

## **A GENERALIZED EARTH MODEL SOFTWARE UTILITY**

Sandy Ballard, Jim Hipp, Andre Encarnacao, Chris Young and Brian Kraus

Sandia National Laboratories

Sponsored by the National Nuclear Security Administration

Award No. DE-AC04-94AL85000/SL12-SignalPropagation-NDD021

### **ABSTRACT**

Implementation and sharing of 3D Earth models by the seismological research community, and evaluation of those models by global monitoring agencies is hampered by the plethora of model parameterizations and formats that have been implemented by model developers. GeoTess is an attempt to address this issue by providing a common model parameterization for multi-dimensional Earth models and a software support system that deals with the construction, population, storage and interrogation of data stored in the model. GeoTess is not limited to any particular type of data; to GeoTess, the data are just 1D arrays of values associated with each node in the grid. The grid and many of the basic grid manipulation algorithms implemented by GeoTess are described in Ballard et al. (2009).

In this paper, we describe the salient features of the model parameterization and supporting software and describe two example applications. The first is called LibCorr3D, which is an application that manages 3D lookup tables for travel time correction information. With this software, the research community can develop 3D velocity models of the Earth, pre-compute station-phase-specific travel times and travel time uncertainties through their model in any manner they deem appropriate and deliver that information to monitoring agencies in a format that the monitoring agencies are prepared to accept.

The package, including source code, is supplied in Java and C++. There are also interfaces to the C++ version written in C and Fortran90. GeoTess has been tested on multiple platforms, including Linux, SunOS, MacOSX and Windows.

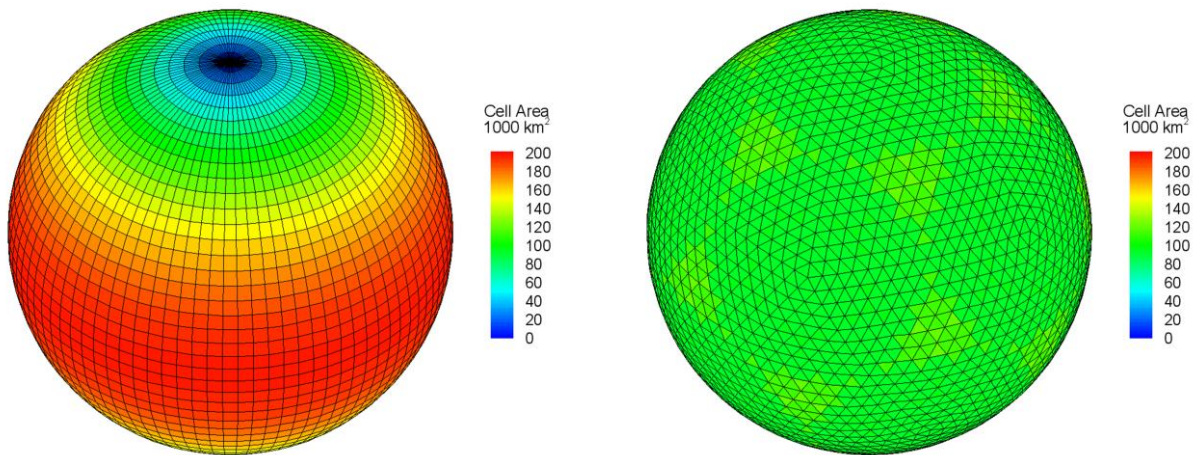
## INTRODUCTION

The use of multi-dimensional global Earth models in both the seismological research community and in the operational systems of global monitoring agencies has become increasingly ubiquitous in recent years and is destined to become even more prevalent in the future. This trend is driven by the significantly enhanced fidelity of these models relative to the real Earth structure and composition, resulting in the promise of more accurate predictions of seismic observations and hence to more accurate and precise estimates of seismic event locations and magnitudes. Use of multi-dimensional global models has become feasible in recent years due to the dramatic increase in commercial off-the-shelf computational resources that are readily available.

Efforts to evaluate multi-dimensional models and incorporate them into operational systems are complicated by the unique formats and software support systems that are implemented by each development team (Reiter et al, 2011). This makes evaluation and implementation of 3D models by monitoring agencies difficult and inefficient and reduces the probability that research products developed by the seismological research community will ever find their way into operational systems.

