ALomax scientific
seismology · scientific software
real-time seismology · earthquake location · optimisation · visualisation · Java · internet
320 Chemin des Indes · 06370 Mouans-Sartoux · France
+33 (0)4 92 92 29 79 anthony@alomax.net www.alomax.net/alss

Use of 3D models within NonLinLoc

Anthony Lomax anthony@alomax.net *ALomax Scientific, Mouans-Sartoux, France.*

gempa GmbH info@gempa.de *Telegrafenberg, 14473 Potsdam, Germany*

Table of Contents

1 Introduction

Global search earthquake location with the NonLinLoc (NLL) software package (http://alomax.net/nlloc; Lomax et al, 2001; Lomax et al; 2014) is made practical and efficient for large models and corresponding travel-time grids by pre-calculating the travel-times between each station and all nodes of the grid and then storing the results on disk as travel-time NLL 3D Grid files. Then the forward calculation during location in the program NLLoc requires only a look-up of needed travel-times from the grid files on disk or in memory.

The NLL 3D Grid format is explained at http://alomax.net/nlloc → Formats → 3D Grid Files. This format supports Cartesian coordinates for 2D (y=d*istance*, *z=depth*) and 3D (*x*, *y*, *z*), and spherical coordinates for 2D (y=d*istance*, *z=depth*) and 3D (*longitude*, *latitude*, *depth*); these two coordinate systems are used for NLL modes non-GLOBAL and GLOBAL, respectively; see http://alomax.net/nlloc → Overview for more details.

For layered, 1D models and for 3D models that can be accurately defined on a cubic grid in Cartesian (x, y, z), coordinates, the NLL package includes a program Grid2Time which calculates travel-times for each station and phase type using a 3D version (Le Meur, 1994; Le Meur, Virieux and Podvin, 1997) of the Eikonal finite-difference scheme of Podvin and Lecomte (1991). Grid2Time uses cubic grids where the grid values p are the product of slowness and length,

$$p=\frac{1}{v}*ds \quad , \qquad (1)$$

where v is the P or S velocity in km/s at the center of the grid cell and ds is the length is the side of each cube in km; this is the format used by the Podvin-Lecomte FD travel-time programs.

For 1D models defined in spherical coordinates, NLL uses the Java package TauPToolkit (http://www.seis.sc.edu/taup) within the NLL Java program TauP_Table_NLL to create 2D, spherically-layered, travel-time grid files (*distance*, *depth*) for each phase type at either a specified station, or a generic, 2D travel-time file that can be used for any station that does not have a specified travel-time file. The generic station travel-time files use a 0km depth or sea-level for any station and are identified by station code "DEFAULT" in the grid file names and header. For an example, see nlloc_global_sample/run_global_sample.sh and nlloc_global_sample/taup/TauP_Table_NLL.sh in the NLL distribution archive NLLbeta_samples.tgz.

For 3D models that cannot be accurately defined on a cubic grid in Cartesian x, y, z, coordinates, and for 3D models in spherical coordinates, software external to the NLL package is required to produce the travel-time NLL 3D Grid files for each phase type at each station to be used by NLLoc.

Some of the procedures outlined below refer to undocumented utilities and scripts; for more information and assistance with these, contact Anthony Lomax at ALomax Scientific.

2 Development of 3D velocity models from an earthquake catalog

An earthquake catalog consists of a database or list of earthquake source parameters, and may contain or reference arrival time picks or waveforms containing signal for the earthquake events. Seismic P and S velocity models can be determined from arrival times or waveforms through an inversion procedure.

For local and regional scale earthquake location, 1D velocity models are routinely determined by direct, linearized or global-search iterative inversion of travel-times, by simultaneous, iterative inversion of arrival-times for the velocity model and hypocenter locations, or by waveform inversion of surface waves and sometimes body waves (e.g. Bormann et al, 2012; Kissling et al, 1994).

Local and regional 3D velocity models for earthquake location are determined mainly using *earthquake tomography* applied to earthquake catalog arrival-times. Travel-time tomography estimates 3D velocity structure based on inversion of the travel-times for many ray paths between multiple earthquake sources and recording stations (e.g. Curtis & Snieder, 2002; Thurber & Ritsema, 2007). Ideally, as with medical tomography, the sources are known well distributed and the ray paths are known and fill the target 3D volume with high ray density and azimuthal coverage. In practice, the earthquake source locations are not precisely known and they depend on the (unknown) velocity structure, and the ray coverage is usually poor to very poor, with earthquake sources localized in a few small clusters or planes, and stations only available at the surface and sometimes sparsely or poorly distributed. Also, 3D tomographic inversions are usually limited to smooth models with relatively large scale velocity variations and no sharp interfaces.

