# Users' manual for DSurfTomo (version 1.4)

Hongjian Fang (fanghj1990@gmail.com) and Huajian Yao (hjyao@ustc.edu.cn)

Last modified: July 18, 2023

## 1, Description

DSurfTomo is a surface wave tomography program that can directly invert surface wave dispersion data to 3D shear wave speed without the intermediate step of constructing the phase or group velocity maps. It adopts the fast marching method (FMM) (Rawlinson et al., 2004) to compute, at each period, surface wave travel times and ray paths between source and receiver pairs. Consequently, it considers the off-great-circle propagation effect of short-period surface waves, which suits better for shallow structure imaging. Users can refer to Fang et al. 2015 for a detailed description of the method. Note that to avoid complexity, the public version of the code implements conventional smoothing and damping regularizations rather than wavelet parameterization with sparsity constraints, as suggested in the paper.

## 2, Installation

This program has been tested successfully on CentOS, Debian, Ubuntu, and MacOS platforms with both gfortran and ifort compilers. The following shows the detailed steps.
First, make a directory for the DSurfTomo code and example:
*mkdir DSurfTomo*
Second, clone the source code from Github with
*cd DSurfTomo*
*git clone https://github.com/HongjianFang/DSurfTomo*
At last, install
*./configure*
After successful compiling, the executable file will be in the 'bin' directory.

## 3, Data preparation (e.g. 'SurfTomo.dat')

The format of dispersion data is as followed:

```
 #  25.148500 121.511100 1 2 0
  25.158529 121.476890 0.7990
  25.133539 121.499190 1.0420
 #  25.158529 121.476890 1 2 0
  25.119850 121.473190 0.6460
 #  25.128920 121.417420 1 2 0
  25.119850 121.473190 0.9430
 #  25.119850 121.473190 1 2 0
  25.090361 121.462250 0.8280
  25.083694 121.435220 1.0870
  25.133539 121.499190 1.3910
  25.102119 121.515930 1.2950
……
```

Lines beginning with '#' represent the sources, followed by source latitude, source longitude, period index (integer), wave type, and velocity type (explain later).
Each source is then followed by the receiver data: the first two columns are the latitude and longitude of the receivers, and the third column is the phase or group velocities (surface wave dispersion measurements).
Period index (integers starting from 1): index of the periods that are listed in the parameter file DSurfTomo.in (explain later)
Wave type (integer): **2 for Rayleigh wave and 1 for Love wave**
Velocity type (integer): **0 for phase velocity and 1 for group velocity**.

The code currently works best for Rayleigh wave phase velocity data. But different kinds of data, including phase or group velocity data for both Rayleigh and Love waves, can be incorporated to jointly invert the 3D isotropic shear velocity model. To that end, all data should be put into a single file with the order of Rayleigh wave phase velocity, Rayleigh wave group velocity, Love wave phase velocity, and Love wave group velocity. Each follows the format mentioned before.

Note:
In the example data set, a Python script "ExtractData.py" in the 'scripts' directory is provided to extract the data from the dispersion data files. Run '*python ExtractData.py > surfdataTB.dat'* to reformat the data. But you can use whatever works for you, as long as the data are in the right format.)
In version 1.4, There is another Python script '**GenerateDataAndInputFiles4DSurfTomo.py'** that can be used to generate both the input file and the dispersion data in the right format. But as always, some slight modifications are necessary, as different cases have different characteristics.

**4, The initial model (MOD)**
The file name of the initial model must be 'MOD', and the content needs to be formatted as:

```
  0.0  0.2  0.4  0.6  0.8  1.1  1.4  1.8  2.5
  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900
0.900  0.900  0.900  0.900
  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900
0.900  0.900  0.900  0.900
  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900
0.900  0.900  0.900  0.900
  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900  0.900
0.900  0.900  0.900  0.900
```
…...

The first line is the locations (in km) of grid points in the vertical direction.
From the second to the last line are shear velocity values, with latitude first, then longitude, followed by depth. Specifically, each row represents shear velocity values at different latitudes at a single longitude and a certain depth. Note that 3D initial models, if available, can be used for the inversion.

Note:
You can use the Python script 'GenerateIniMOD.py' in the 'scripts' directory to output a simple initial model (some modifications regarding the dimensions of the models and others are necessary). Then run '*python GenerateIniMOD.py*'. For 3D initial models, care must be taken to ensure the right order.