GeoTess is an attempt to address some of these issues by providing a common model format designed to facilitate the implementation and sharing of 3D Earth models. GeoTess is composed of a model parameterization for multi-dimensional Earth models and extensible software that implements the construction, population, storage and interrogation of data stored in the model. GeoTess is not limited to any particular type of data; to GeoTess, the data are just 1D arrays of values associated with each node in the grid. The grid and many of the basic grid manipulation algorithms implemented by GeoTess are described in Ballard et al. (2009).

While many Earth models use regular latitude longitude grids to describe the geographic geometry and topology, GeoTess uses a triangular tessellation. These two approaches are compared in Figure 1. While software algorithms that use regular latitude longitude grids are much more straightforward to develop, the grids suffer from severe unintended variability in cell areas, with cell areas approaching zero near the poles. Software for triangular tessellations, on the other hand, is somewhat more complicated to develop but results in grids with much more uniform cell size and approximately 25% fewer vertices.



**Figure 1 – Comparison of a regular latitude longitude grid and a uniform triangular tessellation. In both grids the edge lengths are approximately 4°.**

GeoTess software provides tools to perform the following functions:

- Construct the discretized grid of a model by defining the positions of all the nodes (geometry), and the connectivity of the nodes with one another (topology or connectivity) to form non-overlapping triangles that completely span the unit sphere. Construction of grids with variable resolution in both geographic and radial dimensions is supported.
- Query a model grid for information about the nodes, cells or tessellations.
- Associate data structures with the nodes and/or the cells of the geometry.

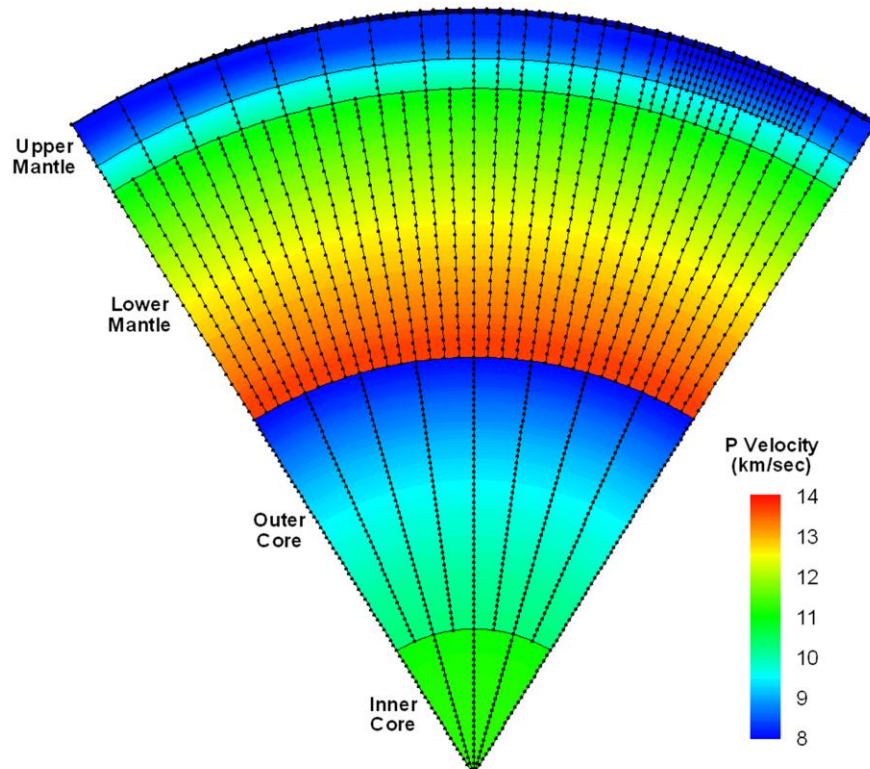
- Modify the data associated with a node or cell.
- Query the model for the data associated with a specified node or cell.
- Rapidly find arbitrary positions within the grid hierarchy (point searching).
- Given an arbitrary position within the model space that does not correspond to a node of the model geometry, calculate a set of interpolation coefficients that can be applied to data at nearby nodes of the model geometry to obtain an estimate of the data value at the point of interest. GeoTess currently implements linear and natural neighbor interpolation algorithms and may implement additional methods in the future.
- Read models from, and write them to, files in ascii, netcdf and a custom binary formats.