To alleviate some of these problems, additional geological and geophysical information can be used, with or independent of tomographic inversion, to construct *deterministic* velocity models which include realistic, 3D geological structures such as basement interfaces and the Mohorovičić discontinuity (e.g. Wagner et al, 2013; Latorre et al., 2016).

In general, the selection of the study volume and inversion method, the data collection and processing, the application of the inversion, the verification and interpretation of the results, and many other aspects of a tomographic study are difficult and time consuming. A thorough and high-quality tomographic inversion on a local or regional scale may require a full PhD thesis or post-doc.

3 Generation of a 3D NonLinLoc velocity model

The primary NLL program for generating velocity model NLL 3D Grids, Vel2Grid (http://alomax.free.fr/nlloc → Vel2Grid), supports 1D and simple 3D models in Cartesian *x*, *y*, *z*, coordinates. An additional NLL program, Vel2Grid3D, generates a 3D, NLL velocity grid file from an external 3D velocity model generated by SimulPS, Simul2000, FDTomo and related inversion codes (e.g. Thurber, 1993; Thurber & Eberhart-Phillips, 1999).

For general and complex 3D models, however, a user of NLL will need to transform their model using external software (or undocumented NLL utilities and scripts, see below) into the cubic NLL 3D Grid format used by Grid2Time and other NLL tools.

3.1 Velocity model NLL 3D Grid structure

The NLL programs and utilities use velocity models in NLL 3D Grid format. When using Vel2Grid or Vel2Grid3D, the 3D Grid dimensions (*num_grid_x, num_grid_y, num_grid_z, dx, dy dz*), spatial origin (*orig_grid_x*, *orig_grid_y, orig_grid_z*) and data types (*gridType*) are specified in the VGGRID control statement; for details, see http://alomax.net/nlloc → Control File → Vel2Grid → VGGRID.

NLL 3D Grid format velocity models contain data types of either velocity, *v*, flagged in NLL control statements by VELOCITY, or the product of slowness and length, *p* (Eq. 1), flagged by SLOW_LEN. The velocity model NLL 3D Grid format model file for input to Grid2Time must consist of cubic cells ($dx = dy = dz$) and the cell values must be SLOW_LEN.

An implicit staggered grids used for the velocity model grids. Thus VELOCITY or SLOW_LEN values must correspond to the slowness at the center of each cell, e.g. the value at grid point *(x, y, z)* should be based on the slowness in the velocity model at $x+dx/2, y + dy/2, z + dz/2$, and the total number of used values is *num_grid_x*-1, *num_grid_y*-1, *num_grid_z*-1, where *num_grid_x*/*y*/*z* are the number of grid nodes in the *x*/*y*/*z* direction specified in a NLL control file. In contrast, the values in the travel-time grids output by Grid2Time represent the travel-times to the corners of each grid cell, and the total number of used values is *num_grid_x*, *num_grid_y*, *num_grid_z*.

For 1D models , the model grid must have *num_grid_x*=2, and the corresponding 2D (*y*=*distance*, *z*=*depth*) travel-time grids output by Grid2Time will have *num_grid_x*=1. The 2D (*y*=*distance*, *z*=*depth*) travel-time grids output by TauP_Table_NLL also have *num_grid_x*=1.

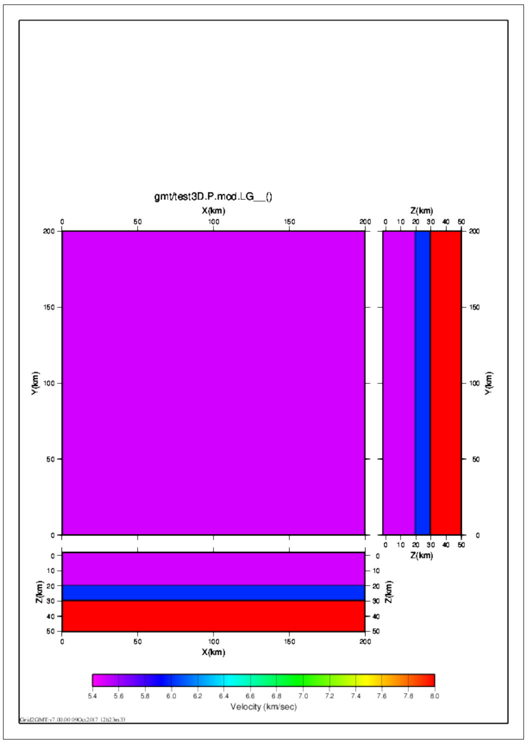
3.2 Creating a 3D model grid from a layered model with Vel2Grid

A 1D, layered model in NLL is normally specified using 2D model and travel-time grids. However, the NLL program Vel2Grid can be used to specify a horizontally layered, 1D model on a 3D grid. This can be useful when developing or testing new 3D grid parameterizations in preparation for implementing a 3D model, but would not be used in general for NLLoc location as it requires much more disk space, and is much less CPU and memory efficient during location than using 2D model and travel-time grids.

