Users' manual for DSurfTomo (version 1.3)

Hongjian Fang (fanghj@mail.ustc.edu.cn) and Huajian Yao (hjyao@ustc.edu.cn)

School of Earth and Space Sciences, University of Science and Technology of China
Last modified: July 4, 2018

1, Description

DSurfTomo is a surface wave inversion program which can directly invert surface wave dispersion data to 3D shear wave speed without the intermediate step of constructing the phase or group velocity maps. The fast marching method (FMM) (Rawlinson et al., 2004) is used to compute, at each period, surface wave travel times and ray paths between sources and receivers. This avoids the assumption of great-circle propagation that is used in most surface wave tomographic studies, but which is not appropriate in complex media. Please refer to Fang et al. (2015 , GJI) for the detail description of the method.

2, Installation

This program has been tested successfully on CentOS, Debian, Ubuntu and MacOS platforms with gfortran/ifort compiler. Detailed steps are as followed.
First, make a directory to put the DSurfTomo code and example in with the command
mkdir DSurfTomo
Then, clone the source code from Github with
cd DSurfTomo
git clone https://github.com/HongjianFang/DSurfTomo
At last, install with
./configure
The executable file will be in the 'bin' directory after successful compiling.

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

Different kinds of data can be incorporated to jointly invert the 3D shear velocity model. 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 as mentioned before.

(In the example data set, I wrote a python script "ExtractData.py" in the script directory to extract the data from the dispersion data files. Run '*python ExtractData.py > surfdataTB.dat'* to prepare the data for this example. But you can use whatever works for you, as long as you prepare in the right format.)

4, The initial model (MOD)

