Geol 483.3

Lab project #2 – Seismic source location

You will study grid search and iterative methods to solve a 2-D seismic (earthquake) location problem. You are given a 13-station array that recorded the following first-arrival times from two earthquakes:

Station	x [km]	y [km]	t_1 [s]	t_2 [s]
1	9.0	24.0	14.189	20.950
2	24.0	13.2	13.679	21.718
3	33.0	4.8	13.491	21.467
4	45.0	10.8	14.406	21.713
5	39.0	27.0	13.075	20.034
6	54.0	30.0	15.234	20.153
7	15.0	39.0	13.270	18.188
8	36.0	42.0	12.239	16.008
9	27.0	48.0	12.835	15.197
10	48.0	48.0	14.574	16.280
11	15.0	42.0	12.624	16.907
12	18.0	15.0	13.496	21.312
13	30.0	36.0	10.578	16.664

This table is also available in Excel format from the lab web page. These arrivals represent P waves, and we will assume that they travel at a constant speed near 6 km/s, which is the typical lower-crustal P-wave velocity.

The goal of this lab is to determine the locations of each of these earthquakes and their times, and also to improve the estimate of the crustal velocity.

Theory

Seismic location is performed for each event independently and based on minimizing the misfit between the observed travel times and those predicted from the model. In a 2D case (only the epicenter coordinates (x,y) are unknown and assuming the hypocenter depth to be zero), the predicted times are:

$$t_i = t_S + \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{V}, \qquad (1)$$

where (x_i,y_i) are the coordinates of i-th station, t_S is the time of the source, and V is the velocity. The total travel-time misfit, measured using the L_2 (also often called the RMS, Root Mean Square) norm is:

$$\Phi\left(x,y\,|\,t_S,V\right)=\sum_i\left(t_i^{observed}-t_i\right)^2=\sum_i\left[t_i^{observed}-\left(t_S+\frac{\sqrt{\left(x-x_i\right)^2+\left(y-y_i\right)^2}}{V}\right)\right]^2. \quad (2)$$

This function of variables (x,y) is called the objective function, cost function, or penalty function. The best source location (x,y) is the one minimizing this objective function.

For simple problems like the one in this lab, this optimum location can be found by computing function (1) at all grid points of interest, contouring it or using color displays, and finding the minimum.

The approach in eq. (2) has a problem, which is the need to estimate parameters V and t_S prior to evaluating function $\Phi(x,y)$. These variables could be included in grid search, but the grid would become 4-D and very computationally expensive. To overcome this problem, we can construct another objective function in 3-D space of parameters (x,y,V) as follows:

- Assuming known (x,y,V), calculate the source times at (x,y) that would predict the observed arrival times at each station i:

$$t_S^{(i)}=t_i-\frac{\sqrt{\left(x-x_i\right)^2+\left(y-y_i\right)^2}}{V}. \quad (3)$$

- Then, for any (x,y), the time of the source can be determined as the mean of all candidate times from each station, and the variance (error) of this time can be obtained from the standard deviation of $\left\{t_S^{(i)}\right\}$:

$$t_S=\sum_{i=1}^{N_{st}}t_S^{(i)}, \qquad \sigma_t\approx s_t=\sqrt{\frac{\sum_{i=1}^{N_{st}}\left(t_S^{(i)}-t_S\right)^2}{N_{st}-1}} \quad (4)$$

- Using this t_S, the objective function for 3-D grid search (the quantity which we are trying to minimize) is the variance of time misfits:

$$\Phi\left(x,y,V\right)=s_t. \quad (5)$$

This time error can also be related to the objective function in eq. (2):

$$\Phi\left(x,y,V\right)=\sqrt{\frac{\Phi\left(x,y\,|\,t_S,V\right)}{N_{st}-1}}. \quad (6)$$

Algorithm

Write Matlab code to perform the 3-D grid search minimizing the function in eq. (5). Implement your algorithm in a modular code like this:

1) Set up the model area and parameters using `global` variable `model`, analogously to Lab 1. Include the earthquake data table in this variable.
2) Modify function `plot_model.m` from Lab 1 to plot an arbitrary grid within the model area, with station locations on top of it.
3) Create functions to calculate travel times from an arbitrary source point to each of the stations using eq. (1).
4) Using the above function, create function `locate()` returning the source time and its standard deviation (eq. (4)) for a fixed value of velocity V.
5) Using function `locate()`, loop through a range of values V and find the velocity minimizing the objective function (5).
6) Create plots for testing and examining various aspects of the problem (gridded values of travel times, objective functions; travel and arrival times vs. source-receiver distances).

Confidence ellipse

There always are some errors in the data, and so you will not be able to achieve zero misfit. From the residual time misfits, estimate the data variance:

$$\sigma^2 = \frac{\sum_{i=1}^{N}\left(t_i^{observed} - t_i\right)^2}{N_{df}} \qquad (2)$$

where N_{df} =(N–2) is the "number of degrees of freedom" (the number of data points (travel times) minus the number of adjusted model parameters; see lecture notes). Square root of this variance (σ) represents an estimate of data (travel time) error.

Random errors in the data lead to errors in location. To measure the uncertainty of location, compute and contour the following function of location (x,y):

$$\chi^2(x,y) = \frac{\sum_{i=1}^{N}\left[t_i^{observed} - t_i(x,y)\right]^2}{\sigma^2}.$$

This function represents the normalized travel-time error associated with a location selected at (x,y). Values of χ^2 are tabulated (Table 1 below and in the lectures) and can be used to determine the probability that the trave-time error for location at (x,y) is still caused by random data errors. To determine the ***confidence ellipse***, you will need to find the contour in $\chi^2(x,y)$ corresponding to 95% percentage points of the distribution (fourth column highlighted by red color in Table 1). The meaning of this selection is that in areas where we observe the value of χ^2 is larger than the one in this column, there exists only 5% probability that this χ^2 comes from a random Gaussian distribution of travel-time errors. Note that the actual shape of this "confidence ellipse" will be far from elliptical and can maybe better described as "banana".

Table 1. χ^2 values at several confidence levels.

N_{df}	At 5%	At 50%	At 95%
5	1.15	4.35	11.07
10	3.94	9.34	18.31
20	10.85	19.34	31.41
50	34.76	49.33	67.5
100	77.03	99.33	124.34

[20%] Write a Matlab/Octave program to perform grid search to find the best location for each earthquake. Make the grid-search area for (x,y) variable. A 100 by 100 grid from about –100 to 150 km in x and -70 to 170 km in y should be adequate for seeing the confidence ellipse, but you can use other values. Try teh range of velocities V from 5 to 6 km/s.

Note that the travel times t_1 ns t_2 in the table above assume the origin times to be *approximately* 0, which is, however, not exactly true. The task of your inversion is to also estimate the origin times. For each earthquake, just assume the origin times to be 0 to begin with, then average the residuals from all stations. This average is your best estimate of the origin time.

To gain some experience first, start with earthquake #2, which is an easier case. You will find the case of earthquake #1 more challenging in the sense that its travel-time errors are larger, and the location is poorer constrained. Its time start will also heavily trade off with the selection of V.

For each earthquake:

1. [20%] Find the best location and *origin* time by making several iterations 1)-6) described above;
2. Obtain the variance (standard deviation) of the time residuals at the best-fitting point using the second eq. (4). This is your estimate of the overall data uncertainty.
3. [10%] Compute χ^2 for each grid point using the expression above. What is the χ^2 for the best-fitting point?
4. Make a contour plot of $\chi^2(x,y)$. Identify those values that are within the 95% confidence range.

5. [5%] Make a plot showing the station locations, the best location, and the range within the 95% confidence region.
6. [5%] Comment on the shapes of the confidence regions, similarities and differences between the two earthquakes.

Hand in:

Codes, plots, and report by email.