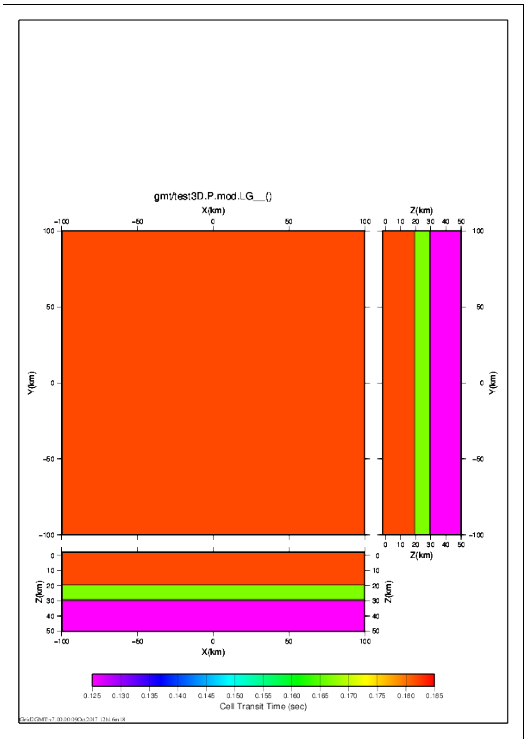
For an example see the file control_layered_3D.in in annex to this report. For a similar example using 2D model and travel-time grids, see the file control_layered_2D.in in annex to this report.

3.3 Verifying a 3D model with Grid2GMT

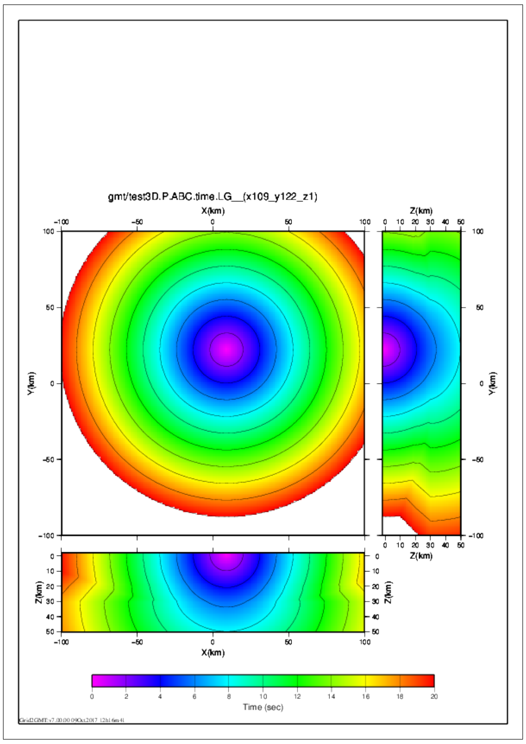
Grid2GMT (http://alomax.free.fr/nlloc → Grid2GMT) supports plotting of NLL Grid files and NLLoc location results. It is important to verify that the geometry, orientation, interface depths and VELOCITY or SLOW_LEN values of a newly created 3D model grid. A 3D model can be verified with Grid2GMT using the G PlotType and the L sections (horizontal section and 2 vertical sections); for examples see the file control_layered_3D.in in annex to this report.



Output of Grid2GMT command in control_layered_3D.in for model grid created with VGGRID type VELOCITY



Output of Grid2GMT command in control_layered_3D.in for model grid created with VGGRID type SLOW_LEN



Output of Grid2GMT command in control_layered_3D.in for travel-time grid output by Grid2Time

A 2D model can be verified with Grid2GMT using the G PlotType and the V sections (vertical sections); for examples see the file control_layered_1D.in in annex to this report.

3.4 Converting an existing SimulPS/FDTomo format 3D velocity model to a NonLinLoc 3D model grid

Existing 3D velocity models in Cartesian *x*, *y*, *z*, coordinates, defined by smooth interpolation between control point nodes and created by velocity inversion programs such as SimulPS, Simul2000 and FDTomo (e.g. Thurber, 1993; Thurber & Eberhart-Phillips, 1999; http://www.geology.wisc.edu/~thurber/simul2000) can be converted to NLL grid format using the NLL program Vel2Grid3D.

Vel2Grid3D uses the following NLL control file statements: Generic: INCLUDE, CONTROL. TRANS; Vel2Grid: VGOUT, VGGRID, VGTYPE; Vel2Grid3D: VGINP, VGCLIP. VGINP sets the input format type (FDTOMO or SIMUL2K) and input velocity grid parameters for SIMUL2K. VGCLIP includes two velocity parameters Vmin and Vmax and supports: 1) if Vmin < Vmax: setting minimum and maximum clip limits for the output velocity values; or 2) if Vmin > Vmax: sharpening of velocity difference across an interface (such as the Moho): if velocity at node below current input node is > Vmax: set the NLL grid point velocity equal to the velocity of the deepest overlying input node with velocity<Vmax. For more details, see (http://alomax.free.fr/nlloc → Vel2Grid3D).

