Sensitivity analysis for micro-seismic events

John C. Bancroft

ABSTRACT

Increasing the accuracy of locating microseismic events is an ongoing objective to validate the location of insertion areas for CO_2 injection or well fracturing. Various methods are available for estimating the location using data collected from the surface or in a borehole. The accuracy of an estimated source location varies with the accuracy of the known geometry of the receivers, the velocity of the medium, and in measuring the arrival times of an event. Two papers have previously been presented for evaluating the sensitivity of locating a microseismic event: a Monte Carlo method that perturbed the geometry and a linear algebra method that used singular value decomposition (SVD).

These papers have been combined to be used as a tool to understand the linear algebra behind the SVD method.

INTRODUCTION

Mote Carlo method

The Apollonius method is an analytic solution that directly computes a microseismic source time and location using the first arrival times at four arbitrarily located receivers Bancroft & Du (2007) and Bancroft et al. (2009).

The traveltime equations for raypaths between a source at (x_0, y_0, z_0) and four arbitrarily located receivers at (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), and (x_4, y_4, z_4) are:

$$\begin{aligned}
(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2 &= v^2(t_1-t_0)^2\\
(x_2-x_0)^2+(y_2-y_0)^2+(z_2-z_0)^2 &= v^2(t_2-t_0)^2\\
(x_3-x_0)^2+(y_3-y_0)^2+(z_3-z_0)^2 &= v^2(t_3-t_0)^2\\
(x_4-x_0)^2+(y_4-y_0)^2+(z_4-z_0)^2 &= v^2(t_4-t_0)^2
\end{aligned}. \qquad (1)$$

We assume that the receivers used in the analytic solution are part of a larger grid system, such as many receivers on the surface or in a well.

The above estimate will produce results that match the machine accuracy when there is no error in the estimate of the arrival times or the locations of the receivers. Assuming that the velocity and location of the receivers are known exactly, we assume there is a noise error in the estimated arrival times (jitter) at the receivers. We then estimate the error in the source location for different levels of jitter.

The first example in Figure 1 shows the Apollonius solution for four receivers arbitrarily located near the surface and the source located at a depth that matches the spread of the receivers. A source clock-time was chosen and the clock-times at the receivers calculated. Gausian random jitter was added to the clock-time of the receivers, then using only the receiver locations and their clock-times, a source clock-time and its location were estimated. This procedure was repeated one-hundred times with a jitter

that had a standard deviation (SD) of 0.1 ms, then the mean and SD of the source location estimated. In Figure 1, the receivers are a green "x", the source location a blue "+", and the red circles are the 100 estimated locations.

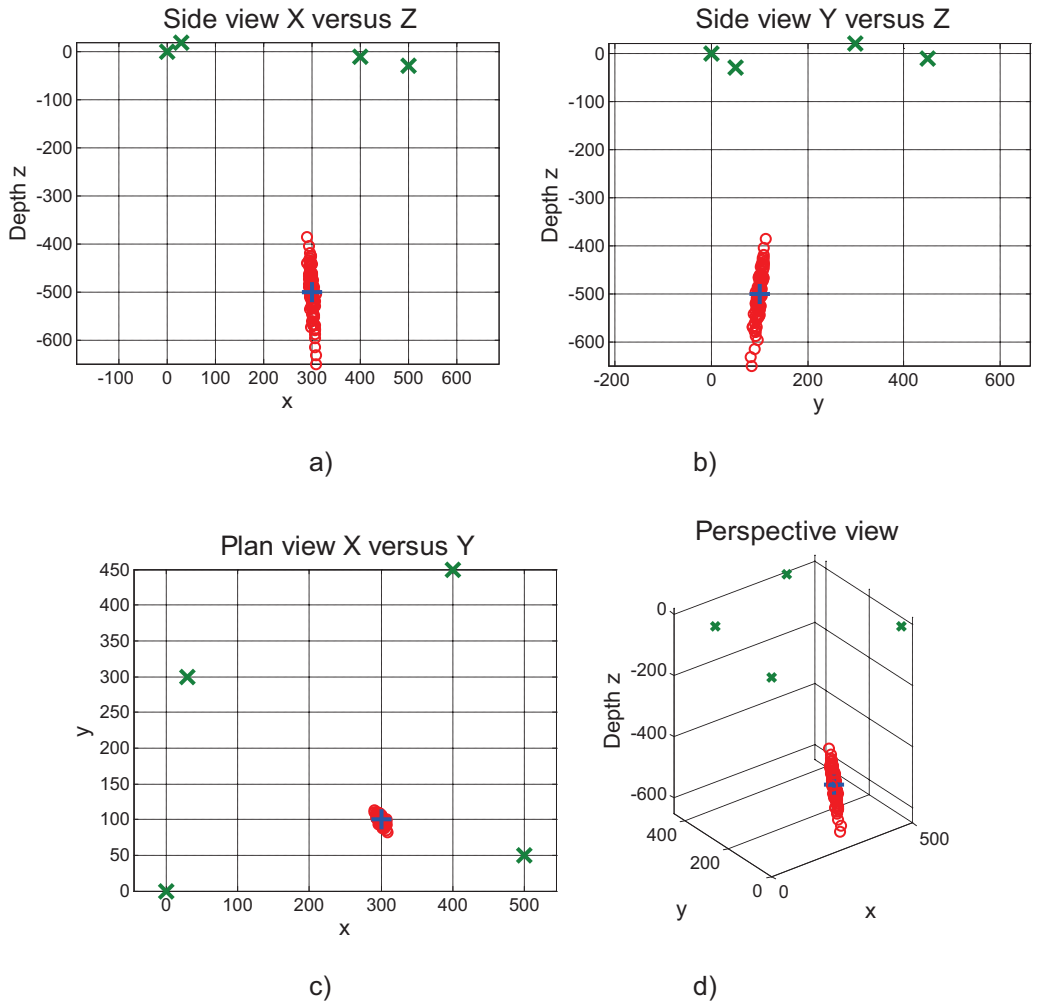


FIG. 1 Four views of an Apollonius solution with four receivers "x" near the surface, the known source "+" and the estimated source locations "o". The standard deviation of the noise was 1 ms.

The above methods required many estimates to establish an estimate of the source location. The following method uses linear algebra to accomplish the same task.

Linear algebra method using SVD

The sensitivity of locating a microseismic event can be computed using linear algebra. The covariance matrix and singular value decomposition (SVD), are used to produce an analytic solution, that is a more sophisticated than the Mote Carlo approach. The results in Figure 2 are a direct computation using this method.

This is the method used as motivation for developing the notes on linear algebra.

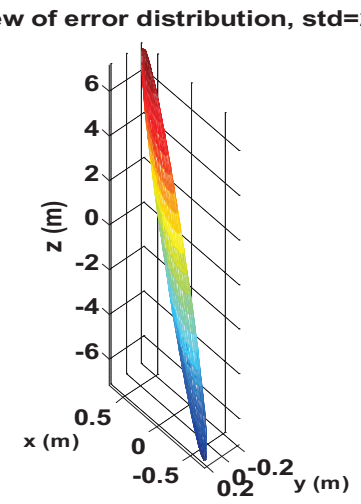


FIG. 2 Ellipsoid of the estimated source location using the SVD method.

REVISITING LINEAR ALGEBRA

Consider the equation of a plane in 3D space of x, y, and z,

$$ax + by + cz = d, \tag{2}$$

where a, b, c, and d are the parameters that define the plane. The values of d/a, d/b, and d/c define the intersection of the plane on the x, y, and z axis. A plane may also be defined using a vector that is normal to the plane and a point on the line that intersects the plane.

One plane is shown in Figure 3a. Two intersecting planes produce a common line as illustrated in (b). A third plane will intersect that line to give one point as illustrated in (c). Note that the planes cannot be parallel if they are to intersect.

