10-NA03

GPUを用いた地震波伝播シミュレーション

竹中博士

(九州大学大学院理学研究院)

概要 本研究では地震波伝播シミュレーションを大規模並列 GPU を用いて高速化すること を目指している。現時点では、完全弾性体の場合の並列 GPU プログラムを開発済みである。 このプログラムは、不規則地形、海底地形(流体層) 不均質構造を扱えるようになってい る。2010年11月から稼働を始めた TSUBAME-2.0 グリッドクラスタにおいて、1200 GPU までの範囲で理想的な場合に近い弱スケーリング性能が得られた。また、1200 GPU で 約 43 TFlops という非常に高い実行性能を達成することができた。

1. 研究の目的と意義

本研究では、非常に高い演算性能を持つ GPU を 複数用いた並列計算によって、地震波、すなわち 地球内部を伝播する弾性波のシミュレーションを 行う計算プログラムを開発する。

地球科学において、地震波データは地球や惑星 の内部を探るうえで基本的なデータとなっている。 また地球では、地震に伴う強い揺れの性質を解明 し強震動を予測することが社会的にも重要な課題 となっている。そのため、理論地震波波形を高速 に計算できる手法を開発することは、地球科学研 究や防災研究を進めていくうえで非常に重要であ る。そこで、本研究では、GPUを応用した地震波 伝播シミュレーションの計算プログラムを開発し、 従来型の CPU を用いた場合よりも高速なシミュ レーションが可能になることを目指す。

本研究で開発するプログラムは、地震波波形を 用いて沈み込み型(巨大)地震の震源域付近を対 象とした地球内部構造推定に応用することを主要 な目的とする。そのためには、媒質内部の波動場 も容易に抽出できる手法であることが望ましい。 そのため本研究では格子計算型の手法、すなわち 食い違い格子型の差分法(FDTD: finite-difference time domain)にもとづいたプログラムを開発す る。この手法は地震波伝播シミュレーションにお ける時間領域解法の標準的な手法の一つとして広 く用いられているものである。なお地震波波形計 算は地震学において基本的なテーマであることか ら、主要目的以外でも、地球や月・惑星の内部構 造推定や強震動の研究など多方面に幅広く役立て ていく予定である。

なお、本課題研究の研究組織は次の表の通りで ある。

大学・機関名	共同研究者	
国立大学法人九州大学		
研究代表者	竹中 博士	
国立大学法人東京工業大学	岡元 太郎	
	青木 尊之	
独立行政法人海洋研究開発機構	中村 武史	

2. 当拠点公募型共同研究として実施した意義

(1) 共同研究を実施した大学名国立大学法人東京工業大学

(2) 共同研究分野

超大規数値計算系応用分野

(3) 当公募型共同研究ならではという事項など

本研究では多数の GPU を用いた大規模並列計 算によって地震波伝播のシミュレーションを行う。 東京工業大学国際学術情報センターの TSUBAME グリッドクラスターは、GPU クラスターとして世 界最高レベルの性能を有している。そのため、本 研究を進めていくうえで最適な環境を利用するこ とができる。

3. 研究成果の詳細

ここでは、本研究で開発した GPU プログラム について紹介する。なお、以下の研究の主要な 部分は TSUBAME-1.2 を利用して得られた結果 (Okamoto et al., in press)にもとづいている。 2010年11月から稼働を始めた TSUBAME-2.0に よる演算性能についても若干の報告を行う。

3.1 差分法の格子点配置

本研究では時間領域の差分法(FDTD: finitedifference time domain)を用いる。この差分法で は媒質の粒子速度($v_i(i = x, y, z)$)と応力(τ_{ij}) とを変数として、図1に示す食い違い格子を用い て計算領域を離散化する。このような食い違い格 子は電磁界シミュレーションでも利用されている。

地震波シミュレーションでこのスキームを用い ると、変位のみを用いた場合に比べて変数が増加 する。しかしこのスキームは、不均質媒質の場合 でも数値計算上の安定性が良いなどの実用上の利 点を多く持っていることから、地震波伝播シミュ レーションにおいて標準的な方法として広く利用 されている。

