TOMO3D: a C++ parallel software package for VTI anisotropic 3-D joint refraction and reflection traveltime tomography - Version 0.2.0

Adrià Meléndez, Jun Korenaga & Alain Miniussi
melendez@icm.csic.es,
adria.melendez.catalan@gmail.com,
jun.korenaga@yale.edu,
alain.miniussi@oca.eu

September 30, 2019

Contents

1 Overview

TOMO3D is a package for VTI anisotropic 3-D joint refraction and reflection traveltime tomography presented by Meléndez et al., [2015] and Meléndez et al., [2019]. It works with active and/or passive traveltime data, and it recovers models for P-wave velocity and optionally reflector depth and/or Thomsen's anisotropic parameters δ and ϵ Thomsen, [1986]. Please address any questions on the usage of this software to the first author. It is based on TOMO2D by Korenaga et al., [2000]. This guide describes the usage of all the contents of this package (see section 4):

- gen_smesh3d - velocity mesh generation.
- edit_smesh3d - velocity mesh editing.
- stat_smesh3d - statistical operations on a velocity mesh.
- tt_inverse3d - do traveltime tomographic inversion (parallelized).

1.1 Current status

Presently TOMO3D package contains the necessary files for the installation of all applications listed above except for stat_smesh3d.

tt_inverse3d is under constant development and new features such as extended parallelization should be introduced in the future.

1.2 Terms and Conditions

The contents of TOMO3D package are free to use for academic purposes only. Industrial use is forbidden unless otherwise stated. The authors are listed immediately after the title of this document and are not liable for any misuse or misfunction of the contents of this package.

Please let us know at melendez@icm.csic.es that you have downloaded a copy of the package and specify if you wish to receive update notifications via e-mail. In the future, we may consider automatic notifications. When presenting your results please cite Meléndez et al., [2015].

2 Input file formats

2.1 Velocity and anisotropy (sheared) grid files

```
nx ny nz vwater vair           - number of nodes in x, y and z, velocity in water and air.
x(1) x(2) ... x(nx)            - nodes' x-coordinates.
y(1) y(2) ... y(ny)            - nodes' y-coordinates.
b(1,1) b(1,2) ... b(1,ny)      - corresponding geological relief (bathymetry or topography).
.        .         .
.        .         .
.        .         .
b(nx,1) b(nx,2) ... b(nx,ny)
z(1) z(2) ... z(nz)            - nodes' z-coordinates.
v(1,1,1) v(1,1,2) ... v(1,1,nz)  - velocity at each node.
v(1,2,1) v(1,2,2) ... v(1,2,nz)
.         .          .
.         .          .
.         .          .
v(1,ny,1) v(1,ny,2) ... v(1,ny,nz)
.          .           .
.          .           .
.          .           .
v(nx,ny,1) v(nx,ny,2) ... v(nx,ny,nz)
```

All coordinates should be in increasing order. The z-coordinate is relative to the seafloor/Earth surface and increases downwards. There is no limit to the number of nodes you can put in one line, as long as it is consistent with the first line and it is within the capacity of your computer. An example is given below:

```
5 5 5 1.5 0.33
0 1 2 3 4
0 1 2 3 4
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 1 2 3 4
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
2 2 2 2 2
```

2.2 Reflector file

The reflector's depth is set with the following format:

nx ny	- number of nodes in x and y directions.
x(1) y(1) d(1,1)	- nodes' x- and y-coordinates, and depth at each node.
. . .	
. . .	
. . .	
x(1) y(ny) d(1,ny)	
. . .	
. . .	