Grid construction is implemented in an application called GeoTessBuilder, which is written in Java and can be run on any computer platform that supports that language. The remaining functions described above are implemented in a software library called GeoTess, which is available in Java and C++ with C and FORTRAN95 interfaces to the C++ library. The libraries have been tested on SunOS, Linux, Mac OSX and Windows operating systems.

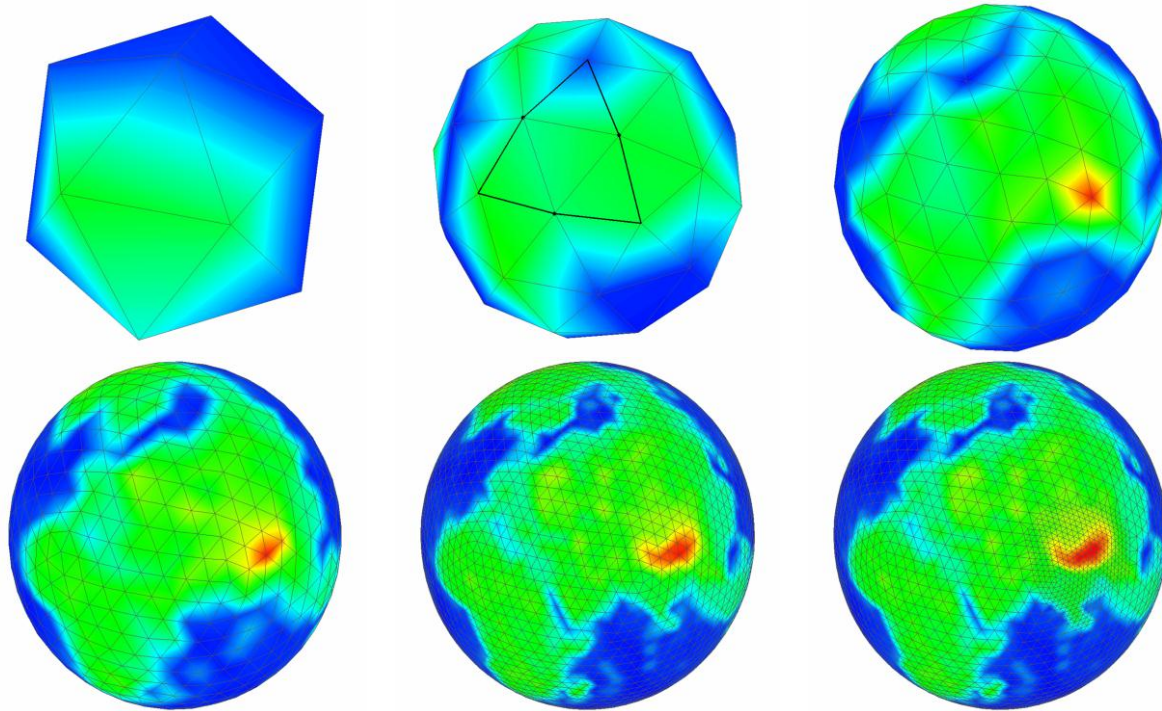
### MODEL COMPONENTS

A GeoTess model is comprised of the following elements:

- A **set of layers** (Figure 2). Each layer spans the entire 2D geographic extent of the model. The boundaries at the top and bottom of a layer may have topography. Within each layer, model data values are continuous, both geographically and radially. Model data values may be discontinuous across layer boundaries. Layers may have zero thickness at some or all geographic locations. An important limitation of the parameterization used by GeoTess is that layer boundaries may not fold back on themselves, i.e., any radial line emanating from the center of the Earth must intersect each layer boundary exactly one time.