なお、本研究で作成したプログラムでは、空間 差分精度は4次精度とした。時間差分は2次精度 とした。



図 1: 本研究で用いた食い違い格子と変数の配置。 図示した格子は単位セルに相当する。

本予稿では完全弾性体の場合のプログラムにつ いて述べる。この場合の支配方程式は次の組にな る。 f_i は力源、 ρ は密度、 λ 、 μ は弾性定数で μ が 剛性率である。この場合、1単位セルあたりのデー タは粒子速度・応力が計9個、物性パラメータが 3個(τ_{ii} と同じ格子点に置く)、合わせて12変数 となる。

$$\rho \frac{\partial v_i}{\partial t} = \frac{\partial \tau_{xi}}{\partial x} + \frac{\partial \tau_{yi}}{\partial y} + \frac{\partial \tau_{zi}}{\partial z} + f_i \qquad (1)$$

$$\frac{\partial \tau_{ij}}{\partial t} = \lambda \, \delta_{ij} \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right) \\
+ \mu \left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_j}{\partial x_j} \right)$$
(2)

食い違い格子を用いるため、粒子速度変数と応 力変数の時刻が時間刻みの半分だけずれることに なる。計算上は、粒子速度と応力を片方ずつ更新 していくことになる(なお、図7も参照)。

3.2 メモリアクセスの効率化

表 1 には、本研究で利用した TSUBAME グリッド クラスタの概要をまとめた。TSUBAME-1.2 で用 いられていた NVIDIA 社の S1070 は 4 基の GPU ユニットを備えた演算用サーバーである。図 2 に は S1070 の GPU ユニット単体の構成を簡略化し て示した。各 GPU ユニットは 8 個の演算コアを 持つマルチプロセッサを 30 個備えており、理論単 精度演算性能は 1036 Gflops である。



図 2: TSUBAME-1.2 の GPU 構成の概念図。

GPU計算ではGPU内部のメモリ(グローバル メモリ)に差分法領域の変数全てを保存し、その 変数に対して演算を行うことになる。グローバル メモリと演算ユニット間のメモリ帯域幅は、通常 の CPU の帯域幅に比べると非常に大きい。例え ば S1070 内の単体 GPU の帯域幅が 102 GB/s、 M2050 の帯域幅が 144 GB/s であるのに対して、 TSUBAME-1.2 の Opteron 880 (dual core) はソ ケットあたり 5.4 GB/s、Intel Xeon X5670 では 最大でソケットあたり 32 GB/s となっている。

しかしそれでもグローバルメモリから演算ユニッ トへのデータ転送には400-600 サイクルの遅延が 発生する。そのため、演算ユニット内部にある高 速な共有メモリを「キャッシュメモリ」として利 用することが性能向上のためには重要である。す なわち、差分法領域の小領域(ブロック)を共有 メモリへ転送し、演算ユニットは共有メモリ上の データを再利用することでグローバルメモリへの

TSUBAME	GPU	#GPUs/node	PCIe	Host CPU	Infiniband
1.2	NVIDIA S1070	2	1.0x8	Opteron 880	$10 { m ~Gbps}$
2.0	NVIDIA M2050	3	2.0x16	Xeon $X5670$	$40 \text{ Gbps} \times 2$

表 1: TSUBAME グリッドクラスタの概要

アクセスを低減することができる(図3)。この データ転送やブロック分割はプログラムで陽に指 定する。



図 3: 共有メモリをキャッシュメモリとして利用す ることの概念図。

ところが、地震波(弾性波)の問題では単位セル あたりの変数が多く、その一方で共有メモリのサ イズは小さいため(S1070の場合 16 kB)、共有メ モリに3次元のブロックを確保することは困難で ある。そこで本研究では、共有メモリとレジスタと を併用して3次元ブロックを確保する(図4)。す なわち、横方向への差分操作が必要になる2次元 平面上のセルのデータは共有メモリに配置し、同 時にこの2次元平面上の各セルに一つのスレッド を割りあてる。こうすると図の縦方向に位置する セルのデータについては縦方向への差分のみが発 生しスレッド間で共有する必要がないので、これ らのデータは各スレッドのレジスタに配置すれば よい。そして、縦方向へ計算を進めていく際に共有 メモリまたはレジスタに一度読み込んだデータを (パイプラインのように)レジスタと共有メモリと の間で交換することによって、グローバルメモリ へのアクセスを低減することができる。このよう なレジスタと共有メモリの利用法は Abdelkhalek et al. (2009) および Micikevicius (2009) によって 音波の問題に適用され、Michéa and Komatitsch (2010) および 岡元・他 (2010)、Okamoto et al. (in press) によって弾性波の問題に適用されている。