Three equations that define three, non-parallel planes are given below in the system of equations (3). When each equation is considered separately, any two variables of x, y, or z can be used to compute the third variable on the plane. However, as a group of three equations we can solve for single values that define the one common point (x, y, z), i.e.,

$$\begin{aligned} a_1x + b_1y + c_1z &= d_1 \\ a_2x + b_2y + c_2z &= d_2 \\ a_3x + b_3y + c_3z &= d_3 \end{aligned}. \tag{3}$$

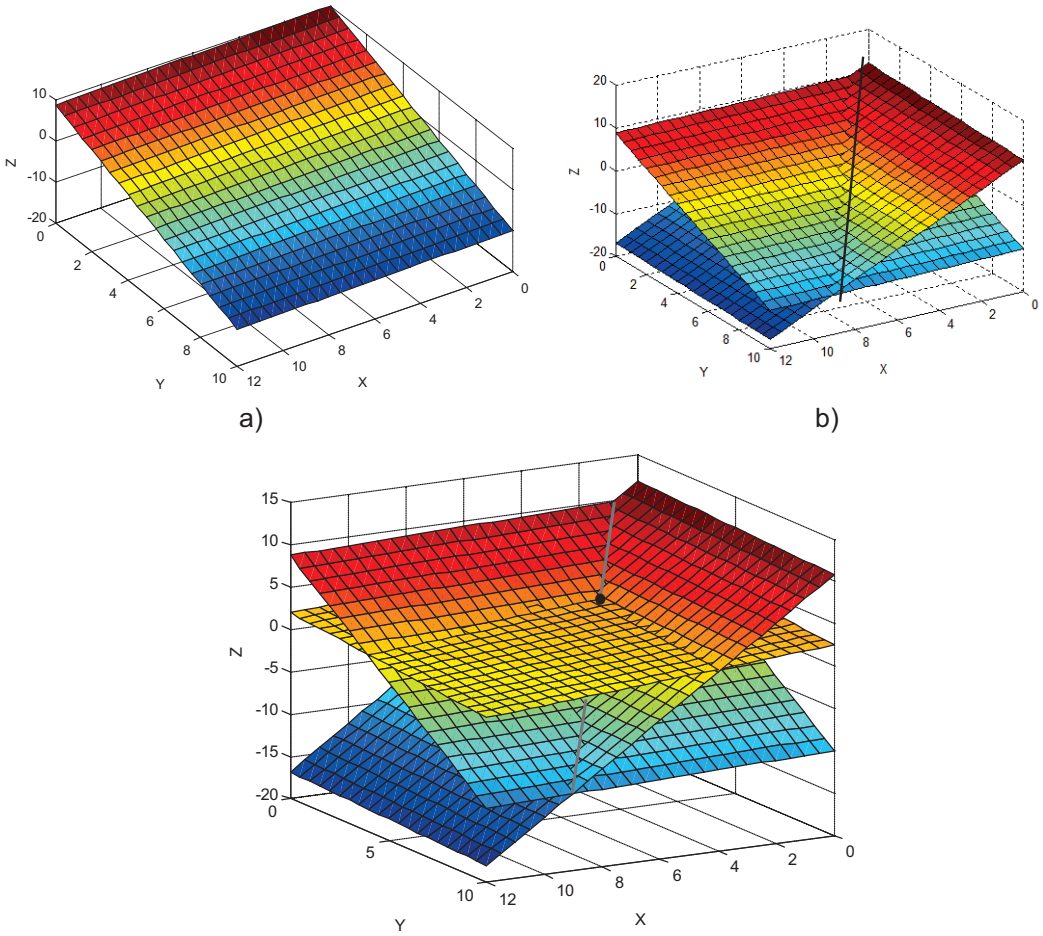


FIG. 3 A 3D view of a) one plane, b) the intersection of two planes to give a line, and c) the intersection of three planes to give one point.

We can solve this system of equations using a number of techniques such as:

- Direct inversion,
- Gausian elimination, or
- Iterative methods.

Direct inversion

The equations in (3) may be written in linear algebra form

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad \text{or} \quad \mathbf{Gx} = \mathbf{d}\,, \tag{4}$$

where

$$\mathbf{G} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \text{ and } \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}. \tag{5}$$

We assume the $\mathbf{G}$ matrix is invertable, and if it is, then we solve for $\mathbf{x}$ using

$$\mathbf{x} = \mathbf{G}^{-1}\mathbf{d} \,. \tag{6}$$

We should discuss things like the rank of $\mathbf{G}$, but if the three equations are linearly independent, then $\mathbf{G}$ will be invertable.

A great thing about linear algebra is that the number of equations and unknown are not limited and can have any reasonable number as long as the number of equations match's the number of unknowns.

Advanced programming languages such as MATLAB take advantage of these simple forms of equations to simplify programming.

The inversion of matrices may be computer intensive and alternative methods may be used to find solutions for $\mathbf{x}$, such as the following iterative methods.

Gauss elimination

We may recall our high school or early university math courses in which we used scaling and subtraction to eliminate x, leaving two equations with y and z. We then eliminate y from these two equations to get one equation in z; the defined the value of z. Working backwards, we then solve to y and finally for x. We accomplish this process by dividing each equation in (3) by the a_i coefficient giving new values for the remaining coefficients, i.e.,

$$\begin{aligned} x + \tilde{b}_1 y + \tilde{c}_1 z &= \tilde{d}_1 \\ x + \tilde{b}_2 y + \tilde{c}_2 z &= \tilde{d}_2 \\ x + \tilde{b}_3 y + \tilde{c}_3 z &= \tilde{d}_3 \end{aligned} \,. \tag{7}$$

Subtracting the second equation from the first, and then the third from the second gives two equations, which can be rescaled to eliminate the b_i coefficients giving

$$\begin{aligned} y + \hat{\tilde{c}}_1 z &= \hat{\tilde{d}}_1 \\ y + \hat{\tilde{c}}_2 z &= \hat{\tilde{d}}_2 \end{aligned} \,. \tag{8}$$

The variable y can now eliminate by another subtraction to give the value for z. Using one equation from (8) we can find the value of y, and from (7) we can get the value of x.

An important point to remember with the exact solution where we have the same number of equations as unknowns is that we can multiply any equation with a scalar. We will see later that we will have to be very careful when scaling equations when using the least squares method.

Iterative method: Jacobi

The Jacobi method separates the **G** matrix into three matrices; the diagonal **D**, upper **U**, and lower **L**, as

$$\begin{bmatrix}\mathbf{G}\end{bmatrix}=\begin{bmatrix}a_1 & b_1 & c_1\\ a_2 & b_2 & c_2\\ a_3 & b_3 & c_3\end{bmatrix}=\begin{bmatrix}a_1 & 0 & 0\\ 0 & b_2 & 0\\ 0 & 0 & c_3\end{bmatrix}+\begin{bmatrix}0 & b_1 & c_1\\ 0 & 0 & c_2\\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\ a_2 & 0 & 0\\ a_3 & b_3 & 0\end{bmatrix}. \qquad (9)$$

$$\begin{bmatrix}\mathbf{G}\end{bmatrix} = \begin{bmatrix}\mathbf{D}\end{bmatrix} + \begin{bmatrix}\mathbf{U}\end{bmatrix} + \begin{bmatrix}\mathbf{L}\end{bmatrix}$$

Our problem them becomes

$$\left(\mathbf{D}+\mathbf{U}+\mathbf{L}\right)\mathbf{x}=\mathbf{d}\,, \qquad (10)$$

or

$$\mathbf{D}\mathbf{x}=\mathbf{d}-\left(\mathbf{U}+\mathbf{L}\right)\mathbf{x}\,. \qquad (11)$$

If we assume the **x** in the right side of equation (11) is an estimated value $\mathbf{x}_{i\text{-}1}$, then we can compute a new value for $\mathbf{x}_i$ on the left side from

$$\mathbf{x}_i=\mathbf{D}^{-1}\left[\mathbf{d}-\left(\mathbf{U}+\mathbf{L}\right)\mathbf{x}_{i-1}\right], \qquad (12)$$

which is now in an iterative form.

For stability, a diagonal element in **D** must be greater than the sum of the remaining elements in the associated row or column. We can improve the stability by adding values to the diagonal. (We may be familiar with adding prewhitening in deconvolution.) This is accomplished with a scale **I** matrix $\lambda\mathbf{I}$ giving

$$\mathbf{x}_{i+1}=\left(\mathbf{D}+\gamma\mathbf{I}\right)^{-1}\left[\mathbf{d}-\left(\mathbf{U}+\mathbf{L}\right)\mathbf{x}_i\right] \qquad (13)$$

The diagonal matrix $\gamma\mathbf{I}$ can be modified to contain values that vary, depending on the data that remain in a particular row or column.

The Jacobi method uses all the previous values if $x_{i\text{-}1}$ to update all the new values x_i. It is possible to use the newly computed values of x_i to compute the remaining values of x_i, that is known as the Gauss-Seidel method.

Iterative method: Gauss-Seidel

Consider equation (7) that is rewritten as an iterative process in the element form

$$\begin{aligned} x_i &= \left(d_1-b_1y_{i-1}-c_1z_{i-1}\right)\\ y_i &= \left(d_2-a_2x_i-c_2z_{i-1}\right)\\ z_i &= \left(d_3-a_3x_i-b_3y_i\right) \end{aligned}\,. \qquad (14)$$

The element x_i is computed using the previous values of $y_{i\text{-}1}$ and $z_{i\text{-}1}$. The next line of the equation now uses the new value of x_i to assist in estimating y_i. The third line uses both of the newly estimated values x_i and y_i to estimate z_i.