**Figure 2 – A slice through a 3D P velocity model. The model consists of a number of layers, such as the Inner Core, Outer Core, etc. Each layer is associated with a separate multi-level tessellation, providing variable resolution in the radial direction. Profiles are defined as a set of nodes, all within a single layer of the model, positioned along a line with constant geographic position. Note the variable resolution in the geographic dimensions in the upper mantle.**



**Figure 3 – Construction of a multi-level tessellation by iterative subdivision of triangles. Each image represents one level and together the levels comprise a single multi-level tessellation.**

- A set of **multi-level tessellations** (Figure 3). Each layer will be associated with one multi-level tessellation but many layers may be associated with each multi-level tessellation, i.e., there is a many-to-one relationship between layers and multi-level tessellations. By associating layers that are deep in the Earth with low resolution multi-level tessellations and layers at shallower levels in the Earth with higher resolution multi-level tessellations, the resolution of the model can be varied radially as necessary to achieve more appropriate sampling.
- The **topology** of each multi-level tessellation will consist of a set of **levels** (see Figure 3), with each level consisting of a set of triangles that spans the surface of a unit sphere, without gaps or overlaps. The triangles on a given tessellation level are obtained by subdivision of the triangles on the previous tessellation level, with the first tessellation level being an icosahedron. Each multi-level tessellation may have variable resolution in the geographic dimensions (i.e. the triangles can be subdivided into smaller triangles arbitrarily).
- The **geometry** of each multi-level tessellation will consist of a set of **vertices** that defines the positions of the corners of the triangles. If a model is comprised of more than one multi-level tessellation, they will share common vertices, to the extent possible.
- **Data arrays.** Each data array is a 1D array of data values that may be of type double, float, long, int, short or byte. All the data arrays in the model must be of the same type and must have the same number of elements.
- **Profiles.** Each profile is composed of a set of monotonically increasing radii and a set of data arrays. Each profile is associated with a single vertex and a single layer in the model. The first and last radii in a profile define the bottom and top of the associated layer at the geographic position of the vertex. Several types of profiles are supported:
  - **N-Point profiles** consist of two or more radii and an equal number of data arrays, with one data array associated with each radius.
  - **Constant value profiles** consist of two radii and a single data array that defines the data values for the entire radial span of the profile.

- **Thin profiles** consist of a single radius and a single data array. They have zero thickness, i.e., the radius of the bottom and top of the profile are equal.
- **Empty profiles** consist of two radii but no data arrays.
- **Surface profiles** consist of only a single data array. They have no radius values. These are used to support 2D models.