Because of there are many variations of the SimulPS, Simul2000 and FDTomo type model formats, it is necessary to edit the Vel2Grid3D.c source code to select a specific sub-format using C **`#define`** statements. The current options are:

1. default, early SimulPs format with **`%5f`** (fixed width) format for reading velocity values and right-handed, East-negative coordinate system
2. Zhang **`%7f`** (fixed width) format for reading velocity values and left-handed, East-positive coordinate system
3. ETH format (**`*f`** variable width, space delimited) for reading velocity values
4. A left-handed, East-positive coordinate system

See the notes about format flags in the Vel2Grid3D.c source code for more details.

3.5 Converting a general, existing 3D velocity model to a NonLinLoc 3D model grid

For general, existing 3D velocity models in Cartesian *x*, *y*, *z*, coordinates, external tools or software provided by the user should be used for conversion or export into SLOW_LEN (sec) units on a cubic grid in the NLL 3D Grid file format used by Grid2Time (http://alomax.net/nlloc → Formats → 3D Grid Files). This typically requires:

1. determination of the NLL grid origin, cell size and number of cells in the *x*, *y*, *z* dimensions;
2. defining the mapping between the existing velocity model coordinates *u*, *v*, *w* and the NLL *x*, *y*, *z* grid coordinates;
3. writing the NLL 3D Grid Data buffer file binary values using a loop over *z*, contained within a loop over *y*, contained within a loop over *x*, applying for each innermost loop the mapping *x*, *y*, *z* to *u*, *v*, *w* to obtain the velocity in the existing model, and writing this VELOCITY or SLOW_LEN value;
4. writing the NLL 3D Grid Header file.

3.6 *Additional, undocumented procedures for creating an NLL 3D model*

There are many possible procedures to produce 3D models with a range of complexity complexity in the model parameterization and velocity field. The following is an overview of (mainly) undocumented procedures that have been used by ALomax Scientific and others.

3.6.1 Cascading NLL grid structure for very large models and travel-time grids

NLL includes a cascading, multi-scale, 3D grid structure which reduces greatly the storage and memory and file size requirement for large, 3D grids, enabling application of NLL to very large model grids. A NLL cascading grid supports a doubling of cell size at specified depths. Thus a cascading grid has the smallest cells at the top of the model, and cells with twice this size (1/8th the volume and memory) below some depth, usually chosen to include strong shallow or crustal complexity in the velocity model and travel-time fields. The doubling of cell size can be repeated at additional depths; for a regional model that extends into the mantle, just two doubling depths can give a saving in memory and file size of a factor of greater than 10.

A NLL grid can be converted from a regular grid to cascading grid with new tool GridCascadingDecimate. When a regular NLL grid is converted to a cascading grid with GridCascadingDecimate, an additional CASCADING_GRID line is added to the grid header file; the information in this line allows NLLoc, Grid2GMT and other NLL programs to transparently and automatically use NLL cascading grids instead of or along with standard NLL grids.

A GridCascadingDecimate command is of the form:

> GridCascadingDecimate ${DEPTHS} ${TIME_GRIDS_PATH}/${GRID_NAME}.P.${STATION}.time.buf ${TIME_GRIDS_PATH}_casc/

where ${DEPTHS} is a list of cell size doubling depths in km, e.g. DEPTHS=20,40,100

Typically, a very large 3D velocity model and travel-times will be constructed using a regular grid to maintain precision of the calculated travel-times. In a final step, the travel-time grids are converted to NLL cascading grids for efficient storage and later location or other NLL processing. To validate the travel-times on a NLL cascading travel-time grid, they can be compared to the travel-times on the corresponding regular grid using the NLL utility GridProc SUB to create the difference grid, which can be plotted with Grid2Time. For example:

```
GridProc SUB ${TIME_GRIDS_PATH}/${GRID_NAME}.P.${STATION}.time $
{TIME_GRIDS_PATH}_casc_diff/cascdiff.${GRID_NAME}.P.${STATION}.time $
{TIME_GRIDS_PATH}/${GRID_NAME}.P.${STATION}.time $
{TIME_GRIDS_PATH}_casc/casc.${GRID_NAME}.P.${STATION}.time
```

```
Grid2GMT ${CONTROL_FILE} $
{TIME_GRIDS_PATH}_casc_diff/cascdiff.${GRID_NAME}.P.$
{STATION}.time.diff gmt/ L G ${GMT_XYZ} -no_plot_contours
```

3.6.2 Creating a 3D model grid from an extruded 2D cross section (2.5D model) specified with polygons with Vel2Grid

Vel2Grid includes control statements VERTEX, EDGE and POLYGON2 to construct a set of points (VERTEX's) which are connected by line segments (EDGE's) which are combined to form polygons (POLYGON2's). Each polygon will contain specified constant or vertical-gradient P and S velocities.

