

## Upgrading/debugging anisotropic tt inverse3d

- Added class file data\_2D3D.cpp (and library data\_2D3D.hpp), which contains different functions which are called from main program "tt\_inverse.cpp" whenever flag "-2" used. This flag is added in the execution line when input data+scripts are provided in 2D, so that they are converted into 3D design (Vp, delta and epsilon models, reflector file and TTT data file). Also outputs will be converted in 2D format if "-2" flag is used.
- In tt\_inverse.cpp:
  - Upgrade: data\_2D3D.hpp library added, and the different calls to the functions for 2D data.
  - Upgrade: added a new error message: program aborts if no damping option is given.
  - Bug in the call to FixDamping function. Solved with a different set of input parameters.
- In inverse\_solve.cpp:
  - Upgrade with necessary changes when using "-2" flag.
- In inverse.cpp (and library inverse.hpp):
  - Bugs in FixDamping function. This bug creates a segmentation fault at some point in the inversion. This was activated if using any of the fixed damping option flags (-DV -DD or -DQ) and no anisotropic models given. The anisotropic option was turned on by default inside this function. Bug solved now, function is rewritten.
  - Upgrade: added function output2d() for conversion of different outputs in 2d style (if flag -2 is added).
- In the following class files: ani3d.cpp, interface3d.cpp and corrlen3d.cpp (and respective libraries):
  - Upgrade: different changes for use of flag "-2" in the 2d anisotropic outputs. Particularly in interface3d.cpp, function outRefl() is now added.
  - Bugs: different corrections in vector and array allocations and different loop constructions when 2d data is used (nx=1 or ny=1). These bugs caused either segmentation faults or accumulation of errors giving as a result a fake bending problem (bending < 0). Bugs solved now.
  - Specifically in corrlen3d.cpp, function DampingWeight3d::at() is adapted to be able to run 2d data (nx=1). With this modification, code works well for the damping option "DQ" in 2D data.
- In bend3d.cpp:
  - Numeric bugs in bending calculation: due to not enough numerical space to find a solution in two loops. The bug is saved by now, giving more space/size for these loops. However the reason underlying this bug remains unknown nowadays. These type of bugs create a fake bending error output (bending<0).
- In smesh3d.cpp:
  - Upgrade for "-2" flag use. Modified function outmesh() and printMasksgrid().
  - Bug?: in functions set(), get() and vget(); whenever velocity model is negative at certain node, there is now a warning instead of an abort or exit (accordingly to tomo2d original design). We chose to let the program run even when certain velocities are negative.
- For running 2D data, Tomo3D is at the moment x3 times slower compared to 2D tomography code. This issue is still not understood.