A hypocentre location method for microseismicity in complex regions

Henry C. Bland and Chad Hogan

ABSTRACT

Complex stratigraphy and topology are challenges for conventional hypocentre location. A method is developed which may be used to determine the hypocentre of seismic and microseismic events located within regions of variable topology and stratigraphy. The program requires a velocity model (supplied as a 3-D grid of values), receiver locations (either on the surface or buried), and a set of observed first-arrival pick times. The velocity field is transformed into a set of travel-time cubes, one per receiver location, computed using a fast-marching eikonal equation solver. Candidate hypocentre locations may be anywhere within the search space defined by the velocity field. Each candidate hypocentre location is evaluated according to a cost function. The simplest cost function is the square of the cumulative errors between observed first-break arrivals and the predicted arrival times based on the velocity field. The global minimum for all candidate locations is considered the best estimate of the hypocentre. The global minimum is determined using a pattern search technique. More complex cost functions may be designed which make use of the seismic trace data so as to reduce the reliance on human-picked first break times. The method is illustrated in a short MATLAB program.

INTRODUCTION

Source location is one of the classic problems in seismology. Most source location efforts are focused on earthquake seismology and nuclear test ban monitoring. Recently, there has been increased interest in source location for passive seismic reservoir monitoring and geohazard monitoring. In all cases the problem is the same – determine the time and place of the seismic event based on seismic arrivals at several seismometer stations. Though deterministic solutions exist for regularly-distributed sets of seismometers (e.g. linear arrays), a hypocentre determination using irregularly spaced seismometers typically requires statistical analysis techniques.

METHOD

We present a simple technique which may be used to compute hypocentres within a region of variable topology and stratigraphy. We assume that the study area is monitored by a set of seismometers, and that a multi-channel recording system records a seismic event at a multiplicity of stations. These stations may be positioned irregularly throughout the region of interest. Given a set of N receivers with locations $\mathbf{x}_i$, first-arrival pick times t_i^{pick}, and a 3-dimensional velocity field $\mathrm{v}(x,y,z)$, we aim to locate the hypocentre position in Cartesian space $x_h = \left[x_h, y_h, z_h\right]$ and find the seismic event time T.

The first stage of the method is to convert the velocity field $\mathrm{v}(x,y,z)$ into a set of N traveltime fields $\tau_i(x,y,z)$ computed from the perspective of each receiver i. The velocity and traveltime fields are both represented by 3-dimensional regular grids with a grid spacing Δx. This operation is computationally expensive, and only needs to be performed once for a given receiver geometry and velocity model. The conversion to traveltime is performed using the Fast Marching Method (FMM) 3-dimensional eikonal equation solver (Sethian and Popovivi, 1999). The resulting set of traveltime cubes defines the travel time (associated with the first-arrival) between each receiver location and any of the grid points in the volume of interest.

Next, we obtain the hypocentre location by finding the source location which offers the best match between observed and computed arrival times. The "best match" is determined via the objective function $f(x, y, z)$. This function evaluates to a small number when the match is good, and larger numbers when the match is poor. We iteratively consider candidate hypocentre locations in discrete *(x, y, z)* space until we find the location with the lowest objective function value. This can be performed using an exhaustive search of the grid space, or through the use of a global optimization technique.

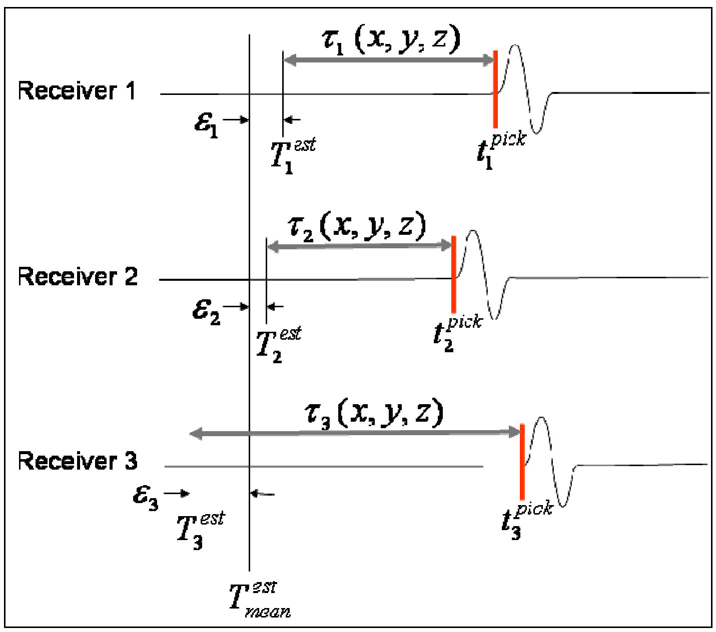

FIG. 1. Illustration time measurements for a set of three receivers.

The objective function $f(x, y, z)$ is computed as follows. For each candidate/receiver combination we compute an estimate of the seismic event time T using

$$T_i^{est} = t_i^{pick} - \tau_i(x, y, z). \tag{1}$$

The mean of these estimates,

$$T_{mean}^{est} = \frac{\sum_{i=1}^{N} T_i^{est}}{N}, \tag{2}$$

provides our best approximation to T for the current candidate location. Next, we compute a series of residuals

$$\varepsilon_i = \left| T_{mean}^{est} - T_i^{est} \right| . \tag{3}$$

Finally, we defined the objective as a function of the residuals: Typically it is either the L1 norm,

$$f(x, y, z) = \sum_{i=1}^{N} \varepsilon_i \ , \tag{4}$$

or the L2 norm,

$$f(x, y, z) = \sqrt{\sum_{i=1}^{N} \varepsilon_i^2} \ . \tag{5}$$

Cross-correlation extension

The algorithm uses a set of pick times which must be selected by careful examination of first arrivals. In some cases, it is very difficult to make-out the first arrival pick times due to noisy or band-limited arrival waveforms. Non-impulsive seismic events also pose a challenge, since they may not have clearly defined first-arrivals. To address these cases, we may consider modifying the objective function to directly interrogate the received waveform data.

VanDecar and Crosson (1990) suggest computing the cross-correlation between all possible pairs of traces to estimate delays and quantify timing uncertainties. After applying a bandpass Chebyshev filter they compute a preliminary arrival time using a single-trace phase-picking algorithm. They go on to compute the trace-to-trace truncated cross-correlation for a pair of traces *i* and j and extract only the peak cross correlation value $\phi_{ij}\left(\tau_{ij}^{\max}\right)$, for the associated delay time $\tau_{ij}^{\max}$. The cross-correlation derived relative delay time is then given by

$$\Delta t_{ij} = t_i^p - t_j^p - \tau_{ij}^{\max} , \tag{6}$$

where t_i^p and t_j^p are the previously auto-picked (approximate) arrival times. The cross correlation coefficient between the *i*th and *j*th is then

$$r_{ij} = \frac{\phi_{ij}\left(\tau_{ij}^{\max}\right)}{\sigma^i \sigma^j} \tag{7}$$

where σ^i and σ^j is the standard deviation of the *i*th and *j*th traces computed over the correlation window located about $\tau_{ij}^{\max}$.

VanDecar and Crosson finally derive a least squares estimation for the estimated pick times for each trace using

$$t_i^{pick} = \tfrac{1}{N}\left(-\sum_{j=1}^{i=1} \Delta t_{ij} + \sum_{j=i+1}^{n} \Delta t_{ji} \right). \qquad (8)$$

We believe that this technique could be incorporated into the calculation of the objective function in Equations 1 through 5, resulting in a method for hypocentre location which does not require hand picking of arrival times.

IMPLEMENTATION

The Fast Marching Method (FMM) of Sethian and Popovici (1999) is designed for orthogonal coordinate systems. It is unconditionally stable which makes it suitable for traveltime computation from complex velocity models: It constructs solutions which are consistent with the exact solution for large gradient jumps in velocity. The algorithm is also able to resolve any overturning propagation wavefronts. Typically, the number of receivers is small and their position is fixed in space. The permanence of the configuration is ideally suited for pre-computing a traveltime cube for each receiver location. The grid size has a large effect on computation time: The FMM algorithm is of the order O(N log N), where N is the total number of grid points. A halving of the grid size increases N by a factor of 8, so careful consideration must be given to both the grid size and the extent of the search space.

Both the FMM eikonal solver and the iterative hypocentre location solver were implemented in Matlab. The FMM eikonal solver runs rather slowly in Matlab due to the presence of large non-vectorized code. A test, executed on a PC with an Athlon 1800+ processor took 60 seconds for a value N=8000 (a cube of 20 elements per side). Even with the inefficiency of Matlab for scalar operations, the algorithm is fast enough to be usable. To further increase the speed of repetitive locations, traveltime cubes are stored to disk for re-use.

The sample implementation uses a “pattern search” method for finding the global minimum objective function value. This method belongs to the family of direct search methods which work without the calculation of a gradient. This is fortunate, as the velocity field is permitted to contain discontinuities.

The algorithm sequentially compares each candidate solution with the “best” obtained up to that time. It uses a strategy for determining the next candidate solution, based on earlier results. A set of fixed direction vectors define the grid search pattern (typically, look north, south, east and west, in the 2-D case). Multiplying these direction vectors by a scalar called the *mesh size* determines the new set of candidate locations. If any of the locations are an improvement, the mesh size grows. If none of the candidates were an improvement, the mesh does not move, but it shrinks.

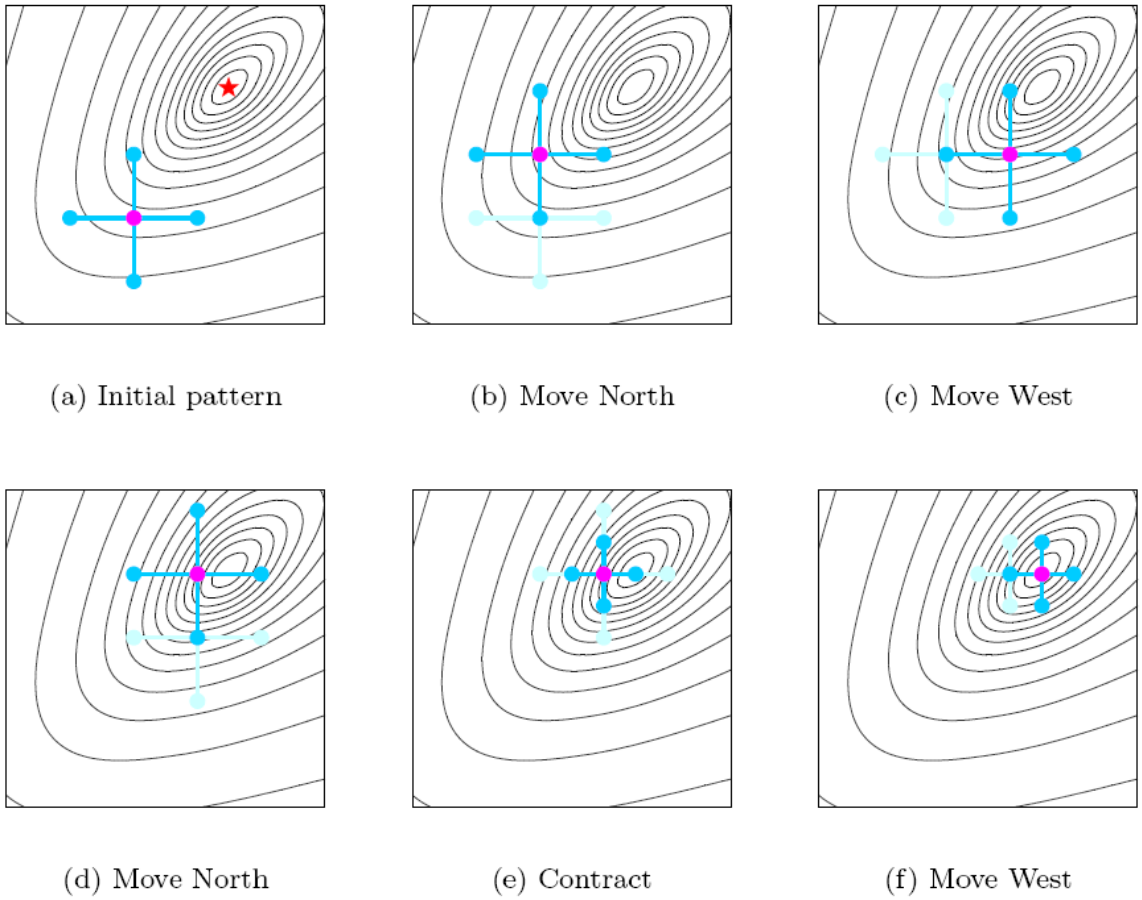

FIG. 2. A sample of the pattern search in action. Here, the global minimum (indicated by a red star in (a)) is obtained through movement and refinement of the pattern search mesh. Illustration after Kolda, Lewis, and Torczon (2003).

Initial tests show that the pattern search works well, particularly if the pattern vectors utilize a maximum positive basis of 2N. In order to obtain the most accurate solution, we allow off-grid interrogation of the traveltime volume with the aid of a 3-dimensional interpolation function (Matlab's *interp3* function).

The structure of the program is such that other optimization techniques may be substituted for the pattern search. Included in the code, is the ability to perform an exhaustive search. It remains untested for large values of N due to its poor speed. One other tested method is the "Genetic Algorithm." Due to this algorithm's complexity, we did not fully evaluate its function in this environment.

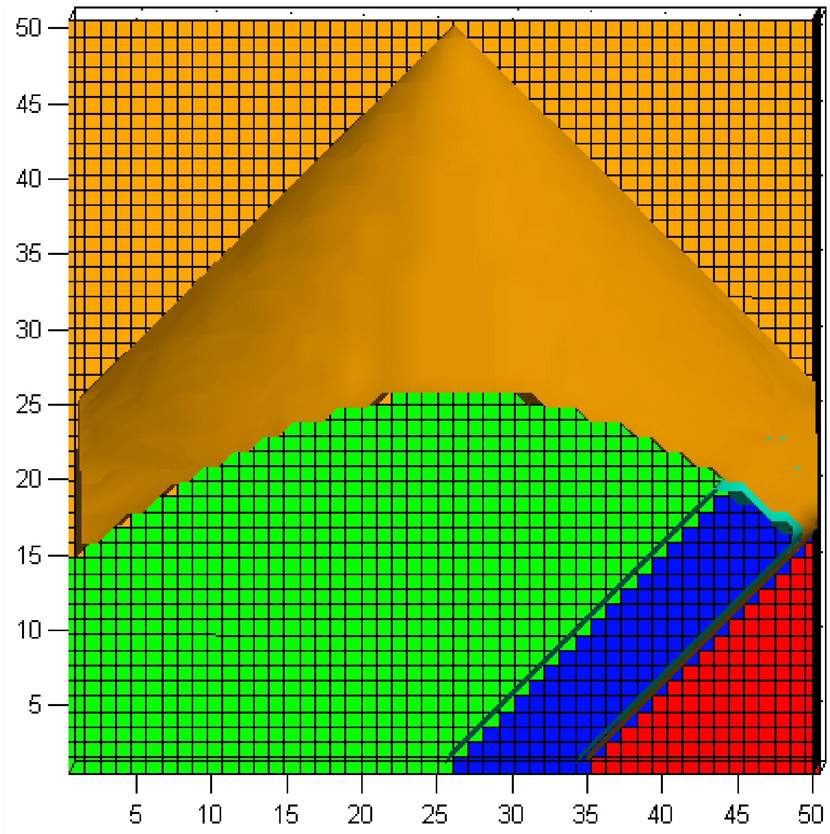

FIG. 2. Tests were performed using a four layer model of a mountain with a thrust fault. The orange region above the surface has air velocity, while the other three layers are assigned appropriate rock velocities.

CONCLUSION

We have presented a technique to compute hypocentres in regions of complex stratigraphy and topology. Based on a preliminary implementation and tests with a synthetic dataset, we believe the method works well. Further testing and validation will be the focus of future work.

REFERENCES

Kolda, T.G., Lewis, R.M., Torczon., V., 2003, Opimization by Direct Search: New Perspectives on Some Classical and Modern Methods: Soc. for Ind and App Mathematics, **45**, No. 3, 385-482

Sethian, J.A., and Popovici, M., 1999, 3-D traveltime computation using the fast marching method, Geophysics, **64**, No. 2, 516-523.

VanDecar, J.,C., and Crosson, R., S., 1990, Determination of teleseismic relative phase arrival times using multi-channel cross-correlation and least squares, Bull of Seis Soc. of America, **80**, No. 1, 150-169.