If the polygons are defined on a vertical, 2D plane, then the Vel2Grid 2DTO3DTRANS control statement can be used to "extrude" this 2D model section into 3D, that is, extend the vertical 2D section into the 3D volume along a horizontal line. By default, the 2D model section will be parallel to the x direction in the 3D model, and the 2D model will be extended along the y direction in the 3D model; thus the point (x, y, z) in the 3D model, for any y, will have the same velocity as the point (x, z) in the 2D model section (see figures in Appendix A). A rotation can be specified to extend the 2D model along any azimuth to form the 3D model. See Appendix A for an example of a simple 2.5D model created with Vel2Grid. For a more realistic and complex 2.5D model representing a subducation zone, see http://alomax.free.fr/nlloc/soft6.00/faq_data/2011__Lopez_Venegas__Building_a_2D_vel_model_with_Vel2Grid_using_polygons__Report.pdf. Both examples courtesy of Alberto M. López Venegas.

3.6.3 Creating a 3D model grid from a 3D model specified with polygonal solids with Vel2Grid

Vel2Grid includes the additional (undocumented) control statement POLYGON3 to construct polygonal solids from sets of POLYGON2's. Each polygonal solid will contain specified constant or vertical-gradient P and S velocities. Creating a 3D model with POLYGON3's is tedious and requires careful planning an bookkeeping. See Appendix B for an example of a 3D model created with Vel2Grid.

For details on the Vel2Grid control statements, see http://alomax.free.fr/nlloc → Control File → Vel2Grid.

3.6.4 Calculating travel-times for complex, 3D models with smooth velocities between sharp interfaces

The Eikonal finite-difference scheme of Podvin and Lecomte (1991) used by Grid2Time requires a regular, cubic grids, which can be very large when a fine grid spacing is needed, and which also gives a stair-step representation of sharp, dipping interfaces. This scheme also computes only first arriving travel-times, and so cannot produce times for most secondary, converted and reflected arrivals, such as the PmP Moho reflection. Some recent earth models consist of both smoothly varying 3D velocities, determined by earthquake tomography, and sharp, 3D interfaces determined by controlled or natural-source seismics analysis. For such models, the freely available, multistage FMM procedure for complex, layered media in spherical coordinates (Rawlinson & Sambridge 2004a, 2005; de Kool et al. 2006) can calculate travel-times for direct, secondary, converted and reflected arrival.

The FMM procedure uses three different grids for computation, called 'velocity', 'interface' and 'propagation grid' (de Kool et al. 2006; http://rses.anu.edu.au/seismology/soft/fmmcode). The velocity grid defines seismic velocities on a regularly spaced 3-D grid in spherical coordinates. The interface grid specifies the location of interfaces, for example, Moho, at depth. It can be irregularly spaced in depth to account for complex interface topographies. The regularly spaced propagation grid is used for computation of travel-times.

NLL includes several undocumented utility programs and scripts to:

1. generate FMM interface grids from an interface file in psuedo-SimulPS format
2. construct FMM velocity grids from SimulPS format models for regions between each interface
3. run FMM travel-time calculation for each station source and specified arrival types
4. convert FMM output arrival time grid to NLL 3D Grid

5. plot results with Grid2GMT

For an example of a application of NLL with FMM travel-times for a complex 3-D velocity model that combines a first-order Moho discontinuity and smoothly varying 3-D seismic velocity information based on local earthquake tomography and controlled-source seismology see Wagner et al (2013).

4 Configuration of NonLinLoc using 3D velocity models

Except for Vel2Grid and Vel2Grid3D, as discussed above, the control statements for configuration of NonLinLoc programs is almost the same for 1D models / 2D travel-time grids as for 3D models / travel-time grids. This is because there is little difference between 1D models / 2D travel-time grids and 3D models / 3D travel-time grids - both cases use the same NLL 3D Grid format. With 1D models / 2D travel-time grids, however, the model grid must have *num_grid_x*=2, and the corresponding travel-time grids output by Grid2Time will have *num_grid_x*=1. A flag TIME2D in the travel-time grid header file indicates that the grid is 2D (travel-times as a function of distance and depth).

Station elevation can be implicitly taken into account for travel-time grids defined in Cartesian coordinates, the since the station is placed at this elevation in the grid for travel-time calculation in Grid2Time. Elevation corrections for models defined in spherical coordinates with 0km depth or sea-level station depth are supported when locations are performed in NLLoc (see http://alomax.net/nlloc → Control File → NLLoc → LOCELEVCORR). Constant travel-time corrections are also supported in NLLoc (see http://alomax.net/nlloc → Control File → NLLoc -> LOCDELAY).