.   .   .
x(nx) y(ny) d(nx,ny)

Example:

```
5 5
0 0 2.5
0 1 2.5
0 2 2.5
0 3 2.5
0 4 2.5
1 0 2.5
1 1 2.5
1 2 2.5
1 3 2.5
1 4 2.5
2 0 2.5
2 1 2.5
2 2 2.5
2 3 2.5
2 4 2.5
3 0 2.5
3 1 2.5
3 2 2.5
3 3 2.5
3 4 2.5
4 0 2.5
4 1 2.5
4 2 2.5
4 3 2.5
4 4 2.5
```

2.3 Traveltime data file

The first line contains only one number, nsrc, which is the number of sources. The rest of the file must contain nsrc packets of traveltime data, each of which has the following format:

n_s
s xs(1) ys(1) zs(1) $n_p(1)$
r xr(1,1) yr(1,1) zr(1,1) code(1,1) time(1,1) dt(1,1)
.
.
.
r xr(1,n_p(1)) yr(1,n_p(1)) zr(1,n_p(1)) code(1,n_p(1)) time(1,n_p(1)) dt(1,n_p(1))
s xs(2) ys(2) zs(2) $n_p(2)$
r xr(2,1) yr(2,1) zr(2,1) code(2,1) time(2,1) dt(2,1)
.
.
.
r xr(2,n_p(2)) yr(2,n_p(2)) zr(2,n_p(2)) code(2,n_p(2)) time(2,n_p(2)) dt(2,n_p(2))

s xs(3) ys(3) zs(3) $n_p(3)$
.
.
.
s $xs(n_s)$ $ys(n_s)$ $zs(n_s)$ $n_p(n_s)$
r $xr(n_s,1)$ $yr(n_s,1)$ $zr(n_s,1)$ $code(n_s,1)$ $time(n_s,1)$ $dt(n_s,1)$
.
.
.
r $xr(n_s,n_p(n_s))$ $yr(n_s,n_p(n_s))$ $zr(n_s,n_p(n_s))$ $code(n_s,n_p(n_s))$ $time(n_s,n_p(n_s))$ $dt(n_s,n_p(n_s))$

with:

- n_s: number of sources.
- flags s and r: source and receiver/pick rows.
- xs(i),ys(i), and zs(i) (i=1,n_s): sources' coordinates.
- n_p(i): number of picks for a particular source.
- xr(i,j),yr(i,j), and zr(i,j) (j=1,n_p(i)): receivers' coordinates.
- code(i,j): ray types (0:refraction, 1:reflection, 2:MSRI[1] - refraction, and 3:MSRI reflection).
- time(i,j): picked traveltimes.
- dt(i,j): estimated pick errors.

For example:

```
1
s 2.5 2.5 0 18
r 1.25 1.25 0 0 1.07313 0.01
r 1.25 2.5 0 0 0.769735 0.01
r 1.25 3.75 0 0 1.07314 0.01
r 2.5 1.25 0 0 0.769735 0.01
r 2.5 2.5 0 0 7.06542e-15 0.01
r 2.5 3.75 0 0 0.769428 0.01
r 3.75 1.25 0 0 1.07314 0.01
r 3.75 2.5 0 0 0.769428 0.01
r 3.75 3.75 0 0 1.07235 0.01
r 1.25 1.25 0 1 1.61775 0.01
r 1.25 2.5 0 1 1.51322 0.01
r 1.25 3.75 0 1 1.618 0.01
r 2.5 1.25 0 1 1.51322 0.01
r 2.5 2.5 0 1 1.39929 0.01
r 2.5 3.75 0 1 1.51312 0.01
r 3.75 1.25 0 1 1.618 0.01
r 3.75 2.5 0 1 1.51312 0.01
r 3.75 3.75 0 1 1.61787 0.01
```

2.4 Correlation length files (velocity, anisotropy and depth)

Correlation lengths for velocity nodes are specified in a manner very similar to the velocity grid format:

