

jh250053

# MPM と FEM による未解明な大規模土砂災害の 数値シミュレーション

寺田賢二郎（東北大学）

## 概要

本課題では、土砂崩れや海底地すべりに起因する津波等の大規模な自然災害リスクを定量的に評価することを目的に、Material Point Method (MPM) を中心とした数値シミュレーション手法の開発を行う。解析手法としては、固相解析に陽的 MPM を採用するほか、流体との連成解析には、陰解法を採用する安定化有限要素法を適用する。今年度は、陰解法ソルバーの高度化を目的として、固体単相の陰的 MPM においてブロック不完全 LU 分解前処理と BiCGStab 法の組み合わせにより、従来用いていた対角スケーリングと GBiCGStab( $s, L$ ) に比べ連立一次方程式の求解において約 6.3 倍の高速化を達成した。また、陽解法の高速化を目的に、MPI 並列に加えて、OpenMP を実装した。加えて、OpenACC も実装することで、GPU 化も実現した。結果として、CPU (OpenMP 72 スレッド) 比で最大約 6.7 倍の高速化を実現した。実大規模斜面崩壊解析への適用では、25 ノードの MPI + GPU 並列計算により既往の MPI 並列計算 (5,760 コア) に比べ約 2.85 倍の高速化が得られることを確認した。

## 1 共同研究に関する情報

### 1.1 共同研究を実施した拠点名

- 東京大学 情報基盤センター
- 東北大学 サイバーサイエンスセンター

### 1.2 課題分野

- 大規模計算科学課題分野

### 1.3 参加研究者の役割分担

- 寺田 賢二郎（東北大学）：研究統括
- 中島 研吾（東京大学）：性能最適化
- 河合 直聡（東北大学）：性能最適化
- 三木 洋平（東京大学）：性能最適化
- 坂口 吉生（富士通）：性能最適化
- 野村 怜佳（東北大学）：3D-2D 連成手法の開発

- 飛弾野 壮真（東北大学）：GPU 高速化 (OpenACC)
- 菅井 理一（東北大学）：陰的 MPM の開発 (VMS 安定化混合定式化)
- 吉田 圭那（東北大学）：陰的・陽的 MPM の開発
- 鈴木 祐太（東北大学）：Hybrid MPM-FEM の開発

## 2 研究の目的と意義

土砂崩れや斜面災害は大きな被害をもたらす自然災害であり、その発生・伝播メカニズムの解明や被害・対策の検討に活用することを目的として、多くの数値計算ツールが開発されてきた。しかしながら、実大規模シミュレーションの実

現には、1) 固体・流体の精細なモデル化・解析と、2) 大規模かつ長時間のシミュレーションを実施できる計算速度が課題となる。

本研究グループは、2021 年に福島県のエビスサーキットで発生した地震による大規模な土砂崩れと、2018 年にインドネシアのスラウェシ島付近で発生した海底地すべりとそれに起因する津波 (Palu 津波) を対象として、現象を詳細に再現できる高精細・高効率な数値シミュレーション手法の開発を目指している。

解析手法として、履歴依存性のある固体土骨格の大変形を表現できる Material Point Method (MPM) を固相解析に採用し、水・空気の流体相解析には SUPG/PSPG 安定化を施した Euler 型 FEM を採用する MPM-FEM ハイブリッド法を用いる。MPM は FEM と類似の数値積分アルゴリズムを採用しているため、固体解析への適用性が高く、FEM との連成も比較的簡単に実装できる。

研究計画全体では、(a) 固体構成則の開発と、固体・流体解析それぞれの安定化、および (b) 大規模並列計算の高速化 (動的負荷分散、GPU 化、線形ソルバーの高度化) の両面から研究を進めている。本研究課題の主な焦点は (b) にあり、これまでに陽解法 MPM に対して動的負荷分散の実装を行い、エビスサーキットの実大規模解析を実施した。今年度 (jh250053) は、線形ソルバーの高速化、および陽的 MPM への GPU 実装によるさらなる高速化を重点的に実施した。