See http://alomax.net/nlloc → Control File for detailed information on NLL program control file statement configuration.

4.1 Optimum NLL grid sizing for accuracy and efficiency

There are no simple rules to determine the NLL grid size (number of *x*, *y*, *z* cells, cell size) for a velocity model and/or corresponding travel-time grid, since there is a trade off between increasing grid size (e.g. finer griding and more accurate travel-times), and increasing disk/memory space and time required for location. The final travel-time grid sizing should satisfy a number of loose requirements:

1. The cell size should be much less than the smallest scale of smooth velocity variation in the velocity model. However, sharp boundaries in velocity that are not horizontal or vertical can only be approximated in a NLL cubic grid with stair-step interfaces, and thus introduce unavoidable errors in travel-times; smaller cell size can help but not fully alleviate this problem.
2. The cell size should be much less than the desired and achievable location uncertainty, as measured by the typical extent of the high confidence region of location PDF's or location error ellipsoids for the best constrained events in a study. The extent of the PDF is related to the prior pick and travel time (LOCGAU, LOCGAU2) uncertainties - these should never be set both unrealistically small or zero. Note that location uncertainty, not precision, is important, the uncertainty is in general much larger than the precision of, for example, the difference in maximum likelihood location for two events with very similar observations.
3. The grid extent in *x*, *y*, *z* should include all stations and target source regions, with large enough margins to fully contain the pdf clouds for moderate to well located events. In the case of generally increasing velocity with depth, or a crust over mantle structure, or other

cases with deep, high-velocity layers, the depth extent of the grid should include several cells below the deepest depth through which first-arrival rays are expected from any source to any station.

4. The maximum total size of all grids needed for location is,
$$S = N_s \times N_p \times \text{num\_grid\_x} \times \text{num\_grid\_y} \times \text{num\_grid\_z} \times 4\text{bytes} \quad , \qquad (2)$$
where N_s is the number of stations and N_p the number of phase types. If multiple events are to be located in a single run of NLLoc, S should be less than about ½ to 2/3 the total memory available (accounting for other running processes and users!), so that all travel-time grids can be kept in memory during location. (see also http://alomax.net/nlloc → Control File → NLLoc → LOCMETH → maxNum3DGridMemory)

For cases where the model grids are too large to satisfy the above requirements see the section *Cascading NLL grid structure for very large models and travel-time grids*.
For cases where there are complex and gently dipping interfaces, see the section *Calculating travel-times for complex, 3D models with smooth velocities between sharp interfaces.*

4.2 Grid2Time

When Grid2Time is applied to a 3D velocity model, GTMODE must be set to GRID3D:

```
# time grid modes

GTMODE GRID3D ANGLES_YES
```

4.3 NLLoc

The forward calculation during location in the program NLLoc requires travel-times grid files created with Grid2Time or otherwise as described earlier. The link with existing travel-time grid files is made in the NLLoc LOCFILES control statement. Note that NLLoc control statements can be in different NLL control files than those used for Vel2Grid and Grid2Time.

When NLLoc is run with 3D travel-time grids, the LOCSEARCH and LOCGRID control statements are usually set to correspond the same grid volume as the corresponding velocity model and travel-time grids. These statements can also be set to a smaller volume that is entirely contained within that of the velocity model and travel-time grids.

The LOCGRID statement corresponding to the VGGRID statement in the example in the file control_layered_3D.in would be:

```
# LOCGRID  num_grid_x  num_grid_y  num_grid_z   orig_grid_x  orig_grid_y
orig_grid_z  d_grid_x d_grid_y d_grid_z  type save_flag

LOCGRID  201 201 53  -100.0 -100.0 -2.0  1.0 1.0 1.0   PROB_DENSITY  SAVE

# above specifies identical 200x200x52km 3D grid as VGGRID statement
```

A compatible LOCSEARCH OCT statement should have approximately the same ratios of init_num_cells_x / y / z as the ratios *num_grid_x* / y / z in LOCGRID, so that the oct-tree cells are approximately cubic. Also, the product of the init_num_cells_x * init_num_cells_y * init_num_cells_z should be much less than max_num_nodes, but large enough so that the smallest dimension of the LOCGRID (usually z, depth) has at least 2-4 init_num_cells:

```
# LOCSEARCH  OCT init_num_cells_x init_num_cells_y init_num_cells_z
min_node_size max_num_nodes num_scatter use_stations_density
stop_on_min_node_size

LOCSEARCH OCT 20 20 5  0.01 25000 1000 0 0

# above has 20 20 5 init_num_cells, which follows ratio of LOCGRID
```

```
num_grid
# and the product 20*20*5=2000 is << 25000  max_num_nodes
```