The file name of the initial model must be 'MOD', the content looks like:

```
   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 locations (in km) of grid points in the vertical direction.
Then followed by shear velocity values. The order is altitude first, then longitude, followed by depth.
Each row represents shear velocity values at different latitude at a single longitude and a certain depth, then followed by next longitude, then depth.
In case of a 3D initial velocity model, note we have boundary values included in this file. The original point will be in the upper northwest corner.

(You can use the python script 'GenerateIniMOD.py' in the 'scripts' directory to output an simple initial model (some modifications are necessary). Then run '*python GenerateIniMOD.py*'. For 3D initial models, remember to follow 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 kind of self-explanatory. However, 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. I usually plot all sources and receivers to decide this origin point to make sure the inverted region fully include all the sources and receivers. Or the program will complain.

The seventh line is the grid interval in Latitude and Longitude.
Again, make sure that all the sources and receivers are in the region of goxd-(nx-3)*dvxd~goxd and gozd~gozd+(ny-3)*dvzd, otherwise the program will complain and stop.

The eighth line is the maximum number of sources and receivers.

The ninth line, "weight" is the balancing parameter between data fitting term and smoothing regularization term, 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 change at each iteration. At the first few iterations, velocity change can reach about 0.4. If it is too large, it means the weight you choose is too small and vice versa. Generally, a few trials will be more than enough to choose a good weight. The velocity change will decrease in the last a few iterations as the inversion converges, or the program will complain by outputting a lot of garbage 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 do you want to transform form grids to layers in order to calculate the depth kernel. I usually set it to 3.

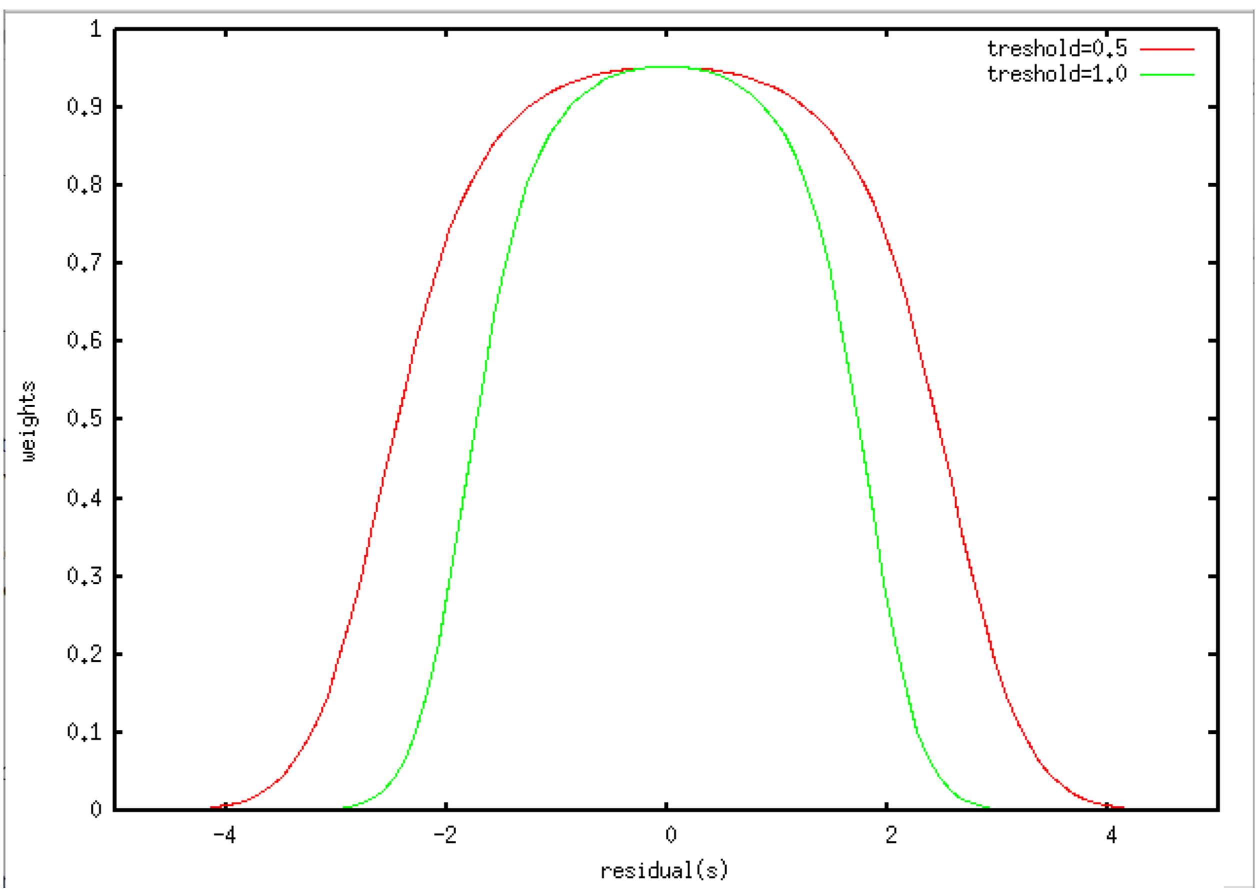
minimum and maximum velocity in the eleventh line are a priori information you have about the study area. You can set a large interval if you don't have this information, it doesn't affect the final result too much.
The twelfth line is the iteration number of the inversion. Usually 10 iterations are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

KmaxRg, kmaxRc, 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 a file named "MOD.true" when the flag is 1, otherwise the program will stop.

(You can also generate 'MOD.true' using 'GenerateTrueMOD.py' in the 'scripts' directory, the script is for 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 do 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 measurements.
The 'threshold' in the last line controls the weight of measurements in order to prevent outliers affect the final result. The weights follow the expression '1.0/(1+0.05*exp(x**2*threshold))', where 'x' is the residual. The following figure shows the weight for two values of threshold. For threshold=0.5, residual larger than 4 seconds will be down weighted to zero.



After preparing these 3 files in a certain directory, you can simple run the program in the same directory with:
../src/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 velocity

"I 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] are some depths of 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
…

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 code is now in Github at https://github.com/HongjianFang/DSurfTomo, feel free to contact me or open an issue on GitHub if you have any questions, I would be happy to help.

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