Seismic Tomography (Part III, Solving Inverse Problems)

Problems in solving generic AX = B

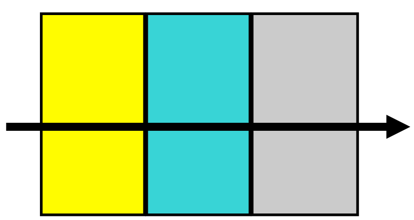
Case 1: There are errors in data such that data cannot be fit perfectly (analog: simple case of fitting a line while there the data actually do not fall on it).

Case 2: There are more equations than unknowns (i.e., say 5 equations and 3 unknowns). This is usually called an ***over-determined system***. (Condition: **the equations are not linearly dependent**).

Case 3: There are fewer equations than unknowns, this is called an ***under -determined*** system. (**no unique solution**).

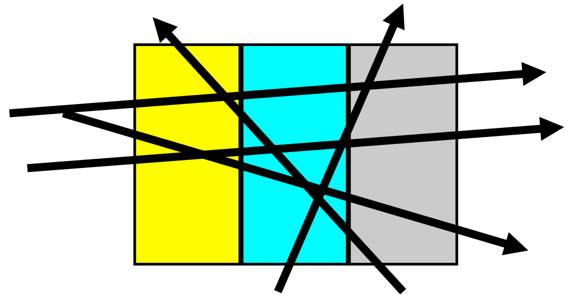
Example from Seismic Structural Study

Under-determined



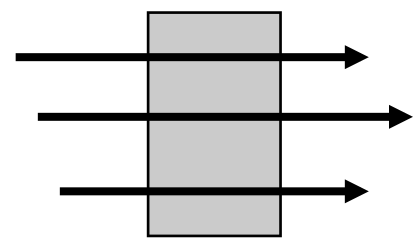
More unknowns than number of observations, can't solve matrix equation exactly, ill conditioned

mix-determined



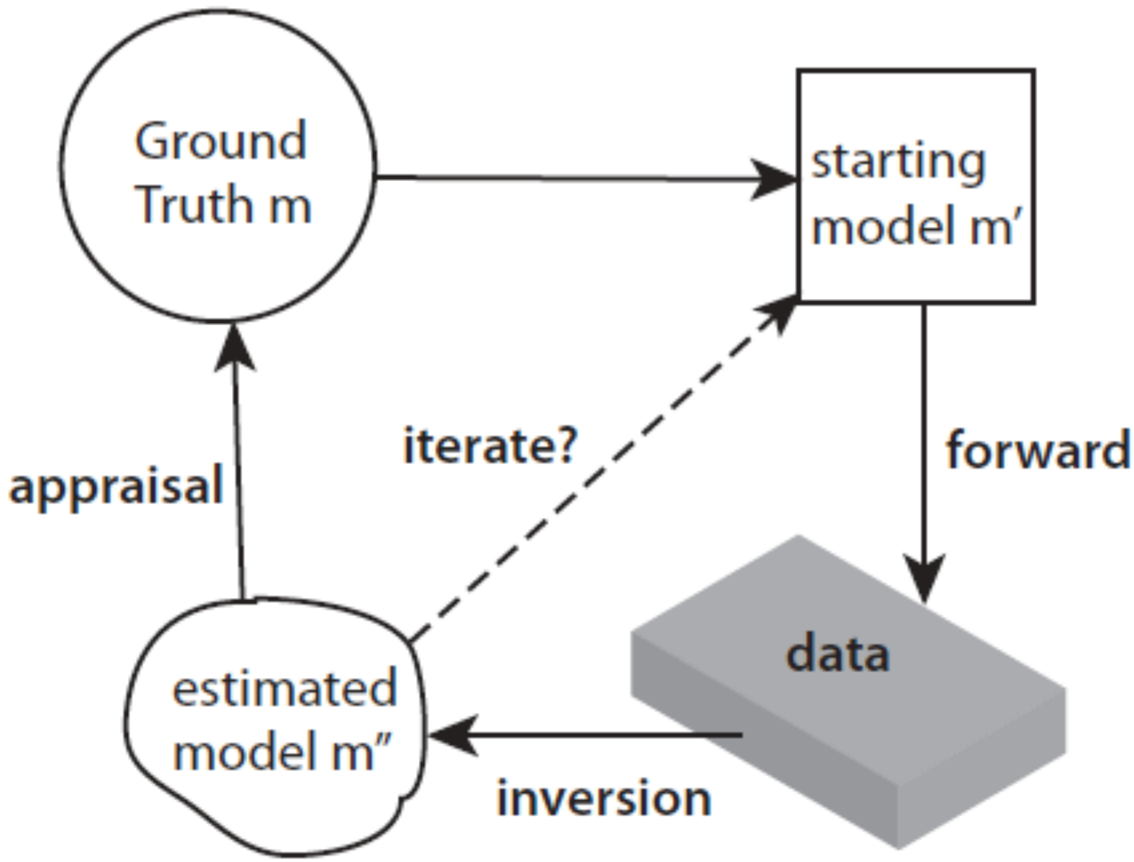
Works well in terms of solving the structures since coverage is good.

