SEISMIC TOMOGRAPHY TUTORIAL

CIDER 2014: MARK PANNING AND VED LEKIC

BACKGROUND INFORMATION –

As seismic waves – both body (P,S) and surface waves – travel across the Earth from source to receiver, they are affected by variations of density and seismic wavespeeds in the Earth's interior. For example, their **travel times** can become advanced or delayed, their **amplitudes** increased or reduced, or they may become dispersed as different frequency components travel at different velocities. If we can calculate how a particular travel-time, amplitude, or phase dispersion measurement depends on Earth structure (i.e. what its sensitivity kernel is), then we can use these measurements to image the Earth's interior. **Seismic tomography** is one way in which measurements from many source-receiver paths can be self-consistently combined to construct a three-dimensional image of structure in the Earth.

Love and Rayleigh waves, collectively referred to as **surface waves,** are particularly useful for constraining structure in the upper mantle. This is because they are readily observed and because the speed of propagation of their different frequency components has a different sensitivity to depth profiles of velocity. In this tutorial, you will be using tens of thousands of measurements of **phase velocity** – the distance a peak or trough at a specific frequency travels in one second – to infer the geographic variations of phase velocity at a particular frequency. **Phase velocity maps** at different frequencies can be combined to make a three-dimensional model of velocity variations in the upper mantle; however, we will not be performing this task in this tutorial.

MATHEMATICAL FORMULATION

How should a phase velocity measurement made on a particular source-receiver path be related to geographic variations in phase velocity? A simple but very useful answer to this question is provided by the **path average approximation** (PAVA, Woodhouse & Dziewonski, 1984), which states that the deviation $\delta c_k(\omega)$ of the phase velocity measured on a particular path *k* from a reference value $c^0(\omega)$ is the average of the variations in geographic phase velocities $\delta V(r,\theta,\phi)$ encountered along the great circle path joining source and receiver. Written as an equation, therefore, we have the following relationship:

$$\frac{\delta c_k(\omega)}{c^0(\omega)} = \frac{1}{X}\int_{SRC}^{RCV} \frac{\delta V(r,\theta,\phi)}{c^0(\omega)}\,ds,$$

where *X* is the distance from source to receiver, ω is the frequency at which the measurements is made, the integration is carried out over the path from source (SRC) to receiver (RCV), and velocity varies with location specified by radius (r), latitude (θ) and longitude (ϕ).

We can proceed by discretizing the space into a finite number (M) of blocks, which allows us to re-write the integration as a summation:

$$\frac{\delta c_k(\omega)}{c^0(\omega)} = \frac{1}{X}\sum_{j=1}^{M} \frac{\delta V_j}{c^0(\omega)} \Delta s_{kj},$$

where *k* is the index identifying the path in question, *j* is the index identifying the block of interest, and the infinitesimal path length element *ds* has been replaced by a finite path length within the block *j*, *Δs*$_{kj}$. Subscripts *k* and *j* must be introduced because the length of a great circle within a block depends both on which block *(j)* and which great circle *(k)* we are considering.

If we measure the phase velocity on N paths, we can write a matrix equation:

$$\begin{bmatrix} \delta c_k/c^0 \\ \vdots \\ \delta c_N/c^0 \end{bmatrix} = \begin{bmatrix} \Delta s_{11}/X & \cdots & \Delta s_{1M}/X \\ \vdots & \ddots & \vdots \\ \Delta s_{N1}/X & \cdots & \Delta s_{NM}/X \end{bmatrix} \begin{bmatrix} \delta V_j/c^0 \\ \vdots \\ \delta V_j/c^0 \end{bmatrix}.$$

We usually refer to the 1 by M vector of **relative velocity perturbations** as the **model vector**, **m**, and the 1 by N vector of measured phase velocities as the **data vector, d**. Using this nomenclature, and written compactly, we have:

$$\frac{\delta c_k}{c^0} = d_k = (\Delta s_{jk}/X)\, \delta V_j/c^0 = G_{kj} m_j \text{ or } \mathbf{d} = \mathbf{Gm}.$$

In order to incorporate information on measurement reliability (data uncertainty), we usually specify a variance σ_k^2 associated with each measurement. Because we want to assign less importance to data with high uncertainty, we incorporate measurement reliability information using a diagonal matrix, C_D^{-1}, whose entries are $1/\sigma_k^2$.