Rewriting equation (14) in the form of

$$\begin{aligned} a_1 x_i &= \left(d_1 \quad - b_1 y_{i-1} - c_1 z_{i-1}\right) \\ a_2 x_i + b_2 y_i &= \left(d_2 \quad - c_2 z_{i-1}\right), \\ a_3 x_i + b_3 y_i + c_3 z_i &= \left(d_3 \quad\right) \end{aligned} \tag{15}$$

we can visualizes the matrix form

$$\left(\mathbf{D}+\mathbf{L}\right)\mathbf{x}_{i+1} = \mathbf{d} - \mathbf{U}\mathbf{x}_i, \tag{16}$$

or in the Gauss-Seidel form as

$$\mathbf{x}_{i+1} = \left(\mathbf{D}+\mathbf{L}\right)^{-1}\left(\mathbf{d} - \mathbf{U}\mathbf{x}_l\right). \tag{17}$$

Now the lower matrix **L** is part of the inverted matrix and improves the rate of convergence over the Jacobi method. Stability becomes an issue, and the latter values of xi may converge faster than the initial values. Reordering or randomizing the elements of **x** may be of value.

Another problem with these iterative methods is that the higher frequencies converge faster than the lower frequencies and may produce a low frequency instability. This problem may be addressed by a Multigrid method.

Comment

The main reason for me repeating this basic information is to emphasize that we scale and add or subtracts the initial equation to get our solution. I can multiply the first equation by 10,000 and it will not affect the solution. We simply scale the other equations accordingly and keep iterating towards the solution. We will now take a big leap into least squares solutions where scaling becomes very important. Usually there are a large number of equations, samples, or observations, more than our unknowns and we cannot arbitrarily scale an equation without affecting the solution.

If all samples or observations were correct, then we could choose the number of equations that equals the unknowns. But our equations are not exact and we now assume there is some error or noise, and that there may be some dependence between them.

A SIMPLE LEAST SQUARES PROBLEM

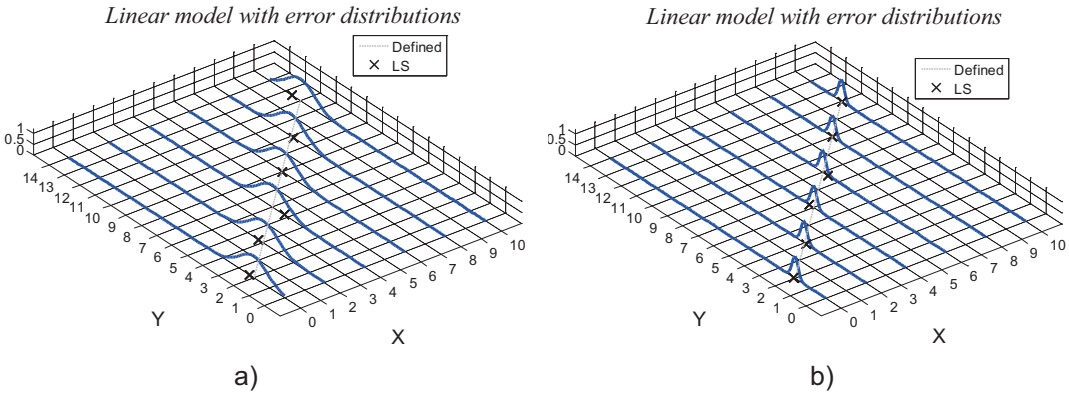
Our model consists of a 2D linear event, taken from a process, where the horizontal component x is defined, and we take readings or make observations of the vertical component y. The equation of the noise free measurements is the conventional equation of a straight line

$$y = mx + c, \tag{18}$$

where m is the slope of the line and c a constant that is the intersection point on the y axis (where $x = 0$). The components m and c are defined by the process but are not known to us, and it is our objective to estimate them.

The actual measurements are contaminated with noise, which can be a result of measurement error or a limitation of the equipment. Assume experience has told us (prior knowledge) that the errors of our measurements or noise have a Gausian distribution, and that we know the standard deviation σ_{y_i}. This error distribution could be defined by the measuring apparatus, or estimated for example by having 100 people measure a temperature at exactly the same time (x_5).

Figure 4 shows a plot of our model in which we take measurements at specific values of x at 0, 2, 4, 6, 8, and 10, or x_i for i = 1 to I =6. The true linear process is defined by the gray line. Blue lines at each x location define the probability in the z direction of the estimated locations of the y values. Two standard deviations of 0.9 and 0.3 are illustrated. Black x's are examples of observed values of y, which do fall in the range illustrated by the distributions. The following examples may have different noise values that may vary with each figure.



a) b)

FIG. 4 Linear model (gray) showing the SD's of a) 0.9 and b) 0.3.

We will initially assume that the distributions will have the same SD's for each set of measurements. We will then introduce a set of measurements where the SD may vary.

USING THE ESTIMATED DATA

Our set of measurements contain six observed values for y_i, with each observation containing a random error e_i controlled by the SD. Figure 5 contains a 2D plot of the data, where, once again, the defined line is gray and the observations are black x's.

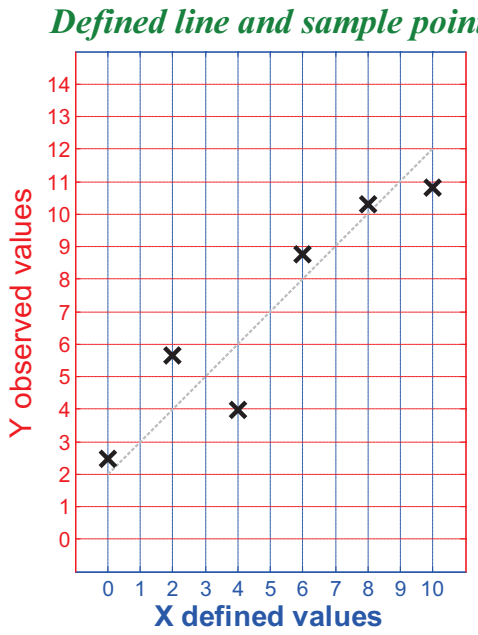


FIG. 5 Display of six observations y_i at six values of x_i.

If we assume estimated values for the slope as $\hat{m}$ and constant $\hat{c}$, then these six observations may be written as linear equations

$$\begin{aligned} y_1 &= \hat{m}x_1 + \hat{c} + e_1 \\ y_2 &= \hat{m}x_2 + \hat{c} + e_2 \\ y_3 &= \hat{m}x_3 + \hat{c} + e_3 \\ &\cdots \\ y_6 &= \hat{m}x_6 + \hat{c} + e_6 \end{aligned} \quad . \tag{19}$$

We write these equations in matrix form

$$\mathbf{Gp} = \mathbf{y} \quad , \tag{20}$$

where **G** is a matrix that relates the geometry of the equations, **p** a vector that containing the unknown parameters m and c, and **y** the observed values of y_i, i.e.,

$$\mathbf{G} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ x_5 & 1 \\ x_6 & 1 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} \hat{m} \\ \hat{c} \end{bmatrix}, \quad \text{and } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix}. \tag{21}$$

We are now in the realm of least squares (LS) solutions as we have more observations than unknowns, and the normal LS equation is

$$\mathbf{G}^{\mathrm{T}}\mathbf{G}\mathbf{p} = \mathbf{G}^{\mathrm{T}}\mathbf{y}, \tag{22}$$

and the standard solution **p** for *m* and *c* becomes

$$\mathbf{p} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{G}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{y}. \tag{23}$$

The result of the LS solution is shown as the red line in Figure 6. Notice that at this point there is no mention of the observation noise (σ_{y_i}) terms in the LS solution.

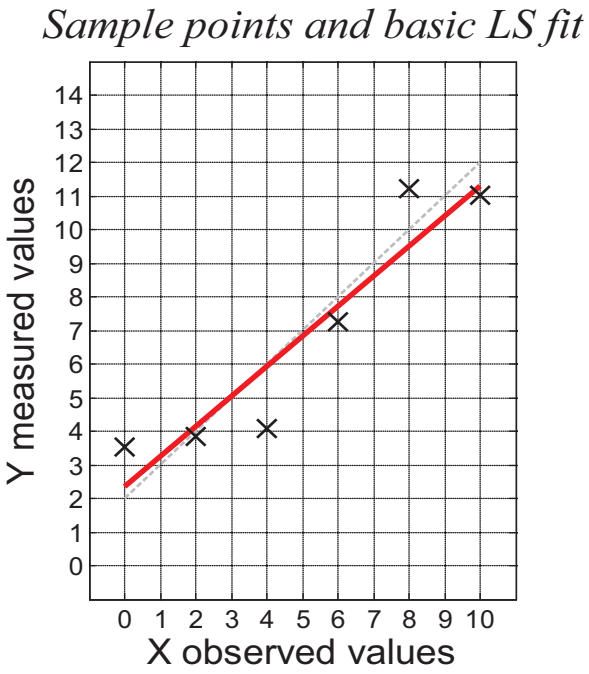


FIG. 6 Least squares solution to six observations.

We can assume any value for the error terms say a σ of 0.1 or 10 in equation (23) and we would still get the same answer. That is because our LS solution had assumed the noise distributions for each observation were all equal, as our original model had assumed.