図 4: 共有メモリとレジスタを併用して3次元ブ ロックを確保することの概念図。

3.3 3 次元領域分割

GPUのグローバルメモリのサイズはあまり大きく ない(S1070の場合4GB、M2050の場合3GB)。 そのため大規模計算のためには計算領域を分割し て複数のGPUのそれぞれに副領域を割り当てる ことが不可欠となる。本研究では大規模な計算で 標準的に利用される3次元領域分割を用いる。副 領域間の通信にはMPIを用いる。



図 5:3 次元領域分割の概念図。青い部分が袖領域 を示す。

3次元領域分割では隣接する副領域との間で「袖 領域」(ghost points)を交換する必要がある。GPU 計算では、この袖領域のメモリ並びに関して問題 が発生する。これは、GPU間で直接データを交換 することはできないことと関係している(図2に 示したように、データを交換するためには、まず ホスト計算機のメモリに GPU からデータを転送 し、そのデータを別のホストに送るという段階を 経なければならない)。



図 6: 袖領域のメモリ並びの概念図。(a) 内部領域 の配列の延長として割り当てた場合。袖領域内部 では数多くの小さなデータが不連続に並んだメモ リ並びになる。(b) 内部領域とは独立の連続なメ モリ領域を割り当てた場合。

袖領域のメモリを割り当てる方法としては、 図 6(a) のように内部領域の配列を延長する方法が しばしば採用される。しかし、袖領域内部ではメモ リ並びは不連続となるため、袖領域のデータをホ スト計算機のメモリに転送する際には cudaMemcpy などのデータ転送関数を細切れのデータごとに繰 り返し呼び出すことになる。そのため袖領域全体 を転送するには非常に長い時間がかかることにな り、この方法はGPU計算では実用にならない。そ こで、本研究では図 6(b) のように内部領域とは 独立の連続なメモリを袖領域に割り当てる。こう すると、一度のデータ転送関数呼び出しで袖領域 全体を GPU からホストに転送することが可能と なるので、データ転送時間を(a)の場合よりも大 幅に短くすることができる。ただし GPU 内部で は、袖領域メモリのデータを図4の小ブロックに 転送するための複雑なマッピングが追加されるた め、その部分の処理が性能に影響をもたらし得る。 この点については後に議論する(§3.5)。

3.4 計算と通信のオーバーラップ

本研究で開発したプログラムの簡略化したフロー チャートを図7に示す。式1および式2に示し たように、食い違い格子では粒子速度成分と応力 成分を交互に時間積分していくことになる。した がって、それぞれの積分演算を行うGPUカーネ ルも別々のものになる。本研究で作成したプログ ラムでは、計算に要する全体の時間を短縮するた めに、内部領域についての積分と袖領域通信とを オーバーラップさせている。そのため、側面プロッ クのみを処理するカーネルと内部ブロックのみを 処理するカーネルの2つのカーネルを作成して利 用している。



図 7: 地震波計算プログラムのフローチャート。赤 い項目が GPU で実行するカーネルを示している。

3.5 単一 GPU での実行性能

ここでは、袖領域の処理(独立なメモリを用いたマ ッピング)を含むプログラムと、処理を含まないプ ログラムの2つを用いて実行性能の比較を行った。 袖領域の処理を含まないプログラムは単一 GPU 計算に限定されるが、GPU を用いた場合の計算 性能の目安を与えるものとして利用する。GPU 用 のプログラムは NVIDIA CUDA C を用いて作成 した。

図 8 にはそれぞれのプログラムの実行性能を 示す。これらは同一のプログラムを異なる GPU で実行して得られたものである。計算性能の数値 (GFlops)は、プログラムのソース・コードに現れ る浮動小数点計算の回数を、計算時間で割り算し て求めたものである。袖領域処理を含まない(単 一 GPU 用の)プログラムの実行性能は、どちら の GPU でも同程度のものになった。これに対し て、袖領域処理を含む(マルチ GPU 用の)プログ ラムでは、GPUによって大きな違いが見られた。 すなわち、S1070では袖領域処理を含むプログラ ムの性能は、袖領域処理を含まないプログラムの 半分程度に低下した。一方、M2050ではそのよう な性能低下がほとんど見られなかった。この違い の原因の一つとしては、キャッシュメモリの有無 の違いが考えられる (M2050 ではキャッシュメモ リが新規に導入された)。袖領域処理による性能 の低下は TSUBAME-1.2 を利用する際に大きな