5 References

Bormann, P., Engdahl, B., Kind, R. (2012): Seismic Wave Propagation and Earth models. - In: Bormann, P. (Ed.), New Manual of Seismological Observatory Practice 2 (NMSOP2), Potsdam : Deutsches GeoForschungsZentrum GFZ, pp. 1—105. DOI: http://doi.org/10.2312/GFZ.NMSOP-2_ch2 http://gfzpublic.gfz-potsdam.de/pubman/item/escidoc:65558:8/component/escidoc:364130/Chapter%202.pdf

Curtis, A., and Snieder, R. (2002). Probing the Earth´s with seismic tomography. In: Lee, W. H. K., Kanamori, H., Jennings, P. C., and Kisslinger, C. (Eds.) (2002). International Handbook of Earthquake and Engineering Seismology, Part A. Academic Press, Amsterdam, 861-874. *https://www.geos.ed.ac.uk/homes/acurtis/Papers/Curtis_Snieder2002.pdf*

de Kool, M., Rawlinson, N. & Sambridge, M., 2006. A practical grid-based method for tracking multiple refraction and reflection phases in three-dimensional heterogeneous media, Geophys. J. Int., 167(1), 253–270.

Kissling, E., W.L. Ellsworth, D. Eberhart-Phillips, and U. Kradolfer: Initial reference models in local earthquake tomography, J. Geophys. Res., 99, 19635-19646, 1994. http://www.seg.ethz.ch/software/velest.html

Le Meur, Helene, 1994. Tomographie tridimensionnelle a partir des temps des premieres arrivees des ondes P et S, application a la region de Patras (Grece), These de Doctorat, Paris VII, France.

Le Meur H, J. Virieux et P. Podvin, 1997. Seismic Tomography of the gulf of Corinth: a comparison of methods, Annali di Geofisica, 40, 6, 1-24.

D. Latorre, F. Mirabella, L. Chiaraluce, F. Trippetta, A. Lomax (2016), Assessment of earthquake locations in 3D deterministic velocity models: a case study from the Altotiberina Near Fault Observatory (Italy), J. Geophys. Res. Solid Earth, 121, doi:10.1002/2016JB013170. https://www.dropbox.com/s/rydqf81chb2c73o/2016__Latorre_et__Assessment_of_earthquake_locations_in_3-D_deterministic_velocity_models_A_case_study_from_the_Altotiberina_Near_Fault_Observatory_Italy__JGR.pdf?dl=0

Lomax, A. and B. A. Bolt (1992). Broadband waveform modelling of anomalous strong ground motion in the 1989 Loma Prieta earthquake using three-dimensional geologic structures, Geophys. Res. Lett., 19, 1963-1966. (PDF: http://alomax.free.fr/articles/1992__Lomax_Bolt__Broadband_waveform_modelling_of_1989_Loma_Prieta__GRL.pdf)

Lomax, A., A. Michelini, A. Curtis (2014), Earthquake Location, Direct, Global-Search Methods, in Encyclopedia of Complexity and System Science, 2nd Edition, Meyers, R. A. (ed.), Springer, New York, pp. 1-33, doi:10.1007/978-3-642-27737-5_150-2. (PDF annexed to this report)

Lomax, A., A. Zollo, P. Capuano, and J. Virieux, (2001), Precise, absolute earthquake location under Somma-Vesuvius volcano using a new 3D velocity model, Geophys. J. Int., 146, 313-331. (PDF annexed to this report)

Podvin, P. and Lecomte, I., 1991, Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools., Geophys. J. Int., 105, 271-284.

Thurber, C.H., 1983. Earthquake locations and three-dimensional crustal structure in the Coyote

Lake area, central California. J. Geophys. Res. 88, 8226-8236.