We will now consider the case when the error distribution can be different for each observation. Figure 7 shows five observations with a σ_{y_i} of 0.9, while the fifth observation has a σ_{y_5} of 0.09. The distributions are now plotted at the observed point, as that is all we know. The LS solution is indifferent to the accuracy of the 5th observation as it treats all observations equally, as evident by the LS solution.

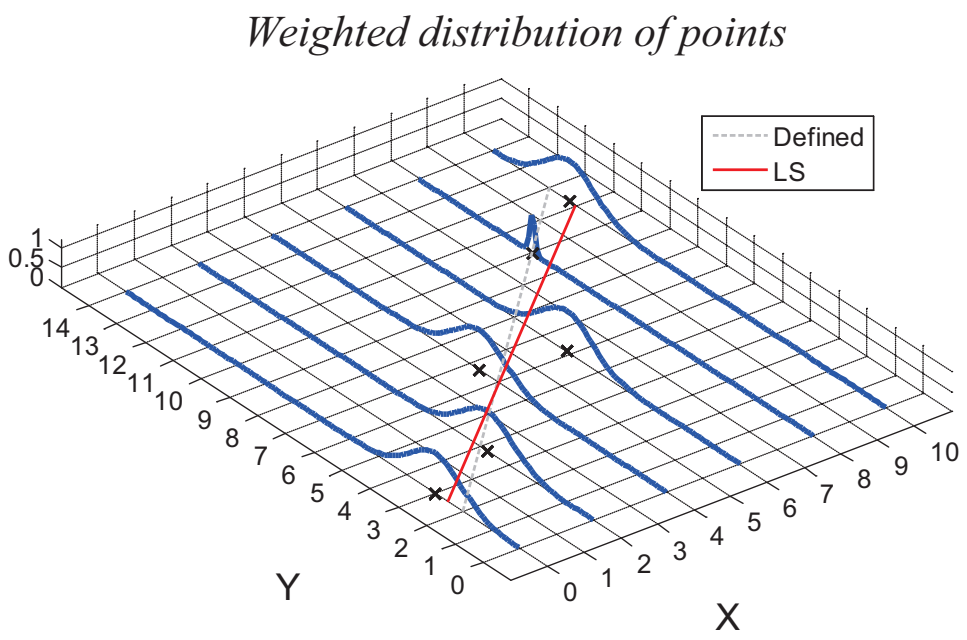


FIG. 7 Data example when the 5th observation is more accurate than the remaining observations. The red line indicates the LS solution is indifferent to the more accurate observation.

SCALING THE EQUATIONS AND THE COVARIANCE MATRIX

We are aware of a more accurate observation, so let us try an experiment where we scale an equation to improve its weight when estimating a solution. If we now assume the fifth equation is more accurate, what would happen if we arbitrarily scaled the equation by a large value L,

$$\begin{aligned} y_1 &= \hat{m}x_1 + \hat{c} + e_1 \\ y_2 &= \hat{m}x_2 + \hat{c} + e_2 \\ &\ldots \\ Ly_5 &= \hat{m}Lx_5 + L\hat{c} + Le_5 \\ y_6 &= \hat{m}x_6 + \hat{c} + e_6 \end{aligned} \qquad (24)$$

We may be tempted to try to find a new point $\left(\ddot{x}_5, \ddot{y}_5\right)$ to replace the original value $\left(x_5, y_5\right)$ by some sort of scaling, but that does not work. The only satisfactory hint at an explanation for the weight is to assume equation 5 is repeated L times,

$$
\begin{aligned}
y_1 &= \hat{m}x_1 + \hat{c} + e_1 \\
y_2 &= \hat{m}x_2 + \hat{c} + e_2 \\
y_3 &= \hat{m}x_3 + \hat{c} + e_2 \\
y_4 &= \hat{m}x_4 + \hat{c} + e_4 \\
y_5 &= \hat{m}x_5 + \hat{c} + e_5 \\
y_5 &= \hat{m}x_5 + \hat{c} + e_5 \\
y_5 &= \hat{m}x_5 + \hat{c} + e_5 \\
&\ldots \\
y_5 &= \hat{m}x_5 + \hat{c} + e_5 \\
y_6 &= \hat{m}x_6 + \hat{c} + e_6
\end{aligned}
\quad . \tag{25}
$$

If we repeated y_5 L times, the new **G** matrix will become very long, and it transpose $\mathbf{G}^{\mathbf{T}}$ very wide, i.e.,

$$
\mathbf{G}^{\mathbf{T}} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_5 & \ldots & x_5 & x_5 & x_6 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \text{ and } \mathbf{G} = \begin{bmatrix} x_1 & 1 \\ x_1 & 1 \\ x_1 & 1 \\ x_1 & 1 \\ x_5 & 1 \\ x_5 & 1 \\ \ldots & 1 \\ x_5 & 1 \\ x_5 & 1 \\ x_1 & 1 \end{bmatrix}. \tag{26}
$$

The $\mathbf{G}^{\mathbf{T}}\mathbf{G}$ matrix becomes a small 2x2 matrix

$$
\mathbf{G}^{\mathbf{T}}\mathbf{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{27}
$$

The coefficients of the $\mathbf{G}^{\mathbf{T}}\mathbf{G}$ matrix will be biased by the repeated values and the left side of the normal equation becomes

$$
\mathbf{G}^{\mathbf{T}}\mathbf{G}\mathbf{p} \tag{28}
$$

.

The right side of the normal equation becomes

$$\mathbf{G}^{\mathrm{T}}\mathbf{y} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_5 & \dots & x_5 & x_5 & x_6 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_5 \\ \dots \\ y_5 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}. \qquad (29)$$

where the elements e and f will be significantly modified by the number of repeated elements. However that is not the case as we should use something like a weight vector $\mathbf{w} = [1, 1, 1, 1, L, 1]'$ to scale the amplitudes, i.e.,

$$\mathbf{wGp} = \mathbf{wy} \qquad (30)$$

which becomes

$$\mathbf{wGp} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ Lx_5 & L \\ x_6 & 1 \end{bmatrix} \text{ and } \mathbf{wy} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ Ly_5 \\ y_6 \end{bmatrix}. \qquad (31)$$

Scaling within the LS solution is different from scaling within the exact solution where scaling has no effect on the solution. Now scaling does have a significant effect and allows us to improve our solution by using information or prior knowledge about the process in the way the observations we collected.

Equations (30) and (31) assumed some form of a dot multiplication with the vector **w**. The linear algebra approach is to multiply each side of the forward equation by a square matrix **W** that contains the inverse of σ_i or $1/\sigma_i$ on the diagonal. This matrix contains zeros, except on the diagonal where the elements are the inverse of the SD σ_i or σ_{ii}. This matrix will scale each equation by the inverse of the SD, and has the effect of making the error distribution to be the same for each equation, which is ideal for the LS method.

$$\mathbf{W} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sigma_5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\sigma_6} \end{bmatrix}. \tag{32}$$

The normal equation for the LS solution becomes

$$\left(\mathbf{WG}\right)^{\mathbf{T}} \mathbf{WGp} = \left(\mathbf{WG}\right)^{\mathbf{T}} \mathbf{Wy}, \tag{33}$$

or

$$\mathbf{G^T W^T WGp} = \mathbf{G^T W^T Wy}. \tag{34}$$

Since $\mathbf{W}$ and $\mathbf{W^T}$ are diagonal matrices, the product $\mathbf{W^T W}$ is also diagonal with the elements of $\mathbf{W}$ squared. The inverse of $\mathbf{W^T W}$ is also a diagonal matrix with elements of $\mathbf{W^T W}$ inverted. The diagonal elements of $\left(\mathbf{W^T W}\right)^{-1}$, which we will call $\mathbf{C_y}$, are the variances or square of the SD.

$$\mathbf{C_y} = \left(\mathbf{W^T W}\right)^{-1} = \begin{bmatrix} \sigma_{11}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{33}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{44}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{55}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{66}^2 \end{bmatrix}. \tag{35}$$

This $\mathbf{C_y}$ matrix may be referred to as the variance matrix, the covariance matrix, or even the variance-covariance matrix. I will use the term covariance matrix that will allow us to add non-diagonal terms that will include some form of a dependence between observations. Assume our observations require different people to make a measurement to eliminate some kind of bias, but somehow two measurements are made by the same person. There is a possibility that the error of those two measurements are related or biased and we would measure that as a covariance. If the same person took the 4th and 6th reading, the covariance matrix would become

$$\mathbf{C}_y = \left(\mathbf{W}^T\mathbf{W}\right)^{-1} = \begin{bmatrix} \sigma_{11}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{22}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{33}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{44}^2 & 0 & \sigma_{6-4} \\ 0 & 0 & 0 & 0 & \sigma_{55}^2 & 0 \\ 0 & 0 & 0 & \sigma_{6-4} & 0 & \sigma_{66}^2 \end{bmatrix}, \tag{36}$$

where the off diagonal elements σ_{6-4} and σ_{4-6} establish the covariance relationship. The normal equation can now be written with the covariance matrix as

$$\mathbf{G}^T\mathbf{C}^{-1}\mathbf{G}\mathbf{p} = \mathbf{G}^T\mathbf{C}^{-1}\mathbf{y}\,. \tag{37}$$

The covariance matrix is very important to us as we can now use statistical information about our measurement system to obtain even more accurate results.

