

GPU クラスタを用いた モンテカルロ地殻変動想定手法の開発

山口 拓真¹・縣 亮一郎²・市村 強³・堀 宗朗⁴・LALITH Wijerathne⁵

¹ 学生 東京大学 工学部 社会基盤学科 (〒113-8654 東京都文京区本郷 3-1)

E-mail:yamaguchi@eri.u-tokyo.ac.jp

² 修士 東京大学大学院 工学系研究科 社会基盤学専攻 (〒113-8654 東京都文京区本郷 3-1)

E-mail:agata@eri.u-tokyo.ac.jp

³ 正会員 東京大学准教授 工学部社会基盤学科 (〒120-0005 東京都文京区弥生 1-1-1)

E-mail:ichimura@eri.u-tokyo.ac.jp

⁴ 正会員 東京大学教授 工学部社会基盤学科 (〒120-0005 東京都文京区弥生 1-1-1)

E-mail:hori@eri.u-tokyo.ac.jp

⁵ 正会員 東京大学准教授 工学部社会基盤学科 (〒120-0005 東京都文京区弥生 1-1-1)

E-mail:lalith@eri.u-tokyo.ac.jp

地震災害においては地殻変動が災害要因となる場合があり、被害想定のために地殻構造モデルを用いた解析手法が求められている。地震時の地殻変動解析では地震シナリオを一意に定められないため、入力データの曖昧さを定量的に評価することが望ましいとされている。本研究では多数回計算を用いたモンテカルロシミュレーションによる地殻変動解析手法を開発した。莫大な計算負荷を克服するために GPU を導入し、ハードウェアに適したアルゴリズムを構築することによって、現実的な時間内で曖昧さを考慮した地殻変動解析を可能とした。

Key Words: crustal deformation, Finite Element Method, GPU computation, uncertainty

1. はじめに

地震時には地殻の変動が大きな被害をもたらすと懸念されている。この被害の軽減を図る上では、地震時の地殻変動をシミュレーションし、想定外の事態を最小限に抑えた被害予測を行うことが重要となる。近年では地殻構造観測・推定技術の発展により、高精度な観測網が利用可能である。一方で高い分解能での解析を行う場合、問題の自由度が大きくなる。既往の研究では断層帯の活動による地表面の上下変位がシミュレーションされてきたが、計算負荷の大きさから、想定される地震シナリオについては数ケースのみを設定して解析を行うという決定的論的側面の強い被害予測に留まっている¹。

本研究では GPU クラスタを用いた、モンテカルロシミュレーションによる地殻変動想定手法を開発する。解析に含まれる各種の演算を高速化することで、現実的な計算時間での多数回の解析を行い、モンテカルロシミュレーションの適用を可能とする。適用例では琵琶湖西岸断層帯を対象領域として、地震時地殻変動による地表面上下変位を評価することにより、提案手法の有用性を示す。

2. 手法

(1) 有限要素法定式化

解析には詳細な地殻データをモデル化するのに適した有限要素法を用いる。本研究においては短い時間スケールを対象とするため、地殻を線形弾性体とみなし、食い違い断層に対する地殻変動解析を行う。基礎方程式を離散化することにより、解析は以下の連立一次方程式に帰着する²。