[1] Meléndez et al., [2013]

```
nx ny nz                    - number of nodes in x, y and z, velocity in water and air.
x(1) x(2) ... x(nx)             - nodes x-coordinates.
y(1) y(2) ... y(ny)             - node's y-coordinates.
b(1,1) b(1,2) ... b(1,ny)             - corresponding geological relief (bathymetry or topography).
.          .            .
.          .            .
.          .            .
b(nx,1) b(nx,2) ... b(nx,ny)
z(1) z(2)  z(nz)                  - nodes z-coordinates.
Lx(1,1,1) Lx(1,1,2) ... Lx(1,1,nz)             - correlation length in x at each node.
Lx(1,2,1) Lx(1,2,2) ... Lx(1,2,nz)
.             .                .
.             .                .
.             .                .
Lx(1,ny,1) Lx(1,ny,2) ... Lx(1,ny,nz)
.              .                 .
.              .                 .
.              .                 .
Lx(nx,ny,1) Lx(nx,ny,2) ... Lx(nx,ny,nz)
Ly(1,1,1) Ly(1,1,2) ... Ly(1,1,nz)             - correlation length in y at each node.
Ly(1,2,1) Ly(1,2,2) ... Ly(1,2,nz)
.             .                .
.             .                .
.             .                .
Ly(1,ny,1) Ly(1,ny,2) ... Ly(1,ny,nz)
.              .                 .
.              .                 .
.              .                 .
Ly(nx,ny,1) Ly(nx,ny,2) ... Ly(nx,ny,nz)
Lz(1,1,1) Lz(1,1,2) ... Lz(1,1,nz)             - correlation length in z at each node.
Lz(1,2,1) Lz(1,2,2) ... Lz(1,2,nz)
.             .                .
.             .                .
.             .                .
Lz(1,ny,1) Lz(1,ny,2) ... Lz(1,ny,nz)
.              .                 .
.              .                 .
.              .                 .
Lz(nx,ny,1) Lz(nx,ny,2) ... Lz(nx,ny,nz)
```

An example is given below:

```
2 2 2
0. 5
0. 5.
0 0
0 0
0. 3.
1. 2.
1. 2.
1. 2.
```

```
1. 2.
1. 2.
1. 2.
1. 2.
1. 2.
1. 2.
1. 2.
1. 2.
1. 2.
```

Correlation lengths for reflector nodes can be specified in a separate file as:

```
nx ny                              - number of nodes in x and y.
x(1) y(1) Lx(1,1) Ly(1,1)          - node's x- and y-coordinates, correlation lengths in x and y.
x(1) y(2) Lx(1,2) Ly(1,2)
.          .           .
.          .           .
.          .           .
x(1) y(ny) Lx(1,ny) Ly(1,ny)
.          .           .
.          .           .
.          .           .
x(nx) y(ny) Lx(nx,ny) Ly(nx,ny)
```

An example is given below:

```
2 2
0. 0. 2. 2.
0. 5. 2. 2.
5 0. 2. 2.
5 5. 2. 2.
```

Note that if a correlation length file for reflector nodes is not provided, tt_inverse3d will sample from horizontal correlation lengths for velocity nodes.

2.5 Variable damping file

Spatially variable damping is implemented by tt_inverse3d -DQdamp_file, which is useful for squeezing tests. The file format is very similar to those for velocity grid and correlation lengths. Analogous files can be input for anisotropic parameters:

```
nx ny nz                      - number of nodes in x, y and z, velocity in water and air.
x(1) x(2) ... x(nx)           - nodes' x-coordinates.
y(1) y(2) ... y(ny)           - nodes' y-coordinates.
b(1,1) b(1,2) ... b(1,ny)     - corresponding geological relief (bathymetry or topography).
.        .          .
.        .          .
.        .          .
b(nx,1) b(nx,2) ... b(nx,ny)
z(1) z(2) ... z(nz)           - nodes' z-coordinates.
```

D(1,1,1) D(1,1,2) ... D(1,1,nz) - velocity at each node.
D(1,2,1) D(1,2,2) ... D(1,2,nz)
. . .
. . .
. . .
D(1,ny,1) D(1,ny,2) ... D(1,ny,nz)
. . .
. . .
. . .
D(nx,ny,1) D(nx,ny,2) ... D(nx,ny,nz)

An example is given below:

```
5 5 5
0 1 2 3 4
0 1 2 3 4
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 1 2 3 4
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
100 100 100 100 100
```

3 Output file formats

Velocity and depth output files follow the same format as the corresponding input files for initial velocity and depth models. Velocity and depth output files are named as *your_inversion_name*.smesh.*#iteration.#dataset* and *your_inversion_name*.refl.*#iteration.#dataset*, respectively. Currently *#dataset* is simply equal to 1, and it does not seem necessary to allow the code to handle more than one dataset in the same run since the same purpose can be achieved by running separate inversions, either simultaneously or consecutively. *your_inversion_name* is set using flag -O.