図 8: 単一 GPU での実行性能。S1070(1GPU) と M2050 とを用いて測定したもの。青色のグラ フは袖領域処理を含まないプログラムによる実行 性能値。赤色のグラフは袖領域処理を含む(マル チ GPU 用の)プログラムによる実行性能値。差 分法の領域サイズはS1070 では 384 × 384 × 384、 M2050 では 320 × 320 × 320 とした。

制約となっていたが、TSUBAME-2.0 ではこの制 約がかなり緩和されたことになる。

3.6 マルチ GPU の実行性能: TSUBAME-1.2

TSUBAME-1.2 で全領域サイズを変えてマルチ GPU 計算を実行したときの性能を図 9 に示す。 すべての領域サイズにおいて、もっとも左側の点 (もっとも GPU 数が少ないケース)では副領域サ イズを同じ大きさ(384×384×384)にしてある。 そのため、もっとも左側の点をつないでいくと弱ス ケーリングの曲線が得られる。弱スケーリング性能 は 120 GPU までの範囲で、理想的な場合(GPU 数に比例)にやや近いものが得られた。120 GPU のときには約 2.2 TFlops という非常に高い実行 性能を達成できた。

一方、各領域サイズの性能曲線で示される強ス ケーリング性能は GPU 数には比例せず、GPU 数 の2/3 乗にほぼ比例することがわかった。袖領域 のサイズは副領域の表面積にほぼ比例し、副領域 の表面積は GPU 数(分割数)の2/3 乗にほぼ比 例することから、強スケーリングに関係した性能 は通信と袖領域の処理で律速されていることがわ かる(詳細は省略するが、側面ブロックのみの計 算時間、内部ブロックのみの計算時間、および通 信に要する時間を個別に測定した結果からもその ことがわかる: Okamoto et al., in press. を参照)。 このことから、プログラム側とクラスタ側の双方 において、袖領域処理や通信処理に要する時間を 短縮することが全体の性能を向上させるうえで重 要であることがわかる。



図 9: TSUBAME-1.2 での実行性能。ホスト CPU での性能は Fortran で作成した差分法プログラム を OpenMP で並列化して、PGI-Fortran でコン パイルし、1 ノード(16 コア)で実行して測定し たものである。そのため、ホスト CPU プログラ ムは1次元領域分割が適用されている。このよう に、CPU でのプログラムは GPU プログラムとは かなり違う構成になっているので、CPU での実行 性能は参考値として見ていただきたい。

3.7 マルチ GPU の実行性能: TSUBAME-2.0

TSUBAME-2.0 で全領域サイズを変えてマルチ GPU計算を実行したときの性能を図 10に示す。こ の例では副領域サイズを固定しているので、グラフ は弱スケーリング性能を示すことになる。図からわ かるように、1200 GPUの場合まで理想的なスケー リングに近い(GPU数にほぼ比例する)実行性能 が得られた。1200 GPUのときには約 43 TFlops という非常に高い実行性能を達成できた。なお、 § 3.5 で議論したように、1 GPUのときの性能が TSUBAME-2.0 では TSUBAME-1.2 のときの約 2 倍になっている。そのため、弱スケーリングの グラフも図 10 ではおよそ2 倍だけ「底上げ」され ている。

このような TSUBAME-2.0 における性能向上

は、袖領域処理に要する時間が短縮されたことと、 GPU-ホスト間(PCIeバス)およびホスト-ホス ト間(Infiniband)のそれぞれのデータ転送性能が 大幅に向上して通信処理に要する時間も短縮され たことが理由であろうと考えられる。



図 10: TSUBAME-2.0 での実行性能。副領域サイ ズを $320 \times 320 \times 320$ に固定して、全領域サイズ を変えて(つまり分割数 =GPU 数を変えて)弱ス ケーリング性能を測定したもの。ホスト CPU で の性能は Fortran で作成した差分法プログラムを OpenMP で並列化して、PGI-Fortran でコンパイ ルし、1 ノード(12 コア)で実行して測定したも のである。そのため、ホスト CPU プログラムは1 次元領域分割が適用されている。図 9 の場合と同 様に、CPU での実行性能は参考値として見ていた だきたい。