The covariance matrix has assumed our noise is Gaussian. If the noise is not Gaussian then other weighting matrices can be used.

Let's return to out example where the 5th observation was made with a more accurate method that had a SD that was 10 times smaller than the other observations, (maybe a digital thermometer was used rather than a glass thermometer). Using this additional information in our least squares solution, we get the following result in Figure 8, where the weighted least squared solution (WLS) (cyan) is closer to the defined line than the original LS solution (red). Note that the WLS line passes very close to the accurate fifth observation.

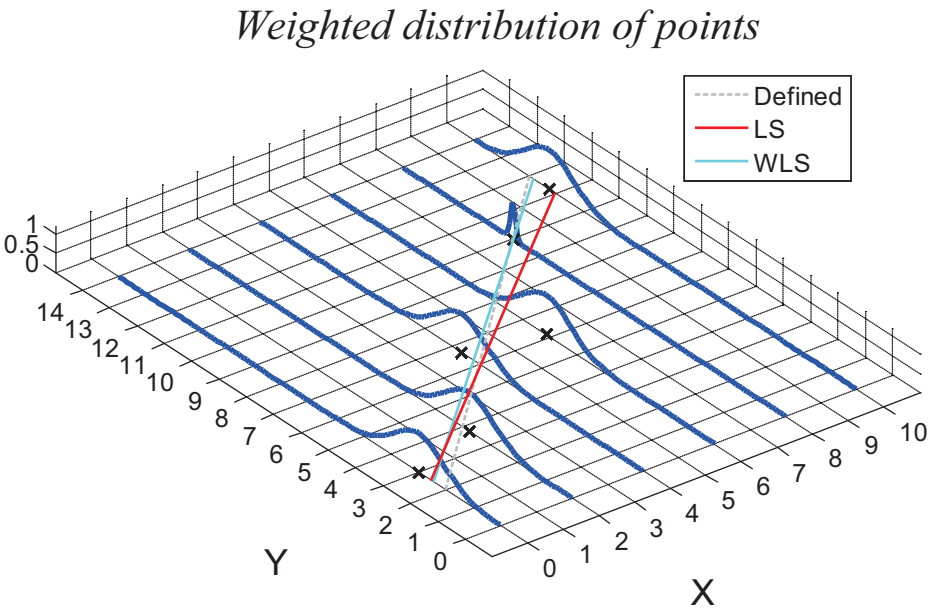


FIG. 8 The WLS solution (cyan) includes the weighting matrix and is a closer match to the true solution (gray line) than the LS solution (red).

Data from the previous example

The following data was produced from the run that produced Figure 7. The data defines the points and their SD's, the covariance matrix, and the solutions for the LS and WLS solutions. Note the improvement in *m* and *c* of the WLS solution.

```
Start least squares example.
*****************************.
  Point  X    Y   Sdv     Noise2  y2
  1      0    2   0.900    0.989   2.989
  2      2    4   0.900   -0.250   3.750
  3      4    6   0.900    0.631   6.631
  4      6    8   0.900   -1.847   6.153
  5      8   10   0.090   -0.032   9.968
  6     10   12   0.900   -0.741  11.259

Covariance matrix2
    1.2346         0         0         0         0         0
         0    1.2346         0         0         0         0
         0         0    1.2346         0         0         0
         0         0         0    1.2346         0         0
         0         0         0         0  123.4568         0
         0         0         0         0         0    1.2346

Least-squares solution of m and c for each method
      Defined   LS solution    WLS solution
 m =  1         0.8504         0.9382
 c =  2         2.5397         2.4422
```

SIMPLIFYING THE LS PROBLEM

We do have a complicated system of matrices and vectors

$$\mathbf{G}^T\mathbf{W}^T\mathbf{W}\mathbf{G}\mathbf{p} = \mathbf{G}^T\mathbf{W}^T\mathbf{W}\mathbf{y}\,. \tag{38}$$

that can be simplified to

$$\mathbf{M}\mathbf{p} = \mathbf{z}\,. \tag{39}$$

where

$$\mathbf{M} = \mathbf{G}^T\mathbf{W}^T\mathbf{W}\mathbf{G}, \text{ and } \mathbf{z} = \mathbf{G}^T\mathbf{W}^T\mathbf{W}\mathbf{y}\,. \tag{40}$$

M is now a square matrix, with a side dimension equal to the parameter vector. It is called the Normal Matrix. The square matrix allows special operations such as finding eigenvalues and eigenvectors, along with singular value decomposition (SDV).

Before we proceed with our goal of estimating the sensitivity of micro seismic data, we need two more tools to work with. The first is eigenvalues and eigenvectors that are standard in most linear algebra courses. The second is singular value decomposition (SVD) that is possibly the most important development in linear algebra.

REVIEW OF EIGENVECTORS AND EIGENVALUES

Consider a matrix **M** that is multiplied by a vector **v** to get a new vector **p**.

$$\mathbf{M}\mathbf{v} = \mathbf{p} \,. \tag{41}$$

The matrix **M** maps the vector **v** to become a new vector **p**. In 3D space, a vector may be defined from the origin to a point. **M** then moves that point to somewhere else. **M** can also move all points to new locations by rotation or scaling.

Matrix scaling

Suppose we have data (x, t) that we want to stretch in the t direction, but leave the other axis the same. This may not be to exciting, but I did need this approach to scale the coordinates of a circle to many different ellipse for a dip moveout (DMO) algorithm. The points on a unit circle were defined by various angles θ, using $x = \cos\theta$ and $z = \sin\theta$. The 2D scaling matrix for one ellipse was

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \tag{42}$$

where the t parameters were all scaled by 2. Consider Figure 9 where three points on a unit circle (1, 0), (0.7, 0.7), and (1, 0) which are mapped to three points on an ellipse.

$$\mathbf{M}\mathbf{v} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0.7 & 0 \\ 0 & 0.7 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.7 & 0 \\ 0 & 1.4 & 2 \end{bmatrix} = \mathbf{p} \,. \tag{43}$$

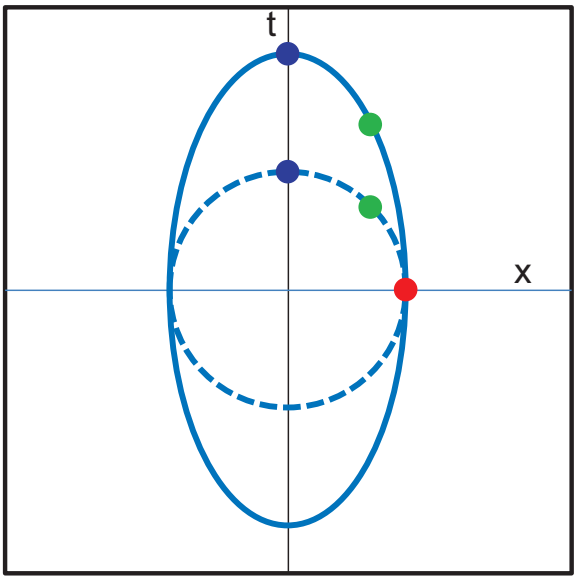


FIG. 9 Three points on a circle are mapped to three points on an ellipse.

Rotational matrix

We might remember a rotational matrix

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{44}$$

where a point can be mapped into a new value on a rotated axis, rotated counterclockwise by angle an θ.

An example of rotation is displayed in Figure 10 where an ellipse is rotated 60 degrees counter clockwise. The black lines map the movement of points from the original blue ellipse to the red ellipse.