# 5, The input file for the inversion ('DSurfTomo.in')

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c INPUT PARAMETERS
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
surfdataTB.dat                  c: data file
18 18 9                         c: nx ny nz (grid number in lat lon and depth direction)
25.2  121.35                    c: goxd gozd (upper left point,[lat,lon])
0.015 0.017                     c: dvxd dvzd (grid interval in lat and lon direction)
20                              c: max(sources, receivers)
4.0  0.1                        c: weight damp
3                               c: sablayers (for computing depth kernel, 2~5)
0.5 2.8                         c: minimum velocity, maximum velocity (a priori information)
10                              c: maximum iteration
0.2                             c: sparsity fraction
26                              c: kmaxRc (followed by periods)
0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0
0                               c: kmaxRg
0                               c: kmaxLc
0                               c: kmaxLg
0                               c: synthetic flag(0:real data,1:synthetic)
0.02                            c: noiselevel
0.5                             c: threshold
```

The file is made as self-explanatory as possible. Some further explanations are:
The sixth line is the origin point in latitude and longitude, note the direction of latitude is from North to South, and from West to East for Longitude. One trick I use is to plot all sources and receivers to decide this origin point to make sure the inverted region contains all the sources and receivers. Otherwise, the program will complain with an error about 'sources outside the study region' and stop.

The seventh line is the grid interval in Latitude and Longitude.
Again, make sure that all the sources and receivers are inside the model region, i.e., their latitudes are all in the range from 'goxd-(nx-3)*dvxd' to 'goxd' and longitudes reside between 'gozd' to 'gozd+(ny-3)*dvzd'.

The eighth line is the maximum number of sources and receivers. For ambient noise tomography, it should be at least the number of stations.

The ninth line, "weight" is the balancing parameter between the data fitting term and the smoothing regularization term.
Note:
Sometimes it is tricky to choose an appropriate one since L-curve does not always work. In my experience, an appropriate weight will lead to reasonable velocity changes at each iteration. At the first few iterations, velocity change can reach 0.2-0.4 km/s. Too large velocity variation means the weight you choose is too small and vice versa. Generally, a few trials will be more than enough to choose a reasonable weight parameter. The velocity change will decrease in the last few iterations as the inversion converges, or the program will complain by outputting information like 'no roots can be found….'.
"damp" is the input parameter for 'lsqr', it controls the amplitude of the inverted parameter.
'sublayers' in the tenth line represents how many sublayers you want to transform from grids to layers in order to calculate the depth kernel. The default value is 3.

The minimum and maximum velocity in the eleventh line are a priori information you have about the study area. You can set a large interval if you don't have this information; it often doesn't affect the final result.
The twelfth line is the iteration number of the inversion. 10 iterations are usually more than enough. The sparsity fraction parameter means how sparsity the sensitivity matrix is, 2-10 percent will be enough for most cases.

KmaxRc, kmaxRg, kmaxLc, kmaxLg means the number of periods for Rayleigh wave phase velocity, Rayleigh wave group velocity, Love wave phase velocity, and Love wave group velocity, respectively. Note if the number of periods is zero, there's no need to write the periods following the number, e.g. the kmaxRg.

"synthetic flag" setting to 0 means inversion using real data, 1 for synthetic data; it needs "MOD.true" when the flag is 1, otherwise the program will stop.
Note:
You can also generate 'MOD.true' using 'GenerateTrueMOD.py' in the 'scripts' directory, the script is for the checkerboard, and modifications are necessary to control the number and size of 'checkers'. The command is '*python GenerateTrueMOD.py*'.

"noise level" means how much Gaussian noise you want to add to your synthetic data. For instance, 0.02 means the random Gaussian noise added has zero mean and a standard deviation of 2% of the synthetic data.
The 'threshold' in the last line represents a threshold with a value between 1 and 2, which is mainly to remove outliers in the data. Setting the threshold to 2 will remove residuals with values larger than 2 times the 75 percentile of all the residuals and those with values smaller than 25 percentile. Put a large value there, e.g. 2.0 or larger, if you have done strict quality control on your dispersion data.

Note:
In version 1.3, the value is chosen based on the expression '1.0/(1+0.05*exp(x**2*threshold))', where 'x' is the residual. The feature is depleted in version 1.4 since it is not straightforward.

After preparing these 3 files in a certain directory, simply run the program in the same directory with:
*../bin/DSurfTomo DSurfTomo.in*

## 6, Output files

The final velocity model (DSurfTomo.inMeasure.dat):
First column: longitude
Second column: latitude
Third column: depth
Fourth column: shear wave velocity

Note:
You can use 'plotslice.gmt' in the 'scripts' directory to quickly check the results with:
*csh plotslice.gmt DSurfTomo.inMeasure.dat depth1 depth2 depth3 depth4*
where depth[1234] is a certain depth from the vertical grids.
You can interpolate them to any depth you want and then plot with your own script.

The raypath distribution of the final model (raypath.out)
# number of ray path segments
latitude longitude
…
Note that in version 1.4, this file is depleted. Users interested in the ray path distribution can either go back to version 1.2 or use FMTOMO to trace the ray paths.

Velocity model at each iteration (DSurfTomo.inMeasure.dat.iter)
The same format as 'DSurfTomo.inMeasure.dat'. These files can be used to check the intermediate results during the running.

Residual of first and last iteration (residual*.dat)
The format is as followed:
Distance ForwardT ObserveT weightedForwardT weightedObserveT weight

The latest version of the code can be found on GitHub at https://github.com/HongjianFang/DSurfTomo, feel free to shoot emails or open issues on GitHub if you have any questions, we are happy to hear from you.

## References

Fang, H., Yao, H., Zhang, H., Huang, Y. C., & van der Hilst, R. D. (2015). Direct inversion of surface wave dispersion for three-dimensional shallow crustal structure based on ray tracing: methodology and application. Geophysical Journal International, 201(3), 1251-1263.
Rawlinson, N. & Sambridge, M., 2004. Wave front evolution in strongly heterogeneous layered media using the fast marching method, *Geophys. J. Int.,* **156**(3), 631–647