When the number of phase velocity measurements is equal to the number of unknowns $(N = M)$, the matrix **G** is square, and the model vector can be calculated by multiplying d by the inverse of **G**, i.e. $(m = G^{-1}C_D^{-1}d)$, provided that the inverse exists (i.e. that the matrix G is not singular). When N ≠ M, we can multiply through by the transpose of G, in which case the solution will be given by: $m = (G^T C_D^{-1} G)^{-1} G^T C_D^{-1} d$. However, if the number of measurements is smaller than the number of unknowns, i.e. $N < M$, the problem is said to be **underdetermined**, and $(G^T C_D^{-1} G)$ will always be singular. Even when the number of measurements is greater than the number of unknowns, i.e. $N > M$ and the problem may be **overdetermined,** the invertibility of $(G^T C_D^{-1} G)$ will depend on how complementary the path coverage of the various source-receiver pairs is. This makes sense, since even a million measurements of phase velocity between Japan and California will not tell you what the phase velocity in the South Atlantic is.

TASK #1: SETTING UP THE DATA VECTOR, DATA COVARIANCE MATRIX, AND BLOCK SIZE

- Run the first block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER (or cmd+enter on a Mac).
- Specify the frequency of observation (ω). Your choices are periods of 50 s, 100 s, or 150 s.
- Specify the appropriate reference phase velocity c^0(ω) in km/sec. For 50 s data, the appropriate value is 3.952 km/s, for 100 s, it is 4.080 km/s, and for 150 s, it is 4.280 km/s.
- Specify the approximate size (in degrees, 1° ≈ 111 km) of the blocks. Note that choosing values smaller than 5.0 might cause the inversion to run extremely slowly.

- Specify the multiplier on the data error estimates. If you want to use the actual reported measurement uncertainties, set the multiplier to 1. Values higher than 1 will act to “degrade” the data, while values less than 1 will force the model that is obtained to predict the measures phase velocities more precisely.
- This code block will read all of the measurements and uncertainties of your chosen dataset and calculate all the elements of the **G** matrix for the dataset and model parameterization of your choice. This may take a few minutes.
- Finally, this block of code will produce a map showing the hit count of the dataset. This is simply a sum of how many individual paths cross a portion of each block of the model. This lets us see the sampling of our dataset.

REGULARIZATION

Tomographers can use a number of procedures for **regularizing** the problem, or, in other words, for obtaining estimates of **m** even when the matrix $(G^T C_D^{-1} G)$ is not invertible. In this tutorial, we will explore the effects of two regularization procedures:

1. Introduction of prior information on model parameters using the model covariance matrix C_M.
2. Singular value decomposition and elimination of eigenvalues smaller than some threshold value.

1. MODEL COVARIANCE MATRICES

Usually, tomographers have other sources of information about phase velocities in the Earth besides those coming from the particular dataset of phase velocity measurements. For example, we might know that $\delta V / c^0$ should be normally distributed with a variance σ^2. Or, we might have reason to believe that the geographic variations of $\delta V / c^0$ are smooth, in the sense that the velocities at locations closer than some distance L will tend to be similar. Mathematically, one example of an *a priori* model covariance matrix that contains both types of information would be:

$$C_{ij} = \sigma^2 \exp\left[-{}^{1}\!/_{2} \left({}^{X_{ij}}\!/_{L} \right)^2 \right],$$

where $X_{ij}$is the distance between blocks *i* and *j*.

Taking account of this prior information, the solution is given by (Tarantola & Valette, 1982): $m = \widetilde{C_M} G^T C_D^{-1} d$, where $\widetilde{C_M} = (G^T C_D^{-1} G + C_M^{-1})^{-1}$ is the posterior model covariance matrix. The advantage of incorporating this prior information is that it will ensure that $\widetilde{C_M}$ exists and that a solution can be obtained. The posterior model covariance matrix is an extremely useful object, because it not only contains information on the uncertainty of the block estimates obtained with the phase velocity measurement data, but also contains information on how much velocities in two different blocks may trade off with each other. We will return to this later in the tutorial.