### 3 当拠点公募型研究として実施した意義

今年度は、東京大学情報基盤センターの Wisteria/BDEC-01 Odyssey と東京大学情報基盤センターと筑波大学計算科学研究センターが共同運営する最先端 HPC 基盤施設の共同利用スーパーコンピュータシステムである Miyabi-G

を使用した。

実大規模の土砂崩れ・海底地すべりシミュレーションに必要な大規模 3 次元解析 (格子数 1800 万以上、粒子数 1900 万以上) は、JHPCN の大規模計算資源なくしては実現できない。また、東京大学の中島・三木 (情報基盤センター) および東北大学の河合 (サイバーサイエンスセンター)、富士通の坂口による計算科学的な知見の提供と協力は、アルゴリズムの高速化に不可欠である。このような文理融合・産学連携体制での研究は、JHPCN の公募型共同研究として実施する意義が高い。

### 4 前年度までに得られた研究成果の概要

本研究は 2022 年度 (jh220019) に開始した継続課題であり、前年度 (jh240040) までに以下の主要な成果を得ている。

#### 4.1 動的負荷分散 (DLB) の実装と有効性検証

MPM による解析では、土砂流動に伴い粒子の空間分布が大きく変化するため、MPI 並列計算において各プロセス間の演算負荷・通信負荷に偏りが生じる問題があった。これに対して、粒子分布に応じて計算領域分割を動的に更新する動的負荷分散 (DLB) を実装し、2 次元・3 次元土柱崩壊解析を通してその有効性を検証した。DLB を適用しない場合の並列化効率が 10~20% であるのに対し、DLB を適用した場合は 60~95% の並列化効率となることを確認した。3 次元解析では DLB 採用時の計算時間が 6,886 秒 (約 1.91 時間) であるのに対し、非採用時は 138,037 秒 (約 38.3 時間) と、計算コストを約 1/20 に抑えることができた。

また、エビスサーキット土砂崩れ (2021 年、福島県) を対象として、DEM (Digital Elevation Model) データに基づく解析領域 (450 m × 125 m × 320 m)、格子一辺 1 m、粒子数 約

1,900 万の実大規模解析を Wisteria/BDEC-01 Odyssey (5,760 コア) を用いて実施した。DLB の適用により、100 秒の全解析を約 6.3 時間で完了した。以上の成果は国際誌 *Engineering with Computers* に掲載された [1]。

#### 4.2 MPM-FEM ハイブリッド法における異領域分割の実装

MPM-FEM ハイブリッド法では、固相 (MPM) と流体相 (Euler 型 FEM) とで、最適な MPI 領域分割パターンが異なる。固相 (土骨格) は計算粒子が領域の一部にのみ存在するため粒子分布に応じた動的負荷分散が有効な一方、流体相 (水・空気) は領域全体に存在するため要素数均等の静的分割が最適となる。そこで、両相で異なる領域分割を同時に用いながら矛盾なく通信を行うアルゴリズムを提案した [2]。水中での土柱崩壊解析 (固相粒子数 12,288 個・流体格子数 16,000) において、DLB 適用により固相の解析時間を削減した。ただし、流体相の計算が大半を占めており、その中でも特に計算時間の多くを占めている線形ソルバーの高度化が必要であることを、明確にした。以上の成果は国際誌 *Computational Mechanics* に掲載された [2]。

## 5 今年度の研究成果の詳細

今年度 (jh250053) は、以下の 2 つの課題を中心に取り組んだ。

### 5.1 固体単相の陰的 MPM: 混合定式化と VMS 安定化

前年度の検討から、固液連成解析において流体相の計算が全体の大半を占め、うち線形方程式のソルバーが約 1/3 を占めることが判明していた。流体相ソルバーの高度化に先立ち、固相で採用する陰的 MPM の定式化と前処理・ソルバーの改善に取り組んだ。

#### 5.1.1 定式化

固体単相の陰的 MPM として、変位  $\mathbf{u}$  と圧力  $p$  を独立変数とする  $\mathbf{u}$ - $p$  混合定式化を採用した [3]。弱形式を線形化して得られる連立一次方程式は、

$$\begin{bmatrix} \mathbf{K}_{uu} & \mathbf{K}_{up} \\ \mathbf{K}_{pu} & \mathbf{K}_{pp} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{u} \\ \Delta p \end{Bmatrix} = - \begin{Bmatrix} \mathbf{R}_u \\ \mathbf{R}_p \end{Bmatrix} \quad (1)$$