3.1 Traveltime residuals files

These files are named as *your_inversion_name*.tres.*#iteration.#source*. Each of them contains the receiver coordinates and the traveltime residual (difference between observed and calculated traveltimes) of all receiver for one particular source in the same order as in the traveltime data file.

3.2 Raypath files

These files are named as *your_inversion_name*.ray.*#iteration.#source*. They contain the raypaths corresponding to all receivers for one particular source separated by lines starting with > symbol followed by: the receiver number, the type of ray, whether it is in water or air (or in the ground).

3.3 Derivative weighted sum files

These files are named as *your_inversion_name*.dws.*#iteration.#dataset* and *your_inversion_name*.dwsr.*#iteration.#dataset*. They respectively contain the velocity and depth kernel values associated to each velocity (*dws*) and depth (*dwsr*) parameter in the model described by its coordinates.

4 Command description

4.1 Manipulating velocity grid files

NAME **gen_smesh3d** - generate a velocity mesh.

SYNOPSIS
  gen_smesh3d [velocity options] [grid options]

DESCRIPTION
  This command generates a velocity grid, a required input file for other programs in the package.
  Note: options for Zelt & Smith [1992] are **not available** at the moment.

OPTIONS
  -**A**v0 -**B**gradient
    - specifies velocity as a function of depth: `v(z) = v0` + `gradient` · `z` (km/s).

  -**C**v.in/ilayer [-**F**jlayer/refl_file]
    - uses v.in of Zelt & Smith [1992] to construct a velocity field. The seafloor layer must be given by ilayer.
    -**F** extracts jlayer as a reflector in refl_file.

  -**H**vfile
    - 1D/2D/3D velocity profile to be hung from the seafloor. File format:

'h' x1 y1 nz z_1 v_1 ... z_nz v_nz
'h' x1 y2 ...
.
.
.
'h' x_nx y1 ...
.
.
.
'h' x_nx y_ny ...
'end'

' ' indicate characters or strings, not variables. nz can vary for each "h" line. Last velocity provided will be used if input model is not extensive enough.

1D: 1 line. Keep x1 and y1 constant.
2D: nx (or ny) lines. Keep y1 (or x1) constant. y-dimension (or x-) is generated by repeating the 2D model.
3D: nx·ny lines. x1 and y1 are **not** exchangeable in vfile, **follow format strictly**.

-**N**nx/ny/nz -**D**xmax/ymax/zmax
- specifies a uniform spacing grid with nx, ny and nz nodes, spanning from 0 to xmax, from 0 to ymax, and from 0 to zmax (km).

-**X**xfile -**Y**yfile -**Z**zfile [-**T**tfile]
- specifies a variable spacing grid, as defined by xfile, yfile and zfile. Optional tfile specifies variable bathymetry (km).

-**E**dx/dy -**Z**zfile
- creates a grid based on v.in Zelt & Smith [1992] given in -**C** option, with a (nearly) uniform horizontal spacing of dx and dy (km), and a variable vertical spacing as defined by zfile.

NAME **edit_smesh3d** - edit a velocity mesh.

SYNOPSIS
edit_smesh3d grid_file -**C**cmd [-**L**vcorr_file -**U**upper_file]

DESCRIPTION
This program may be useful when performing synthetic tests, for instance to add anomalies to your background model.

OPTIONS
-**Ca**
- set all velocities to horizontal average.

-**Cp**grid
- paste grid on the original grid.

-**CP**prof
- paste 1-D profile given by prof.

-**Cs**x/y/z
- apply Gaussian smoothing operator with an window of x, y, and z (km).

-**Cr**mx/my/mz

- refine mesh by mx for x-direction, my for y-direction, and by mz for z-direction.

-CcA/x/y/z
- add checkerboard pattern with amplitude A (%), horizontal cycles x and y km, and vertical cycle z km.