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad (1)$$

ここで、 \mathbf{K} 、 \mathbf{u} 、 \mathbf{f} はそれぞれ全体剛性マトリクス、変位ベクトル、外力ベクトルである。

(2) 有限要素モデル生成手法

三次元有限要素モデル生成では、成層構造の数値標高データと物性値を入力とする。モデル生成時間を短縮するためにバックグラウンド構造格子を利用しており、これにより剛性マトリクスの計算も高速化される。モデルは四面体二次要素で構成されており、側面と底面に無限要素を用いることで半無限性が導入される。

```

1 :  $\mathbf{r} \leftarrow \sum \mathbf{K}^i \mathbf{u}^i$ 
2 : MPI通信による $\mathbf{r}$ の袖通信
3 :  $\mathbf{r} \leftarrow \mathbf{f} - \mathbf{r}$ 
4 :  $\beta \leftarrow 0$ 
5 :  $i \leftarrow 1$ 
6 : do while  $\|\mathbf{r}\|^2 / \|\mathbf{f}\|^2 \leq \epsilon$ 
7 :  $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{M}}^{-1} \mathbf{r}$ 
8 :  $\bar{\mathbf{r}}_c \leftarrow \bar{\mathbf{P}}^T \mathbf{r}$ 
9 :  $\bar{\mathbf{u}}_c \leftarrow \bar{\mathbf{P}}^T \mathbf{u}$ 
10 :  $\bar{\mathbf{K}}_c \bar{\mathbf{u}}_c = \bar{\mathbf{r}}_c$ をアルゴリズム2によって求解
11 :  $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{P}} \bar{\mathbf{u}}_c$ 
12 :  $\bar{\mathbf{K}} \bar{\mathbf{u}} = \mathbf{r}$ をアルゴリズム2によって求解
13 :  $\mathbf{z} \leftarrow \bar{\mathbf{u}}$ 
14 : if  $i > 1$  then
15 :    $\gamma \leftarrow (\mathbf{z}, \mathbf{q})$ 
16 :   MPI通信による $\gamma$ の総和計算
17 :    $\beta \leftarrow \gamma / \rho$ 
18 : end if
19 :  $\mathbf{p} \leftarrow \mathbf{z} + \beta \mathbf{p}$ 
20 :  $\mathbf{q} \leftarrow \sum \mathbf{K}^i \mathbf{p}^i$ 
21 : MPI通信による $\mathbf{q}$ の袖通信
22 :  $\rho \leftarrow (\mathbf{z}, \mathbf{r})$ 
23 : MPI通信による $\rho$ の総和計算
24 :  $\gamma \leftarrow (\mathbf{p}, \mathbf{q})$ 
25 : MPI通信による $\gamma$ の総和計算
26 :  $\alpha \leftarrow \rho / \gamma$ 
27 :  $\mathbf{q} \leftarrow -\alpha \mathbf{q}$ 
28 :  $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{q}$ 
29 :  $\mathbf{u} \leftarrow \mathbf{u} + \alpha \mathbf{p}$ 
30 :  $i \leftarrow i + 1$ 
31 : end while

```

表-1 連立1次方程式求解アルゴリズム1

(3) 連立一次方程式求解方法

本研究では我々のグループで開発された、並列計算用共役勾配法によるソルバー³⁾を利用する。表-1 および表-2 にアルゴリズムを示す。ソルバーは通常用いられる共役勾配法のソルバーと比較してデータ通信量および計算コストが削減されるよう、複数の手法から構成されている。以下に導入された各手法について述べる。

a) 可変的前処理

可変的前処理とは、通常では反復間で不変となる行列を前処理に用いる代わりに、共役勾配法の前処理方程式 $\mathbf{K}\mathbf{r} = \mathbf{z}$ を別のソルバーによって解く方法である。この前処理方程式の求解においても共役勾配法を利用するため、ソルバー内では元々の反復の内部に、前処理のための反復が含まれる構成となっている。以降、前処理方程式を解くための反復を内部反復、元々の反復を外部反復と呼ぶ。表-1 が外部反復、表-2 が内部反復にあたるアルゴリズムである。内部反復は前処理であるため、求解にある程度の誤差が許容される。そこで、次のような手法の適用が可能となる。

```

1 :  $\bar{\mathbf{e}}_c \leftarrow \bar{\mathbf{K}} \bar{\mathbf{u}}$ 
2 : MPI通信による $\bar{\mathbf{e}}_c$ の袖通信
3 :  $\bar{\mathbf{e}}_c \leftarrow \bar{\mathbf{r}} - \bar{\mathbf{e}}_c$ 
4 :  $\beta \leftarrow 0$ 
5 :  $i \leftarrow 1$ 
6 : do while  $\|\bar{\mathbf{e}}\|^2 / \|\bar{\mathbf{r}}\|^2 \leq \epsilon$ 
7 :  $\bar{\mathbf{z}} \leftarrow \bar{\mathbf{M}}^{-1} \bar{\mathbf{e}}$ 
8 :  $\bar{\rho}_a \leftarrow (\bar{\mathbf{z}}, \bar{\mathbf{e}})$ 
9 : MPI通信による $\bar{\rho}_a$ の総和計算
10 : if  $i > 1$  then
11 :    $\beta \leftarrow \bar{\rho}_a / \bar{\rho}_b$ 
12 : end if
13 :  $\bar{\mathbf{p}} \leftarrow \bar{\mathbf{z}} + \beta \bar{\mathbf{p}}$ 
14 :  $\bar{\mathbf{q}} \leftarrow \bar{\mathbf{K}} \bar{\mathbf{p}}$ 
15 : MPI通信による $\bar{\mathbf{q}}$ の袖通信
16 :  $\bar{\gamma} \leftarrow (\bar{\mathbf{p}}, \bar{\mathbf{q}})$ 
17 : MPI通信による $\bar{\gamma}$ の総和計算
18 :  $\bar{\alpha} \leftarrow \bar{\rho}_a / \bar{\gamma}$ 
19 :  $\bar{\rho}_a \leftarrow \bar{\rho}_a$ 
20 :  $\bar{\mathbf{e}} \leftarrow \bar{\mathbf{e}} - \bar{\alpha} \bar{\mathbf{q}}$ 
21 :  $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{u}} + \bar{\alpha} \bar{\mathbf{q}}$ 
22 :  $i \leftarrow i + 1$ 
23 : end while