over-determined



e.g., 1 model coef (unknown) but many observations.
Worse yet when they conflict with each other

Processes in seismic inversions in general



Liu & Gu, Tectonophys, 2012

Simple Inverse Solver for Simple Problems

Cramer's rule:

Suppose

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}$$

Consider determinant

$$D = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

Now multiply D by x (consider some x value), by a property of the determinants, multiplication by a constant x = multiplication of a given column by x.

$$xD = \begin{vmatrix} a_1x & b_1 & c_1 \\ a_2x & b_2 & c_2 \\ a_3x & b_3 & c_3 \end{vmatrix}$$

Property 2: adding a constant times a column to a given column does not change determinant,

$$xD = \begin{vmatrix} a_1 x + b_1 y + c_1 z & b_1 & c_1 \\ a_2 x + b_2 y + c_2 z & b_2 & c_2 \\ a_3 x + b_3 y + c_3 z & b_3 & c_3 \end{vmatrix} = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}$$

Then, follow same procedure, if D !=0, d != 0,

$$x = \frac{\begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}}{D} \qquad y = \frac{\begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}}{D} \qquad z = \frac{\begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}}{D}$$

<u>Something good about Cramer's Rule:</u>

(1) It is simpler to understand

(2) It can easily extract one solution element, say x, without having to solve simultaneously for y and z.

(3) The so called "D" matrix can really be some A matrix multiplied by its transpose, i.e., **D=A^{T}A, in other words, this is equally applicable to least-squares problem.**

Other solvers:

Common matrix factorization methods

(1) ***Gaussian Elimination*** and Backsubstitution:

$$\begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a'_{33} & a'_{34} \\ 0 & 0 & 0 & a'_{44} \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

(2) ***LU Decomposition***: **L**=lower triangular **U**=upper triangular

Key: write $\mathbf{A = L * U}$

So in a 4 x 4,

$$\begin{bmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix} \bullet \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & 0 & b_{33} & b_{34} \\ 0 & 0 & 0 & b_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

$$\mathbf{A \bullet x = (L \bullet U) \bullet x = L \bullet (U \bullet x) = b}$$

First solve $\mathbf{L \bullet y = b}$

then solve $\mathbf{y = U \bullet x}$

Advantage: can use Gauss Jordan Elimination on triangular matrices!

(3) ***Singular value decomposition*** (SVD): useful to deal with set of equations that are either singular or close to singular, where LU and Gaussian Elimination fail to get an answer for.

Ideal for solving least-squares problems.