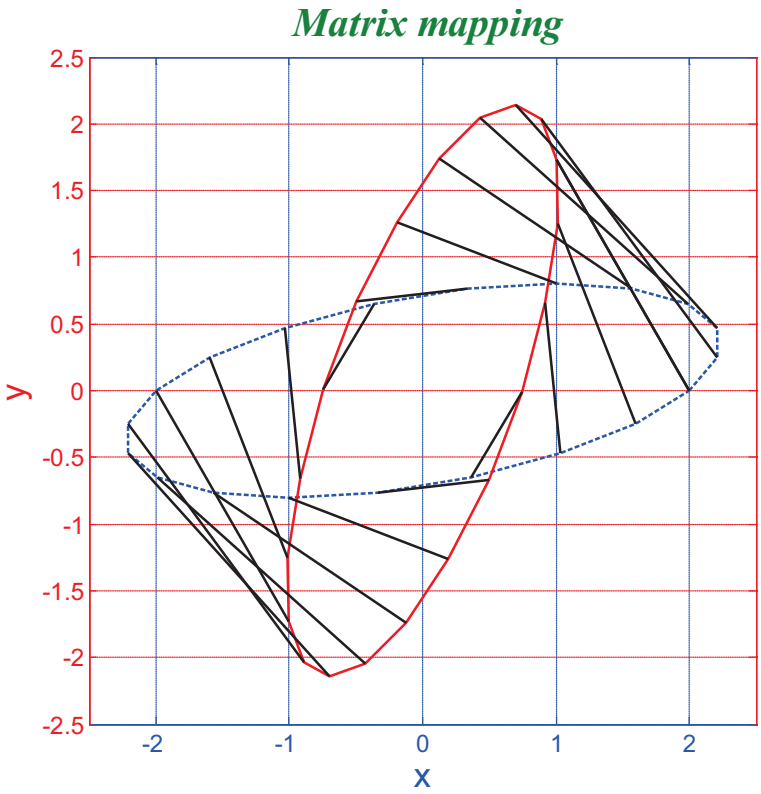


FIG. 10 Rotation of a blue ellipse counter clock-wise 60 degrees to the red ellipse.

Scaling with shear is illustrated in Figure 11 that used the matrix

$$\mathbf{M} = \begin{bmatrix} 2 & 1 \\ 0 & 0.5 \end{bmatrix}. \tag{45}$$

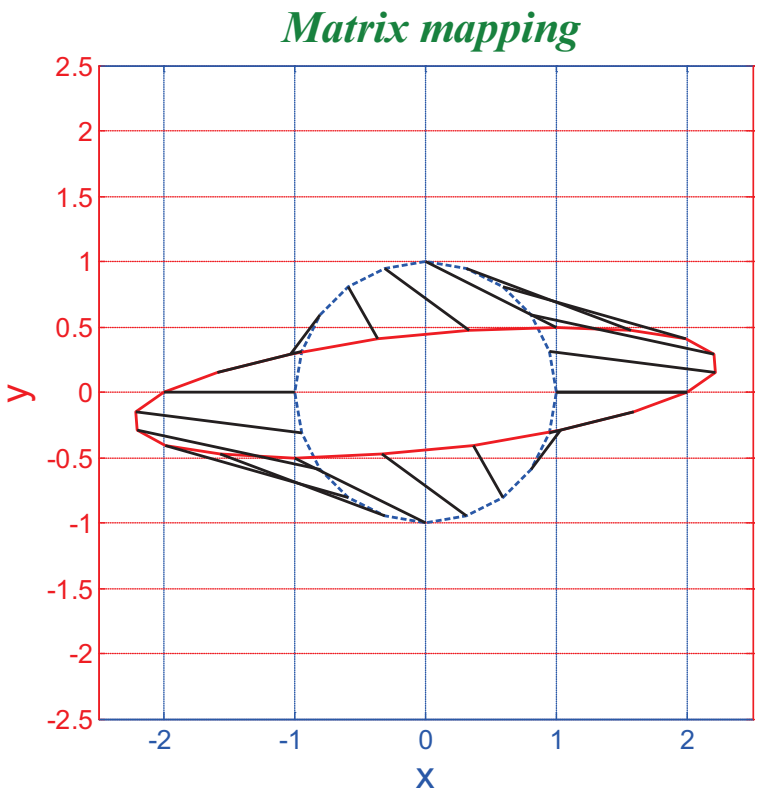
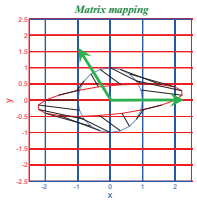


FIG. 11 Matrix scaling with shear.

Figure 11 has some nice properties. Consider two lines that scale the data by extending points away from the origin as indicated in the icon below.



These green vectors are very special vectors and are unique for M. The operation of **M** on the green vectors **Mv** is the same as scaling the green vectors λ**v**. These vectors **v** are called eigenvectors. The change in magnitude of the eigenvectors λ is called the eigenvalue. These vectors will have even more importance when we consider the least squares solution that has an **M** matrix that is symmetric, derived from $\mathbf{G}^T\mathbf{G}$. In this case, the eigenvectors will be orthogonal, and have real eigenvalues.

Eigenvectors and eigenvalues

We can write equations for the eigenvectors **x**, and the eigenvalues λ, as

$$\mathbf{Mx} = \lambda\mathbf{x}, \tag{46}$$

where we see that the matrix operation only scales this special vector by λ.

We find **x** and λ by first finding the eigenvalues λ using.

$$\mathbf{M}\mathbf{x} - \lambda\mathbf{x} = \left(\mathbf{M} - \lambda\mathbf{I}\right)\mathbf{x} = 0\,. \tag{47}$$

where **I** is the identity matrix. Now consider

$$\left(\mathbf{M} - \lambda\mathbf{I}\right)^{-1}\left(\mathbf{M} - \lambda\mathbf{I}\right)\mathbf{x} = \mathbf{x} = 0\,. \tag{48}$$

The only case where this is possible is when the matrix $\mathbf{M} - \lambda\mathbf{I}$ is not invertible, and that means the determinant is zero,

$$\det\left(\mathbf{M} - \lambda\mathbf{I}\right) = 0\,. \tag{49}$$

For a 2D system,

$$\det\left(\mathbf{M} - \lambda\mathbf{I}\right) = \begin{bmatrix} m_{11} - \lambda & m_{21} \\ m_{21} & m_{22} - \lambda \end{bmatrix} = \left(m_{11} - \lambda\right)\left(m_{22} - \lambda\right) - m_{21}m_{21} = 0\,, \tag{50}$$

giving the quadratic equation

$$a\lambda^2 + b\lambda + c = 0\,, \tag{51}$$

that can be solved for one or two values of λ, that can be real and complex.

The eigenvectors can be found using the eigenvalues equation

$$\begin{bmatrix} m_{11} & m_{21} \\ m_{21} & m_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda\begin{bmatrix} x \\ y \end{bmatrix}. \tag{52}$$

or

$$\begin{aligned} \left(m_{11} - \lambda\right)x + m_{12}y &= 0 \\ \left(m_{21} - \lambda\right)x + m_{22}y &= 0 \end{aligned}. \tag{53}$$

Notice that in both these equations, the eigenvectors pass through the origin, or

$$y = mx\,. \tag{54}$$

We have shown the solutions for a 2D case: for a 3D system, the characteristic equation will be of third order, and an *n*D system will have an n^{th} order characteristic equation. There are exact solutions up to a 5^{th} order equation, but then iterative methods are required to solve the higher order equations for the eigenvalues.

Some properties of eigenvalues and eigenvectors:

1. $\det\left(\mathbf{M}\right) = \prod\lambda_i = \lambda_1\lambda_2\lambda_3...\lambda_n$.
2. if **M** is symmetric or Hermitian, $\left(\mathbf{M} = \mathbf{M}^{\mathbf{T}}\ \text{or}\ m_{ij} = m_{ji}^{*}\right)$, a result of **M** = $\mathbf{G}^{\mathbf{T}}\mathbf{G}$, all eigenvalues are real.

3. the eigenvector of a symmetric matrix are orthogonal, (only for distinct eigenvalues).
4. if **M** is diagonal, upper triangular or lower triangular, the Eigenvalues are the diagonal entries of **M.**
5. if **M** in Unitary, $\left(\mathbf{M}^{-1} = \mathbf{M}^{T}\right)$, the magnitude of all eigenvalues equal 1, $|\lambda| = 1$
6. if **M** is a rotational matrix, the eigenvalues will be complex.
7. if **M** can be written as $\mathbf{M} = \mathbf{Q}\Lambda\mathbf{Q}^{-1}$ were **Q** contains the eigenvectors, then Λ is a diagonal matrix with the eigenvalues on the diagonal (more in this later).

The following examples ion Figure 12 are copied from the website below and illustrate various matrix operations on vectors in 2D space.

