

事前走時計算をもとにした 3 次元速度構造における迅速な震源決定手法について

関口 渉次*

Quick Hypocenter Determination in Three-Dimensional Velocity Structure, Using the Pre-Calculated Travel Time Data

Shoji SEKIGUCHI

Earthquake Research Division,
National Research Institute for Earth Science and Disaster Prevention, Japan
seki@bosai.go.jp

Abstract

Calculating travel time in three-dimensional velocity structure is the most time-consuming process of hypocenter determination. If we calculate the travel times in advance, it is possible to determine the hypocenter very quickly. We divide the target area into rectangular cells. The travel times between all seismic receivers and all grid points of the target area are calculated. Traveltime and slowness vector at any location within the cell is obtained by interpolating the traveltimes of surrounding grid points. The cell size should be small enough for obtaining precise travel time by the interpolation. We apply this method to the real dataset obtained at the Kanto-Tokai area, Japan. We estimate the cell size. We try HD (Hard Disc drive), SSD (Solid State Device) and memory to store the traveltime data of three-dimensional velocity structure and we find that the speeds of hypocenter determination by HD, SSD and memory are approximately 10, 100, and 1000 times faster than the ordinary hypocenter determination method., respectively.

Key words : Hypocenter determination, Three-dimensional velocity structure, Pre-calculated travel time, SSD

1. はじめに

これまで定常業務的な地震データの処理では、1 次元速度構造を使った震源決定を行うのが一般的であり、3 次元速度構造を用いた震源決定は計算時間が掛かるため実施されてこなかった。3 次元速度構造下での震源決定には、震源と観測点間の走時計算が必要となり、この走時計算の部分が計算全体のほとんどを占める。そこで、走時計算をあらかじめ実施し、その結果を保存しておき、必要なときにそこから走時の値を求めるようにした場合、計算の迅速化が可能になると期待できる。今回の研究では、この考え方にしたがって、実際のデータを用いながら、具体的にどのような手順を踏めば計算速度を短縮することができ、業務的な処理として実行可能かを示す。

2. 方法

走時計算をするために、まず、対象領域を格子状に分割し、各観測点から各格子点までの走時を計算する (図 1)。

以下では分割された矩形領域を cell と呼ぶことにする。対象領域の任意の位置での走時はそれが属する cell の隅にある格子点での走時の値から内挿して得る。震源決定に必要な slowness vector 即ち走時の空間座標についての 1 階微分値も同様に内挿により得る。

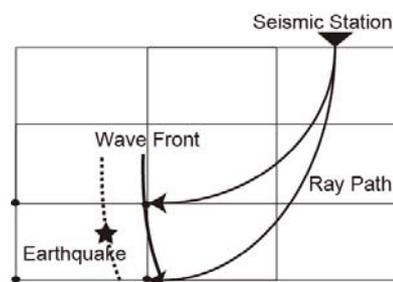


図 1 波線を観測点から格子点まで計算する。

Fig. 1 Rays are traced from the receiver to the grid points.

*独立行政法人 防災科学技術研究所 地震研究部門

走時計算には、2 点間の ray tracing でよく用いられている Um and Thurber (1987) の pseudo-bending method を用いることにした。

走時と slowness vector の内挿には、3 次元の cubic spline 関数を使用することとした (例えば、de Boor, 2001)。これは、節点において 2 階微分値まで連続となる滑らかな関数である。節点を格子点に一致されると、全格子点での走時値から、内挿に必要な spline 関数の係数を算出でき、その数は格子点毎に 8 個となる。保存しておくのは、観測点毎の全格子点でのこの係数データである。

このようにして計算した理論走時と観測走時の残差が最小になるように震源パラメータ 4 個 (時刻, 位置) についての方程式を作り、LAPACK の DGELS サブルーチン (Anderson, et al., 1995) を用いて解くことにより震源決定を行う。得られた震源位置を用いて理論走時を再度計算し、震源再決定を行うというプロセスを何回か繰り返し、走時残差が変化しなくなったら計算を終了する。

3. 格子点間隔

内挿された値が十分な精度を持つためには格子点の間隔は狭い方がよい。一方狭いほど格子点の数が増加し保存すべきデータ量も増加する。そこで、実際の速度構造において、観測点から震源まで直接走時を計算したものと、内挿によって得られたものとを比較して、その誤差が十分小さくなるように格子点間隔を決めることにする。

対象領域は関東東海地方 34°N - 37°N, 136°E - 141°E, 40 km - 140 km とした (図 2)。観測点は旧関東東海地震観測網の 124 点である。比較のために 1 次元速度構造の場合も計算してみる。1 次元速度構造は APE システム (例えば、浜田, 1988) にて震源決定に用いられていた鶴川ほか (1984) のものを使用する。

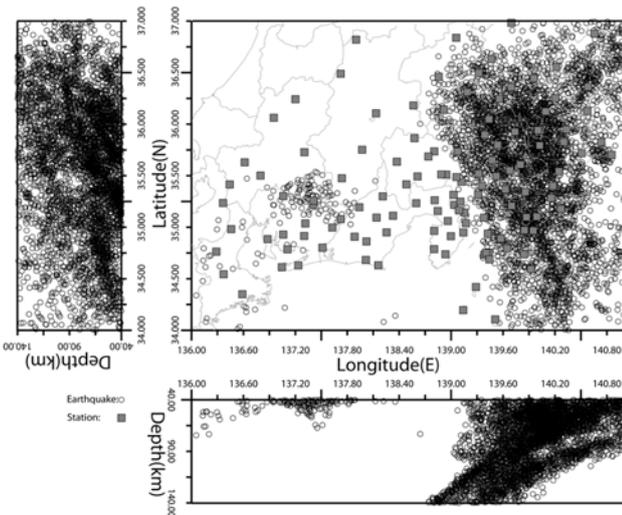


図 2 対象領域の中の地震 (○印) と観測点 (灰色□) の分布

Fig. 2 Distribution of earthquakes (open circles) and seismic stations (grey squares), used in this study.

3 次元速度構造は Sekiguchi (2006) において得られたモデルを用いることとした (図 3)。関東東海地震観測網の走時データを用い、波線の方向と密度に応じて速度内挿多項式の次数を変えられるようにして求めたものである。具体的には、cell size は 0.125° x 0.125° x 10 km で、damping parameter:1, 最適 AIC モデルである。

格子点間隔を変えながらこれらの速度構造について、観測点から cell の中心点まで直接計算した走時と、周囲の格子点の値から内挿して得られた走時の値の差を表 1 に示す。1 次元速度構造では、図 2 に示した領域全域に対して計算したが、3 次元速度構造では、計算量が膨大になるため 35.5°N - 36.0°N x 139.5°E - 140.0°E x 40 km - 65 km の領域に限ることとした (図 3 の太線矩形領域)。3 次元速度構造で pseudo-bending method による ray tracing に失敗する場合がいくつかあったが、その場合でも ray path の初期位置を 1 次元速度構造で得られたものとして計算を開始するとすべて成功した。

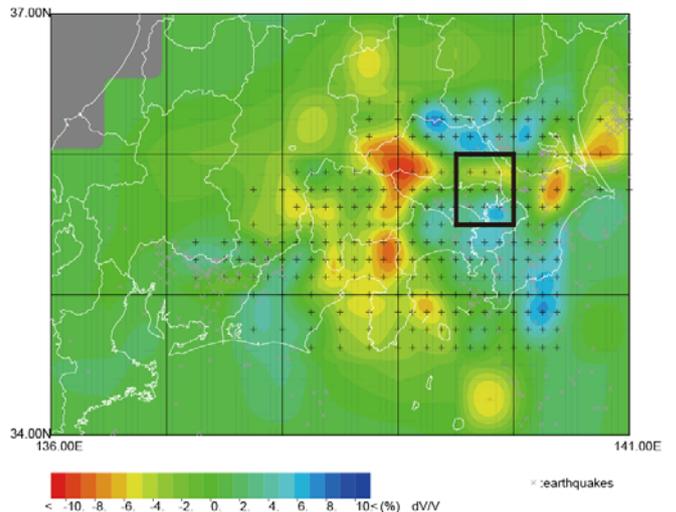


図 3 Sekiguchi (2006) による 3 次元速度構造の深さ 40km での断面図。太枠四角で囲まれた領域は格子点間隔を見積もるために使用した範囲を示す。

Fig. 3 Three-dimensional velocity distribution at the depth of 40km (Sekiguchi, 2006). The square area trimmed with bold line indicates the area where the cell size is examined.

表 1 格子点間隔 (cell size) 毎の走時の内挿誤差

Table 1 Interpolated travelttime errors of each cell size.

	Cell Size	Interpolation Error (sec)
1D Velocity Structure	0.25° x 0.25° x 25 km	0.0206
	0.125° x 0.125° x 12.5km	0.0068
3D Velocity Structure	0.125° x 0.125° x 12.5km	0.0277
	0.05° x 0.05° x 5km	0.0116

1 次元速度構造と 3 次元速度構造を比較すると、3 次元速度構造の方が、同じ格子点間隔では大きな誤差となっている。地震波形のサンプリングは 100 Hz 前後のものがほとんどと思われるので、誤差は 0.01sec 程度以下であれば十分である。即ち、1 次元速度構造では、格子点間隔 0.125° x 0.125° x 12.5 km、3 次元速度構造では、0.05° x 0.05° x 5 km である。

4. 実データへの適用

ここで使用する地震観測点と地震を図 2 に示す。地震の数は 4,392 個である。なお、この図に示してある地震の位置は APE システムにおいて決定されたものである。今回の研究では、到着時刻読み取り値は P 波のみとし、1 地震当たりの平均 P 波読み取り個数は約 40 個となっている。

震源決定の際の初期震源は、1 次元速度構造の場合は、最も早く地震波が到達した観測点の直下で深さは対象領域の中心（今回の場合は 90 km）とした。3 次元速度構造の場合は、APE システムで決定された位置を初期震源とした。O-C の残差について繰り返し計算の前回との差が 0.01sec 以下になったら収束したと判断し計算を終了させる。

計算環境は、計算機：DELL Precision T5400 (Intel Xeon 2.66GHz, SATA HD), OS: Linux (Fedora 10 and 11), Fortran Compiler: GNU gfortran である。保存する媒体については、今回は 3 次元速度構造の場合正味のデータ量は約 1GBytes (倍精度) なので計算機メモリに保存できた。即ちプログラム内の配列変数として扱えた。実際にはもっと広い領域や多くの観測点の場合があるので、容量の大きい外部記憶媒体の利用も考えなくてはならない。そこで、HD (Hard Disc drive) の場合、SSD (Solid State Device) の場合の 2 通りについても計算を実行した。SSD はランダムリード (512KB) 時 241.7MB/sec の転送速度を持つとして販売されている現時点では入出力の早い部類の製品を使った。外部記憶装置の場合、Linux の ext2 ファイルシステムを利用して

・ /cell 緯度/cell 経度/cell 深さ/ファイル
というディレクトリ構造でデータを binary で保存した。一つのファイルから必要な全データが取得できるようにファイル内には cell の 8 格子点の全観測点についてのデータが格納されている。即ち正味のデータ量の 8 倍の量を読み込みの迅速化のために重複して保存している。

以上のようにして計算した場合の 1 地震当たりの処理時間を表 2 に示す。参考のために pseudo-bending method を使った通常の計算方法で震源決定した場合の処理時間も示してある。3 次元速度構造の場合、通常方法に比べて処理時間は大雑把に言って HD, SSD, memory の順に 1/10, 1/100, 1/1000 と早くなっている。HD を使っているのはそれほど処理速度が速くならない。SSD と memory の場合、1 次元速度構造の場合に匹敵する速さが実現できている。

日本全国で 1 日およそ 100 個の地震が発生しているの

で、1 年分 109,500 個の地震を処理するとした場合、この結果を使って処理時間を概算してみると、通常方法では 54 時間、SSD では 24 分、メモリでは 2 分 34 秒かかると見積ることが出来る。

表 2 各保存装置に対する 1 地震当たり処理時間

Table 2 Mean process time of hypocenter determination of each data storage device.

保存装置	1 次元速度構造	3 次元速度構造
HD	0.0678 (1/8)	0.3155 (1/6)
SSD	0.01486 (1/36)	0.01326 (1/134)
Memory	0.001672 (1/317)	0.001402 (1/1270)
(通常の計算法)	0.5301 (1)	1.7807 (1)

単位：秒 (但し、括弧内は通常の計算法に対する相対値)
Unit: second (relative values to those of ordinary traveltimes calculation shown in parentheses).

今回の例のように走時を計算するための保存すべきデータ量が少なくメモリ上に収まるならばメモリを使用するのが最善である。データ量が増えメモリ上に収まらなくなった場合は、SSD を利用できる。現在 100GB 程度のもので比較的安価に入手できるので、その容量までは納められる。さらにデータ量が増え SSD に収まらなくなった場合は、容量的には HD しか使えないが、HD の場合処理速度が SSD ほど早くないので、むしろ対象領域を分割しそれぞれのデータ量を 1 台の SSD に収まるようにし複数の計算機で処理をするなど工夫した方がよいだろう。

5. 議論

データ保存方法として、データベースソフトを利用することが考えられる。そこで、SQLITE3 というデータベースソフトを使ってデータをデータベース化し HD に保存してみた。検索を高速化するために緯度、経度、深さに関して index を作成した。しかしながら、処理速度は HD と同程度だった。

Linux には tmpfs というメモリ上に仮想ファイルシステムを作成する機能がある。通常のファイルシステムと同様に扱える利便性がある上にメモリ上に保存するのでメモリ並みの処理速度を期待したが、実際は SSD と同程度の速さであった。

既に述べたように P 波走時の事前計算には pseudo-bending method を使った。計算時間は、1 次元速度構造で 1 時間 14 分、3 次元速度構造で 3.4 日掛かっている。一度計算しておけばよいので、この程度の処理時間は許容できると思われる。

一度事前に計算しておくことの利点として以下のことが挙げられる。今回のデータでは全ての ray path について計算に成功したが、複雑な構造の場合には、失敗する場合も出てくるであろう。そのような時でも、個々の事例について原因を調べ計算し直す時間があるので可能な限りの走時データを事前に用意するこ

とが出来る。その結果、震源決定の時に走時データの不足あるいは誤りによって失敗する頻度が低くなり、定常業務的な処理により適したものとなる。一方、通常の計算方法では、その都度対応する、個々の事例についてプログラミング内で対処できるようにする等、非常に煩雑になる。

pseudo-bending method は領域全体を隈無く走査しているわけではないので複雑な3次元不均質構造においては、走時の local minimum に陥っている可能性は常に存在している。それを回避するには、差分法(例えば Vidale,1990) や最短経路法(例えば Moser,1991) など根本的に違った原理に基づく手法を適用するしか方法はない。これは、今後の検討課題であろう。

ray tracing には、paraxial ray approximation を使った方法(Červený *et al.*, 1984) もある。この方法では波面が2次曲面で表現されているので、曲面を表すパラメータを保存すれば spline 関数を使うことなく直接外挿により領域内の任意の点での走時が算出できる。そこで、今回の領域について、この方法を使って走時計算を試みた。しかしながら、3次元速度構造の場合、全波線の1割程度について収束せず計算に失敗した。pseudo-bending method が two-point ray tracing なのに対して、この手法は ray shooting method の1種なので成功率が低くなったと考えられる。

6. まとめ

今回の手法では、領域を分割し各格子点まで事前に走時計算をし、震源決定の際にはそこからの内挿によって任意の位置の走時を算出するとしている。十分な精度の走時を得るには、格子点間隔を、今回のデータセットの場合、1次元速度構造では $0.125^\circ \times 0.125^\circ \times 12.5 \text{ km}$ に、3次元速度構造の場合には、 $0.05^\circ \times 0.05^\circ \times 5 \text{ km}$ にすれば良いことがわかった。

事前計算した走時データを保存するために、メモリあるいは SSD を利用できる。そうすれば通常の計算方法に比べて1,000倍あるいは100倍迅速な3次元速度構造下の震源決定が可能になることがわかった。

また、事前に計算することにより、より完全な走時データセットの作成が容易になり震源決定処理中の失敗の頻度が下がることが期待できる。

したがって、このような方法をデータ処理システムの定常業務として実施することは十分可能と思われる。

謝辞

読者によるコメントは本論文を改善する上で有益であった。記して感謝する。

参考文献

- 1) Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen (1995) : LAPACK user's guide, second edition, 325 pp., Society for Industrial and Applied Mathematics, Philadelphia.
- 2) Červený, V., L., Klimeš, and I. Pšenčík (1984) : Paraxial ray approximations in the computation of seismic wavefields in inhomogeneous media. *Geophys. J. R. Astron. Soc.*, **79**, 89-104.
- 3) de Boor, C. (2001) : A practical guide to splines, revised edition (Applied Mathematical Sciences; 27), 346 pp., Springer-Verlag, New York.
- 4) 浜田和郎 (1988) : 地震前兆解析システム開発の背景とその意義. 国立防災科学技術センター研究報告, No.41, 31-34.
- 5) Moser, T.J. (1991) : Shortest path calculation of seismic rays. *Geophysics*, **56**, 59-67.
- 6) Sekiguchi, S. (2006) : Hierarchical traveltimes tomography. *Geophys. J. Int.*, **166**, 1105-1124.
- 7) 鶴川元雄・石田瑞穂・松村正三・笠原敬司 (1984) : 関東・東海地域地震観測網による震源決定方法について. 国立防災科学技術センター研究速報, No.53, 1-88.
- 8) Um, J., and C.H. Thurber (1987) : A fast algorithm for two-point seismic ray tracing. *Bull. Seismol. Soc. Am.*, **77**, 972-986.
- 9) Vidale, J.E. (1990) : Finite-difference calculation of traveltimes in three dimensions. *Geophysics*, **55**, 521-526.

(原稿受理：2010年3月4日)

要 旨

3次元速度構造において震源決定をする際に走時を計算する過程が最も時間を消費する。もし事前に走時を計算しておけば震源決定は非常に迅速になる可能性がある。本研究では対象領域を矩形領域 (cell とよぶ) に分割し、観測点から格子点まで走時を計算する。cell 内の走時および slowness vector は cell の隅の 8 点の走時の値から内挿することより取得する。格子点間隔は十分な精度で内挿した走時が得られるように短くなければならない。この方法を関東東海地域で得られた実際の走時データに適用した。3次元速度構造の場合について事前計算した走時データをハードディスク, SSD, メモリにそれぞれ保存して、震源決定を行った。その結果、処理速度は通常の計算方法に比較して、ハードディスク, SSD, メモリの順に 10 倍, 100 倍, 1,000 倍と早くなることがわかった。

キーワード：震源決定, 3次元速度構造, 事前計算した走時, SSD