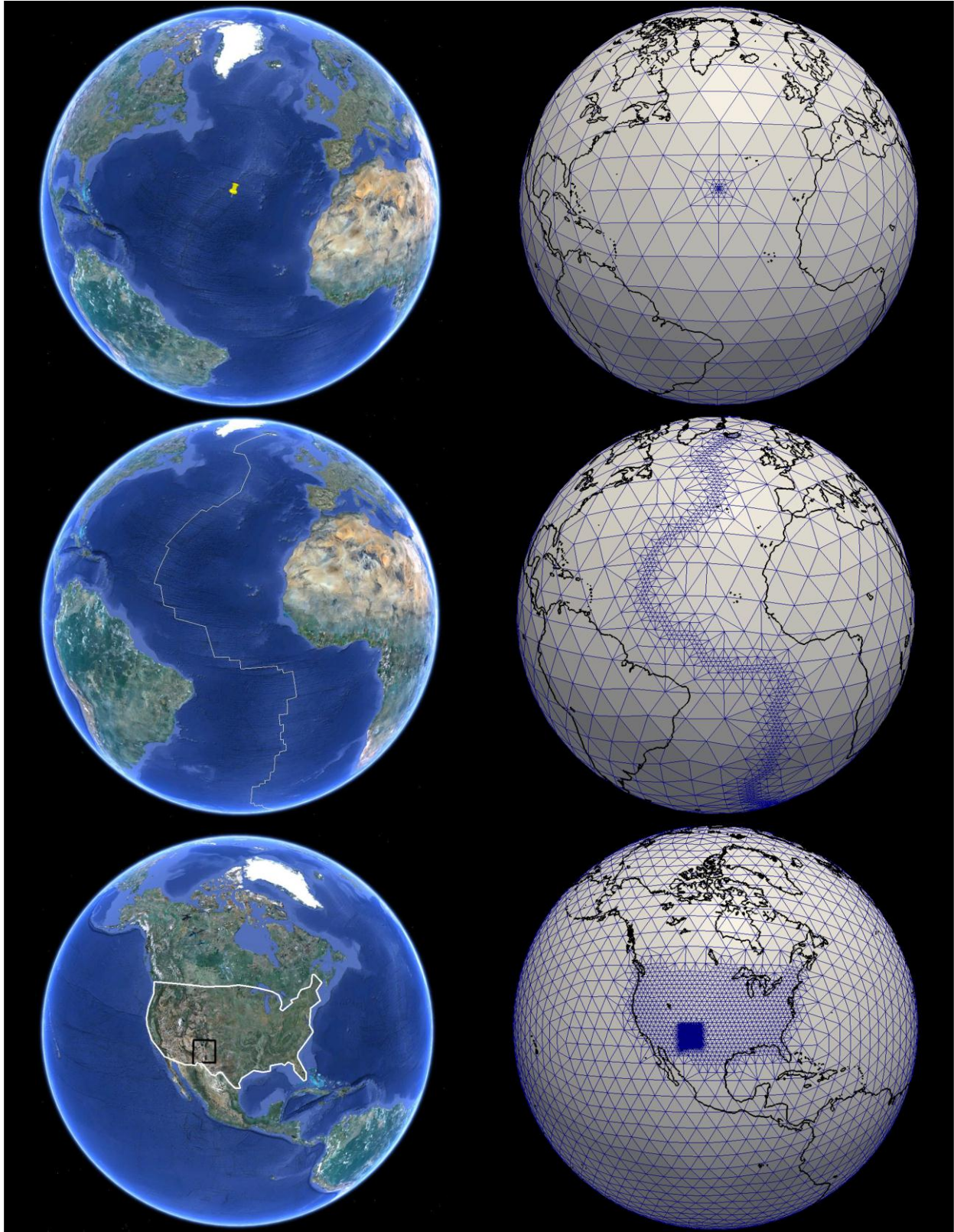
The data values within a profile are continuous.

- A **2D array of Profiles** with  $nVertices \times nLayers$  elements. The first index refers to one of the vertices of the model geometry and the second index refers to one of the layers of the model. For a given vertex index, the 1D array of profiles contains a profile for each layer of the model, stored in order of increasing radius. The last radius of each profile in a 1D profile array must be equal to the first radius of the next profile in the same 1D profile array. While the data values within a single profile are continuous, data values may be discontinuous across profile (i.e. layer) boundaries.
- **Radial interpolators** that interpolate data values in profiles. These include linear interpolators, cubic spline interpolators, and potentially others.
- **2D interpolators** that interpolate values in the 2 geographic dimensions. These include linear interpolators that interpolate values within a single triangle of the 2D tessellations, and higher order interpolators that provide continuous spatial derivatives of the data values.
- **(2+1)D interpolators** that combine 1D and 2D interpolators to interpolate data in 3D. They first use a 1D interpolator to interpolate values at a specified radius in a neighborhood of profile arrays, and then apply a 2D interpolator to those values to find an interpolated value at the desired 3D location.

## **GRID GENERATION**

As mentioned above, grid generation is accomplished with a software application called GeoTessBuilder, which operates in two fundamental modes. In the first, GeoTessBuilder will construct a new 2D geographic grid from scratch. Specification of the radii and data values along the profiles of the model is performed separately by user-generated software (examples are provided with the code). To build a new grid in this manner, the user first specifies a default grid resolution (the edge length of the triangles, in degrees). If nothing further is specified, then a uniform tessellation with that resolution will be constructed. The user may also specify an arbitrary collection of points, paths and polygons, or any combination thereof. GeoTessBuilder will refine the grid in and around those points, paths and polygons, to whatever grid resolution the user requests (Figure 4). The points, paths and polygons can be specified in ascii text files, or can be accessed from kmz files generated by Google Earth.