Express A in the following form:

$$\begin{pmatrix} & & \\ & \mathbf{A} & \\ & & \end{pmatrix} = \begin{pmatrix} & & \\ & \mathbf{U} & \\ & & \end{pmatrix} \cdot \begin{pmatrix} w_1 & & \\ & w_2 & \\ & & w_3 \end{pmatrix} \cdot \begin{pmatrix} & & \\ & \mathbf{V}^{\mathbf{T}} & \\ & & \end{pmatrix}$$

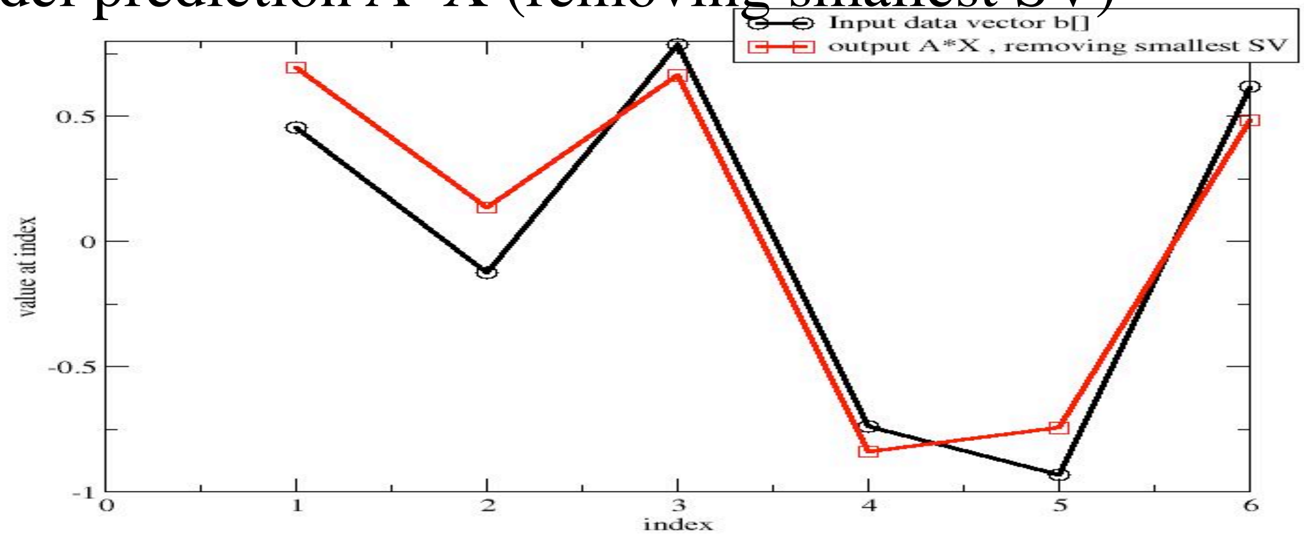
U and **V** are orthogonal matrices (meaning each row/column vector is orthogonal). If **A** is square matrix, say 3 x 3, then **U V** and **W** are all 3 x 3 matrices.

orthogonal matrices-→ **Inverse=Transpose.** So **U** and **V** are no problems and inverse of **W** is just **1/W**,

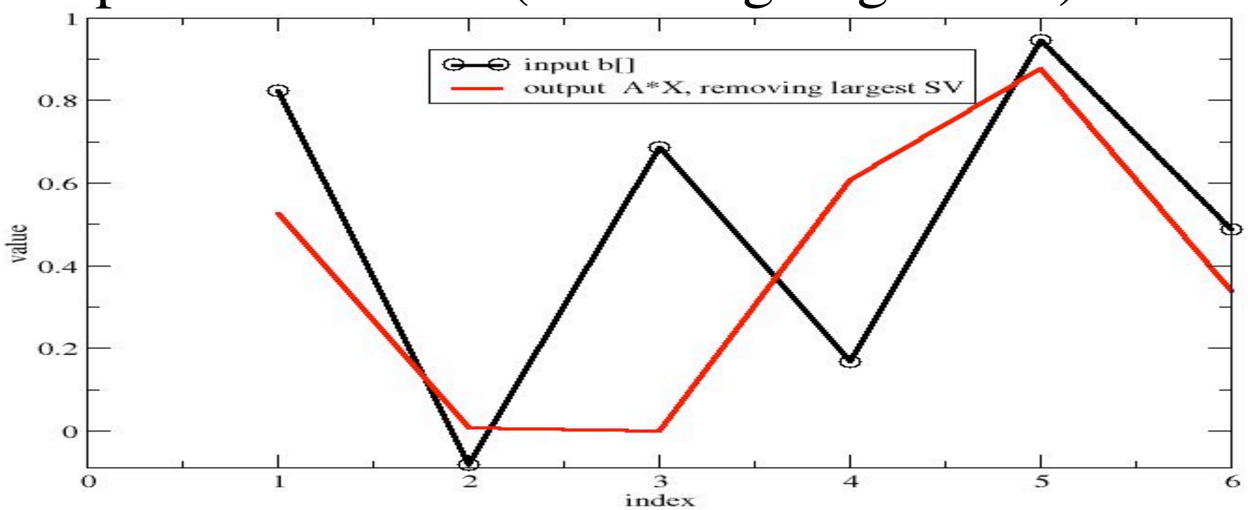
$$\mathbf{A^{-1} = V * [diag(1/w_j)]*U^T}$$