の形で表され、Newton-Raphson 法によって収束計算を実施する。任意の基底関数に対して LBB 条件を満足しない問題を避けるため、Variational Multiscale (VMS) 法による安定化を適用した。VMS 法では解を有限要素スケールの成分と細スケール成分の和で表現し、細スケール成分を残差で近似することで安定化項を導出する。この手法は流体解析に用いている SUPG/PSPG 安定化有限要素法に近い安定化手法であり、整合性を持つ定式化となっている。

#### 5.1.2 前処理・線形ソルバーの高度化

従来の前処理は対角スケーリングのみであり、ソルバーは GBiCGStab( $s, L$ ) 法 (Krylov 部分空間法) を使用していた。係数行列の非対称性と条件数の悪化から、反復回数が多く収束性に問題があった。今年度は、ブロック不完全 LU 分解を前処理に採用し、ソルバーを BiCGStab 法に変更した。具体的には、係数行列をブロック分割して各ブロックに対する不完全 LU 分解を前処理として適用することで、係数行列の条件数を大幅に改善した。ただし、係数行列の非対角優位性が強く、ブロック不完全 LU 分解の破綻が起こるため、対角成分に定数を足す正則化を前処理行列にのみ施した。また、本検証を通して、ブロックサイズは 4、正則化係数  $1.0 \times 10^{-2}$  とした。

この検証として、Cook's membrane 問題 (ヤング係数 240.565 MPa, ポアソン比 0.4999 の微圧縮性材料, 格子長 8.0, 4.0, 2.0, 1.0 mm の 4 種, 1 格子あたり  $4 \times 4$  粒子, 2 次 Extended

B-spline (EBS) 基底関数) を用いた検証を行った。表 1 にそれぞれの前処理・ソルバーの組み合わせにおける計算時間を示す。ブロック不完全 LU 分解前処理 + BiCGStab の組み合わせは, GBiCGStab( $s, L$ ) + 対角スケーリングに比べて反復回数を約 500 回から約 30 回に削減し (図 1), 総計算時間を 635.6s から 100.8s へと約 6.3 倍短縮した。

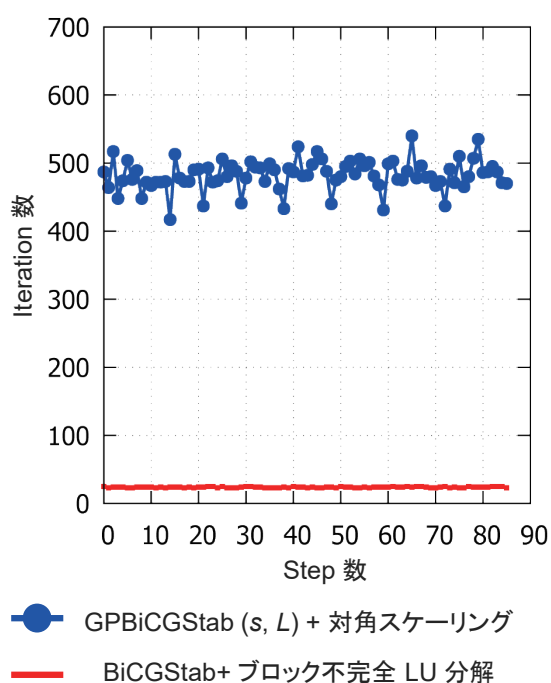


図 1 Cook's membrane 問題での線形ソルバーの反復回数の比較

表 1 Cook's membrane 問題での計算時間比較

時間 [s]	GBiCGStab( $s, L$ ) + 対角スケーリング	BiCGStab + ブロック不完全 LU 分解
Total	635.6	100.8
Solver + 前処理	568.1	28.07
前処理	0.168	13.29
Solver	567.4	14.25

