**Mid-term Test**

Consider the following problem in cross-well tomography.

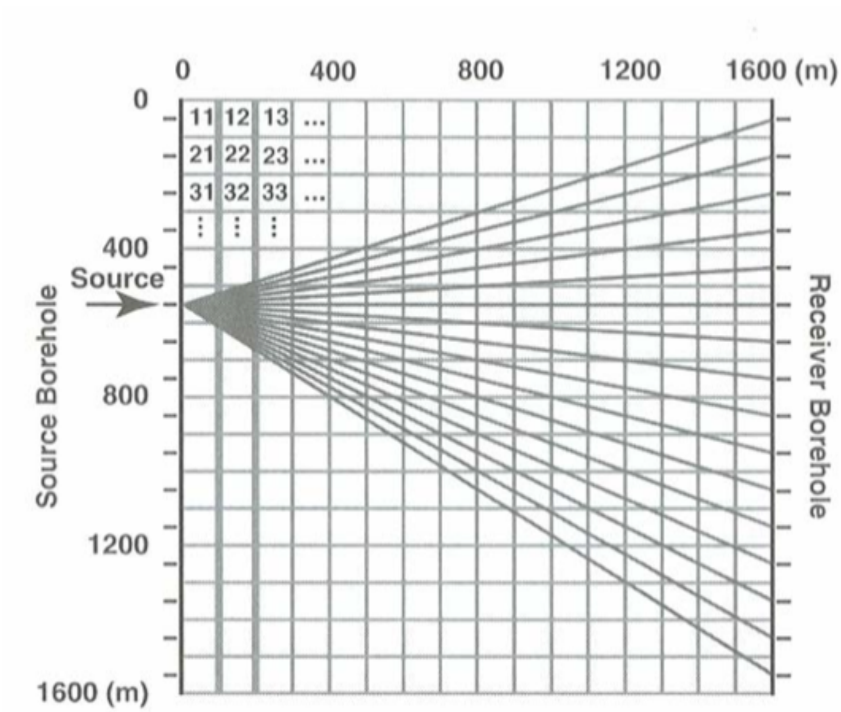


Figure 1 Cross-well tomography problem, showing block discretization, block numbering convention, and one set of straight of source-receive ray paths.

Two vertical wells are located 1600 meters apart. A seismic source is inserted in one well at depths of 50, 150,..., 1550 m. A string of receivers is inserted in the other well at depths of 50, 150,..., 1550 m. See Figure 1. For each source-receiver pair, a travel time is recorded, with a measurement standard deviation of 0.5 msec. There are 256 ray paths and 256 corresponding data points. We wish to determine the velocity structure in the two-dimensional plane between the two wells.

Discretizing the problem into a 16 by 16 grid of 100 meters by 100 meters blocks gives 256 model parameters. The $G$ matrix and data $d$ are provided. Use the following methods to derive a model

(1) Standard least squares

(2) Damped least squares – try different regularization weights to see what works the best

(3) first-order Tikhonov regularization, and

(3) second-order Tikhonov regularization

total number of model parameters = 256. (16x16 model).

data sets:

cross_well_g_matrix.txt

cross_well_d_vector.txt

Matrix G is 25X256 written as a column vector of length 256*256
Length of data vector is 256..

**NOTE:**

**You may want to use *pcolor* followed by *contour* for plotting the models**
**Use axis ij command to plot the vertical axis increasing downward**

Recall the objective function with Tokhonov (smoothing) term

$$F(\mathbf{m}) = (\mathbf{d} - \mathbf{Gm})^T(\mathbf{d} - \mathbf{Gm}) + \alpha^2(\mathbf{Dm})^T(\mathbf{Dm})$$

$$F(\mathbf{m}) = (\mathbf{d} - \mathbf{Gm})^T(\mathbf{d} - \mathbf{Gm}) + \alpha^2(\mathbf{m})^T\mathbf{Wm}$$

$\alpha$ **is the regularization** weight and **D** is the regularization matrix (smoothness operator); **D=I** for zeroth-order or damped least squares case.

Because this is a two-dimensional problem, you will need to implement a finite-difference approximation to first or second derivative in both directions in the smoothness matrix. The ***D*** matrix can be generated using the following MATLAB code:

## First derivative case:

```
m=16;n=16;

D=zeros(m*n,m*n);

for i=1:m*n

   for j=1:m*n

      if i==j

          D(i,j)=-2;

    end

      if j==i+1

          D(i,j)=1;

       end

      if j==i+n

          D(i,j)=1;

       end

   end

end
```

## Second Derivative Case

```
L=zeros(14*14,256);

k=1;

for i=2:15

for j=2:15

        M=zeros(16,16);

        M(i,j)=-4;

        M(i,j+1)=1;

        M(i,j-1)=1;

        M(i+1,j)=1;

        M(i-1,j)=1;

        L(k,:)=(reshape(M,256,1))';

        k=k+1;

end

end
```