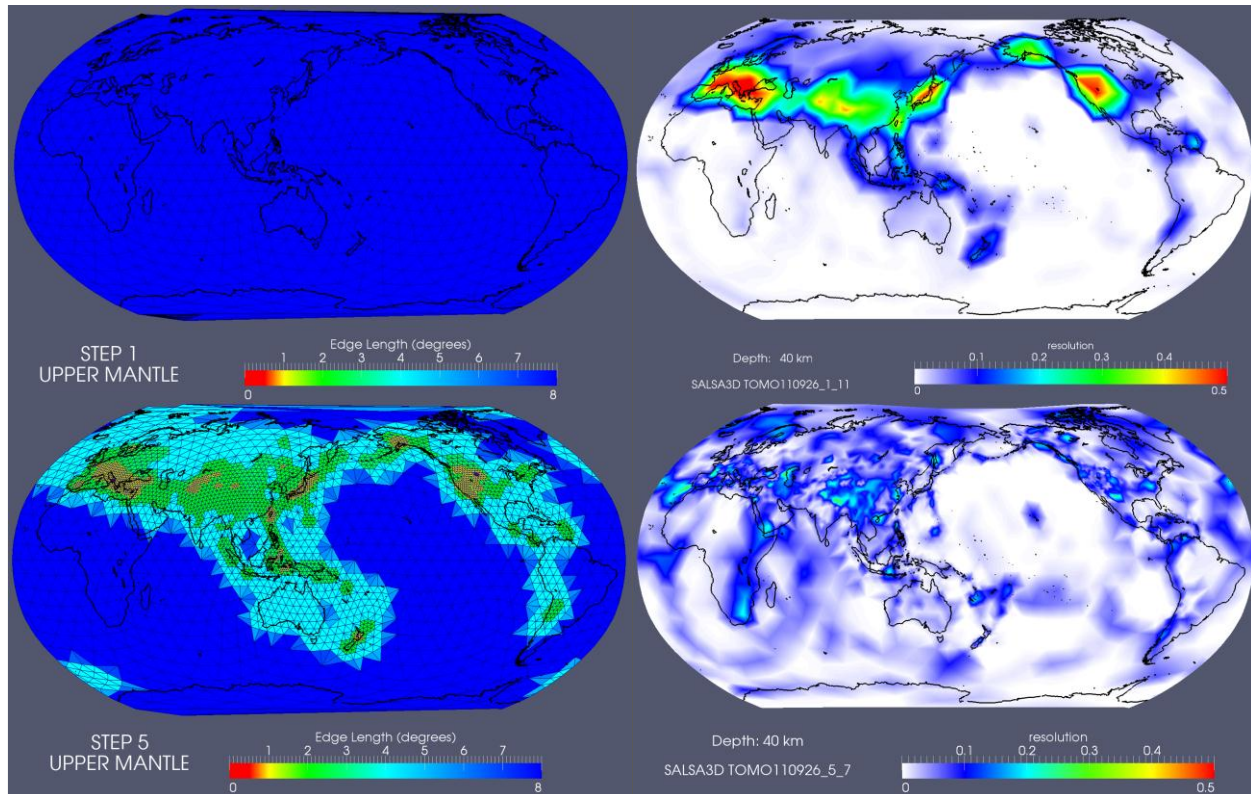
In the second mode of operation, GeoTessBuilder will construct a new 3D grid by refining an existing 3D GeoTess grid. The user supplies an existing grid and a list of grid node indexes. GeoTessBuilder will add additional grid nodes around the specified grid nodes, effectively doubling the spatial resolution around the supplied grid nodes. Refinement is performed in both the geographic and radial dimensions.