加えて, 図 2 に示すように, 変位の収束性についても, FEM の参照解と MPM の結果が小

さい格子サイズへと変更するにしたがって良好に一致することを確認しており, 解析の妥当性を確認した。

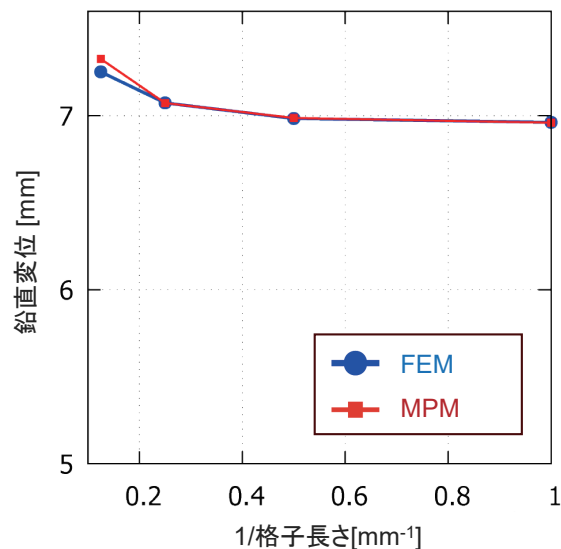


図 2 Cook's membrane 問題での変位の収束性

## 5.2 OpenACC による GPU-MPM の実装と大規模解析への適用

陽的 MPM の GPU 高速化を目的として, Fortran をベースとして OpenACC を用いた GPGPU 実装を行い, Miyabi-G 上で性能評価を実施した。

本実装では, GH200 Grace-Hopper Superchip 環境においてコンパイルオプションに `-gpu=cc90,mem:unified:nomanagedalloc` を用いて OpenACC による GPU 実行を行った。mem:unified は NVLink-C2C を介して CPU・GPU 双方のメモリを互いにアクセス可能などとして双方向に使用できる設定であり, nomanagedalloc は動的確保を暗黙的に cudaMallocManaged へ置き換えないことを指定するものである。これにより, ページフォルトをトリガーとしたページサイズごとの自動 CPU-GPU 間データ転送が起こらないようになり, 手間をかけることなく性能向上が期待できる。

### 5.2.1 実装方針

OpenACC は GPU 向けの指示型の並列化手法であり, OpenMP コードを比較的容易に置き換えられる利点がある. 本研究で使用しているコードは MPI 並列のみ実装済みであり, OpenMP が未実装であったため, そちらを実装したうえで, OpenACC の実装を行った. 本研究では, タイムループの中身のほぼすべてに対して OpenMP/OpenACC を実装した (図 3). その中でも特に重要となる MPM の処理は以下の 3 つである.

- P2N: 粒子から格子点へのマッピング (Algorithm 1 に相当する処理)
- 運動方程式の求解 (Algorithm 1 に相当する処理)
- 格子点から粒子への補間 (Algorithm 2 に相当する処理)

Algorithm 1 に相当する処理では, 複数の粒子が同一格子点に寄与するため競合状態 (race condition) が発生する問題がある. 本研究では各格子点に寄与する粒子をリスト化することで競合状態を回避する Algorithm 3 を採用することでこれを回避した. この方法は, ノードごとにコネクティビティを有する粒子のリストを保持する必要があり, 大量のメモリを消費するという欠点がある. これに対し, Miyabi-G の GPU では, 1 GPU あたりのメモリ容量が 96 GB と大きいため, この方法の欠点がある程度補うことができている. そのほかの方法として, 色分けによる回避方法も実施されているが, こちらについては今後の課題として検討している.

