Development of a Direct Search Software Package for Locating Poorly Constrained Earthquakes*

W. H. K. Lee and D. A. Dodge

- Presented by Willie Lee at the 2006 Annual Meeting of the Seismological Society of America, 18-22 April, 2006 in San Francisco, CA, USA. The abstract by W.H.K. Lee and L.M. Baker was published in *Seism. Res. Lett.*, vol. 77, p. 291-292, 2006.
- This report was published in *Seismology Technical Report* Vol.. *44*, p. 6-27, Central Weather Bureau, Taipei, Taiwan (ISSN 1815-4328).

Abstract

Most commonly used algorithms for locating earthquakes on computers are based on an inverse formulation (Geiger, 1912). Numerous software implementations have been made using the Geiger method, which applies the Gauss-Newton nonlinear optimization technique to find the origin time and hypocenter by iterative linearized steps starting from a trial solution. Recent advances in earthquake location methods are mostly concentrated on obtaining the best relative locations for a group of earthquakes using high-quality data (including waveforms) recorded by a dense seismic network (e.g., Richards et al., 2006). However, Geiger-like location programs do not work well for poorly constrained earthquakes, because the available arrival times may not be sufficient to solve the inverse problem, and the chosen trial solution may lead to a local minimum.

A physical problem involving observations is much easier solved by a forward or direct formulation, if the large amounts of computations required can be managed. Recently, computers became fast enough that the forward approach has been explored (e.g., Sambridge and Kennett, 2001; Oye and Roth, 2003). Instead of a brute grid search, we use the downhill simplex algorithm to search the neighborhood of thousands of grid points that coarsely cover the solution space. The downhill simplex algorithm was chosen for its robustness (Press et al., 1986). Computation is intensive, but a robust solution can be found in about 100 seconds on a Pentium 4 PC for a typical case, along with a 3-D residuals map for visualizing the solution's uncertainty. This direct approach has three major advantages over the inverse formulation: (1) a global minimum in the solution space can be obtained, (2) the computation is simple and straight forward, and (3) it can be adapted to perform either a L1-norm or L2-norm minimization.

We are now developing a general software package (JLOC written in Java) with 3-D visualization (using MATLAB), specifically for locating earthquakes before 1963 when

the available phase data were highly variable in quality, the distributions of the recording stations were highly non-uniform, and many earthquakes occurred outside existing seismic networks.

Introduction

So far, most commonly used algorithms for locating earthquakes on computers are based on an inverse formulation, first published by L. C. Geiger (1912). Numerous software implementations have been made using the Geiger method, which applies the Gauss-Newton nonlinear optimization technique to find the origin time and hypocenter by iterative linearization steps starting from a trial solution.

The travel time residuals (i.e., observed minus predicted from a given velocity model) of the first P-wave (and sometimes the S-wave and later phases) are minimized, usually in the least squares sense (L2 norm), and occasionally in L1 norm. Waldhauser and Ellsworth (2000) introduced the "double- difference" algorithm, which minimizes the residuals of travel times differences for pairs of earthquakes observed at common stations by iteratively adjusting the vector connecting the hypocenters. Similar to the Joint Hypocentral Determination, JHD, (Pujol, 2003), the double-difference algorithm improves "relative" earthquake locations and works well for a good set of arrival times with large numbers of stations. For a comprehensive review of recent earthquake location developments, see Richards et al. (2006). In brief, they are all variants of the Geiger method based on an inverse formulation, and they are mostly concentrated on obtaining the best *relative* accuracy of the hypocenters for data (including waveforms) recorded in a dense seismic network.

The mathematics of the inverse formulation is elegant as shown in the next section, and it works well for a good seismic network with stations surrounding the epicenters.

However, all existing location programs work poorly for earthquakes outside a seismic network, because the available arrival times are not sufficient to solve the problem mathematically as shown below (greatly condensed from Lee and Stewart, 1981 p. 105-139). We will use bold face symbols to denote vectors or matrices.

The Least Squares Method and Nonlinear Optimization

In the least squares method, we attempt to minimize the errors of fit (or residuals) at a set of m data points where the coordinates of the kth data point are $(\mathbf{x})_k$, k =1, 2,…, m., and **x** is a vector of n independent variables, i.e.,

$$\mathbf{x} = (x_1, x_2, \ldots, x_\mathrm{n})^\mathrm{T} \qquad (1)$$

where the superscript T denote a vector transpose. The objective function for the least squares problem is

$$F(\mathbf{x}) = \sum [r_k(\mathbf{x})]^2 \qquad (2)$$

where the summation is from k = 1 to m, and $r_k(\mathbf{x})$ denotes the evaluation of residual at the kth data point. We may consider these residuals as components of a vector in the m-dimensional Euclidean space and Equation (1) becomes

$$F(\mathbf{x}) = \mathbf{r}^\mathrm{T}\,\mathbf{r} \qquad (3)$$

Taylor expansion of this objective function is

$$F(\mathbf{x} + \boldsymbol{\delta}\mathbf{x}) = F(\mathbf{x}) + \mathbf{g}^\mathrm{T}\,\boldsymbol{\delta}\mathbf{x} + \tfrac{1}{2}\,\boldsymbol{\delta}\mathbf{x}^\mathrm{T}\,\mathbf{H}\,\boldsymbol{\delta}\mathbf{x} + \ldots \qquad (4)$$

Where **g** is the gradient vector and **H,** the Hessian matrix, and it can be shown that

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{-}\ \mathbf{H}^{-1}\,\mathbf{g} \qquad (5)$$

To find the gradient vector **g,** we perform partial differentiation on Equation (2) with respect to x_i , i = 1, 2, …, n, and obtain

$$\partial F(\mathbf{x})/\,\partial x_i = \sum 2\ r_k(\mathbf{x})\ [\partial r_k(\mathbf{x})\ /\partial x_i],\ \ i = 1, 2, \ldots., n \qquad (6)$$

and the summation is from k = 1 to m. In matrix notation,

$$\mathbf{g} = 2\ \mathbf{A}^\mathrm{T}\ \mathbf{r} \qquad (7)$$

where **A** is the Jacobian matrix, whose elements are defined by

$$A_{ki} = \partial r_k / \partial x_i, \qquad k = 1, 2, \ldots, m, \quad \text{and} \quad i = 1, 2, \ldots, n \tag{8}$$

To find the Hessian matrix **H**, we perform partial differentiation on the set of *n* equations in Equation (6) with respect to x_j , for $j = 1, 2, \ldots, n$, assuming that $r_k(\mathbf{x})$, $k = 1$ to *m*, have continuous second derivatives, and obtain

$$\mathbf{H} \cong 2\, \mathbf{A}^T \mathbf{A} \tag{9}$$

by ignoring the cross derivative terms. Hence,

$$\boldsymbol{\delta}\mathbf{x} = -\,[\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T\ \mathbf{r} \tag{10}$$

The Geiger's method is essentially applying the least squares method and using the above Gauss-Newton algorithm to solve the earthquake location problem. Starting from a trial origin time and hypocenter, the adjustment vector **δx** as given in Equation (10) is solved. A new origin time and hypocenter is then obtained and the same procedure is repeated again until some cutoff criteria are met. However, the Jacobian matrix as given in Equation (8) is often ill-conditioned for giving a meaningful inverse, and if the trial solution is not chosen appropriately, the iterative procedure converges to a local minimum, rather than the global minimum.

For the earthquake location problem, the 4 independent variables are: time *t*, and coordinates *x*, *y*, and *z*. The Jacobian matrix **A** may be written with column vectors as its elements:

$$\mathbf{A} = (\,\mathbf{V_1} \quad \mathbf{V_2} \quad \mathbf{V_3} \quad \mathbf{V_4}\,) \tag{11}$$

where

$$\mathbf{V_1} = (\,1 \qquad 1 \quad \bullet\ \bullet\ \bullet \quad 1\ \,)^T \tag{12}$$

$$\mathbf{V_2} = (\partial t_1/\partial x \quad \partial t_2/\partial x \quad \bullet\ \bullet\ \bullet \quad \partial t_m/\partial x\,)^T \tag{13}$$

$$\mathbf{V_3} = (\partial t_1/\partial y \quad \partial t_2/\partial y \quad \bullet\ \bullet\ \bullet \quad \partial t_m/\partial y\,)^T \tag{14}$$

$$\mathbf{V_4} = (\partial t_1/\partial z \quad \partial t_2/\partial z \quad \bullet\ \bullet\ \bullet \quad \partial t_m/\partial z\,)^T \tag{15}$$

where the travel times for m stations are denoted by: $t_{1,}\ t_2$, …, t_m . Since the travel time derivatives with respect to time are 1, vector $\mathbf{V_1}$ is a unit vector. Let us recall that the determinant of a matrix is zero if any of its columns is multiple of another column. Since the first column of the Jacobian matrix is all 1's, it is easy for other columns of A to be a multiple of it. E.g., if an earthquake occurs outside the seismic network, it is likely that the elements of the $\partial t/\partial$x column, and the corresponding elements of the $\partial t/\partial$y will be nearly proportional to each other. In other words, we do have adequate observed data to solve the matrix for meaningful adjustments.

Although the Geiger method was published in 1912, computations are too laborious until the electronic computer era in the 1950s. There are many pitfalls in solving a problem by the inverse approach primarily because no one has yet found a fool-proof technique to guarantee a true solution in nonlinear optimization since Gauss time (about two hundred years ago). Unlike the Fermat's last theorem (which was solved recently after more than 300 hundred years' efforts), many experts in optimization consider the guarantee for a global minimum is unsolvable in an inverse problem.

Some Direct Methods for Earthquake Location

Almost all physical problems involving observations are formulated as inverse problems simply because solving problems by the method of least squares become so standard (since Gauss first popularized it) that few scientists ever question it. After electronic computers became available in the late 1950s, a few visionary scientists realized that it is much easier to solve a physical problem involving observations by the forward formulation. Unfortunately it will involve large amounts of computations and computers were far too slow at that time. However, it was adequate for solving most inverse problems. When W. H. K. Lee examined the earthquake location problem from both the mathematical and computation points of view in the late 1960s, he realized that the computers were about 5 orders of magnitude too slow for the forward formulation and thus had to wait.

By the early 2000, computer speed has increased about 10,000 times faster that in the 1960s, Lee began laying a plan to attach the earthquake location problem by the forward formulation approach. The least squares method of Gauss assumes that the observational errors have a normal (or Gaussian) distribution. However, this assumption is not appropriate for earthquake arrival times, which often have large outliers.

Lee and Doug Dodge then began investigating the simplex algorithm (Press et al., 1986) developed for minimizing the L1 norm (rather than the L2 norm in least squares). Recently, they are developing a forward simplex search software for earthquake location, and showed that it is practical to use it for relocating large numbers of earthquakes, provided a fast multi-processor PC is available.

Several seismologists have been exploring direct search methods, e.g., Sambridge (2003) provided a software package for the neighborhood algorithm (NA) as a direct search method for nonlinear inversion. NA had been applied to seismic event location by Sambridge and Kennett (2001), and to automated seismic event location by Oye and Roth (2003).

The JLOC Computer Program: Overview

The JLOC program is an earthquake location program that computes earthquake source location parameters using forward modeling and the IASPEI91 Earth model. It uses the TauP traveltime calculator *Crotwell, H. P., T. J. Owens, and J. Ritsema (1999).* Locations can be calculated using a grid search followed by a simplex finishing step or using the simplex procedure alone. The objective function evaluated by the locator at each point in the solution space is an L1 norm, so the locator is relatively insensitive to large outlier observations at a few stations. In grid-search mode, the locator also has the ability to remove a subset of observations with the largest residuals prior to the simplex finishing step. The locator has an interactive mode in which the operator can visually inspect residual distributions and add or remove observations from the solution.

The only required argument is the name of the input phase file. If not specified, the output file, summary file, and residual file names will all be based on the input file name with extensions “out”, “sum”, and “res” respectively. If the control file is not specified, a default set of parameters will be used. The Box below show below is a portion of an input file for the JLOC program. Comments begin with an asterisk. The hypocenter line is required and must precede the observation lines.

```
* File = 1920.157.04.21.35.pha
*
* Auth      Year Mo Dy Hr Mn Sec    Latitude  Longitude  Depth
  Doug      1920 06 05 04 21 35.62  24.0284  122.1706    0.0
*
* Code  Phase  Timedef  Ttime  Deltim    Latitude  Longitude  Elev.  Delta  Azimuth Residual
*
  HWA0   S-P      d      7.100   1.000     23.967    121.617   0.018    0.51    263.2    -0.000
  HWA0   P        n     52.380   1.000     23.967    121.617   0.018    0.51    263.2    42.603
  HWA0   S        n     61.380  10.000     23.967    121.617   0.018    0.51    263.2    44.503
  TAP0   S-P      d     16.200   1.000     25.033    121.517   0.018    1.16    329.4    -0.000
  TAP0   P        n     23.980   1.000     25.033    121.517   0.018    1.16    329.4     1.672
  TAP0   S        n     40.180   1.000     25.033    121.517   0.018    1.16    329.4     1.672
  TAP    P        n     24.380   1.000     25.039    121.506   0.006    1.17    329.1     1.877
  TCU0   S-P      d     16.800   1.000     24.150    120.683   0.018    1.36    275.4    -2.188
  TCU0   P        n      5.680   1.000     24.150    120.683   0.018    1.36    275.4   -20.467
  TCU0   S        n     22.480   1.000     24.150    120.683   0.018    1.36    275.4   -22.655
  TAI0   S-P      d     28.700   1.000     23.000    120.217   0.018    2.06    240.7     1.289
  TAI0   P        n     24.380   1.000     23.000    120.217   0.018    2.06    240.7   -11.526
  TAI0   S        n     53.080   1.000     23.000    120.217   0.018    2.06    240.7   -10.237
```

Each observation line consists of:

- Station code
- Phase

- Timedef (one of ‘n’, ‘d’) – – ‘d’ for use in location, ‘n’ not to be used.
- Travel time (time in seconds relative to the origin time for this phase.
- Deltim (the standard error of this observation.)
- Latitude
- Longitude
- Elevation
- Delta (degrees) – – epicentral distance.
- Azimuth (degrees) – – azimuth angle to station with respect to North.
- Residual (seconds)

The last four columns are parsed but not currently used by the program. They are included because the ISS Summary published these values. The order of the columns must be as shown above, but column position is not important. It is only necessary that there be white space between the columns.

If JLOC is started in non-interactive mode it will simply calculate a new solution and write the results to the summary file, the output file, and the residual file. The summary file contains a 1-line summary of the hypocentral parameters:

- Date
- Day of Year
- Time
- Lat
- Lon
- Depth
- RMS
- MAR (Mean Absolute Residual)
- Nass (Number of associated phases)
- Ndef (Number of defining phases)

- Azgap
- Azgap2 (Secondary azgap)
- Sigmalat (Standard error of the latitude)
- Sigmalon (Standard error of the longitude)
- SigmaZ (Standard error of the depth)
- Semi-major (Length of the semi-major axis (km) of the 95% confidence ellipse)
- Semi-minor (Length of the semi-minor axis (km) of the 95% confidence ellipse)
- Strike (Strike of the semi-major axis of the 95% confidence ellipse)

The output file contains information from various stages of the solution, the final solution hypocentral parameters, and the final state of the observations in the solution. The residual file contains the residuals corresponding to each point evaluated during the grid search operation.

JLOC: Interactive Operation

When JLOC is invoked using the –i command line option, it starts up in interactive mode. In this mode the user can easily see the distribution of residuals. The user can also interact with the program to change the defining status of selected observations and see the effect on the source location. In interactive mode, JLOC reads the input file and displays the data on a world map (Figure 1) without attempting a source location. Initially, the map is centered on the epicenter given in the input file and the entire world is displayed using an equal area azimuthal projection.

The stations are displayed as colored triangles with color determined by the defining status of observations at the station and by the residuals of defining observations.

Stations with no defining observations are shown in light gray. Stations with one or more defining observations have their color determined by the observation with the largest residual. The color scale runs from pure green for residual values of 0 sec to pure yellow for residual values of 10 sec, and to pure red for residual values of 20 sec. Right-clicking on any station opens a popup menu that allows the user to change the defining status of each observation at that station.

The user can zoom in on a map detail by dragging with the left mouse button (Figure 2). The right mouse button un-zooms the map incrementally. By holding the shift key down the user can pan the map.

The JLOC Toolbar

Most of the functions in JLOC's interactive mode are controlled using the toolbar displayed at the top of the dialog. Three most important toolbar are:

The Grid Search Button:

When clicked, performs a new grid search for the minimum residual location using a grid centered on the current epicenter. By default, the grid search is over a region 6 degrees around the current epicenter using a step size of 1 degree. This can be altered by changing values in the control file read by JLOC on startup. The grid search always incorporates a simplex finishing step, so normally the only reason to use the grid search is if you think the program has gotten stuck in a local minimum.

The Simplex Solution Button:

When clicked, performs a Simplex search for the location with the minimum residual value. The control parameters for the simplex search can be modified by editing the control file read by JLOC on startup. After each simplex search is complete, the map is updated with the new solution parameters. The formal 95% confidence ellipse is also displayed (See Figure 3 below). This ellipse is calculated using the assumption that the data are uncorrelated and have equal variance. In fact, the data variance used in generating the ellipse is the average of the deltim squared values, so this statistic has room for improvement.

The Display Residual Map Button:

This is a toggle button that when selected causes computation of residuals over a region surrounding a new location. The residuals are displayed as a color contour map superimposed over the map. The residual values are displayed using a color map in which white corresponds to a residual of zero and black corresponds to the largest residual computed (see Figure 4).

Visualization using MATLAB

JLOC writes a grid of residual values for the final solution at user-specified spacing to a file for further processing. We use MATLAB to visualize these results. The JLOC grid of residual values is converted to a 3-D 50 x 50 x 50 grid (~1 MB). To geographically locate the study area, we project the surface of the 3-D residuals grid onto a Mercator map of the nearby coastline, colored by the residual value, in seconds (Figure 5).

To visualize the structure of the residual data values, we compute vertical slices of the data through the final hypocenter and isosurfaces for the data at 1, 2, 4, and 8 seconds. Figure 6 shows a 3-D view of the slices. The X-axis is longitude, the Y-axis is latitude, and the Z-axis is km depth. Color is residual value, in seconds; the colorbar legend is on the right. The surface coastline fragment is drawn in black.

Figure 7 adds the isosurfaces to Figure 6, which are also colored by residual value. The minimum residual value is 2.106 sec, so the only isosurfaces are for 4 and 8 seconds. The slices have transparency that varies with residual value: the larger the residual value, the more transparent the slice becomes. The isosurfaces are opaque and are illuminated for better discrimination by an artificial light source from above and left at infinity. The crescent shape in the northwest portion of the solution that can be seen in Figure 5 is clearly seen in the structure of the 4 second isosurface (yellow).

Conclusions

We demonstrated that a general software package (JLOC written in Java) can locate earthquakes that have phase data of highly variable in quality, or have the recording stations highly non-uniform in distribution, or occur outside existing seismic networks. Because earthquakes offshore of Taiwan are located outside of the CWB current seismic network, this JLOC software will help CWB to find more reliable locations of these earthquakes in their routine operation, and to relocate old earthquakes.

Forward formulation can solve many more seismological problems, including seismic tomography. Many seismological problems involve the Greens function, and Greens function is an inverse operator, as pointed out by Lanczos (1961). The forward formulation will usher a new era in research because computers are now fast enough to make the forward approach practical.

Acknowledgements.

We are grateful to Joe Fletcher and Steve Kirby for their comments and suggestions, and Larry Baker for developing the visualization code.

References

Crotwell, H. P., T. J. Owens, and J. Ritsema (1999). The TauP Toolkit: Flexible travel time and ray path utilities, *Seismological Research Letters*, **70(**2), 154-160.

Geiger, L. C. (1912). Probability method for the determination of earthquake epicenters from the arrival time only, *Bull. St. Louis Univ*., **8**, 60-71.

Lanczos, C. (1961). *Linear Differential Operators*, Van Nostrand, London.

Lee, W. H. K., and S. W. Stewart (1981). *Principles and Applications of Microearthquake Networks*, Academic Press, New York.

Oye, V., and M. Roth (2003). Automated seismic event location for hydrocarbon reservoirs, *Computers & Geosci*., **29**, 851-863.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1986). *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge.

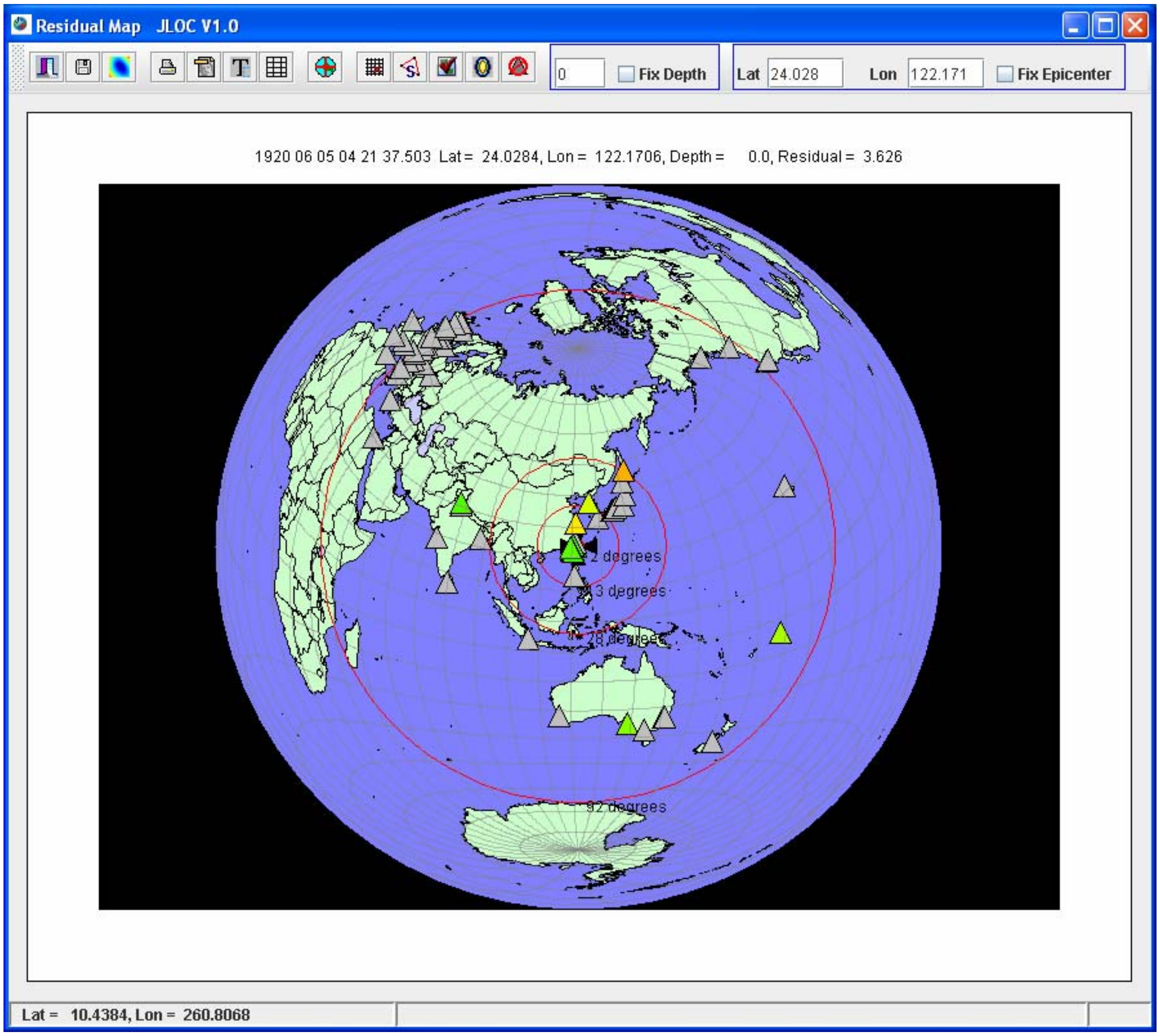
Pujol, J. (2003). Software for joint hypocentral determination using local events, in: *International Handbook of Earthquake and Engineering Seismology,* edited by W. H. K. Lee, H. Kanamori, P. C. Jennings, and C. Kisslinger, Part B, p. 1621-1623, Academic Press, San Diego.

Richards, P. G., F. Waldhauser, D. Schaff, and W. Y. Kim (2006). The applicability of modern methods of earthquake location, in: *Advances on Studies of Heterogeneities in the Earth's Lithosphere: The Keiiti Aki Volume II*, edited by Y. Ben-Zion and W. H. K. Lee, Pageoph Topical Volumes, Birkhauser Verlag, Basel.

Sambridge, M. (2003). Nonlinear inversion by direct search using the neighbourhood algorithm, in: *International Handbook of Earthquake and Engineering Seismology*, edited by W. H. K. Lee, H. Kanamori, P. C. Jennings, and C. Kisslinger, Part B, p. 1635-1637, Academic Press, San Diego.

Sambridge, M., and B. Kennett (2001). Seismic event location: nonlinear inversion by using a neighbourhood algorithm, *Pure Appl. Geophys.*, **158**, 241-257.

Waldhauser, F., and W. L. Ellsworth (2000). A double-difference earthquake location algorithm: Method and application to the northern Hayward fault, *Bull. Seism. Soc. Am.*, **90**, 1353-1368.



Residual Map JLOC V1.0

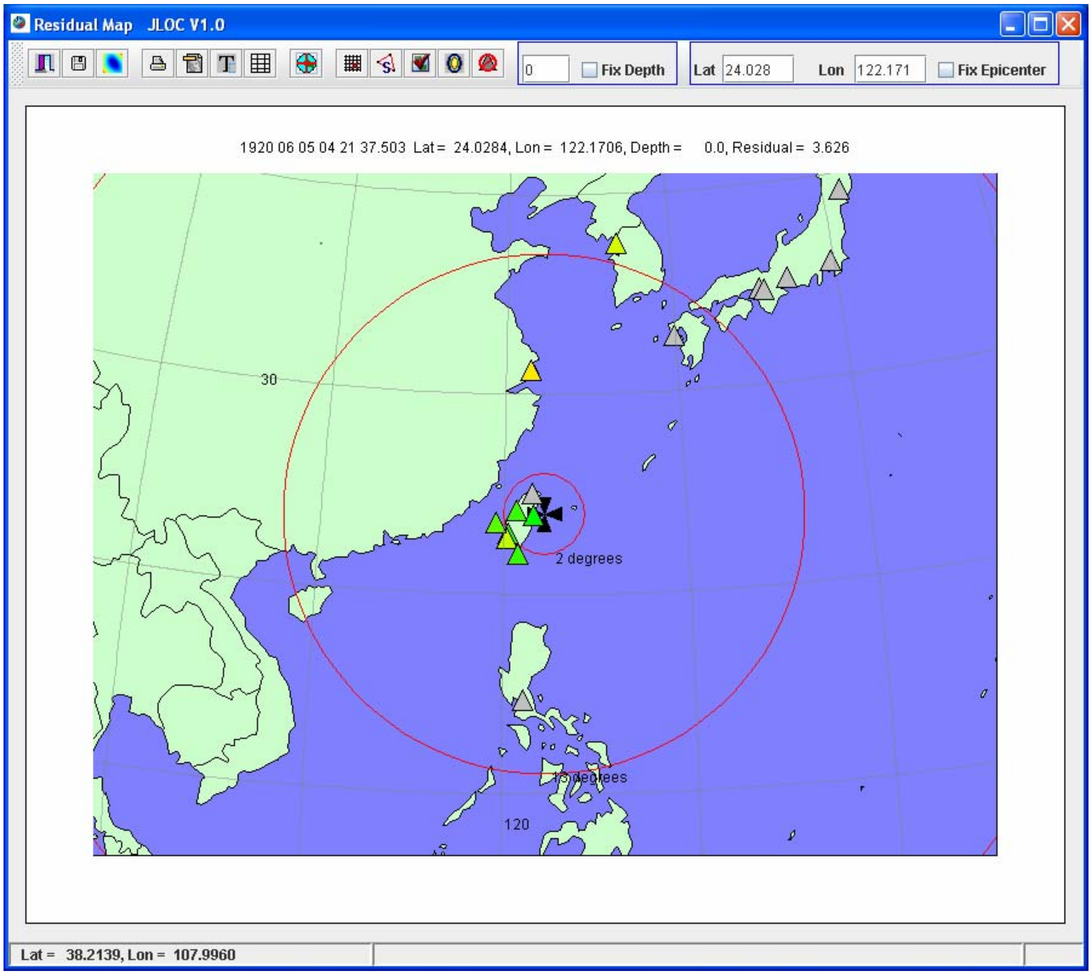
0 □ Fix Depth

Lat 24.028 Lon 122.171 □ Fix Epicenter

```
1920 06 05 04 21 37.503  Lat =  24.0284, Lon =  122.1706, Depth =      0.0, Residual =  3.626
```

```
Lat =   10.4384, Lon =  260.8068
```

Figure 1. The interactive residual map displayed when JLOC is invoked in interactive mode.



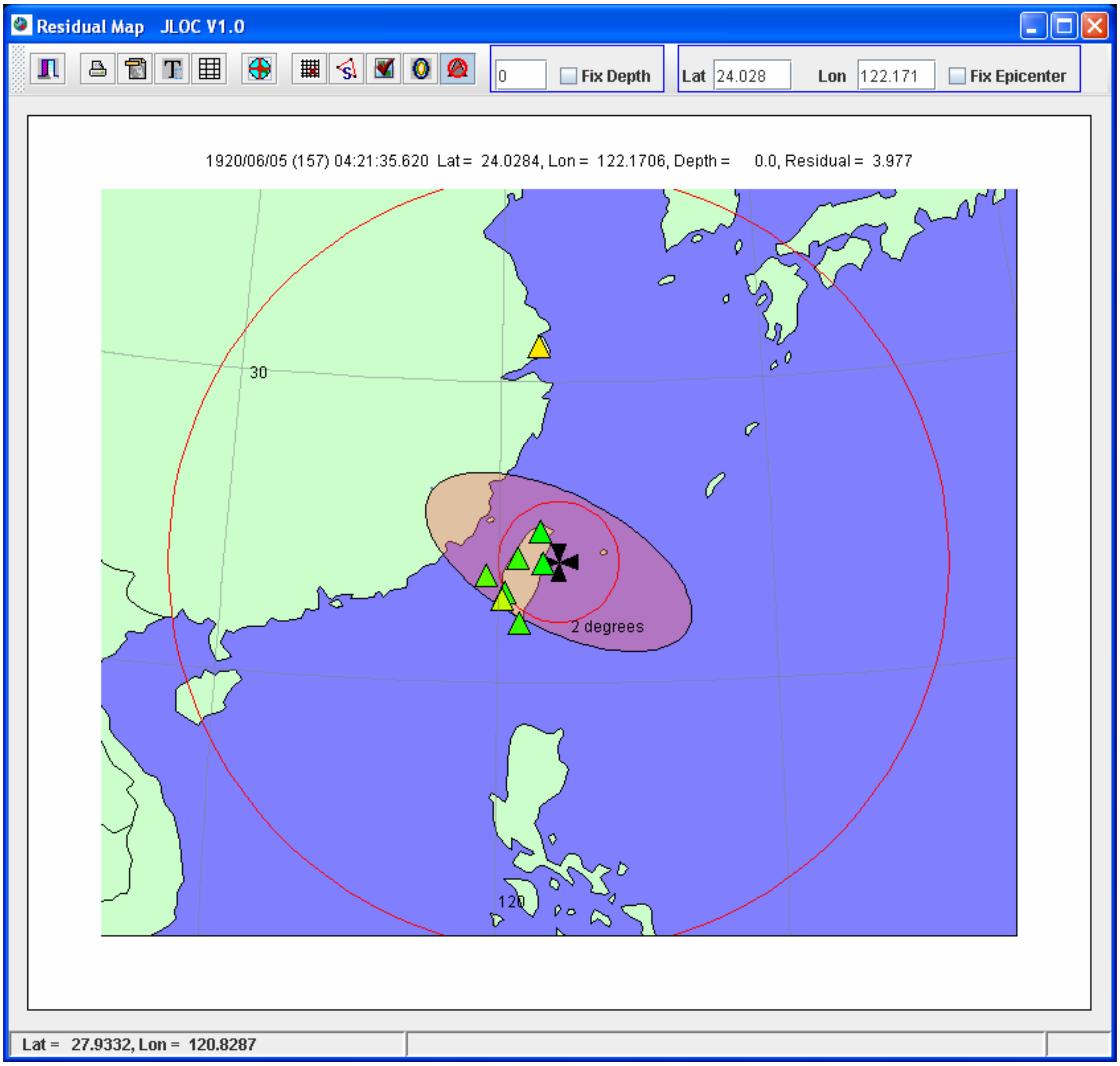
0 □ Fix Depth

Lat 24.028 Lon 122.171 □ Fix Epicenter

```
1920 06 05 04 21 37.503  Lat =  24.0284, Lon =  122.1706, Depth =      0.0, Residual =  3.626
```

```
Lat =   38.2139, Lon =  107.9960
```

Figure 2. Example map after zoom in to a region around the epicenter.



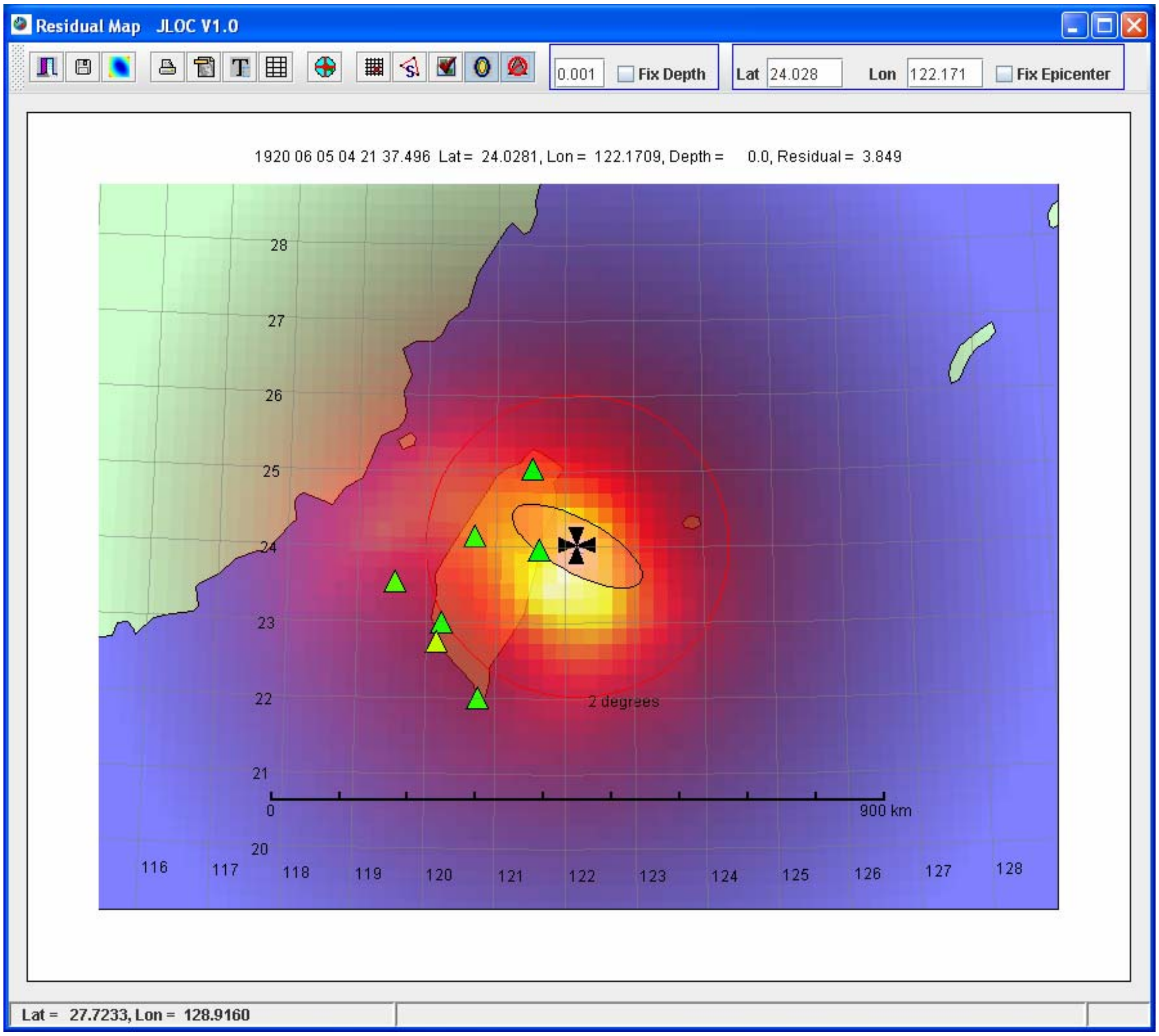
0 □ Fix Depth

Lat 24.028 Lon 122.171 □ Fix Epicenter

```
1920/06/05 (157) 04:21:35.620  Lat =  24.0284, Lon =  122.1706, Depth =      0.0, Residual =  3.977
```

```
Lat =   27.9332, Lon =  120.8287
```

Figure 3. Example map after zoom in to a region around the epicenter with the formal 95% confidence ellipse.



Residual Map JLOC V1.0

0.001 □ Fix Depth

Lat 24.028 Lon 122.171 □ Fix Epicenter

1920 06 05 04 21 37.496 Lat = 24.0281, Lon = 122.1709, Depth = 0.0, Residual = 3.849

Lat = 27.7233, Lon = 128.9160

Figure 4. Example map with contour map of residuals superimposed on the solution.

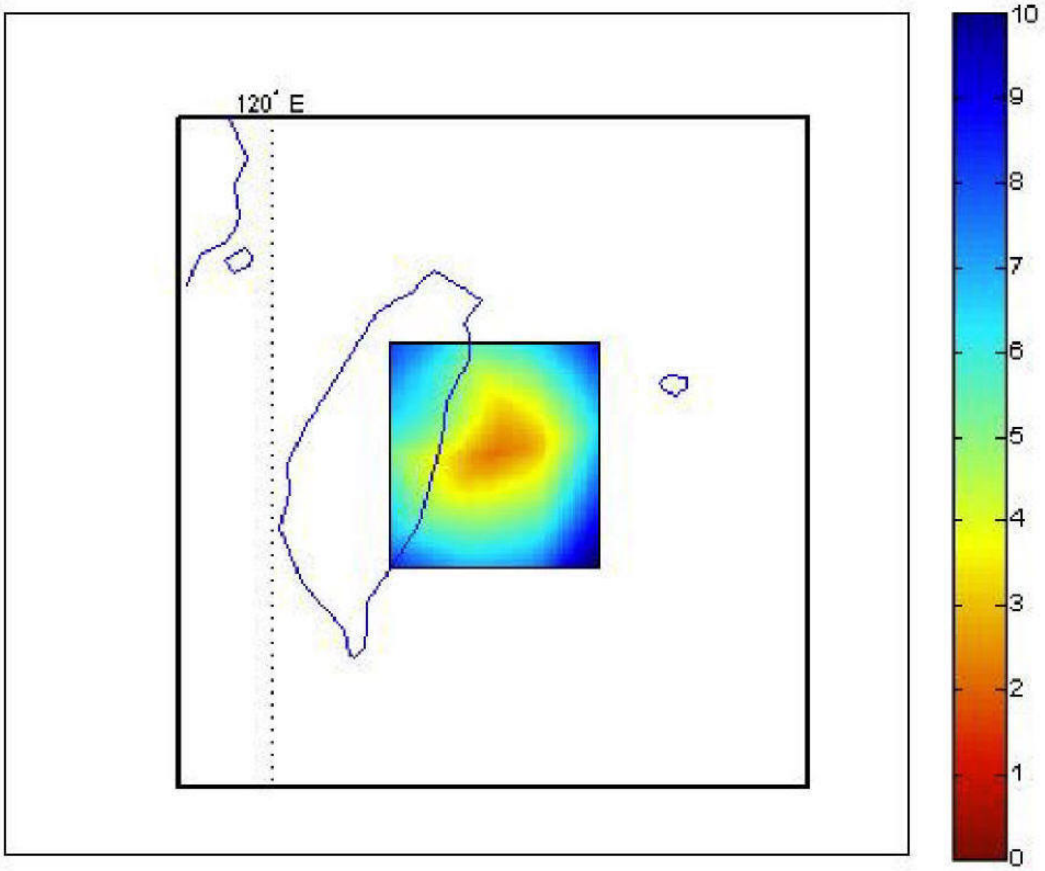


Figure 5. Mercator map of the study area and residual values at the surface. Color is residual value, in seconds; the colorbar legend is on the right. The world coastline data file and the map projection functions used are from the MATLAB Mapping Toolbox.

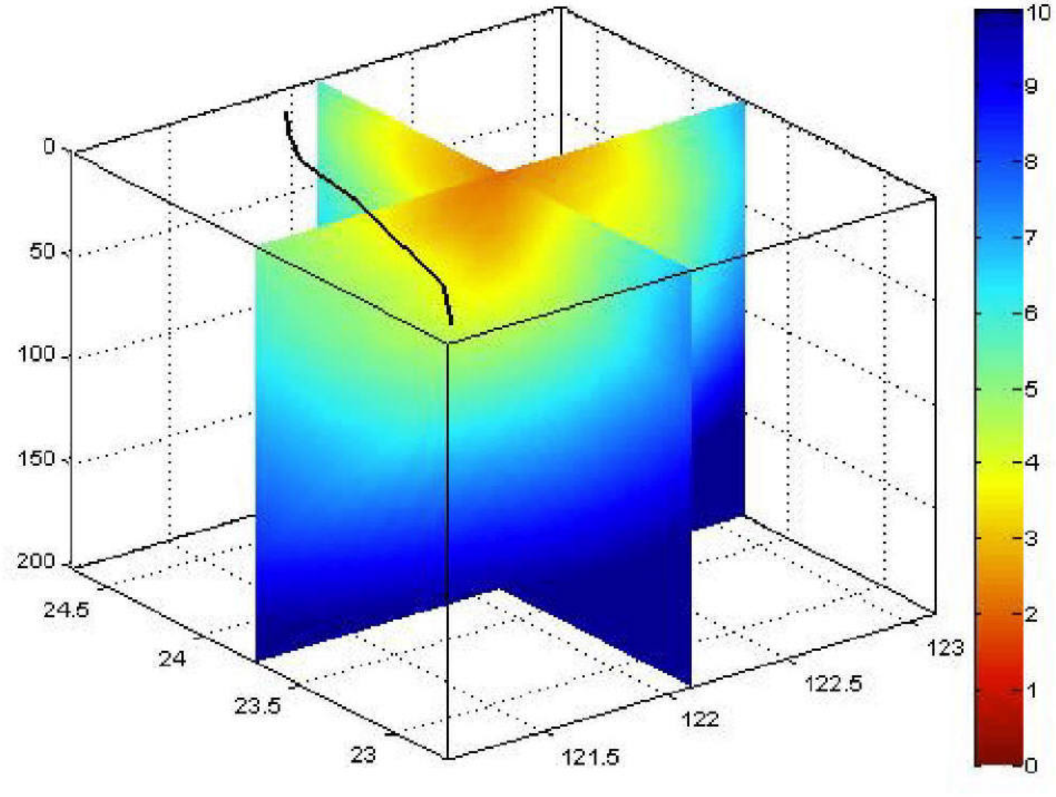


Figure 6. 3-D view of vertical slices through the final solution (see text for explanations).

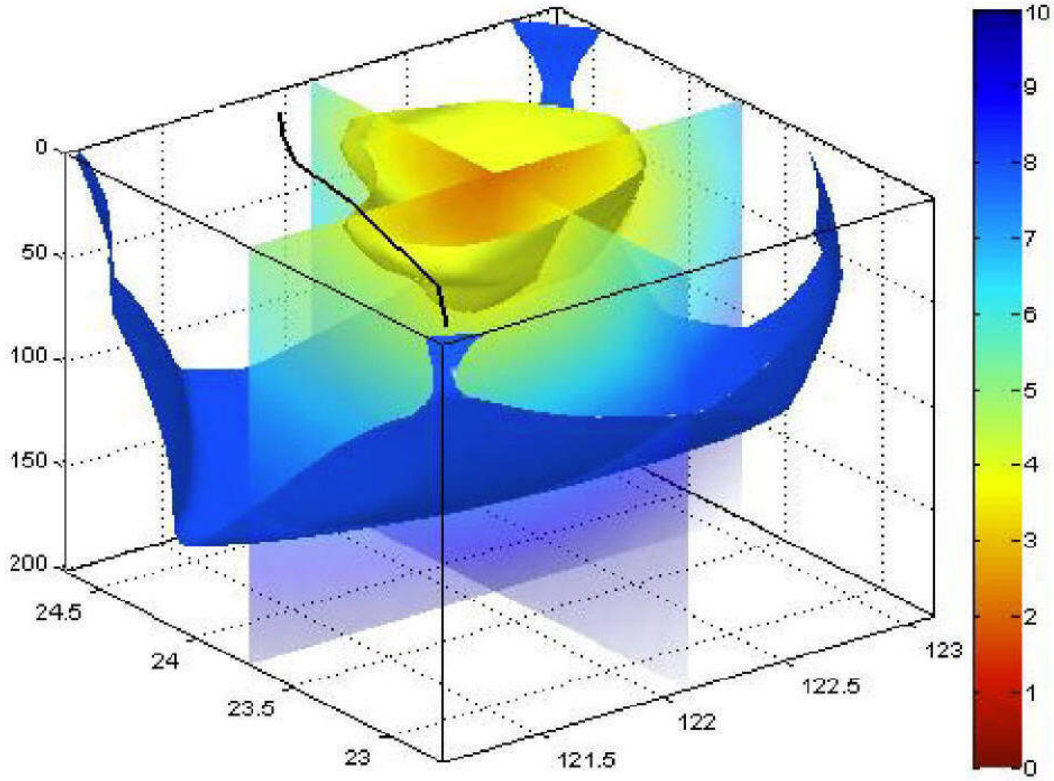


Figure 7. 3-D view of vertical slices and isosurfaces (see text for explanation).