-CdA/xmin/xmax/ymin/ymax/zmin/zmax
- add a rectangular anomaly with amplitude A (%).

-CgA/x0/y0/z0/Lx/Ly/Lz
- add a Gaussian anomaly of $A \cdot exp[-(x-x0)/Lx-(y-y0)/Ly-(z-z0)/Lz]$ (%).

-Cl
- remove low velocity zone.

-CRseed/A/nrand
- randomize the velocity field.

-CSseedA/xmin/xmax/dx/ymin/ymax/dy/zmin/zmax/dz
- another randomization.

-CGseed/A/N/xmin/xmax/ymin/ymax/zmin/zmax
- yet another randomization.

-Cmv/refl_file
- set velocities below refl_file to v.

-Lvcorr_file
- set correlation length file used by **-Cs**.

-Uupper_file
- set upper limit depth for edit operations.

NAME **stat_smesh3d** - do some statistical operations for velocity grid(s) or reflector(s). **Not available yet**.

SYNOPSIS

stat_smesh3d **-L**list file **-C**cmd [**-R**n]
stat_smesh3d **-M**mesh **-D**cmd [**-T**topb **-B**botb **-m**midb **-P**TPcorr **-U**vrepl **-X**xmin/xmax **-Y**ymin/ymax **-x**cxmin/cxmax **-y**cymin/cymax **-t**ctopb **-b**cbotb]

DESCRIPTION

Performs statistical operations. It may be used for Monte Carlo uncertainty analysis.

OPTIONS

-Llist file
- specifies a list of velocity grid (or reflector) files.

-Ccmd
- sets an operation for a list of grids (or reflectors).

  -Ca
  - takes ensemble average.

  -Crave_file
  - calculates standard deviation from ave_file.

-Rn
- assumes reflector of n nodes, instead of velocity grid.

-Mgrid

- specifies a velocity grid file.

-Dcmd
- sets an operation for a single grid.

-Daavex/wlen
- takes horizontal average at x=avex with a window of wlen (km)

-Dbxmin/xmax/dx/wlen
- takes horizontal and vertical average with a window of wlen (km), from xmin to xmax with increment dx.

-Ttopb file
- sets top boundary by topb file.

-Bbotb file
- sets bottom boundary by botb file.

-mmidb file
- sets middle boundary by midb file.

-PTref/Pref/dVdT/dVdP/a/b
- applies temperature and pressure corrections to the reference condition of Tref (C) and Pref (MPa). Temperature profile is calculated as T az b where z is depth beneath seafloor.

-Uvrepl
- sets all velocities lower than vrepl to vrepl.

-Xxmin/xmax
- sets x range for operation.

-Yymin/ymax
- sets y range for operation.

-xcxmin/cxmax **-y**cxmin/cymax **-t**ctopb_file **-b**cbotb_file
- sets the region to be skipped by operation.

4.2 Traveltime inversion and forward solver

NAME **tt_inverse3d -** traveltime inversion.

SYNOPSIS
tt_inverse3d **-M**vel_file **-G**data_file **-N**xorder/yorder/zorder/clen/nintp/tol1/tol2 **-F**refl_file **-A** **-d**delta_file **-e**epsilon_file **-H** **-K** **-k** **-x** **-y** **-O**out_fn_root **-o**level **-l** **-P** **-R**crit_chi **-Q**lsqr_tol **-s**bound **-W**d_weight **-V**level **-L**logfile **-CV**vcorr_file **-CD**dcorr_file **-Cd**delta_corr_file **-Ce**epsilon_corr_file **−−nb-threads** number_MP_threads **−−placement**
[iteration options] [smoothing options] [damping options]

DESCRIPTION
This command is an implementation of 3-D joint refraction and reflection traveltime tomography.

OPTIONS
-Mgrid_file
- specifies a velocity grid file.

-Gdata_file
- specifies a traveltime data file.

-Nxorder/yorder/zorder/clen/nintp/tol1/tol2 (see tt_forward).

-Frefl_file
(see tt_forward).