```

表-2 連立1次方程式求解アルゴリズム2

b) マルチグリッド法

内部反復に対して、マルチグリッド法を適用することによって、従来の前処理付き共役勾配法と比較して計算量を削減することができる。本ソルバーでは幾何的マルチグリッド法を適用し、元になる四面体二次要素モデル (fine model) と同一に生成した、四面体一次要素モデル (coarse model) を利用する。coarse model を用いて $\mathbf{K}_c \mathbf{r}_c = \mathbf{z}_c$ を解くことにより、fine model における $\mathbf{K}\mathbf{r} = \mathbf{z}$ の収束性を高めることができる。また coarse model は fine model と比較して自由度が少ないため、計算コストを抑えることができる。アルゴリズム内では、グリッド間での変数の変換は、coarse model から fine model へのマッピング行列 $\bar{\mathbf{P}}$ を用いてそれぞれ計算される。

c) 精度混合演算

先述の通り、前処理は高い精度での演算を必要としない。そこで、内部反復は単精度演算によって計算を行うこととする。これにより単精度演算を導入した箇所についてはメモリアクセス量と通信量を半減することができ、計算時間の削減につながると考えられる。一方で外部反復における演算は倍精度によって行われるため、最終的な精度は十分に保ちつつ、計算を高速化することが可能となる。アルゴリズム内でオーバーラインを付した変数は単精度によって格納される。

(4) GPU クラスタによる高速化手法

本研究では CPU と GPU のヘテロコンピューティングによって解析の更なる高速化を試みる。共役勾配法内の可変的前処理は計算量が大きく、前処理部の高速化が解析全体の高速化に寄与する。また、前処理部に含まれる演算は単精度演算であり、一般に GPU は倍精度演算と比較して単精度演算の性能が大きく上回る。そこで本研究では単精度演算による前処理部分を GPU にポーティングする。

GPU の有するメモリは数 GB～数十 GB 程度であるため、大規模問題を解析するためには複数枚の GPU による並列性の向上が不可欠である。そこで解析領域を Metis により分割し、各領域に GPU を 1 枚ずつ割り当てる。領域間での通信コントロールには MPI を利用する。また CPU による計算部では MPI と OpenMP とのハイブリッド並列化を導入し、計算ノード内の計算資源をすべて使用して解析を行う。GPU のプログラミング環境は CUDA Fortran⁴⁾を使用する。NVIDIA 製 GPU に対する詳細なチューニングを行うことによって、ハードウェアの性能を最大限に引き出すことができる。

GPU が処理する各種演算では、疎行列ベクトル積が計算負荷の大部分を占める。GPU は不連続なメモリアクセスに不向きなため、演算性能を発揮させるには適切なデータ構造が必要となる。本解析では疎行列内の非零要素の分布を考慮すると、HYB 形式⁵⁾が適した格納形式となった。本解析では節点ごとにブロック化した上で行列データを格納する。これによりメモリアクセスが効率化し、解析全体の高速化が実現した。具体的な計算時間については次章で示す。