The diagonal elements of W are singular values, the larger, the more important for the large-scale properties of the matrix. So naturally, damping (smoothing) can be done by selectively throw out smaller singular values.

Model prediction A*X (removing smallest SV)

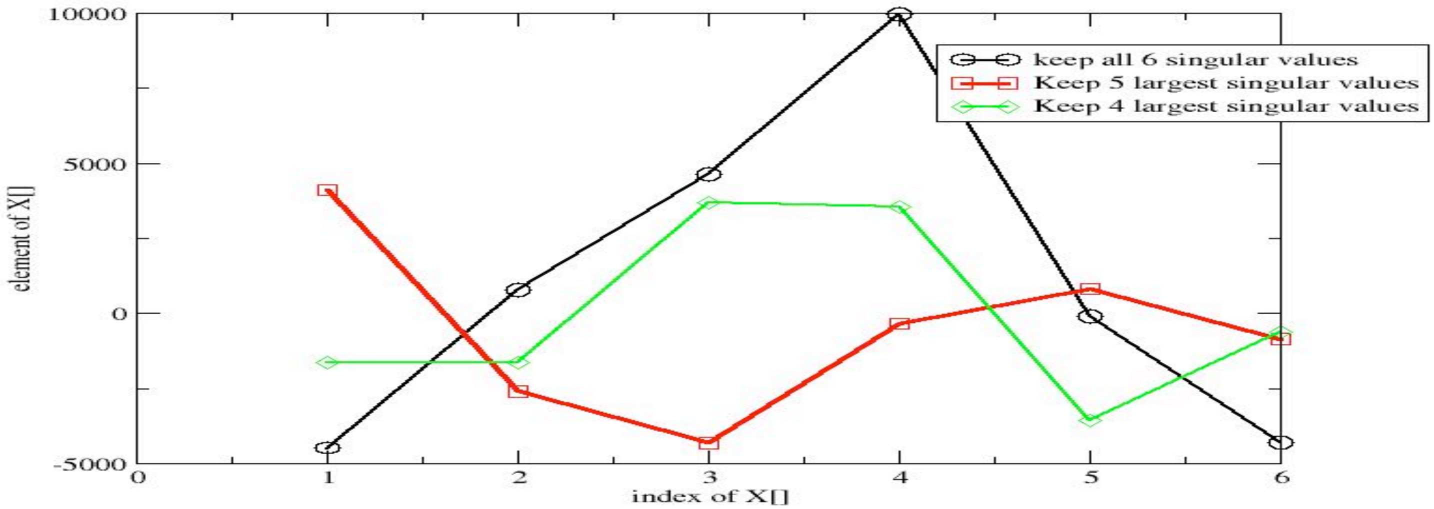


Model prediction A*X (removing largest SV)



We can see that a large change (poor fit) happens if we remove the largest SV, the change is minor if we remove the smallest SV.