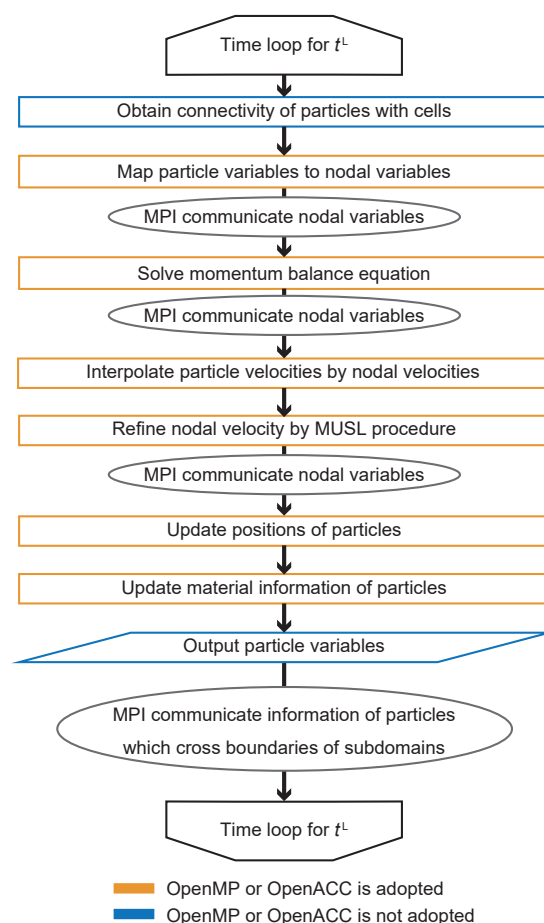


図 3 OpenMP/OpenACC を実装した MPM のフローチャート

---

#### Algorithm 1 P2N — scatter via particle loop

---

- 1:  $\Phi_I = 0$  for all  $I$
  - 2: **for**  $p = 1$  **to**  $n_p$  **do**
  - 3:   **for** each node  $I$  in the support of  $p$  **do**
  - 4:      $m_I \Phi_I += N_{Ip} m_p \Phi_p$
  - 5:     (atomic operation is required)
  - 6:   **end for**
  - 7: **end for**
  - 8:  $\Phi_I = m_I \Phi_I / m_I$  for all  $I$
-

**Algorithm 2** N2P — gather via particle loop

```

1:  $\Phi_p = 0$  for all  $p$ 
2: for  $p = 1$  to  $n_p$  do
3:   for each node  $I$  in the support of  $p$  do
4:      $\Phi_p += N_{Ip} \Phi_I$ 
5:   end for
6: end for

```

**Algorithm 3** N2G — gather via noda loop

```

1:  $\Phi_I = 0$  for all  $I$ 
2: for  $I = 1$  to  $n_I$  do
3:   for each node  $p$  in the support of  $I$  do
4:      $m_I \Phi_I += N_{Ip} m_p \Phi_p$ 
5:   end for
6: end for
7:  $\Phi_I = m_I \Phi_I / m_I$  for all  $I$ 

```

## 5.2.2 片持ち梁の振動解析による性能評価

解析領域  $4.0\text{ m} \times 1.0\text{ m} \times 1.0\text{ m}$  の片持ち梁モデル (図 4) を用い、その格子数と粒子数を  $\times 1 \sim \times 7$  に拡大した 5 種のモデル (表 2) で 10,000 ステップの計算を実施した。計算時間は、ファイル入出力を含み、解析の開始から終了までの全体の時間を MPI.Wtime で計測した。解析に用いた材料パラメータは表 3 に示す。

まず、基準となる図 4 に示すサイズのモデルについて、OpenMP (スレッド並列数 1, 2, 4, 8, 24, 36, 72) と OpenACC との計算時間の比較を行う。ただし、MPI と OpenMP によるハイブリッド並列を含めると実行ケース数が膨大になるため、この例題では MPI 並列は行わない。図-5 に解析時間、図-6 に OpenMP での並列化効率を示す。解析時間のグラフから、OpenMP においてもスレッド数を増やすことで計算時間が短縮されることがわかる。その一方、OpenACC を利用したケースは今回使用した計算機での 1 ノードあたりの最大スレッド数であ

表 2 各サイズのモデルの格子点数と粒子数

サイズ	格子点数	粒子数
$\times 1$	3864	$3.20 \times 10^4$
$\times 2$	25069	$2.56 \times 10^5$
$\times 3$	78624	$8.64 \times 10^5$
$\times 5$	342784	$4.00 \times 10^6$
$\times 7$	916344	$1.0976 \times 10^7$