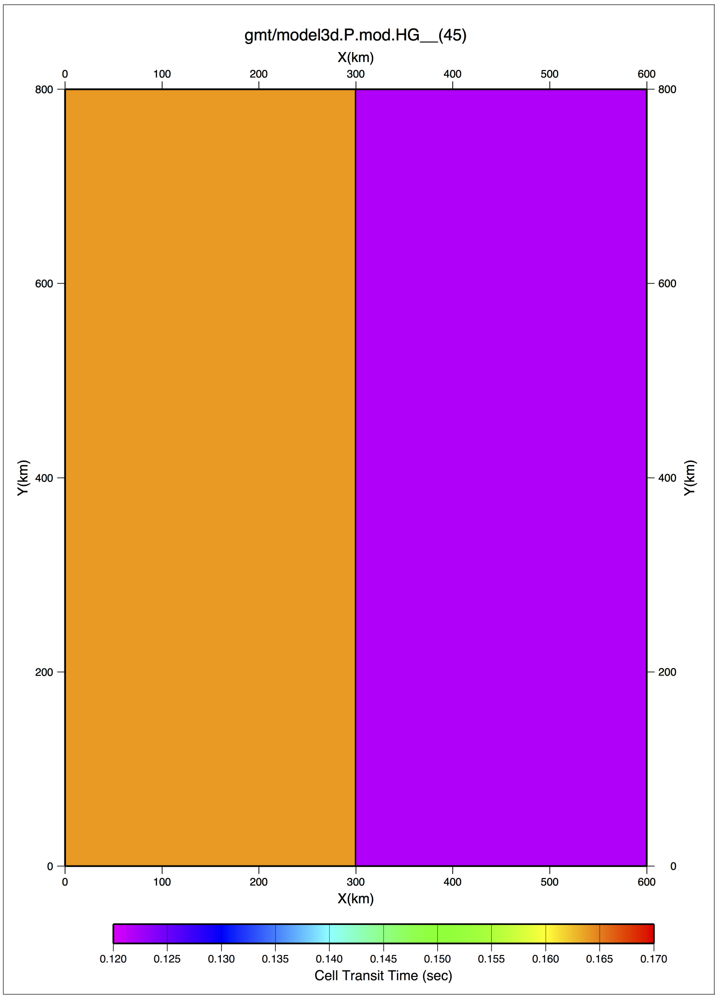
Thurber, C.H. and Eberhart-Phillips, D., 1999. Local earthquake tomography with flexible gridding. Computers and Geoscience 23, 809-818.

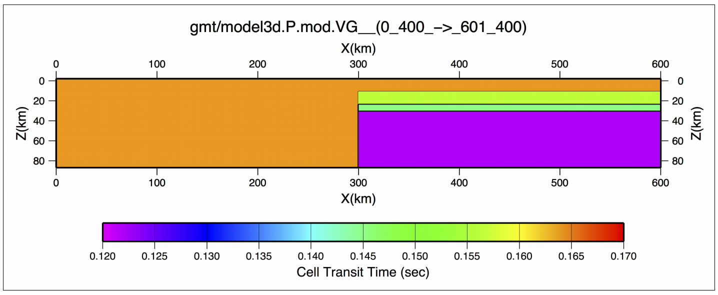
Thurber, T. C., and Ritsema, J. (2007). Theory and observation – seismic tomography and inverse methods. In: Schubert, G. (ed.). Treatise on Geophysics, Elsevier Ltd., Oxford, Vol. 1, 323-360. mrw.elsevier.com/tgp2/Example1.pdf

Wagner, M., S. Husen, A. Lomax, E. Kissling, and D. Giardini. (2013), High-precision earthquake locations in Switzerland using regional secondary arrivals in a 3D velocity model, Geophys. J. Int., 193, 1589-1607, doi: 10.1093/gji/ggt052. (PDF annexed to this report)

6 Appendix A

The following figures show a simple 2.5D model constructed by extruding a 2D polygon model into 3D using the Vel2Grid 2DTO3DTRANS control statement. The corresponding NLL control file 2DTO3DTRANS_control.in is an annex to this report.





xxx

7 Appendix B

The file calif_sfbay3D.in (annex to this report) contains the NLL control statements to define a complicated, 3D polygonal solid model for the San Francisco Bay Area (Lomax and Bolt, 1992):

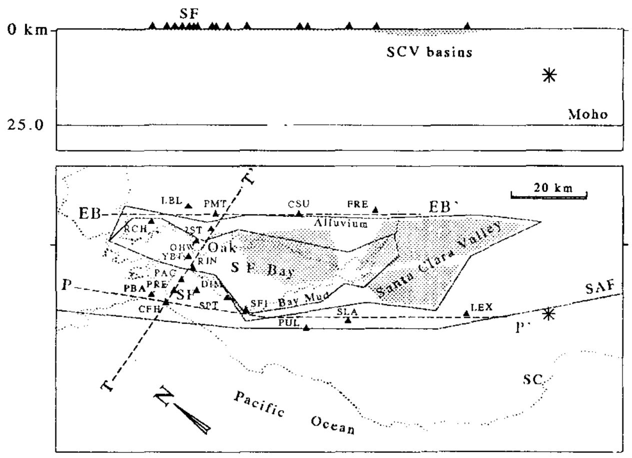


Fig. 1. (upper) cross section and (lower) map view of the SF Bay region showing the Loma Prieta source (star), strong motion stations (triangles), San Andreas Fault (SAF), San Francisco (SF), Oakland (Oak), Santa Cruz (SC) and limits of 3-D velocity model elements. Dashed lines show synthetic seismogram lines. The surface extent of low velocity alluvium and bay muds is indicated by the outer and inner polygons around the SF Bay, respectively; the three shaded areas indicate low velocity basin regions deeper than 300 m. The shaded region on the cross-section indicates the depth extent of the alluvial basins under the Santa Clara Valley (SCV basins).

Schematic diagram of the 3D model defined by control statements in the file calif_sfbay3D.in