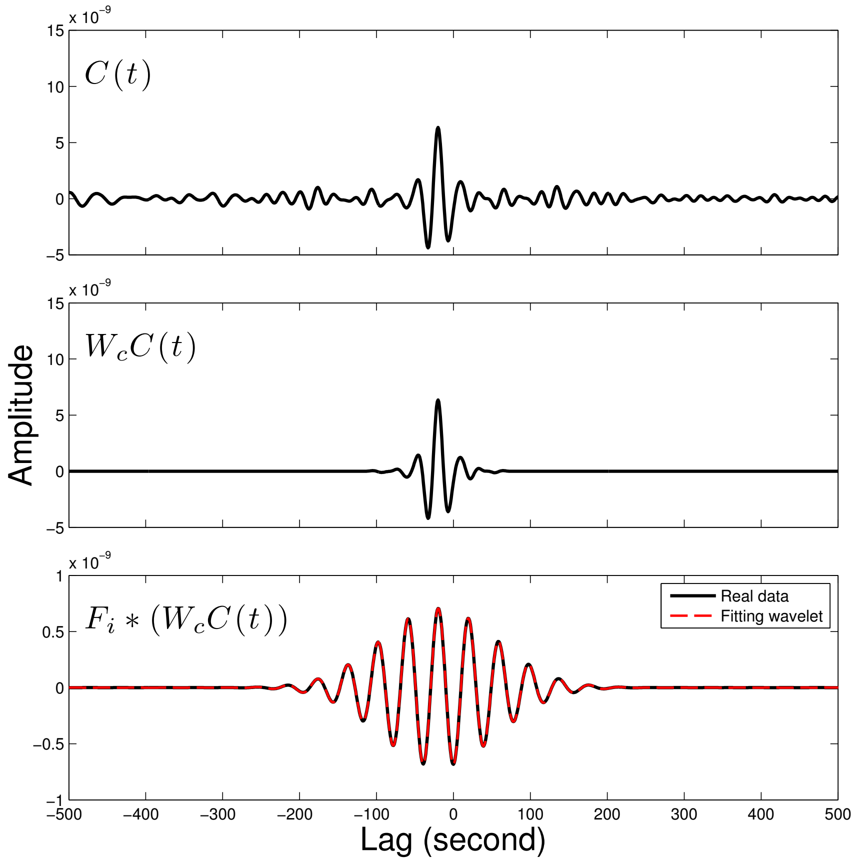

Figure 4: *Demonstration of the cross-correlation procedures*

```
parameters.minstadist = 5;
parameters.maxstadist = 200;
parameters.is_rm_resp = 0;
parameters.refv = 4;
parameters.refphv = ones(size(parameters.periods))*4;
parameters.min_width = 0.06;
parameters.max_width = 0.10;
parameters.wintaperlength = 30;
parameters.prefilter = [10,200];
parameters.xcor_win_halflength = 100;
parameters.Nfit = 2;
parameters.Ncircle = 5;
parameters.cohere_tol = 0.5;
parameters.tp_tol = 10;
```

minstadist : a small number just to avoid the station cross-correlate with itself.

maxstadist : the cross-correlation range. This distance should be at least twice of the average station spacing to get a good tomography result. However, should be smaller than 3-4 times of the surface wave wavelength at highest frequency.

is_rm_resp : you want to have it off (0).

refv : a rough estimation of the average group velocity at center frequency band (about 40s). It does not need to be accurate. 4 is a good number for all teleseismic projects.

refphv : a reference phase velocity to correct the cycle-skipping. It does not matter much if the station are close to each other comparing to the surface-wave wavelength. However, if the high frequency bands experiencing cycle-skipping problem, you may need to provide a more accurate number here.

min_width,max_width : define the band-width of the narrow-band Gaussian filters applied on the cross-correlation waveforms. The shape of Gaussian functions for the default setup (0.06 and 0.10) is shown in Figure 5.

wintaperlength : the taper length of the window function applied on the original waveform. Because usually this window function is a few hundred seconds long, no strong (long) taper is needed. (Figure 3)

prefilter : the pre-filter applied on the original waveforms. The band-pass should be wider than your interested frequency range.

xcor_win_halflength : the length of the cross-correlation window function. Should be longer than a few periods of your lowest frequency band. (Figure 4)

Nfit : Number of from the center used to fit the five-parameter wavelet. Usually 2 or 3 is good for all project.