表 3 片持ち梁の振動解析におけるパラメータ

パラメータ	記号	値
Young 率 [kPa]	$E$	$1.0 \times 10^3$
Poisson 比 [-]	$\nu$	0.30
密度 [kg/m <sup>3</sup> ]	$\rho$	$1.05 \times 10^3$
重力加速度 [m/s <sup>2</sup> ]	$g$	5.0

る 72 スレッドを用いた OpenMP よりも短い計算時間となっており、約 5 倍の高速化が得られている。また、並列化効率のグラフから、スレッド数が増えるにつれて効率が低下していることがわかる。そのため、CPU のみの解析の場合でも MPI 並列と組み合わせることでさらなる高速化が見込まれる。

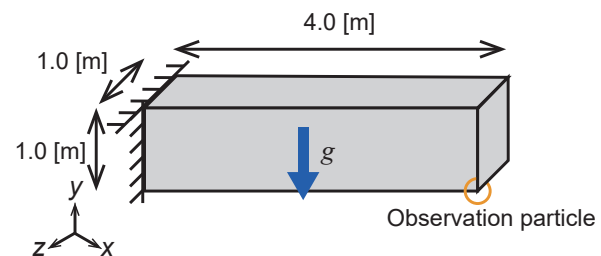


図 4 片持ち梁の解析モデル

次に、図-4 に示すサイズのモデルを基準として、格子数と粒子数を  $\times 1 \sim \times 7$  に拡大したモデルに関する解析の計算時間の比較を行う。ただし、解析ケース数の削減のため、CPU による OpenMP スレッド並列数は 72 とし、GPU を