**Figure 4 – Grid refinement options. On the left are shown points, paths and polygons generated in Google Earth. GeoTessBuilder can read the Google Earth kmz files and refine the grid in or around the points, paths or polygons.**

## SALSA3D MODEL

We offer two examples of models based on the GeoTess parameterization. The first is SALSA3D, a global P velocity model of the Earth (Begnaud et al., this volume; Begnaud et al., 2011). The SALSA3D model is deployed on a grid that was generated with an adaptive grid refinement algorithm. The starting model consisted of a triangular tessellation with uniform  $8^\circ$  edge lengths (Figure 5a). After an iterative tomographic inversion procedure had converged, the model resolution matrix was computed (Figure 5b). All grid nodes where the diagonal of the model resolution matrix was greater than 0.3 were refined to effectively double the spatial resolution of the model in the vicinity of the nodes with high model resolution. This procedure: tomography followed by model resolution calculation and adaptive grid refinement, was repeated 4 times, at the end of which the grid had spatial resolution as low as  $\frac{1}{2}^\circ$  (Figure 5c) and the model resolution was everywhere less than 0.3 (Figure 5d). This procedure resulted in a model with approximately an order of magnitude fewer nodes than would have been the case if the spatial resolution had been  $\frac{1}{2}^\circ$  everywhere.

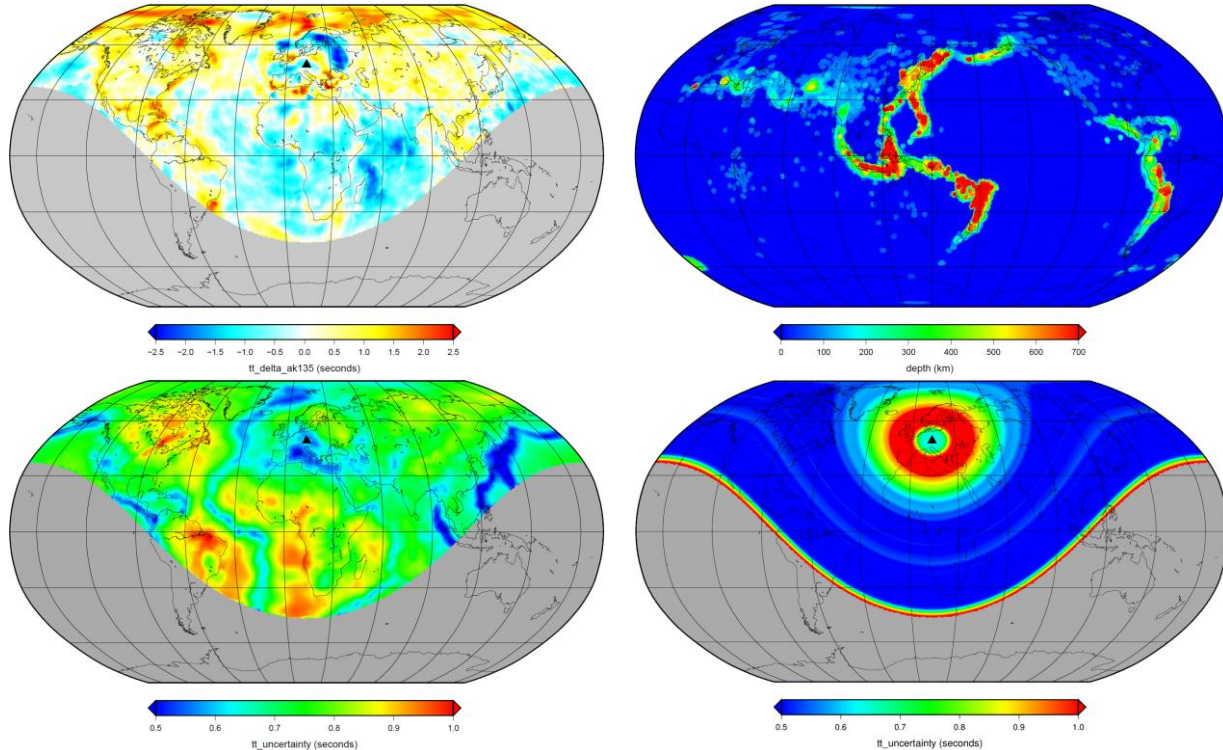


**Figure 5 – SALSA3D grid refinement.**

## LIBCORR3D

LibCorr3D is a software library built upon GeoTess that is used to manage station-phase specific predictions of seismic attributes, and uncertainties of those predictions. For every station in a network, every phase of interest, and every attribute of interest (e.g. travel time, horizontal slowness, station-event azimuth, etc.) a separate GeoTess model is generated which stores 2 quantities: the predicted value and the uncertainty of the prediction. A seismic event location code would use these models by looking up predictions and model uncertainties, rather than computing them on the fly. An alternative strategy would be that LibCorr3D would store corrections to values computed with a 1D model, such as ak135, rather than the predictions themselves, and the locator would add the corrections to the values it calculated with the 1D model.