Solution vector X[] elements



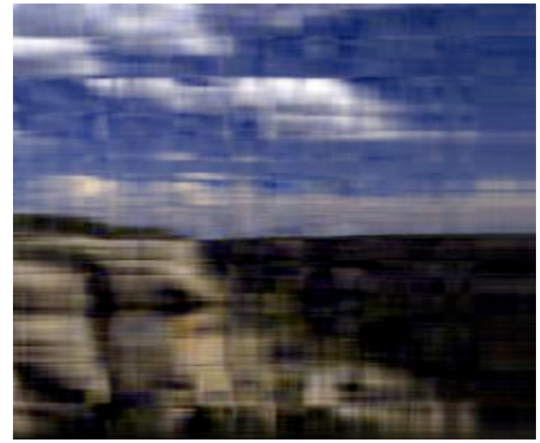
Black --- removing none
Red ---- keep 5 largest SV
Green --- Keep 4 largest SV

Generally, we see that the solution size decreased, SORT of similar to the damping (regularization) process in our lab 9, but SVD approach is not as predictable as damping. It really depends on the solution vector **X** and nature of **A**. The solutions can change pretty dramatically (even though fitting of the data vector doesn't) by removing singular values. Imagine this operation of removal as changing (or zeroing out) some equations in our equation set.