3.8 不規則陸上地形・海底地形や不均質構造の効果

本研究で用いている差分法では、媒質の不均質性は 物性パラメータを空間的に変化させることで取り 入れることができる。ただし、不規則な地形(自由 表面)や海底地形(固体・液体境界)を差分法に導 入する際には物性定数の与え方や差分式の設定に 注意する必要がある(詳しくは 岡元・竹中(2005) を参照)。本研究では、我々のグループによる方法 を用いて不規則地形・海底地形の効果を導入した (Nakamura et al., 2009; Takenaka et al., 2009; 中 村・他, in press)。 ここではプログラムや計算手法の妥当性を検証 するために、傾斜した固体・液体境界の場合の波形 を示す。この計算では中村・他(in press)の case 3 と同じ構造モデルを用いた(図11)。総セル数は70 億を超えるような大規模計算となっている。このモ デルでは境界面は傾斜しているが平面なので、半 解析的な波形計算方法(離散波数積分法、DWM) による波形と比較することができる。ただし、離 散波数積分法では傾斜した面を扱うことはできな いので、差分法波形に座標変換を施してから比較 することになる。そのようにして波形を比較した ものが図 12 である。差分法波形(赤色)と離散波 数法波形(青色)はほぼ一致しており、プログラ ムや手法の妥当性を確認できる。



図 11: 計算に用いた構造モデル。上の図の青い部 分が液体層(海水)を示している。数値は1辺あた りの単位セル数であり、総セル数は70億を超える。 下の図は震源と境界面上にある観測点を通る断面 で見たときのそれぞれの位置を示す。格子間隔は 25 m、時間間隔は0.002 s,時間ステップ数は7500 である。この計算は TSUBAME-1.2 の 100 GPU を利用して実行した。計算に要した時間は約4800 秒(1.9 TFlops)であった。P 波速度は液体(海 水を想定)が1.5 km/s、固体が 5.2 km/s、固体の S 波速度は 3.0 km/s、密度は液体が 1.05 g/cm³、 固体が 2.3 g/cm³ と設定した。震源は Z 軸方向 (下向き)の single force で、時間関数は f(t) = $(1 - \cos(2\pi t))$ (0 < t < 1s) である。

さらに現在、より複雑で現実的な場合の計算を行 うためのプログラムの準備を進めているところで ある。そのような場合の一例として、伊豆半島付近 に仮想的な震源を置いた場合に地震波がどのように 伝播していくかを可視化したものを示す(図13)。 この計算では地表地形と海底地形を取り入れて、



図 12: 固体・液体境界での固体側の変位波形。X,Y は境界面に沿った変位、Z は境界面に垂直な向き の変位である。

海水層、堆積物層、地殻という不規則な境界面を 持つ3層構造モデルを用いている。このモデルは 3層のみを用いているなどの面で実際の構造を簡 略化したものであり、実際のデータと比較するた めにはより現実的な構造モデルを設定して計算に 取り入れていく必要がある。



図 13: 伊豆半島付近に仮想的な震源を置いた場合 について、地震波伝播の様子を可視化したもの。 自由表面での上下動変位の大きさと向きをカラー スケールで表現した。図は発震時から 6.8 秒後の 変位場の様子を表す。この計算は1 台の M2050 を用いて実行した。格子間隔は 200 m、格子数は 640 × 640 × 128、震源の深さは約 10 km である。 構造モデル設定にあたっては、地震調査研究推進 本部「長周期地震動予測地図」(2009 年試作版)の 構造モデルの一部を利用させていただいた。

4. これまでの進捗状況と今後の展望

課題申請時の研究計画の項目と対比させて、進捗 状況と展望をまとめる。 4.1 不均質地球内部構造の導入および不規則な地 表地形・海底地形の導入

上に述べたように、これらはすでにマルチ GPU プログラムに導入し、計算例などを発表した (Okamoto et al., 2010 AGU Fall Meeting 発表 など)。

4.2 非弾性減衰の導入

単一 GPU プログラムでは、非弾性メカニズムが1 要素の場合について実装を行い、計算例などを発 表した(岡元・他、2010年地震学会秋季大会)。 マルチ GPU プログラムについては実装作業を進 めているところである。