-A
(see tt_forward).

-dgrid_file
- specifies a δ grid file.

-egrid_file
- specifies an ϵ grid file.

-f
- select only-forward mode: generate synthetic traveltimes and raypaths.

-i
- must be used alongside -f: performs inversion for the first iteration.

-rvred
- sets reduction velocity for travel time output.

-Llogfile
- sets log file, with the output format as: 1. the number of iteration, 2. the number of set, 3. the number of rejected data, 4. RMS traveltime misfit, 5. initial χ^2, 6. the number of valid refraction data, 7. RMS traveltime misfit (refraction), 8. initial χ^2 (refraction) 9. the number of valid reflection data, 10. RMS traveltime misfit (reflection), 11. initial χ^2 (reflection), 12. the number of valid MSRI refraction data, 13. RMS traveltime misfit (MSRI refraction), 14. initial χ^2 (MSRI refraction), 15. the number of valid MSRI reflection data, 16. RMS traveltime misfit (MSRI reflection), 17. initial χ^2 (MSRI reflection), 18. CPU time used for graph solution, 19. CPU time used for bending solution, 20. smoothing weight for velocity nodes, 21. smoothing weight for depth nodes, 22. smoothing weight for δ nodes, 23. smoothing weight for ϵ nodes, 24. damping weight for velocity nodes, 25. damping weight for depth nodes, 26. damping weight for δ nodes, 27. damping weight for ϵ nodes, 28. the number of LSQR calls, 29. the total number of LSQR iteration, 30. CPU time used for LSQR, 31. predicted χ^2 based on LSQR solution, 32. average velocity perturbation for velocity model, 33. average depth perturbation, 34. average δ perturbation, 35. average ϵ perturbation, 36. roughness of velocity nodes in x direction, 37. roughness of velocity nodes in y direction, 38. roughness of velocity nodes in z direction, 39. roughness of depth nodes in x direction, 40. roughness of depth nodes in y direction, 41. roughness of δ nodes in x direction, 42. roughness of δ nodes in y direction, 43. roughness of δ nodes in z direction, 44. roughness of ϵ nodes in x direction, 45. roughness of ϵ nodes in y direction, 46. roughness of ϵ nodes in z direction.

-Oout_fn_root
- sets file name root for output files.

-o
- sets output level (print out travel time residual for level >=1; print out ray paths for level >=2).

-l
- prints out the final model only.

-K
- prints out velocity DWS for each iteration.

-k
- prints out depth DWS for each iteration.

-x
- prints out δ DWS for each iteration.

-y
- prints out ϵ DWS for each iteration.

-P
- sets pure jumping strategy.

-Rcrit_chi
- sets critical χ for robust inversion.

-Qlsqr_tol
- sets tolerance for LSQR algorithm.

-s[bound file]
- applies 3-D filter after every iteration. The upper bound for filtering can be set by bound file.

-Wd_weight
- sets depth kernel weighting factor.

-V[level]
- sets verbose level.

−−**nb-threads** [number_MP_threads]
- sets number of MP threads for each MPI process.

−−**placement**
- outputs the placement of each MPI process and MP thread.

-CVcorr_file
- sets correlation length file for velocity nodes.

-CDcorr_file
- sets correlation length file for reflector.

-Cdcorr_file
- sets correlation length file for δ nodes.

-Cecorr_file
- sets correlation length file for ϵ nodes.

Type-1 iteration options: many iterations with a single set of parameters.

-Initer
- sets the number of maximum iterations.

-Jtarget_chi2
- sets target χ^2.

-SVwsv
- applies velocity smoothing with weighting factor wsv.

-SDwsd
- applies depth smoothing with weighting factor wsv.

-Sdwsad
- applies δ smoothing with weighting factor wsad.

-Sewsae
- applies ϵ smoothing with weighting factor wsae.

Type-2 iteration options: single iteration with many sets of parameters.

-SVwsv_min/wsv_max/dw [-**XV**]
- tries velocity smoothing with weighting factor varying from wsv_min to wsv_max with an increment of dw. With **-XV**, smoothing weights will be raised to the power of 10.