3. 適用例

本手法による、琵琶湖西岸断層帯における地表面変位応答シミュレーションを行う。GPU による高速解析を用いて、多数の震源に対する応答関数をそれぞれ計算する。得られた関数を重ね合わせることですべり量の大きさと分布を様々に変化させた多数の地震シナリオを作成し、地表面変位応答の評価を行う。

(1) 問題設定

モデル化の対象範囲は 240km×240km×300km とする。入力データは J-SHIS の深層地盤データ⁶⁾を用いており、境界面は 33 面上の格子点の DEM データとなる。これらのデータから、図-1、図-2 に示される有限要素モデルが形成される。生成されたモデルの自由度は 423,052,212 で、四面体要素数は 104,110,586 となった。地震シナリオについては長方形の断層面モデルを設定し、内部の格子点に点震源を分布させることで、断層帯のすべり量分

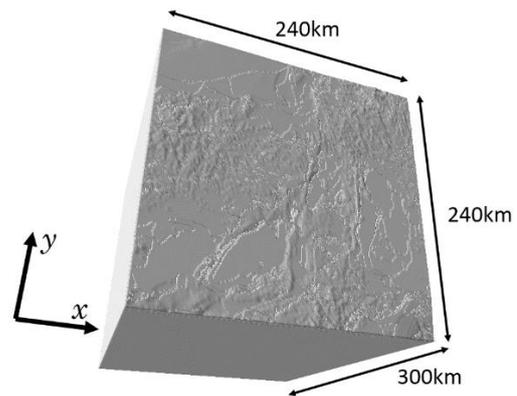


図-1 有限要素法モデル：全体図

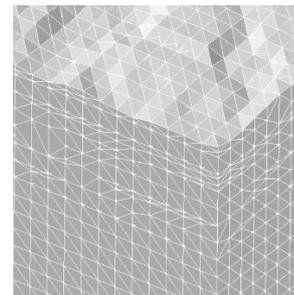


図-2 有限要素法モデル：拡大図

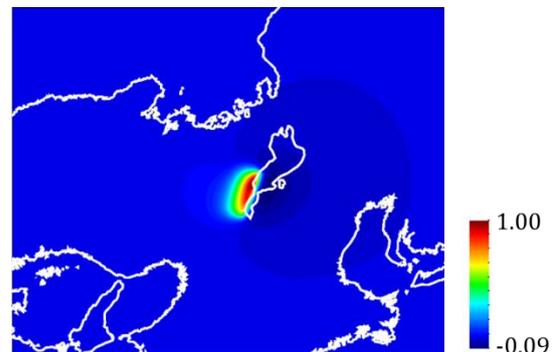


図-3 上下変位平均値

布を再現する。まず各格子点に対して点震源を近似した節点力を入力し、それに対応するグリーン関数をそれぞれ求める。今回設定した断層面モデル上には 541 の格子点が存在したため、541 回の地殻変動解析が必要となる。解析においては NVIDIA Tesla K40 が 64 枚搭載された GPU クラスタを利用した。次に得られたグリーン関数を確率的に足し合わせることで、地震シナリオの 1 つを再現することができる。本適用例では 10000 ケースの地震シナリオを作成する。