4.3 GPU 内部共有メモリ利用の効率化

上に述べたように、共有メモリとレジスタを併 用する方法をマルチ GPU プログラムに実装した (Okamoto et al., in press. 他)。

4.4 複数 GPU 計算

上に述べたように、袖領域処理を含むマルチ GPU プログラムを開発し、1200 GPU までの範囲で実 行性能を測定した。さらに、メモリを節約するた めに、物性パラメータについてインデックス配列 を用いる手法もマルチ GPU プログラムに実装済 みである。非弾性減衰を導入すると単位セルあた りの変数の個数が数倍以上に増加するため、イン デックス配列の実装は非弾性減衰の実装の準備と 位置づけられる。

以上、課題申請時の各研究項目の多くが現時点 までに達成されたと考えている。現時点では未達 成の項目(マルチ GPU プログラムへの非弾性減 衰効果の実装)についても、単一 GPU プログラ ムでの非弾性効果の実装や、マルチ GPU プログ ラムへのインデックス配列実装などの準備が進ん でおり、導入のめどがたっていると考えている。

5. 研究成果リスト

- (1) 学術論文
- Okamoto, T., H. Takenaka, T. Nakamura, and T. Aoki, Accelerating Large-Scale Simula-

tion of Seismic Wave Propagation by Multi-GPUs and Three-Dimensional Domain Decomposition, *Earth, Planets and Space*, in press.

(3) 国際会議発表

- Okamoto, T., H. Takenaka, T. Nakamura, and T. Aoki, Accelerating simulation of seismic wave propagation by multi–GPUs, *Ameri*can Geophysical Union 2010 Fall Meeting, IN44A–03, San Francisco, USA, 2010年12 月16日. (招待講演)
- Okamoto, T., H. Takenaka, T. Nakamura, and T. Aoki, Accelerating simulation of seismic wave propagation by multi GPUs and 3D decomposition, 国際シンポジウム GPU Solutions to Multiscale Problems in Science and Engineering, ハルビン(中国), 2010年7月 27日(招待講演)

(4) 国内会議発表

岡元太郎・竹中博士・中村武史・青木尊之, GPU を用いた地震波伝播シミュレーション(3):地 形・流体層と非弾性減衰の導入, 地震学会 2010 年秋季大会, D32-08, 広島県広島市, 2010 年 10月 29日.

参考文献

- Graves, R. W., Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences, Bull. Seism. Soc. Am., 86, 1091–1106, 1996.
- Michéa, D., and D. Komatitsch, Accelerating a threedimensional finite-difference wave propagation code using GPU graphics cards, *Geophys. J. Int.*, doi: 10.1111/j.1365-246X.2010.04616.x, 2010.
- Micikevicius, P., 3D finite-difference computation on GPUs using CUDA, in GPGPU-2: Proceedings of the 2nd Workshop on General Purpose Processing on Graphics Processing Units, pp. 79–84, Washington DC, USA, 2009.
- Nakamura, T., H. Takenaka, T. Okamoto, and Y. Kaneda, Finite-difference simulation of strong motion from a sub-oceanic earthquake: modeling effects of land and ocean-bottom topographies, *American Geophysical Union, Fall Meeting*, S43B-1981, 2009.
- 中村武史・竹中博士・岡元太郎・金田義行, 流体-固体境界 周辺における3次元地震波動場の差分解法に関する数 値実験, 地震, 第2輯, in press.

- 岡元太郎・竹中博士,速度・応力型差分法での固体・流体境 界の扱いについて,地震,第2輯,57,355-364,2005.
- Okamoto, T., H. Takenaka, and T. Nakamura, Simulation of seismic wave propagation by GPU, Proceedings of Symposium on Advanced Computing Systems and Infrastructures, 141–142, 2010.
- Takenaka, H., T. Nakamura, T. Okamoto, and Y. Kaneda, A unified approach implementing land and ocean-bottom topographies in the staggered-grid finite-difference method for seismic wave modeling, Proceedings of the 9th SEGJ International Symposium, CD-ROM Paper No.37, 2009.
- 地震調查研究推進本部地震調查委員会編集、「長周期地震動 予測地図」2009 年試作版、2009 年 9 月発行.