Figure 6 illustrates the contents of a typical model for station GERES located in Germany. The difference in travel time computed using SALSA3D and ak135 is illustrated in the upper left. In the lower left is the uncertainty of the travel time predictions calculated by SALSA3D (see Hipp et al., this volume, for more information). Standard 1D



**Figure 6 – LibCorr3D model for station GERES located in Germany. a) Difference between travel times computed with SALSA3D and ak135. b) Depth of the bottom of all the LibCorr3D models (not stored in the model). c) Path dependent travel time uncertainty. d) 1D distance dependent model uncertainty (for comparison only, not stored in the model).**

distance dependent travel time uncertainty is shown in the lower right for comparison with the path dependent travel time uncertainties computed with SALSA3D. While the surface values of predicted travel time and uncertainty have been emphasized, it is important to note that these values are stored in a 3D grid. The upper right image in Figure 6 is the depth of the bottom of the LibCorr3D models. Over most of the globe, the model is only one node thick (blue in Figure 6) but the model extends deeper than any recorded earthquake where warranted by recorded seismicity.

This strategy has implications for computational performance. A typical model for a single station-phase-attribute, stored at a resolution of  $0.5^\circ$  horizontally and 30 km radially, will require roughly 50 MB of memory once it is loaded into a computer. For a network of approximately 60 stations, a single phase (first P), and two attributes (travel time and horizontal slowness), this will require approximately 6 GB of RAM to store, which is a modest amount of memory on a modern computer. To meet operational performance requirements, it will be necessary that all the models be loaded into memory and retained there for as long as they are being used. This is because the load time for the models can be substantial (several 10s of seconds up to a minute or so to load an entire 60-station network). Once the models are loaded, the time required to lookup a value in the collection of models is minimal, much less than the time required to compute a 1D or 2½D prediction on the fly. For this strategy to work on a shared computer resource where multiple jobs are running that all require access to the models, it will be desirable to run either the LibCorr3D lookup table facility or the locator itself in a service oriented mode. This implies that a single instance of LibCorr3D (or the locator) is running as a service on a computer and multiple independent jobs running at the same time can access the common lookup table or locator service.

## **SUMMARY**

GeoTess is a model parameterization and software support library that manages the construction, population, storage and interrogation of data stored in 3D Earth models. It is our hope that monitoring agencies will familiarize themselves with GeoTess and implement the ability to accept 3D models in GeoTess format. If this happens, and members of the research community develop the ability to deliver data products in this format, it would greatly facilitate the sharing of 3D Earth model information between model developers and monitoring agencies and would



likely increase the probability that research products are properly evaluated and ultimately implemented by monitoring agencies.

### **ACKNOWLEDGEMENTS**

Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

### **REFERENCES**

- Ballard, S, J. R. Hipp, C. J. Young, 2009, Efficient and accurate calculation of ray theory seismic travel time through variable resolution 3D earth models, *Seismol. Res. Lett.*, **80**, 989-999.
- Begnaud, M. L., S. Ballard, C. J. Young, J. R. Hipp, M. C. Chang, A. V. Encarnacao, C. A. Rowe, W. S. Phillips, L. K. Steck (2011). A Global 3D P-Velocity Model of the Earth's Crust and Mantle for Improved Event Location, in *Proceedings of the 2011 Monitoring Research Review: Ground-Based Nuclear Explosion Monitoring Technologies*, LA-UR-11-04823, Vol. 1, pp. 237-248.
- Reiter, D., W. Rodi, and S. Myers (2011), Estimating the Uncertainty and Predictive Capabilities of Three-Dimensional Earth Models, 2011 Monitoring Research Review, in *Proceedings of the 2011 Monitoring Research Review: Ground-Based Nuclear Explosion Monitoring Technologies*, LA-UR-11-04823, Vol. 1, pp. 132-140.