2D Image compression: use small number of SV to recover the original image

Keep all

Keep 5 largest



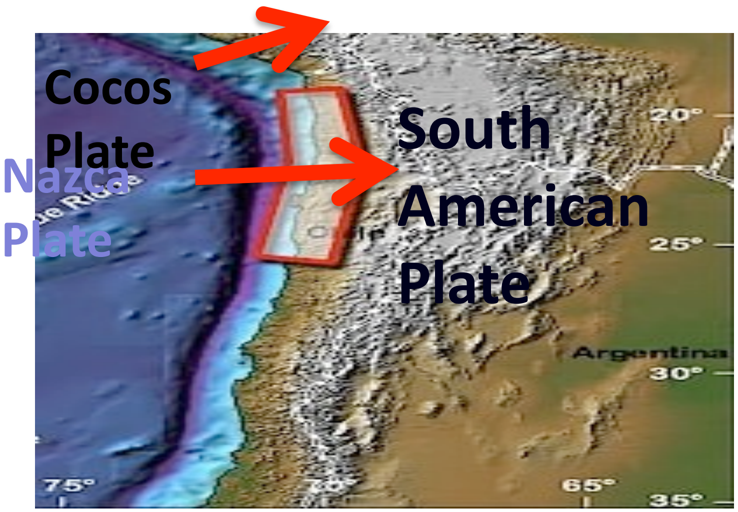
Keep 10 largest

Keep 30 largest

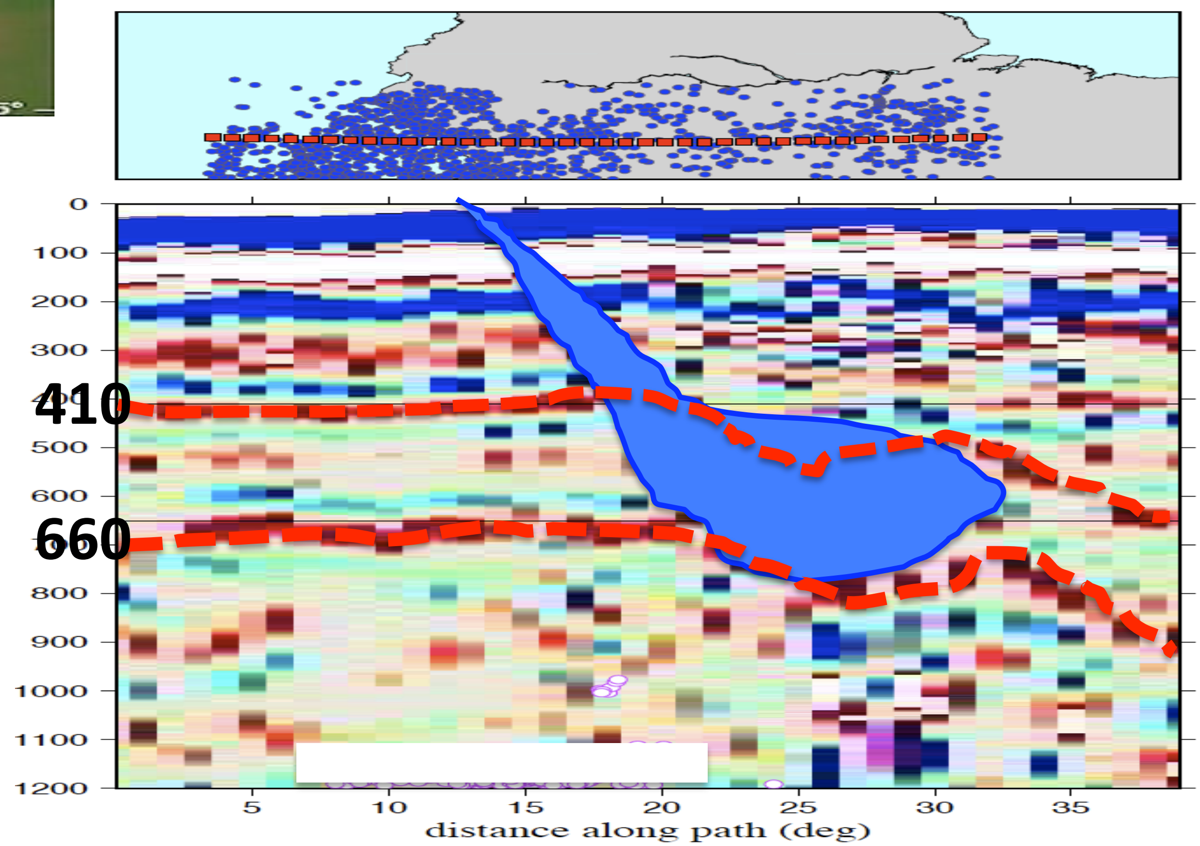