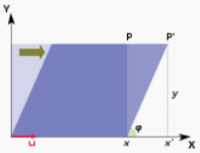
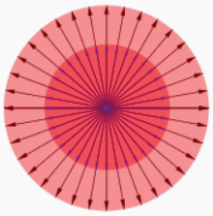
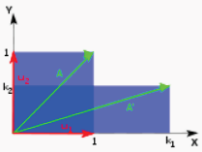
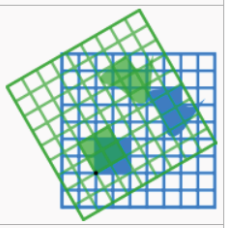
	horizontal shear	scaling	unequal scaling	counterclockwise rotation by ϕ
illustration				
matrix	$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}$	$\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$	$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$
characteristic equation	$\lambda^2 - 2\lambda + 1 = (1-\lambda)^2 = 0$	$\lambda^2 - 2\lambda k + k^2 = (\lambda - k)^2 = 0$	$(\lambda - k_1)(\lambda - k_2) = 0$	$\lambda^2 - 2\lambda\cos\varphi + 1 = 0$
eigenvalues λ_i	$\lambda_1 = 1$	$\lambda_1 = k$	$\lambda_1 = k_1,\ \lambda_2 = k_2$	$\lambda_{1,2} = \cos\varphi \pm i\sin\varphi = e^{\pm i\varphi}$
algebraic and geometric multiplicities	$n_1 = 2,\ m_1 = 1$	$n_1 = 2,\ m_1 = 2$	$n_1 = m_1 = 1,\ n_2 = m_2 = 1$	$n_1 = m_1 = 1,\ n_2 = m_2 = 1$
eigenvectors	$\mathbf{u}_1 = (1,0)$	$\mathbf{u}_1 = (1,0), \mathbf{u}_2 = (0,1)$	$\mathbf{u}_1 = (1,0), \mathbf{u}_2 = (0,1)$	$\mathbf{u}_1 = \begin{bmatrix} 1 \\ -i \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} 1 \\ i \end{bmatrix}.$

FIG. 12 Examples from: http://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

I have described the use of eigenvalues and eigenvectors using simple vectors. The dimensions of M can be very large, and their use extends well beyond this simplifying explanation as expressed in the following quote:

"…the concept of *direction* loses its ordinary meaning, …These ideas often are extended to more general situations, where scalars are elements of any field, vectors are elements of any vector space, and linear transformations may or may not be represented by matrix multiplication. For example, instead of real numbers, scalars may be complex numbers; instead of arrows, vectors may be functions or frequencies; instead of matrix multiplication, linear transformations may be operators such as the derivative from calculus. These are only a few of countless examples where eigenvectors and eigenvalues are important."

Taken from: http://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

INTRODUCING THE SVD METHOD

The singular value decomposition (SVD) method was used to estimate 3D hypocenter location and origin time of a microseismic event. Given the variance of observed first-break arrival time, a 3D error distribution of the hypocenter location is calculated. It is shown that uncertainty in the vertical direction is much larger than the in horizontal directions.

First-break arrival time can be used to estimate location of a microseismic event and its origin time. Figure 13 shows the geometry of ray paths from source to surface receivers, constant RMS velocity is assumed for simplification.

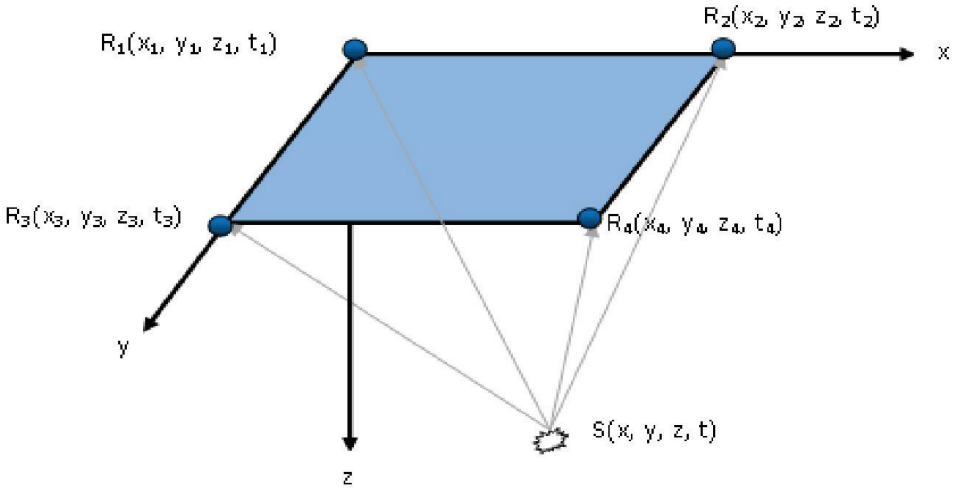


FIG. 13 Illustration of ray paths from source to receivers.

$S(x, y, z, t)$ is a microseismic event with origin clock-time t and location (x, y, z). $R_i(x_i, y_i, z_i, t_i)$ is a surface receiver at (x_i, y_i, z_i) with observed first-break arrival time t_i. A constant RMS velocity is assumed.

Given the first-break arrival times $t_1, t_2, t_3, t_4, \ldots, t_m$ at receivers $R_1, R_2, R_3, R_4, \ldots, R_m$, the location and origin time of a microseismic event can be estimated (Bancroft and Du, 2006). Suppose that all receivers start to record data at arbitrary clock-time t_0, receiver R_i (x_i, y_i, z_i) recorded first-break arrival time at t_i. The travel time from hypocenter to receivers satisfy

$$v^2 (t_i - t_0)^2 = (x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 \tag{55}$$

In order to change above quadratic equations into a linear parameter estimation problem, we apply the above equation in receiver R_{i-1},

$$v^2 (t_{i-1} - t_0)^2 = (x_{i-1} - x)^2 + (y_{i-1} - y)^2 + (z_{i-1} - z)^2 \tag{56}$$

Subtract equations with $i = 2$ from i=1 to get:

$$\begin{aligned} &2\left(x_i - x_{i-1}\right)x + 2\left(y_i - y_{i-1}\right)x + 2\left(z_i - z_{i-1}\right)z - 2v^2\left(t_i - t_{i-1}\right)t_0 \\ &= v^2(t_i^{\,2} - t_{i-1}^{\,2}) + x_i^{\,2} + y_i^{\,2} + z_i^{\,2} - (x_{i-1}^{\,2} + y_{i-1}^{\,2} + z_{i-1}^{\,2}) \end{aligned} \tag{57}$$

If we continue this process and write the results in matrix form, we get

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & 2(z_2 - z_1) & -2v^2(t_2 - t_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) & 2(z_3 - z_2) & -2v^2(t_3 - t_2) \\ 2(x_4 - x_3) & 2(y_4 - y_3) & 2(z_4 - z_3) & -2v^2(t_4 - t_3) \\ \dots & & & \\ 2(x_m - x_{m-1}) & 2(y_m - y_{m-1}) & 2(z_m - z_{m-1}) & -2(t_m - t_{m-1}) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t_0 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{\mathrm{m}} \end{bmatrix}, \tag{58}$$

where

$$d_i = v^2\left(t_i^{\,2} - t_{i-1}^{\,2}\right) + x_i^{\,2} + y_i^{\,2} + z_i^{\,2} - \left(x_{i-1}^{\,2} + y_{i-1}^{\,2} + z_{i-1}^{\,2}\right) \tag{59}$$

For convenience we denote the coefficient matrix at the left-side as **G**, the parameter vector to be estimated as **m**, and known data at the right-side as **d**,

$$\mathbf{Gm} = \mathbf{d}. \tag{60}$$

To solve this linear regression problem the singular value decomposition (SVD) method (Aster et al., 2005) is used in this paper. First, **G** is factored into

$$\mathbf{G} = \mathbf{USV}^{\mathbf{T}} \tag{61}$$

where **U** is an *m* by *m* orthogonal matrix with columns that are unit basis vectors, **S** is an *m* by 4 diagonal matrix with nonnegative diagonal elements called singular values, **V** is a 4 by 4 orthogonal matrix with columns that are basis vectors and $^{\mathbf{T}}$ means transpose.

If only the first *p* singular values are nonzero, then we can partition **S** as

$$\mathbf{S} = \begin{bmatrix} \mathbf{S_p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{62}$$

then the solution to m will be

$$\mathbf{m} = \mathbf{V_p S_p^{-1} U_p^T d}, \tag{63}$$

where $\mathbf{V_p}$ and $\mathbf{U_p}$ mean the first p columns of **V** and **U**.

Assume that measurement errors of t_i are independent and normally distributed, standard deviations σ_i can be incorporated into the solution by weighting **G** and **d**,

$$\mathbf{G}_w = diag(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_m})\mathbf{G}\,, \tag{64}$$

$$\mathbf{d}_w = diag(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_m})\mathbf{d}\,. \tag{65}$$

If the standard deviation for all t_i are identical, then the covariance **C** for the estimated parameter vector **m** can be calculated by

$$\mathbf{C} = \sigma^2 \mathbf{V_p} \mathbf{S_p^{-2}} \mathbf{V_p^T}\,. \tag{66}$$

The covariance matrix **C** can be used to estimate 95% confidence intervals for individual parameters which is given by

$$\boldsymbol{m} \pm 1.96\ diag(\mathbf{C})^{1/2}\,, \tag{67}$$

If we consider combinations of multi-parameters, the confidence region is a 3D ellipsoid. This ellipsoid can be calculated by diagonalizing the inverse of the covariance, $\mathbf{C}^{-1}$,

$$\mathbf{C}^{-1} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T\,, \tag{68}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of positive eigenvalues, and the columns of **P** are orthonormal eigenvectors. The i^{th} semimajor error ellipsoid axis direction is defined by **P**, i.e., its length l is determined by

$$l = \Delta / \sqrt{\Lambda_{i,i}}\,. \tag{69}$$

Example

We use 10 surface receivers and constant RMS velocity v=3000m/s to estimate the location of a microseismic event and its error distribution. First-break arrival times were perturbed with a Gaussian distribution of zero mean and standard deviation of 10ms and 3ms respectively. The 10ms uncertainty is considered an upper bound when observing first-break arrival times (Eisner et al., 2009).