-SDwsd_min/wsd_max/dw [-**XD**]
- tries depth smoothing with weighting factor varying from wsd_min to wsd_max with an increment of dw. With **-XD**, smoothing weights will be raised to the power of 10.

-Sdwsad_min/wsad_max/dw [-**Xd**]
- tries δ smoothing with weighting factor varying from wsad_min to wsad_max with an increment of dw. With **-Xd**, smoothing weights will be raised to the power of 10.

-Sewsae_min/wsae_max/dw [-**Xe**]
- tries ϵ smoothing with weighting factor varying from wsae_min to wsae_max with an increment of dw. With **-Xe**, smoothing weights will be raised to the power of 10.

-TVmax_dv
- applies velocity damping with maximum velocity perturbation of max_dv (%).

-TDmax_dd
- applies depth damping with maximum depth perturbation of max_dd (%).

-Tdmax_dad
- applies δ damping with maximum δ perturbation of max_dad (%).

-Temax_dad
- applies ϵ damping with maximum ϵ perturbation of max_dae (%).

-DVwdv
- applies velocity damping with weighting factor wdv.

-DDwdd
- applies depth damping with weighting factor wdd.

-Ddwdd
- applies δ damping with weighting factor wdad.

-Dewdd
- applies ϵ damping with weighting factor wdae.

-DQdamp_file
- applied velocity damping with spatially variable weighting factor specified by damp file (for squeezing). -**DR**damp_file
- applied δ damping with spatially variable weighting factor specified by damp file (for squeezing). **-DS**damp_file
- applied ϵ damping with spatially variable weighting factor specified by damp file (for squeezing).

Note: It is important to understand the parallelization scheme to properly run inversion jobs on your parallel environment. For a detailed explanation on the current parallelization of the code please see Meléndez, [2014].

4.2.1 Ray codes in runtime output file

The following symbols indicate the different refracted and reflected rays that the code can trace depending on the locations of source and receiver in your input data file.

Refractions - Reflections:

* - #: receiver is in the water layer, and source is on the Earth surface or in the subsurface.
ˆ - /: receiver is on the Earth surface or in the subsurface, and source is in the water layer.
˜ - |: receiver and source are in the water layer.
. - +: receiver and source are on the Earth surface or in the subsurface.

5 References

Korenaga, J., Holbrook, W.S., Kent, G.M., Kelemen, P.B., Detrick, R.S., Larsen, H.-C., Hopper, J.R. & Dahl-Jensen, T. (2000). Crustal structure of the southeast Greenland margin from joint refraction and reflection seismic tomography, *J. Geophys. Res.*, 105, 21591-21614.

Meléndez, A. (2014). *Development of a New Parallel Code for 3-D JointRefraction and Reflection Travel-Time Tomography of Wide-Angle Seismic Data Synthetic and Real Data Applications to the Study of Subduction Zones*, PhD thesis, Institut de Cincies del Mar (CSIC), Universitat de Barcelona.

Meléndez, A., Sallarès, V., Ranero, C.R. & Kormann, J. (2013). Origin of water layer multiple phases with anomalously high amplitude in near-seafloor wide-angle seismic recordings, *Geophys. J. Int.*, 196, 243-252.

Meléndez, A., Korenaga, J., Sallarès, V., Miniussi, A. & Ranero, C.R. (2015). TOMO3D: 3-D joint refraction and reflection traveltime tomography parallel code for active-source seismic data - synthetic test, *Geophys. J. Int.*, 203, 158-174.

Meléndez, A., Jiménez, C. E., Sallarès, V.& Ranero, C.R. (2019). Anisotropic P-wave traveltime tomography implementing Thomsen's weak approximation in TOMO3D, *Solid Earth*, in press.

Thomsen, L. (1986). Weak elastic anisotropy, *Geophysics*, 51, 1954-1966.

Zelt, C.A. & Smith, R.B. (1992). Seismic traveltime inversion for 2-D crustal velocity structure, *Geophys. J. Int.*, 108, 16-34.