Keep 60 largest

Keep 80 largest

Example of 2D SVD Noise Reduction



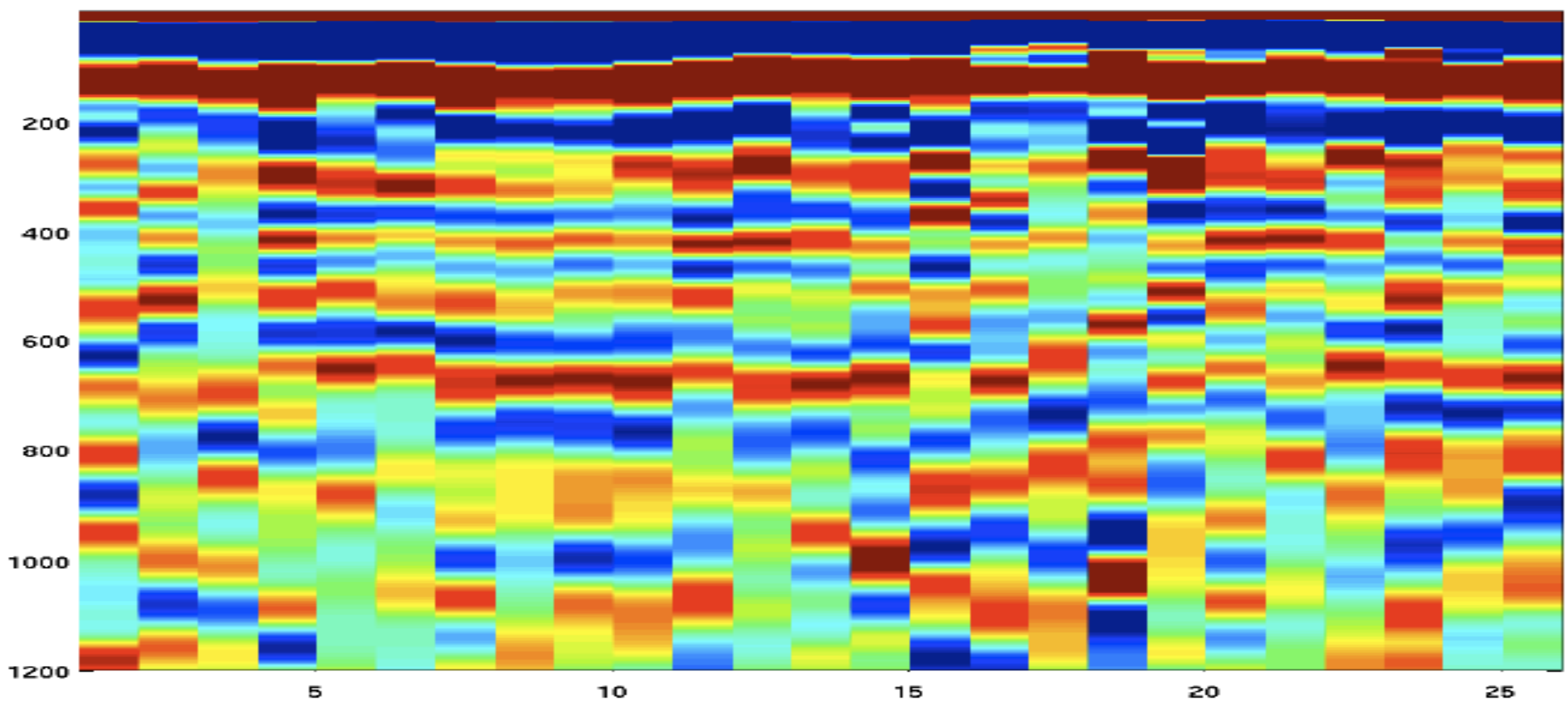
South American Subduction System (the subducting slab is depressing the phase boundary near the base of the upper mantle)