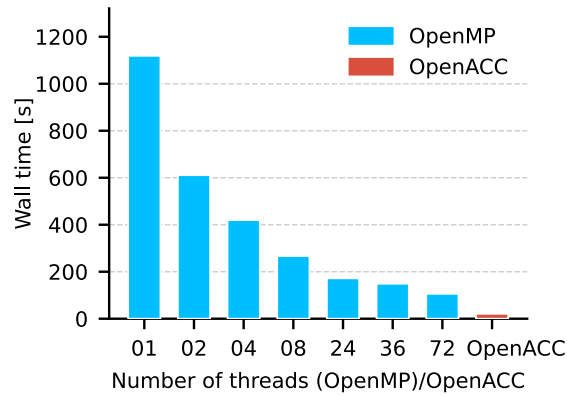


図5 各スレッド数での CPU と GPU による計算時間の比較

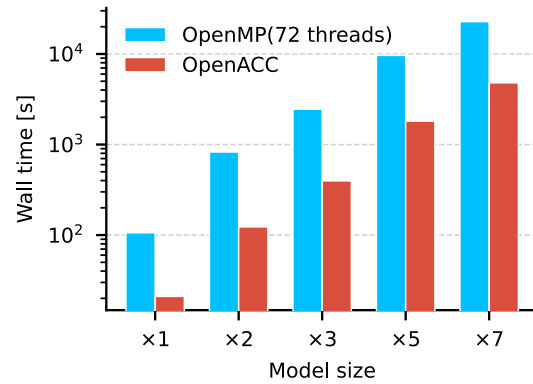


図7 各モデルサイズでの OpenMP (72 スレッド) と OpenACC による計算時間の比較

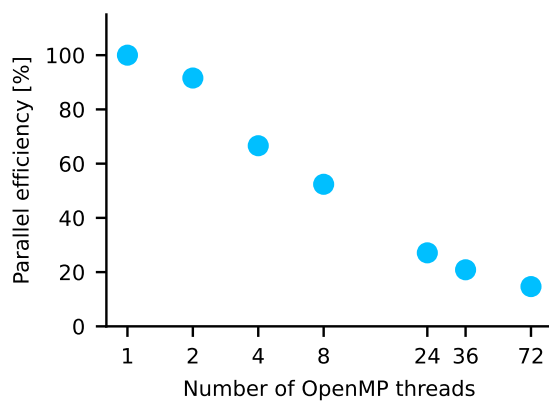


図6 各スレッド数での OpenMP の並列化効率

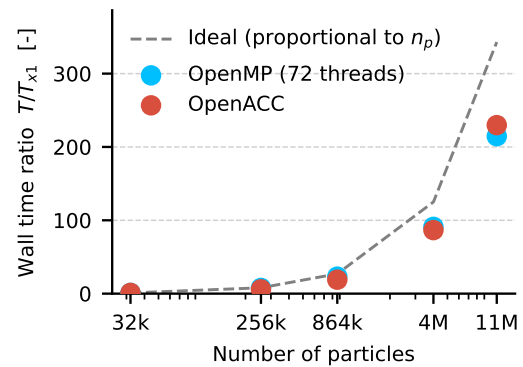


図8 サイズ ×1 に対する各サイズの計算時間の比較

用いた OpenACC による計算時間と比較する。解析時間を示す図-7 から、サイズ数が増大しても常に OpenACC を用いたケースの方が短い計算時間となっており、サイズ ×2 のケースが約 6.7 倍で最大の高速化が得られており、サイズが最大の ×7 のケースでも約 4.7 倍の高速化が得られている。

最後に、サイズ ×1 に対する各サイズの計算時間を図-8 に示す。ただし、領域サイズが  $n$  倍になると格子点数は  $n^3$  倍、粒子数も  $n^3$  倍になることに加えて、陽解法 MPM では、計算時間は格子点数よりも粒子数に比例して増加する傾

向があるため、図-8 の横軸はサイズの倍率ではなく、粒子数で示した。この図から、864000 個の粒子を用いたサイズ ×3 のモデルまでは、概ねサイズの増大に伴う計算時間の増大が見られるが、それ以降ではスレッド数に対して粒子数が多くなったために、並列化性能が飽和したとみられる。このことから、概ね 10 万粒子程度を 1 GPU で処理するのが、本コードにおいては最も効率的であると考えられる。

### 5.2.3 実大規模斜面崩壊解析

Hidano ら [1] の解析条件を模擬した実大規模斜面崩壊解析を実施した。ただし、パラメータを一部変更しており、WLS を用いた B-spline 形

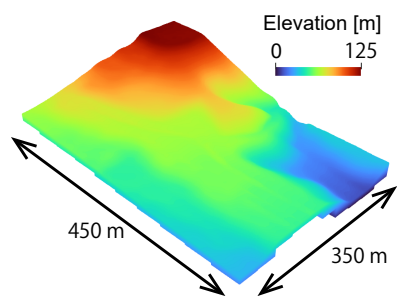


図9 実大規模斜面崩壊解析のモデル

状関数を用いたため、Open-knot B-spline 形状関数を用いた既往研究 [1] の解析とは完全に同一の条件ではないことに留意されたいが、そのほかの条件は準拠した。解析モデルは、図 9 に示すように、格子数 18,241,146、粒子数 17,876,648 であり、DEM データに基づく地形上の表層・基層の 2 層構造 (表 4) を用いた。構成則は、弾性モデルに、Hencky 超弾性を採用し、降伏基準に Drucker-Prager を採用した。仮想的な地震動として  $x, y, z$  方向に振幅  $9.8 \text{ m/s}^2$ 、周波数  $1 \text{ Hz}$  の正弦波加速度を入力 (時刻  $10 \sim 90 \text{ s}$ ) し、時間刻み幅  $\Delta t = 1.0 \times 10^{-3} \text{ s}$  で  $100,000$  ステップ ( $100 \text{ s}$ ) を計算した。

表 4 実大規模斜面崩壊解析のパラメータ

パラメータ		表層	基層
密度 [ $\text{kg/m}^3$ ]	$\rho$	2500	2500
間隙率 [-]	$\theta$	0.5	0.5
ヤング率 [MPa]	$E$	4.0	$1.0 \times 10^3$
Poisson 比 [-]	$\nu$	0.3	0.3
内部摩擦角 [ $^\circ$ ]	$\phi$	30.0	55.0
ダイレイタンス角 [ $^\circ$ ]	$\psi$	0.0	55.0
粘着力 [kPa]	$c$	5.0	300.0