(2) 解析結果

10000 ケースの地震シナリオに対して、地表面における上下方向変位の平均値を表したものが図-3 である。

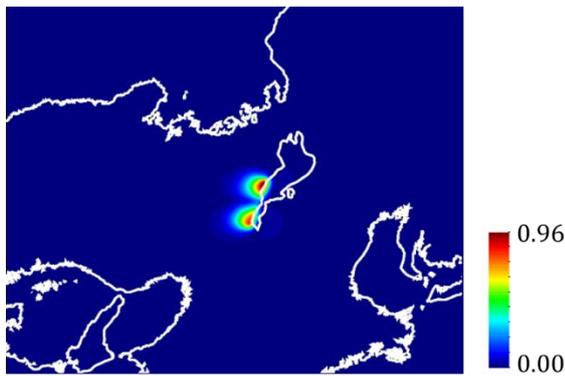


図-4 上下変位標準偏差

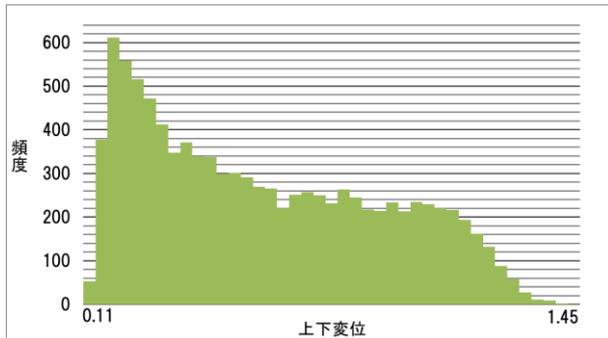


図-5 上下変位分布図

また図-4は標準偏差を示したものである。さらに1地点について上下変位分布を取ったところ、図-5が得られた。これらの値については、図-3における最大値を1として正規化を行っている。得られた分布は裾が重く、モンテカルロシミュレーションの有効性が示されたといえる。地殻変動解析での計算時間は、格子点1点について34.8秒となり、541点全てに対しては5時間18分47秒となった。この計算をCPUのみで行った場合、グリーン関数1つにつき242.9秒を必要とし、GPUの導入によって計算時間が1/7に短縮された。

4. おわりに

本研究では、大自由度となる地殻変動解析を多数回計算するモンテカルロシミュレーションにより、統計的な被害予測を行うための手法を開発した。莫大な計算負荷を克服するためにGPUクラスタを導入し、ソルバー内の演算をGPUに計算させた。これによりソルバー内の計算時間は1/7に短縮された。適用例では4億程度の自由度を持つ領域に対して、地殻変動におけるモンテカルロシミュレーションに必要な解析が5時間程度で計算可能となった。これらのシミュレーション結果の、他の経済影響評価との検討は今後の課題とする。今後は解析高速化手法を利用して、地震時断層すべりの推定のような逆解析を行うことを予定している。

参考文献

- 1) 川崎一朗 他：琵琶湖西岸断層帯南部の仮想地震による地殻変動と琵琶湖疏水，歴史都市防災論文集，vol.6，pp.97-103，2012.
- 2) 日本計算工学会編，竹内則雄，榎山和男，寺田賢二郎共著：計算力学有限要素法の基礎，森北出版株式会社，2003.
- 3) Ichimura, Tsuyoshi, et al. "Physics-based urban earthquake simulation enhanced by 10.7 BlnDOF×30 K time-step unstructured FE non-linear seismic wave simulation." Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE Press, pp.15-26, 2014.
- 4) Shane Cook: CUDA Programming A Developer's Guide to Parallel Computing with GPUs, Morgan Kaufmann, 2013.
- 5) Li, K., Yang, W., & Li, K. Performance analysis and optimization for SpMV on GPU using probabilistic modeling. IEEE Transactions on Parallel and Distributed Systems, 26(1), pp. 196-205, 2015.
- 6) 地震ハザードステーション，<http://www.j-shis.bosai.go.jp/>

(20???. ???.?? 受付)

A MONTE CARLO ANALYSIS METHOD OF CRUSTAL DEFORMATION COMPUTATION ON GPU CLUSTERS

Takuma YAMAGUCHI, Ryoichiro AGATA, Tsuyoshi ICHIMURA, Muneo HORI and Lalith Wijerathne

Crustal deformation could be a significant disaster factor. To make the efficient counter plan, we need to simulate co-seismic crustal deformation and estimate damages. In the simulation, it is desirable to use crustal deformation observation data and crustal structure data at high resolution. In addition, uncertainties of inputs should be considered.

In this study, we developed the analysis method of crustal deformation quantitatively considering uncertainties by MC (Monte Carlo) method. To handle massive computation cost, we developed the fast numerical simulation method of crustal deformation based on the finite element method with GPU (Graphics Processing Units) computation. We reconstructed the algorithm so that calculation in the analyses was suitable for GPU computation. By introduction of GPUs, computation time was significantly decreased and we enabled to perform simulations many times in a realistic time.