Courtesy of
Sean Contenti

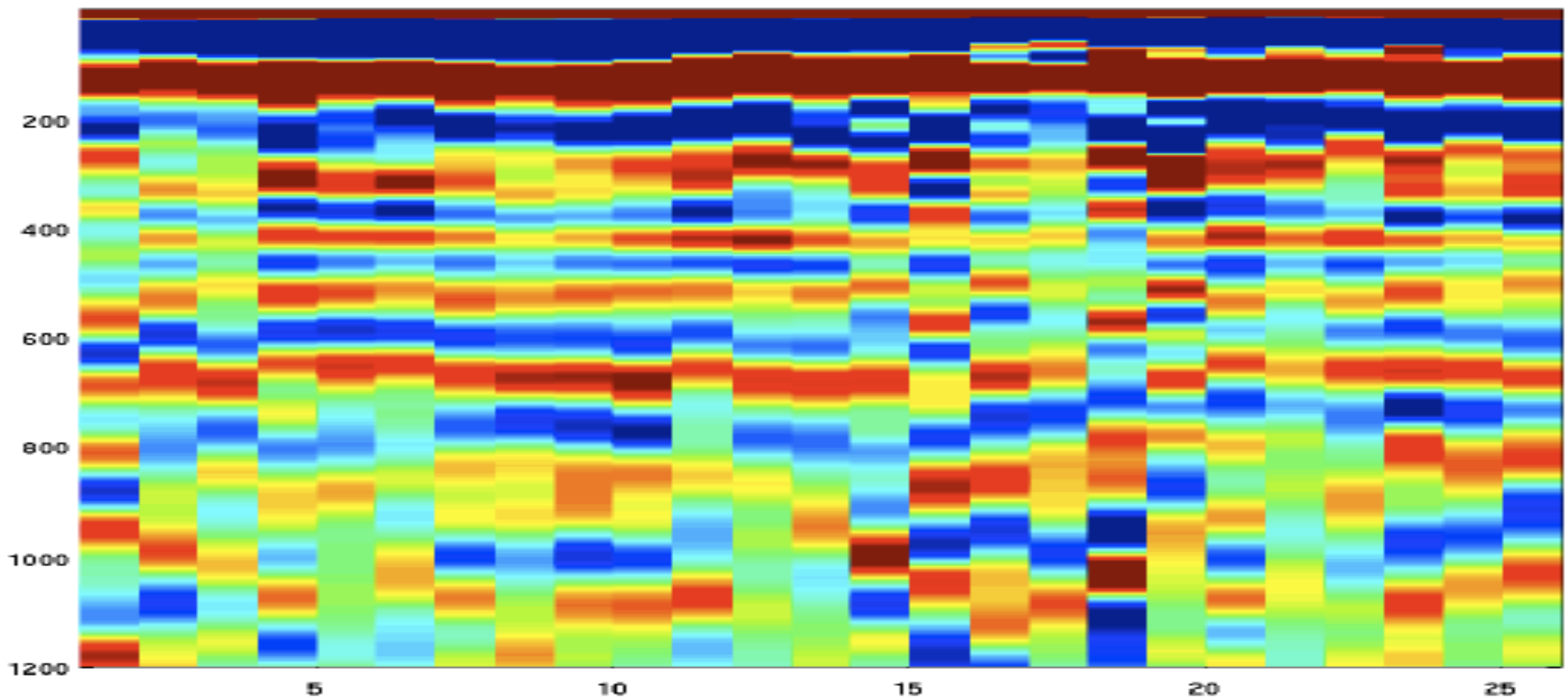
Result using all 26 Eigenvalues

Pretty noisy with small-scale high amplitudes, both vertical and horizontal structures are visible.

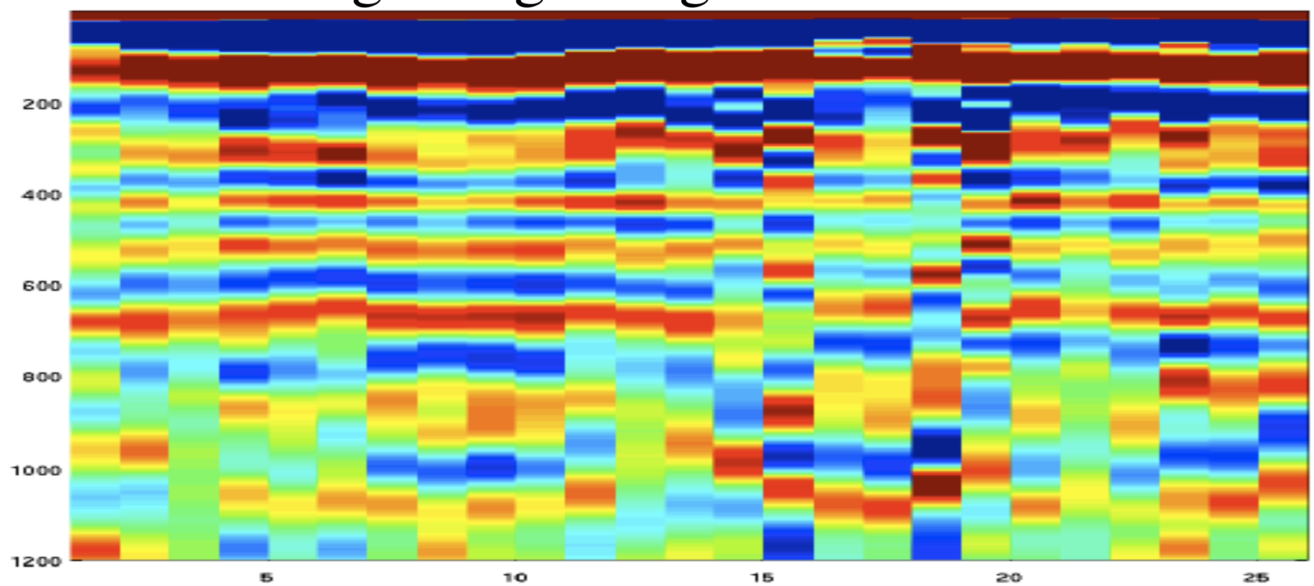


Result using 10 largest Eigenvalues

Some of the higher frequency components are removed from the system. Image appears more linear.



Retaining 7 largest eigenvalues



Retaining 5 largest eigenvalues