比較対象とする Hidano ら [1] で実施した動的負荷分散を適用した MPI 並列計算での、使用したコア数は 5,760 コアであり、全体の計算時間は  $22,745 \text{ s}$  であった。MPI + OpenACC のハイブリッド並列で 9 ノード ( $x, z$  方向に  $3 \times 3$  分割) および 25 ノード ( $5 \times 5$  分割) の計算を

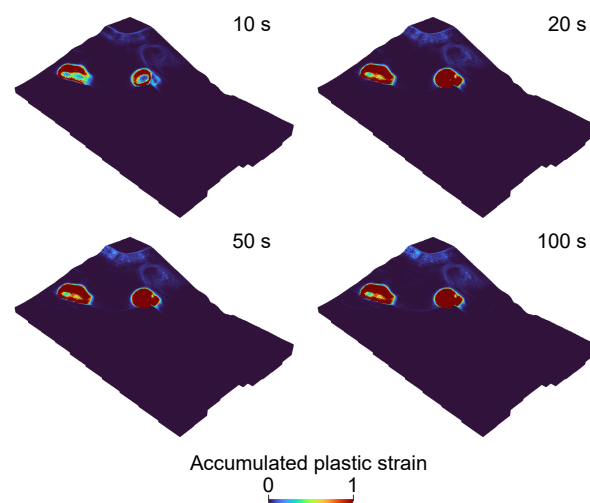


図 10 実大規模斜面崩壊解析の累積塑性ひずみの分布

Miyabi-G で実施した結果、それぞれの計算時間は約  $1.69 \times 10^4$  秒 (約 4.6 時間)、約  $7.97 \times 10^3$  秒 (2.2 時間) であり、約 1.34 倍、約 2.85 倍の高速化が得られた。解析結果は図 10 に示す通りであり、地震動入力に伴い時刻  $20 \text{ s}$  ごろまで表層部に塑性ひずみが集積する傾向が確認された。ただし、その後の進展については解析パラメータの見直しや、より現象に即したモデル開発が今後の課題となっている。

## 6 進捗状況の自己評価と今後の展望

今年度 (jh250053) の申請時計画は、(1) 効率化した固体相-流体相連成解析による 3 次元解析 (Palu 海底地すべりおよび起因津波) と、(2) エビスサーキット土砂災害の再現解析の 2 点であった。

計画 (1) については、流体相ソルバー高度化の前段として固体単相の陰的 MPM におけるブロック不完全 LU 分解前処理 + BiCGStab を実装し、Cook's membrane 問題での有効性を確認した (約 6.3 倍の高速化)。固液連成の Hybrid MPM-FEM への適用と Palu 海底地すべりの

実大規模解析については、引き続き計算アルゴリズムの見直しと合わせて取り組む予定である。

計画 (2) については、陽的 MPM の OpenACC による GPU 実装を完了し、Miyabi-G を用いた片持ち梁の性能評価（最大 6.7 倍の高速化）および実大規模斜面崩壊解析（25 ノードで 2.85 倍の高速化）を達成した。今後は、GPU 化を見据えた OpenMP の最適化、現象を捉えたモデルの開発に取り組む予定である。

## 参考文献

- [1] S. Hidano, S. Pan, K. Yoshida, R. Nomura, Y. Miki, M. Kawai, S. Moriguchi, K. Nakajima, K. Terada, B-spline-based material point method with dynamic load balancing technique for large-scale simulation, *Engineering with Computers*, Vol. 41, pp. 1619–1636, 2025.
- [2] S. Pan, S. Hidano, R. Nomura, L. Jing, Y. Miki, M. Kawai, S. Moriguchi, K. Nakajima, K. Terada, Speeding up MPM calculations in MPM–FEM hybrid method incorporating inconsistent domain decomposition patterns for solid and fluid phases, *Computational Mechanics*, 2025.
- [3] R. Sugai, R. Nomura, S. Moriguchi, K. Terada, Extended B-spline-based mixed material point method stabilized by the variational multiscale method for compressible and nearly incompressible hyperelastic materials, *Advances in Computational Science and Engineering*, Vol. 4, pp. 85–118, 2025.