Choosing a small value of variance σ^2 is equivalent to strongly **damping** the model vector toward the starting model (in essence, doing our best to force $\delta V_j / c^0 \rightarrow 0$). When L is set to 0, the model covariance matrix becomes a diagonal matrix, and its application is called **norm damping**. On the other hand, setting L to be a large distance will result in a very laterally smooth model, and force $\delta V_i - \delta V_j \rightarrow 0$ across nearby blocks.

TASK #2: CARRY OUT TARANTOLA & VALETTE INVERSION

- Specify a value for *a priori* model variance (we will use the same value for all blocks). Smaller numbers will yield more highly damped models.
- Specify a value for L (in degrees, remembering that 1° ≈ 111 km). Setting L = 0 will result in pure norm damping, whereas any positive value will incorporate smoothing in the solution.
- Run the second block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER (or CMD+ENTER).
- To plot up the maps of phase velocity obtained with your choice of input parameters, run the third block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER (or CMD+ENTER).
- Experiment with different values of L and σ^2, and see how the resulting phase velocity maps change. Some combinations of L and σ^2 will yield crazy phase velocity maps (this is equivalent to not introducing a sufficient amount of prior information); generally, you will need to decrease the σ^2 as you increase L.

2. SINGULAR VALUE DECOMPOSITION

An alternative approach toward obtaining a model estimate m even when $(G^T C_D^{-1} G)$ is not invertible is to perform a singular value decomposition. This is a tool taken from linear algebra in which we decompose the N x M **G** matrix as $\mathbf{G} = \mathbf{U\Lambda V}^{\mathrm{T}}$, where **U** is an N x N square matrix, $\mathbf{\Lambda}$ is an N x M matrix, and **V** is an M x M square matrix. This decomposition is chosen to represent G in the coordinate system defined by a series of N eigenvectors $\overrightarrow{U_J}$ which "span" the data space (linear algebra term for saying that any possible data vector can be represented as a linear combination of the columns of **U**), and M eigenvectors $\overrightarrow{V_J}$, which span the model space, each of which corresponds to a singular value λ_j, which are the Λ_{jj}, or diagonal, elements of $\mathbf{\Lambda}$. If, as is typical in tomography, there are more data than model parameters (N>M), there will be M singular values, and the remaining rows of $\mathbf{\Lambda}$ will be zeros. We can therefore write

$$G = \begin{pmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ \overrightarrow{U_1} & \overrightarrow{U_2} & \dots & \overrightarrow{U_N} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_M \\ \vdots & \vdots & \dots & 0 \\ 0 & \dots & \dots & 0 \end{pmatrix} \begin{pmatrix} \leftarrow & \overrightarrow{V_1} & \rightarrow \\ \leftarrow & \overrightarrow{V_2} & \rightarrow \\ \leftarrow & \vdots & \rightarrow \\ \leftarrow & \overrightarrow{V_M} & \rightarrow \end{pmatrix}.$$

There are a lot of interesting things about this decomposition which are beyond the scope of this tutorial, but the important thing in this context is that inverting is easy, because the matrix inverses of U and V are simply their transposes, i.e. $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$, where I is the identity matrix. With this representation, therefore, we can write an expression for the desired inverse matrix:

$$\mathbf{d} = \mathbf{Gm}$$

$$\mathbf{d} = \mathbf{U\Lambda V}^{\mathrm{T}}\mathbf{m}$$

$$\mathbf{m}_{\text{est}} = \mathbf{V}_{\mathrm{p}}\mathbf{\Lambda}_{\mathrm{p}}^{-1}\mathbf{U}_{\mathrm{p}}^{\mathrm{T}}\mathbf{d},$$

where $\mathbf{m}_{\text{est}}$ is our estimate of the true model, and the *p* subscript indicates we've reduced the number of columns of V and U to the smaller of N or M (usually M for most tomographic problems), which makes $\mathbf{\Lambda}$ square and invertible, so that we can write

$$\mathbf{V}_\mathrm{p}\mathbf{\Lambda}_\mathrm{p}^{-1}\mathbf{U}_\mathrm{p}^\mathrm{T} = \begin{pmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ \overrightarrow{V_1} & \overrightarrow{V_2} & \dots & \overrightarrow{V_M} \\ \downarrow & \downarrow & \downarrow & \downarrow \end{pmatrix} \begin{pmatrix} 1/\lambda_1 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\lambda_M \end{pmatrix} \begin{pmatrix} \leftarrow & \overrightarrow{U_1} & \rightarrow \\ \leftarrow & \overrightarrow{U_2} & \rightarrow \\ \leftarrow & \vdots & \rightarrow \\ \leftarrow & \overrightarrow{U_M} & \rightarrow \end{pmatrix}$$

From this expression, we can immediately see that the desired inverse will not exist if any of the eigenvalues $\lambda_j \rightarrow 0$, because this will result in an undefined value due to dividing by zero. Even if the values are only very close to zero, the inverse of the singular values will be very large, and have a very large influence on our solution.

To correct for this problem, we will do something that may seem quite unusual. First, we will arrange the eigenvalues in decreasing order so that $\lambda_j > \lambda_{j+1}$. Then, for all eigenvalues smaller than some threshold value, i.e. $\lambda_j < \varepsilon\lambda_1$, we will set $1/\lambda_j \rightarrow 0$. Doing so allows us to approximate the inverse, and obtain a matrix analogous to the $\widetilde{C_M}$ in the Tarantola & Valette formalism. What does this mathematical procedure do to the problem of using phase velocity measurements to estimate phase velocities in geographic blocks across the globe? Conceptually, you can think about it this way: We are unable to resolve velocity variations in the combination of blocks represented by the eigenvector $\overrightarrow{V_j}$ that corresponds to those $\lambda_j < \varepsilon\lambda_1$. By setting $1/\lambda_j \rightarrow 0$, we are telling the inversion to simply ignore these combinations of blocks, and estimate velocities in blocks where such an estimate is feasible given the phase measurement data.

Choosing larger threshold values (e.g. $\varepsilon = 0.5$) will result in fewer effective model parameters being estimated (less variation in $\delta V_j/c^0$), but does not have a direct relationship to norm damping or smoothing the way that specifying the *a priori* model covariance matrix does. Note that in the supplied code, we weight the rows of the G matrix and the elements of the data vector by the inverse data variance, C_D^{-1}, similar to how we used this weighting in the Tarantola and Valette style inversion.

TASK #3: CARRY OUT SVD-BASED INVERSION

- Specify a threshold value (ε) for deciding for which eigenvalues λ_j we will be setting $1/\lambda_j \rightarrow 0$. Consider a wide range of threshold values from 0.0001 to 0.5.
- Run the fifth block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER (or CMD+ENTER).
- To plot up the maps of phase velocity obtained with your choice of threshold value, run the sixth block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER (or CMD+ENTER).
- The first time you run this, the singular value decomposition may take several minutes to run, but future runs will use saved decompositions and run much more quickly.
- Experiment with different values of ε, and see how the resulting phase velocity maps change.

QUANTIFYING UNCERTAINTY

In science, measurements and inferences are only meaningful if they are associated with an uncertainty. There are three main sources of error that can affect our estimated phase velocity maps:

1. *Errors due to data uncertainty* – Measurements of average phase velocity on a particular source-receiver path are inherently uncertain. In this tutorial – and commonly in global seismology – we assume that this measurement uncertainty is normally distributed and uncorrelated from path to path.
2. *Errors due to regularization* – The introduction of smoothing/damping (a priori model covariance) and SVD thresholding both affect our estimated phase velocity maps.

3. *Errors due to theoretical approximations* – The phase velocity measured on a particular path ($\delta c_k/c^0$) depends on geographic variations in local wavespeed ($\delta V_j/c^0$) in a more complicated way than described by PAVA. Therefore, this theoretical approximation will introduce additional error into our phase velocity maps. Mathematically, these errors can be partially modeled by introducing additional terms in the data covariance matrix, and are equivalent to adding noise to our measurements. **Finite frequency kernels** can partially alleviate errors due to theoretical approximations. Standard uncertainty analysis techniques – such as those we will perform in this tutorial – neglect this source of error.

Here, we will estimate phase velocity map uncertainty in two ways.

POSTERIOR MODEL COVARIANCE MATRIX

The diagonal entries of the posterior covariance matrix $\widetilde{C_M}$ are the variances of the block velocity estimates ($\delta V_j/c^0$). By taking the square root of the diagonal entries and plotting them at the geographic location of the block to which they are associated, we can obtain a map of uncertainty on our phase velocity estimates. The off-diagonal entries of $\widetilde{C_M}$ contain information on trade-offs between parameters. In other words, they provide a quantitative answer to the question: if I change the velocity in block A by some amount, by how much does velocity in block B have to change? If there are M blocks given our chosen block size, then there are M*(M-1)/2 pairs of blocks to analyze for tradeoffs. Visualizing this information is not easy.

RESOLUTION MATRIX

A standard way of visualizing how well constrained our phase velocity maps are is to use a resolution matrix approach. In this approach, we start with an exact, arbitrary, phase velocity map. Commonly, a checkerboard pattern **m** is chosen, in which $\delta V_j/c^0$ alternates between $\pm\gamma$ with some characteristic wavelength. (NB: this choice is in many ways suboptimal, due to the fact that it only probes the retrieval of structure at the chosen wavelength.) Then, artificial data are calculated using this checkerboard pattern $d'_k = G_{kj}m_j$.

Using this simulated and data **d'**, we can obtain a model estimate **m'** in the same way we did using the actual phase velocity measurements: $m' = \widetilde{C_M}G^TC_D^{-1}d' = \widetilde{C_M}G^TC_D^{-1}Gm$. Equivalently, we can write **m = Rm'**, where **R** is called the **resolution matrix,** and is given by:

$$R = \widetilde{C_M}G^TC_D^{-1}Gm$$

By comparing the estimate **m'** with the exact input pattern **m**, we can get a sense of how well our input exact model is resolved. To quantify how well any possible structure is resolved, we simply have to multiply its model vector by the resolution matrix.

QUANTIFYING UNCERTAINTY DUE TO MEASUREMENT ERROR

To simulate the effects of measurement uncertainty, random numbers (ϵ_k) drawn from normal distributions with specified variance can be added to the synthetic data vector, $d'_k = G_{kj}m_j + \epsilon_k$. Alternatively, the random numbers can be drawn from a different distribution, in order to quantify the error introduced by the assumption that data uncertainty is normally distributed.

TASK #4: ESTIMATE UNCERTAINTIES

- Choose a spherical harmonic angular order *l* for generating a global checkerboard pattern. The length scale (in degrees) corresponding to a particular *l* is 180/*l*. If you set the angular order that is too high for your chosen block size, the input checkerboard will not have the desired appearance.
- Plot up the posterior model uncertainties as well as the input and output checkerboard patterns for your choice of inversion parameters:
  - For Tarantola and Valette inversion, run the fourth block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER.
  - For SVD-thresholding inversion, run the seventh block of *InverseTutorial.m* by clicking inside it and typing CTRL+ENTER.
- Each of these blocks will plot up an input and output checkerboard from the resolution test, as well as a map of estimated model error.
- Systematically explore how changing the inversion parameters (ε, σ^2 and L) affects the estimates of uncertainty and resolving power. Pay close attention to scale bars when comparing error maps from different runs!
- How does the resolution and error compare with the hit count plot you made back in task #1?

FINAL STEP: CLEANUP

When you are done with the tutorial, be sure to quickly run block 8 of InverseTutorial.m in order to remove some items from your matlab path that were added by the script all the way back in block 1..

ACKNOWLEDGMENTS

This tutorial was based on a tutorial and fortran codes for surface wave phase velocity inversion developed by Guy Masters, of the Scripps Institute for Oceanography at the University of California, Santa Cruz. The surface wave phase velocity measurements were made by the research group of Göran Ekström, currently at Lamont-Doherty Earth Observatory at Columbia University. Plotting is accomplished in this tutorial by the M_Map package of matlab functions (http://www.eos.ubc.ca/~rich/map.html), which is supplied with the tutorial.