Ncircle : Number of cycles searched for cycle-skipping. 5 is good for most of the teleseismic projects. For very high frequency dataset, you may need a larger number.

cohere_tol : One of the key parameters the user should pay attention. It defines the minimum coherence that is required between the two station waveforms to pass the data QC (quality control). We found in the teleseismic project, 0.5-0.6 is a good choice.

tp_rol : another QC parameter. For each event, we fit an average phase velocity for the entire array by assuming the straight ray path. And we verify the misfit of each phase delay measurement to this average phase velocity. All the measurements with the misfit larger than the tp_tol are discarded as bad measurements. This number is good for USArray. The user may change it to a smaller number for smaller arrays. (Figure 6)

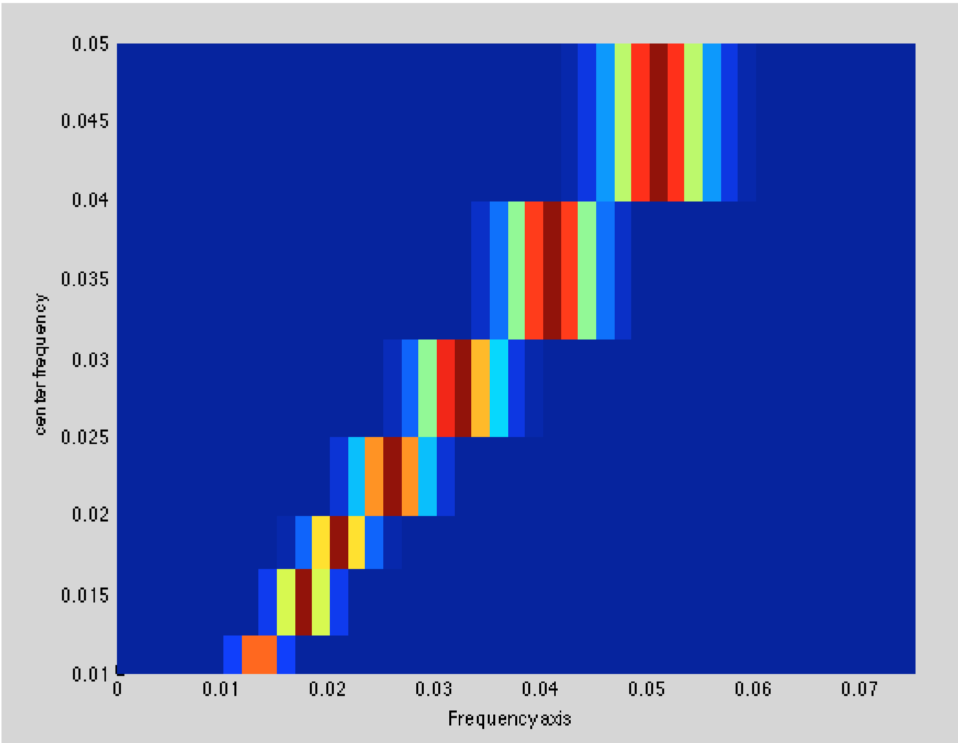

Figure 5: *The gains of Gaussian filters*

7.3 eikonal_eq.m

This script performs the slowness vector inversion based on Eikonal equation:

$$\delta\tau_p = \int_{r_i} \vec{S}(\vec{r}) \cdot d\vec{r}$$

where $\delta\tau_p$ is the phase delay measurements from gsdfmain.m. The parameters to set up in this sections are:

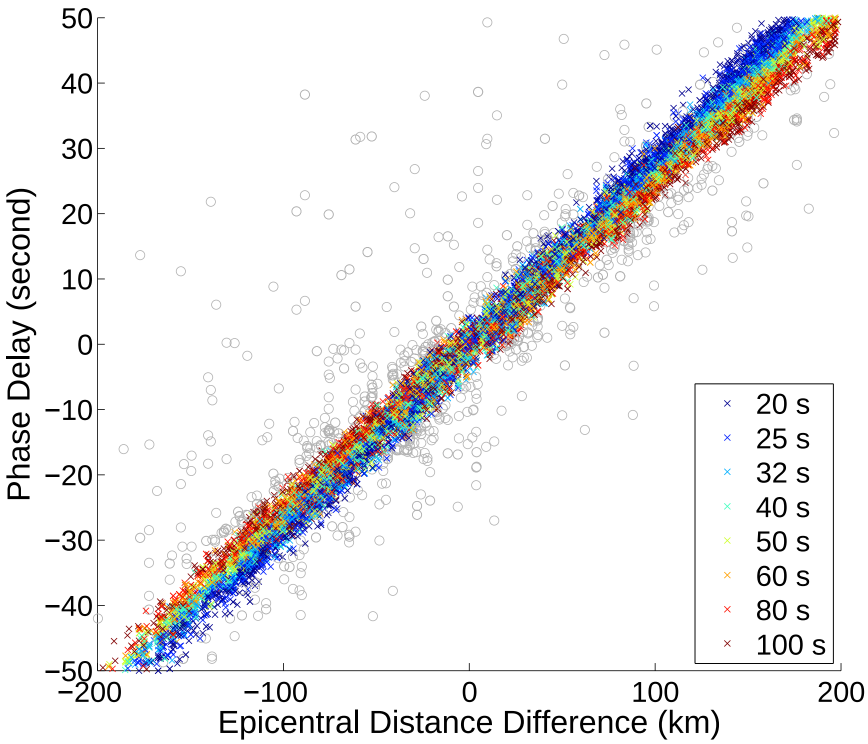

Figure 6: *Relative phase delays against the epicentral distance differences of all the station pairs within 200 km for the same records shown in Fig.2. Crosses with different color represent the measurements at different frequencies, and the grey circles represent the unqualified measurements that are discarded by choosing tp_tol. An faster move-out at the lower frequencies demonstrates the average phase velocity dispersion across the array.*

```
parameters.smweight_array = 3*[0.4 0.3 0.2 0.2 0.2 0.5 1 2];
parameters.raydensetol=deg2km(parameters.gridsize)*2;
parameters.Tdumpweight = 0;
parameters.Rdumpweight = 0;
parameters.fiterrtol = 3;
parameters.isRsmooth = 1;
parameters.dterrtol = 2;
parameters.inverse_err_tol = 2;
```

smweight_array : one of the parameter that should already be changed from project to project. This array of parameters control the smoothing weight of the phase velocity inversion at each period defined in parameters.periods. The ratio defined here is good for tele-seismic project. In most cases the user only need to change the constant before it.

raydensetol : the parameter controls the minimum ray density in each grid. If the ray density is smaller, the result of that grid will be set to NaN.

Tdumpweight : by putting a small number on it, you can force the wave propagates along the great circle path by forcing the projection of the slowness vector on the tangential direction to be zero. It may be useful in some extreme conditions, otherwise should be set to 0 to allow for ray bending.

Rdumpweight : you can also force the slowness on the radical component has the value of refphv. Again, it may be useful in some extreme conditions, otherwise should be set to 0 in most cases.

fiterrtol : error allowed in the wavelet fitting, please leave as it is.

isRsmooth : apply smoothing kernel on the NS-WE direction or on the RT direction. Should be 1 for most cases.

dterrtol : one of the QC parameters in this step. The slowness inversion is performed twice. After the first run, the misfits of the inversion are calculated for all the measurements, and the measurements with misfits larger than this value in seconds are discarded.

inverse_err_tol : the other QC parameter in this step. Same as dterrtol, but with the unit of standard deviations.

7.4 helmholtz_eq.m

This script applies amplitude corrections on the apparent phase velocity from eikonal_eq.m. It also reads in the stack_phv.m output so you may need to run stack_phv.m first, as described in the main_driver.m.

```
parameters.min_amp_tol = 0.1;
parameters.amp_var_tol = 2;
parameters.alpha_range = [1 1];
parameters.alpha_search_grid = 0.1;
```

min_amp_tol : QC parameter. The program will calculate the median amplitude of all the stations and discards the stations with amplitude smaller than the median amplitude times this value. (only the amplitude measurement is discarded. The phase measurements from these stations are still valid)

amp_var_tol : with the range defined by maxstadist, the median amplitude A_m among the stations is calculated. The stations with amplitude larger than A_m times or smaller than A_m divides this value are discarded.

alpha_range, alpha_search_grid : just leave as it is.

7.5 stack_phv.m and stack_helm.m

Scripts to stack the results from Eikonal tomography (eikonal/*) and Helmholtz tomography (helmholtz/*). In the beginning of both scripts, there is an variable isfigure can be changed to 1 if you want to have the result plotted.

The adjustable parameters here are:

```
parameters.min_csgoodratio=0.3;
parameters.min_phv_tol = 3;
parameters.max_phv_tol = 5;
parameters.is_raydense_weight = 1;
parameters.min_event_num = 10;
parameters.err_std_tol = 4;
parameters.issmoothmap = 1;
parameters.smooth_wavelength = 0.25;
```

min_csgoodratio : discard the events with the number of the good measurements smaller than the number of bad measurements times this value.

min_phv_tol : discard the grid of a event with phase velocity smaller than this value.

max_phv_tol : discard the grid of a event with phase velocity larger than this value.

is_raydense_weight : whether weight the stacking by raydensity. You may want to try both. For regions with large azimuthal anisotropy, it is suggested to be turned off.

min_event_num : for the final result, the grids with the number of events being stacked smaller than this value are set to NaN.

err_std_tol : the stacking is performed twice. After the first stacking, the difference between the phase velocity from single event and the stacked phase velocity is calculated, and the data points with the difference larger than this number times the standard deviations are discarded before the second stack.

issmoothmap : whether to perform one more step of smoothing for the final result. It is suggested.

smooth_wavelength : the range of the final smoothing is defined by this value times the average wavelength of the surface wave at each frequency.

8 Reference

Jin, G., and J. B. Gaherty (2014), Surface Wave Measurement Based on Cross-correlation , Geophys. J. Int, submitted.