Figure 14 and Figure 15 are 2D views of error distributions with 2 ms and 10 ms standard deviations respectively. It shows that the uncertainty in the vertical direction is much larger than in the horizontal directions. For example, when the deviation is 10ms, the error in z direction is about 60 meters which is larger than the 8 meters of the x direction.

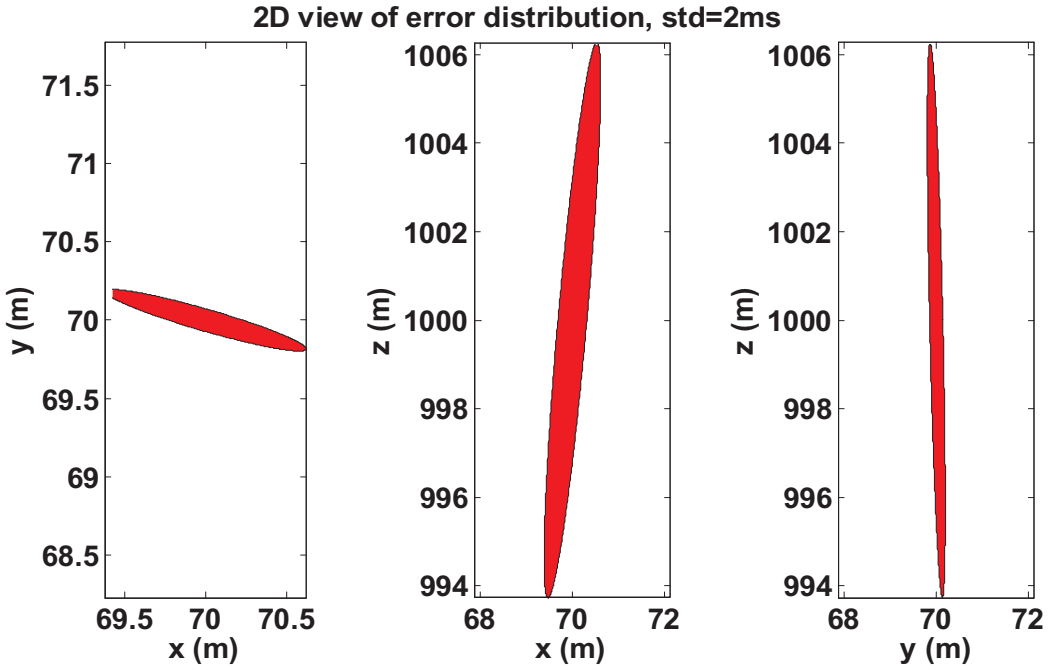


FIG. 14 2D view of error distribution of source with std=2ms.

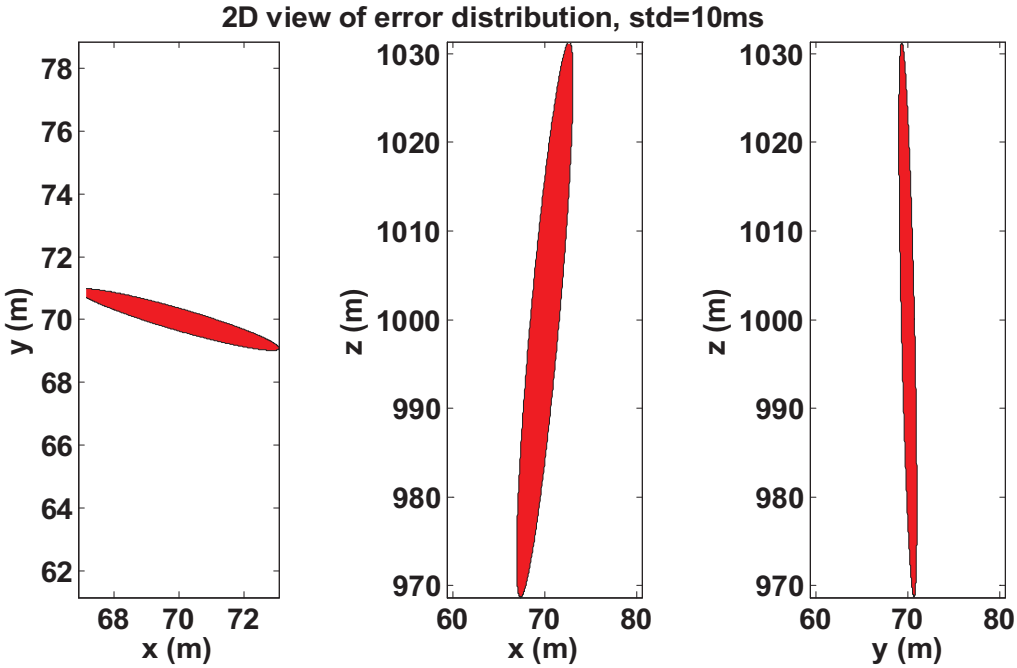


FIG. 15 2D view of error distribution of source with std=10ms.

Figure 16 is a 3D view of the error distribution of a source with a standard deviation equal to 2 ms and 10 ms respectively.

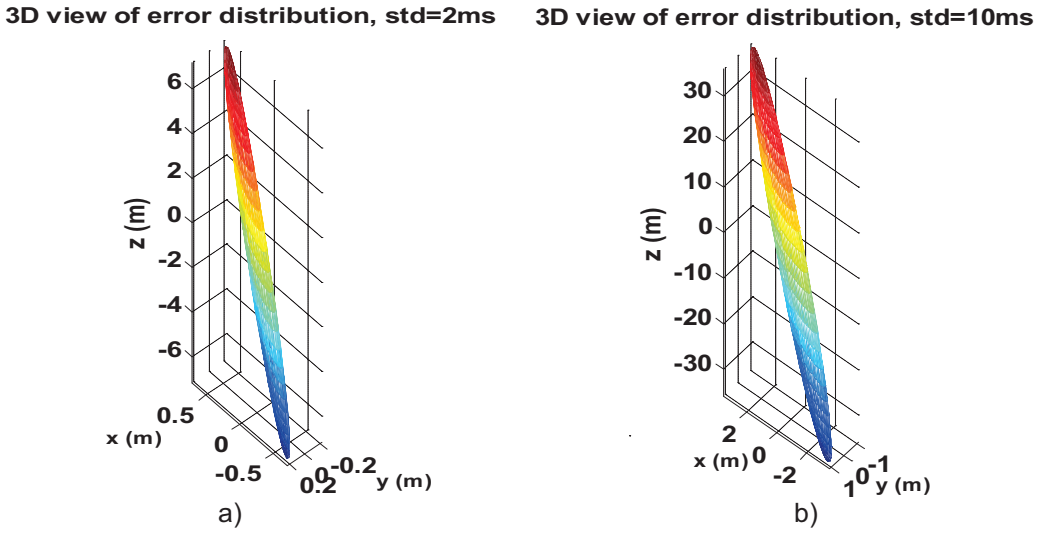


FIG. 16 3D view of the error distribution of a source a) with std=2ms and b) with std=10ms.

CONCLUSION

Two methods for estimating the sensitivity of a microseismic event were presented. The intent is to use these examples to illustrate the linear algebra of the second method with visual examples. The linear algebra processes of model building, scaling, covariance matrix, least squares (LS), singular value decomposition (SVD), and sensitivity were presented.

Given first-break arrival time at each receiver, the SVD method can be used to estimate a microseismic event and its origin time. If the standard deviations are known, the error distribution of the source can be estimated.

ACKNOWLEDGEMENTS

We wish to thank the sponsors of the Consortium for Research in Elastic Wave Exploration Seismology (CREWES) for supporting this research.

REFERENCES

Aster, R.C., Borchers, B. and Thurber, C.H., 2005, Parameter estimation and inverse problems: 15-64, Elsevier Academic Press.

Bancroft, J. C., Wong, J., Han, L., 2009, Sensitivity measurements for locating microseismic events, CREWES Research Report, Vol. 21.

Bancroft, J.C. and Du, X., 2006, Locating microseismic events and traveltime mapping using locally spherical wavefronts: CREWES Research Report, 18.

Eisner, L. et al., 2009, Uncertainties in passive seismic monitoring: The Leading Edge, 6, 648-655.

Qiao, B., and Bancroft, J. C., 2009, Error distribution when using first-break arrival times to locate microseismic events, CREWES Research Report, Vol. 21.

http://www.uwlax.edu/faculty/will/svd/projections/index.html

http://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors