

津波発生伝播の大規模3次元シミュレーション

古村孝志

東京大学大学院情報学環総合防災情報研究センター／地震研究所

1 はじめに

津波は、地震により生まれる海底面の地殻変動が海水を持ち上げ・下げし、これが重力を復元力に持つ重力波として海面を伝播する現象である。マグニチュード (M) 8クラスの巨大地震により生まれる海水面の変動は高々1～2メートル程度にすぎないが、200×100平方キロメートル以上に広がる広大な海面の盛り上がりや津波として沿岸に押し寄せると、5～10メートルを超える大津波に成長する。

海に囲まれた日本では、繰り返し津波の被害を受けてきた。たとえば、1896年明治三陸地震 (M8.5) では、20～30 mを超える巨大津波が東北～北海道の太平洋岸に襲来し、2万人以上の犠牲者が出た。ちなみにこの地震は、強い揺れを伴わずに突然津波が襲来する「津波地震」であった。その後、100名を超える犠牲者を出す津波災害として、1933年昭和三陸地震、1944年東南海地震、1946年南海地震、1983年日本海中部地震、1993年北海道南西沖地震がある。1960年のチリ地震 (M9.5) では、地球の裏側で生まれた津波が1日かけて日本の太平洋沿岸に襲来し、北海道から沖縄にかけて死者・行方不明者142名を出す大惨事となった。チリ北西部では2010年3月にもM8.8の地震が発生し、日本の太平洋沿岸に17年ぶりに大津波警報が発表された。

今後30年以内には、駿河湾から日向灘にいたる南海トラフにおいて、東海・東南海・南海地震の発生が心配されており、その規模はM8.0～8.7になる。房総半島から日向灘にいたる広い範囲では、近いところでは地震から5分後に5～10 mを超える大津波が押し寄せる可能性が高い。こうした巨大地震による津波予測の高精度化と災害軽減に向け、津波シミュレーションの高度化が急務である。これには津波の発生伝播を支配する地下構造と地震断層モデルの高精度化とともに、大規模並列シミュレーションの実用化に期待するところが大きい。

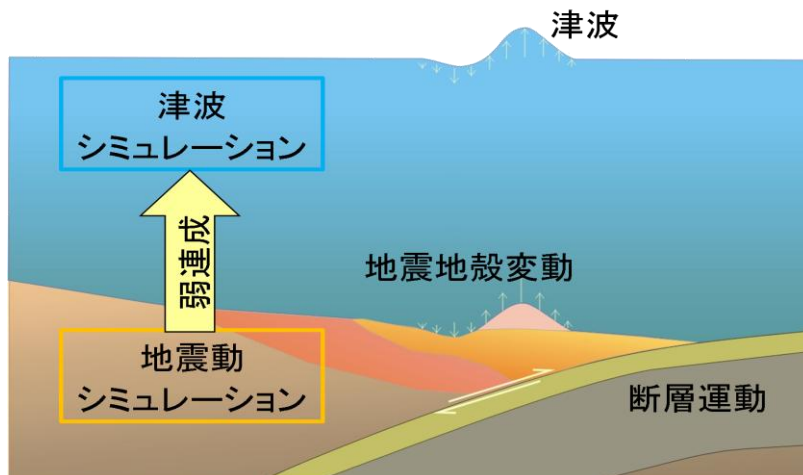


図1 地震—津波連成シミュレーションの模式図。

表1 近年の日本の津波被害。

発生年	地震名	M	死者・行方不明者数
1896 (明 29)	明治三陸地震	8.5	21,259
1933 (昭 8)	昭和三陸地震	8.1	3,064
1944 (昭 19)	東南海地震	7.9	1,223
1946 (昭 21)	南海地震	8.0	1,443
1960 (昭 35)	チリ地震	9.5	142
1983 (昭 58)	日本海中部地震	7.7	104
1993 (平 5)	北海道南西沖	7.8	230

2. 津波シミュレーション

津波シミュレーションでは、まず地震の震源断層モデルと地下構造モデルを用いて海底地殻変動量を計算し、次に複雑な海底地形において発生・伝播する津波を計算する。

地震断層運動による地震波の発生・伝播と地殻変動の発生は、以下の運動方程式と応力-歪みの構成方程式を用いて評価することができる：

$$\begin{aligned} \rho \ddot{u}_x &= \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + f_x, \\ \rho \ddot{u}_y &= \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + f_y, \quad (1) \\ \rho \ddot{u}_z &= \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + f_z. \end{aligned}$$

$$\sigma_{pq} = \lambda(e_{xx} + e_{yy} + e_{zz})\delta_{pq} + 2\mu e_{pq}, \quad (p, q = x, y, z). \quad (2)$$

また、歪みは以下で表される：

$$\begin{aligned}
e_{xy} &= \frac{1}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right), e_{xz} = \frac{1}{2} \left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} \right), \\
e_{yz} &= \frac{1}{2} \left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \right), \\
e_{xx} &= \frac{\partial u_x}{\partial x}, e_{yy} = \frac{\partial u_y}{\partial y}, e_{zz} = \frac{\partial u_z}{\partial z}.
\end{aligned} \tag{3}$$

地震発生から時間が経過し、地震波が広がった後には震源断層のまわりに永久変位が残り、特に震源が浅い場合には地表面（海底面）が大きく隆起・沈降する。こうして生まれた海底面の変動は、海水を押し上げ・下げし、海面が上昇・沈降する。震源域の上に生じた海水の盛り上がり・下がり、やがて津波として伝播を開始する。こうした津波の発生・伝播は、地震動シミュレーションから求められた海底面の変動を入力として、以下の3次元ナビエーストックス式：

$$\begin{aligned}
\frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} + u_z \frac{\partial u_x}{\partial z} &= -\frac{\partial p}{\partial x} + \nu \left[\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right] + g_x \\
\frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_y}{\partial z} &= -\frac{\partial p}{\partial y} + \nu \left[\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \right] + g_y \\
\frac{\partial u_z}{\partial t} + u_x \frac{\partial u_z}{\partial x} + u_y \frac{\partial u_z}{\partial y} + u_z \frac{\partial u_z}{\partial z} &= -\frac{\partial p}{\partial z} + \nu \left[\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \right] + g_z,
\end{aligned} \tag{4}$$

および連続の式：

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0, \tag{2}$$

を用いて評価することができる。

地震と津波のシミュレーションの連成は、海底面の時間変動（地動速度）の境界条件を共有することにより実現する。まず、地震動計算(1)-(3)から求められた海底地殻変動の時間変化（地動速度）を求め、これを用いて津波計算における海底面直上の海水の流速を置き換える（図1）。ここでは、地震から津波への一方向の連成（弱連成）を考え、津波が地震動や地殻変動に与える影響は小さいので考えない。これら運動方程式とナビエーストックス式は差分法に基づき陽的に計算する。

2. 大規模地震—津波シミュレーションの課題

運動方程式とナビエーストックス式の3次元差分法計算には大きな計算資源を必要とするため、これまで主として地球シミュレータ（ES）等のベクトル型計算機を用いて行ってきた。特に、運動方程式の差分法計算はベクトル計算機の演算機構に良く適合し、長いベクトル長を持つベクトルレジスタと高いメモリバンド幅により、特にプログラム書法を意識しなくても理論性能の50%を超える実効性能を達成することができた。

ところが、近年の超並列計算機を代表各であるスカラー型計算機は、高いCPU性能に対してメモリバンド幅が狭く、その性能を引き出すためには特別なコードチューニングが不可欠である。これは、スカラー型計算機だけの問題ではなく、近年のベクトル計算機にも当てはまる。

たとえば 2008 年に更新された新型地球シミュレータ (ES2) では、CPU 性能が前機種 (ES) の 12.8 倍に高まった一方で、メモリバンド幅が ES の 4 Byte/FLOPS から ES2 では 2.5 Byte/FLOPS にまで大きく低下し、地震動シミュレーションの実効性能も 30 %以下にまで低下した。

本研究では、大規模地震—津波シミュレーションの実現に向け、運動方程式の差分法計算を対象に、ベクトル型計算機 (ES2) とスカラー型計算機 (T2K) の演算性能の調査と、そして二つのタイプの計算機を対象としたキャッシュメモリ利用と MPI チューニングを進めた。

2-1. ベクトル型計算機とスカラー型計算機

キャッシュメモリが差分法計算性能に与える影響を見るために、地震動シミュレーションコード (Seism3D) についてモデルサイズと演算速度 (GFLOPS 値) の関係を調査した (図 2)。なお、計算は ES と T2K の 1 CPU を用いて行った。

ES では、モデルサイズが大きくなり、ベクトル長の増長とともに演算速度が加速し、やがて一定の演算速度 (およそ 3.4 GFLOPS) に落ち着く。このときの実効性能は ES の理論性能 (8 GFLOPS/CPU) の 42.5 %である。一方、T2K ではモデルサイズの増大とともに演算速度が低下する逆の傾向が出る。最小モデルの計算では 4.1 GFLOPS の演算速度が得られ、T2K の理論性能 (9.2 GFLOPS/CPU) を考えると実効性能は 44.5 %になる。しかし、大モデルでは 0.5 GFLOPS に低下し、実効性は 5.4 %と見積もられる。T2K では、モデルサイズが小さい場合には、その大部分がキャッシュメモリ内に収まり、CPU との高速データアクセスが可能である。しかし、通常の地震動計算では、1 CPU あたりのメモリ使用量は数百 MB になり、わずか数 MB のキャッシュメモリにはとうてい収まらない。モデルサイズの増大とともにキャッシュ溢れが起き、メインメモリとのアクセスが増えるために計算速度が急激に低下する。

T2K のメモリバンド幅は 0.2 Byte/FLOPS 程度と考えられ、CPU とメインメモリとのアクセスには数十ウェイトの待ち時間が生じるなどペナルティが大きい。したがって、この限られたキャッシュメモリを最大限に活用するコードチューニングが T2K の性能を引き出す鍵となる。

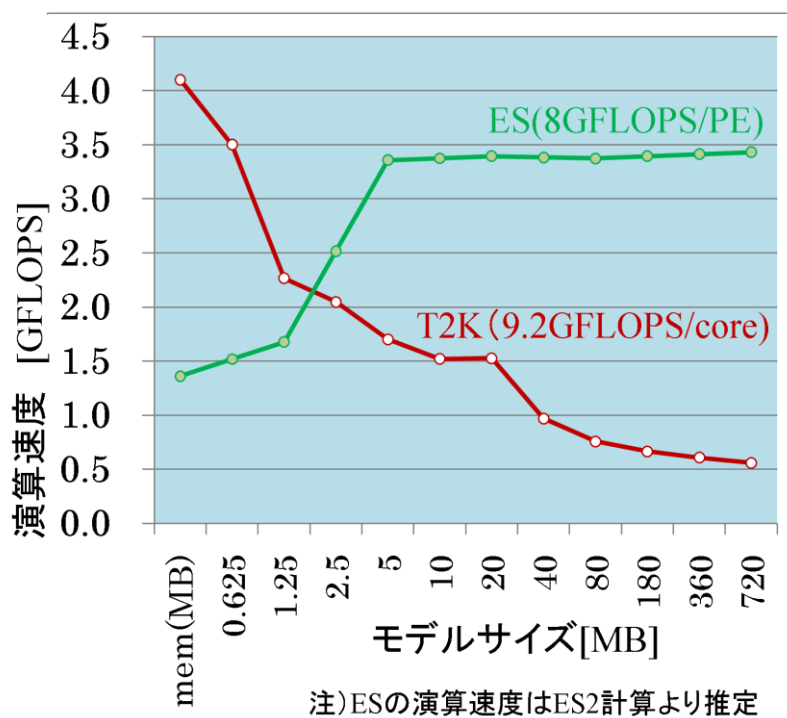


図2 地震動シミュレーションコードにおけるモデルサイズと演算速度の変化。ES と T2K の比較。

2-2. 最適化 Fortran コンパイラの性能比較

T2K では純正の Hitachi 製 Fortran コンパイラのほか、Intel 製コンパイラと PGI 製コンパイラの3つが利用できる。これらの最適化性能を比較するために、Seism3D の1CPU 演算速度をモデルサイズを変えて比較した(図3)。同一プログラムでも、使用するコンパイラにより演算速度が最大3割程度も大きく異なること、その傾向はモデルサイズによっても大きく異なることがわかる。たとえば、小さなモデルでは Intel コンパイラが優位であるが、大きなモデルでは Hitachi コンパイラや PGI コンパイラの性能が高い。

このように、T2K での演算性能はコンパイラ最適化処理に大きく依存していることがわかる。同時に、各社のコンパイラ性能の大きなばらつきを考えると、最適化にはまだ改善の余地があり、現時点では T2K の CPU 性能を十分に出し切れていない感じもする。なお、本テストは 2008 年 5 月に行ったものであり、それ以降のコンパイラのバージョンアップにより状況が改善されている可能性もある。

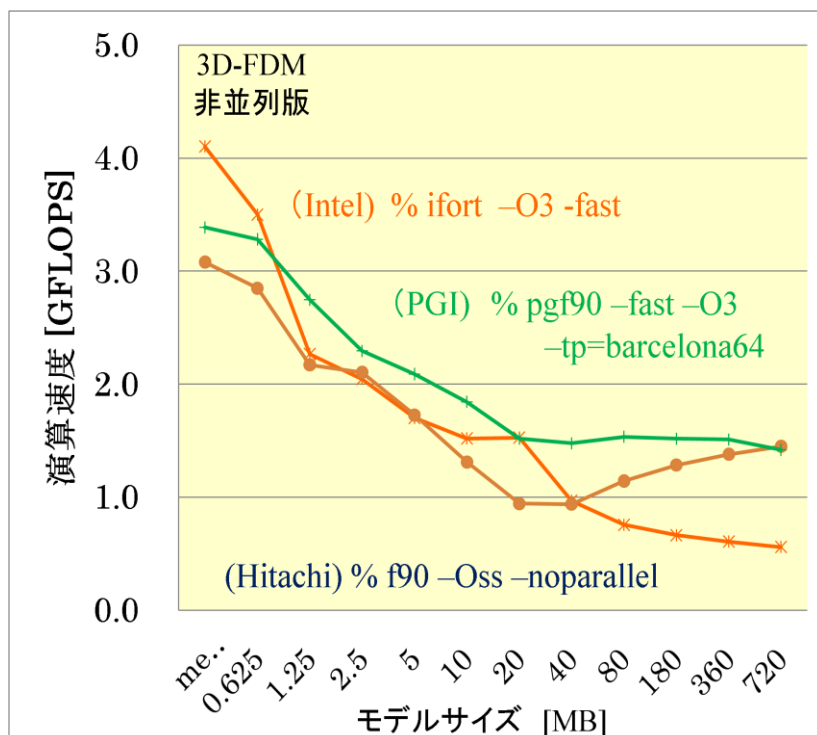


図3 地震動シミュレーションコードのコンパイラによる演算速度の比較。

3. T2K 向けコードチューニング

以上より、スカラー型計算機の性能を得るためには、キャッシュメモリ利用の最適化に向けたコードチューニングが必須であり、かつ最適化コンパイラに頼るだけでなく計算順序の変更やアルゴリズム自体の変更など、コードの大幅な修正が必要である。

このような観点から進めた T2K 向けチューニングの結果を、同様の作業を ES2 に対して行った結果とあわせて比較する。

3-1. 演算の結合

T2K の狭いメモリバンド幅の問題をカバーするために、運動方程式の差分法計算におけるメインメモリからのデータロード回数の節約を考える。例えば、方程式(3)における変数 u_x に対する差分法演算は $\partial u_x / \partial x, \partial u_x / \partial y, \partial u_x / \partial z$ の3つが必要である。これまでの計算では、3つの計算をサブルーチンコールにより行い、その都度メモリから u_x をロードして差分法演算を行ってきた。いっぽう、 u_x をメモリからロードした後に、 $\partial u_x / \partial x, \partial u_x / \partial y, \partial u_x / \partial z$ の3つの計算を連続して行なうことにより、ロード回数を3回から1回に節約することができる(図3)。こうして、3つのサブルーチンを一つに結合した結果、Seism3Dの演算性能が4%高速化(本微分演算部分だけでなくSeism3D計算全体にかかる時間。以下同様。)できた(表2)。なお、本チューニングの効果はES2ではより大きく計算が14%高速化した。

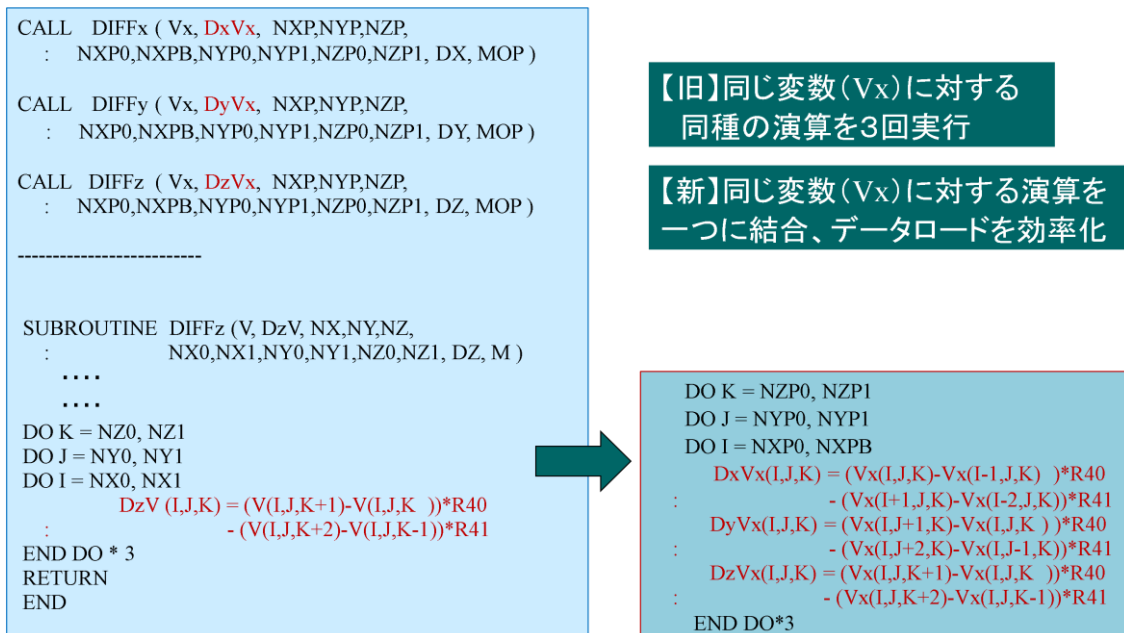


図3 3回の微分演算の結合。

	修正前[s]	修正後[s]	性能比
T2K	650.8	624.8	1.04
ES2	7.2	7.1	1.14

表2 3回の微分演算の結合による計算時間と演算性能の変化(T2KとES2)。

3-2. 変数の結合と共有化

歪みの式(4)では微係数 $\partial u_x / \partial y$ と $\partial u_y / \partial x$ の二つが足しあわされた形で用いられ、これらの微係数は他の運動方程式の計算において単独で用いられることはない。そこで、二つを最初か

ら1変数にまとめることにより、今後データロード回数を2回から1回に減らすことができる(図4)。このような結合可能な変数は6組あり、T2Kにおいて演算性能をさらに16%改善させることができた。いっぽう、ES2での性能向上は4%にとどまり、先に述べたチューニングとともに、計算機により効果は大きく異なる結果となった(表3)。

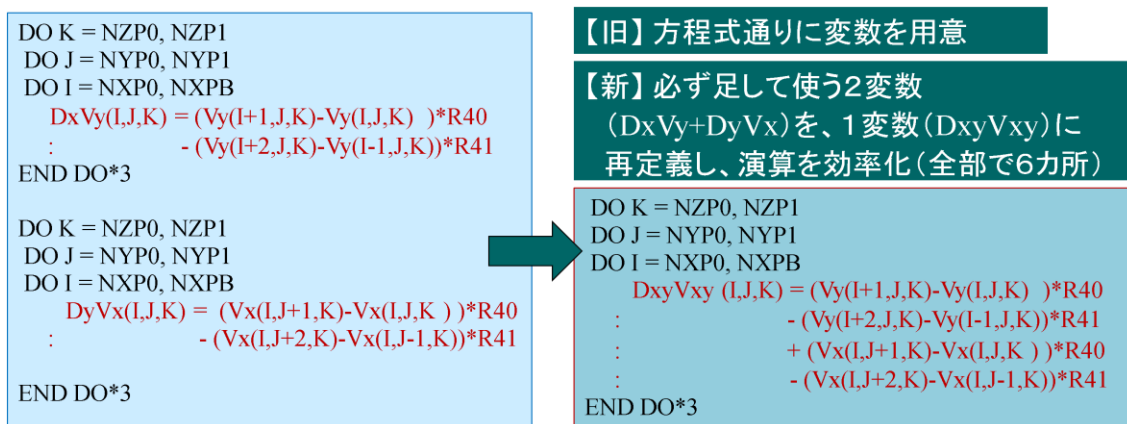


図4 変数の結合と共有化。

	修正前[s]	修正後[s]	性能比
T2K	624.8	536.7	1.16
ES2	7.2	6.9	1.04

表3 変数の結合と共有化による演算時間と性能の比較。

3-3. MPI 通信バッファの記述

領域分割に基づく並列計算では、各CPUに割り当てられた小領域の周囲(袖領域)のデータを隣接するCPUとMPI通信を用いて交換する必要がある。このとき、袖領域のデータはメモリ上には必ずしも連続して配置されていないため、いったん通信バッファにコピーしてからMPI_ISEND/IRECV関数を用いた送受信が必要になる。

この操作のために、これまでMPI2に用意されているMPI_TYPE_CREATE_SYBARRAY()等の関数を用いて「派生データ型宣言」を行い、この型を用いて、飛び飛びに並んだ袖領域内のデータを通信バッファにコピーする作業を行ってきた(図5)。こうして自ら通信バッファにパック/アンパックする手間が省け、プログラムにおいて本作業を隠蔽する効果があり非常に便利な機能ではあるいっぽう、MPI内部でどのような処理が行われているかが不明である不安があった。

試しに、本関数を用いずに、通信バッファのパック/アンパック操作を自らプログラミングしたところ、計算時間が4%短縮した。また、ES2では16%の性能向上が見られた(表4)。プログラムが長くなるなど助長性が増すが、複数の通信バッファをひとまとめにするなど細かい操作も可能になる。


```

CALL MPI_TYPE_CREATE_SUBARRAY(3, ISIZE, ISUBSIZE,
    ISTART, MPI_ORDER_FORTRAN, MPI_REAL, ITYPE_K, IERR)
CALL MPI_TYPE_COMMIT(ITYPE_K, IERR)
-----
CALL MPI_ISEND ( Vx, 1, ITYPE_K, KUP, 1,
    MPI_COMM_WORLD, ITAG1, IERR )

```

**【旧】MPI2の「派生データ型」宣言
 を利用した通信バッファ操作**

```

DO K = 1, NL2
DO J = 1, NYP
DO I = 1, NXP
    SBUFF_x (I,J,K) = Vx (I,J,NZP-NL2+K)
END DO
END DO
END DO
IBSIZE = NXP*NYP*NL2

CALL MPI_ISEND ( SBUFF_x, IBSIZE, MPI_REAL,
:   KUP, 1, MPI_COMM_WORLD, ITAG1, IERR )

```

**【新】通信バッファを自作、パック・
 アンパックを明示的に記述**

図5 派生データ型を用いた通信バッファの利用と、通信バッファの明示的記述。

	修正前[s]	修正後[s]	性能比
T2K	650.8	624.8	1.04
ES2	9.7	8.3	1.16

表4 通信バッファの明示的記述による演算時間の短縮効果。

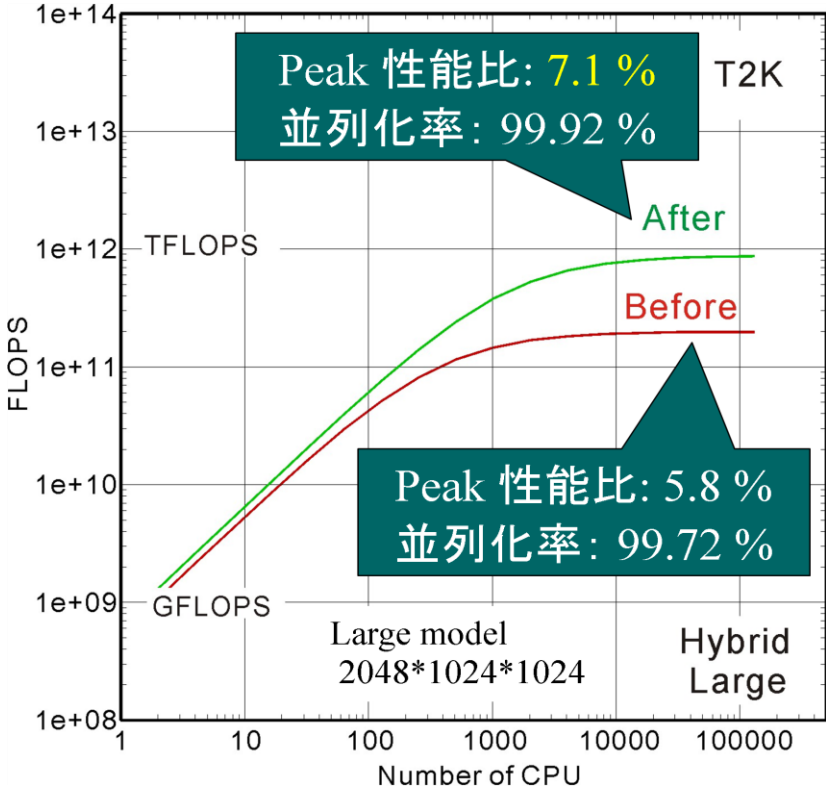


図6 T2Kにおける地震波動伝播シミュレーションの並列計算性能。チューニング前と後。

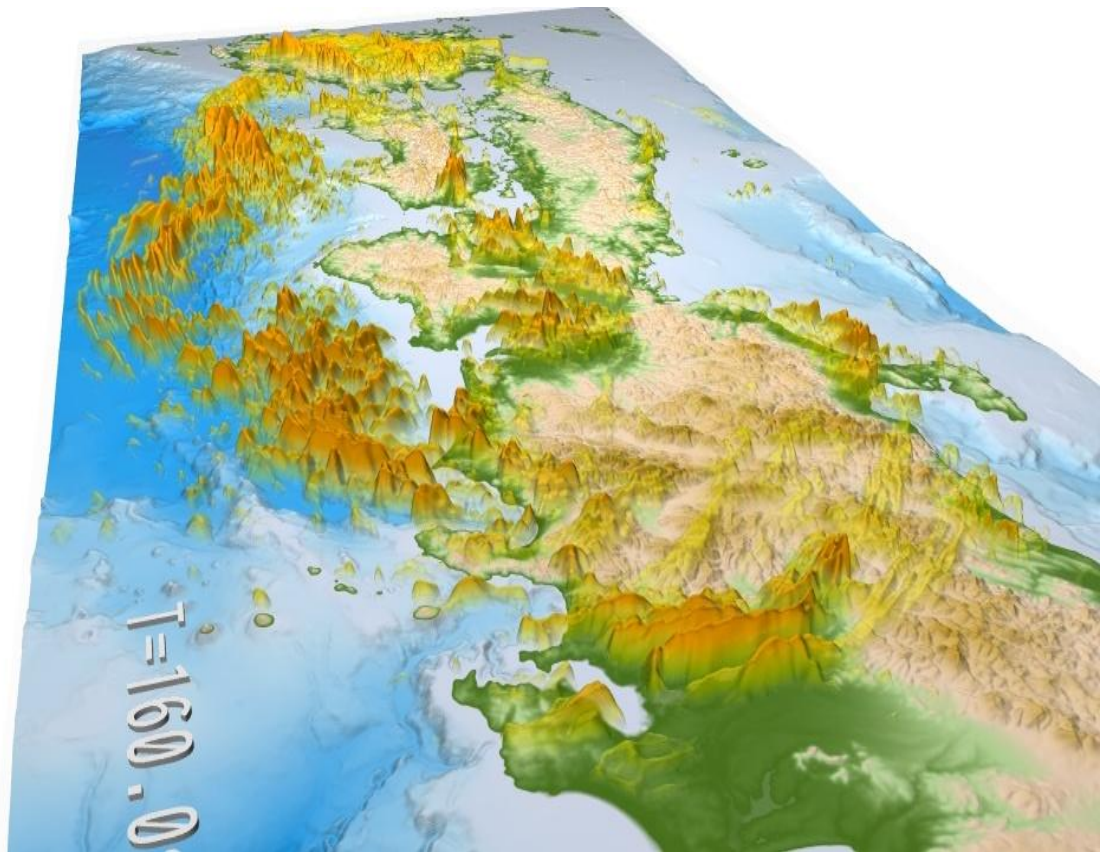


図8 1707年宝永地震の地震動シミュレーション。

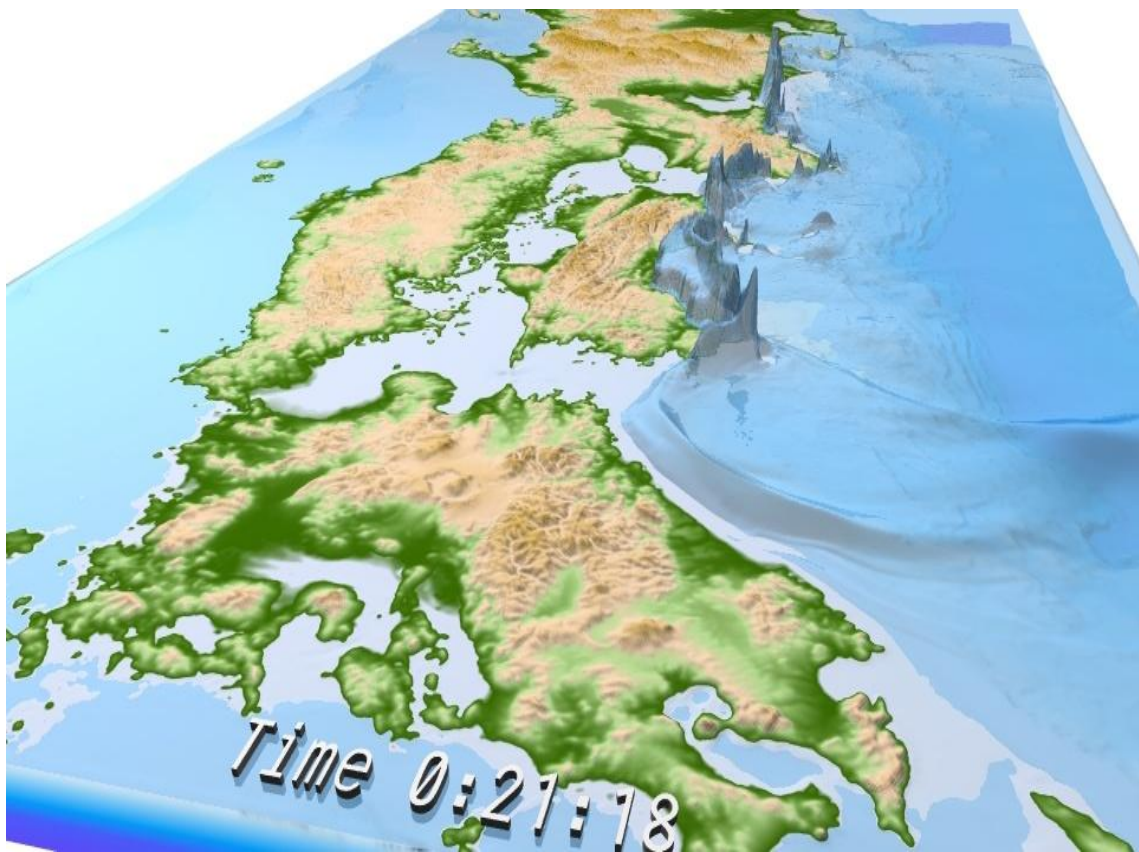


図9 1707年宝永地震の津波シミュレーション。

4. まとめと課題

最後に、本地震—津波シミュレーションコードを用いて実施した、1707年宝永地震の地震—津波シミュレーションの例を示す。本シミュレーションの計算領域は、水平方向に 1200×600 平方キロメートル、そして鉛直方向128キロメートルであり、これを $3000 \times 1500 \times 640$ 格子に離散化した。地表付近の地震波（S波）の最小速度は1.6 km/sとし、周波数1 Hzまでの地震波伝播を評価した（図8）。次に、本計算から求められた地震地殻変動を入力データとして、津波計算（図9）を行った。なお、地震動計算に要した時間は、T2Kの64ノード（1024CPU）計算において12.2時間、ES2の32ノード（256CPU）計算では2.0時間であった。なお、並列計算は、ノード内・ノード間ともにMPIを用いた並列計算（Flat MPI）にて行った。T2Kにおける実効性能はチューニング前の5.8%から7.1%にまで高めることができた。また、並列化率も99.72%から99.92%に向上し、数千CPU以上の大規模並列計算の実用化の目処を得た。