

jh150048-NA26

## 動的負荷分散による GPU スパコンを用いた粒子法の大規模シミュレーション手法の開発 II - 流体・構造連成計算への適用 -

青木 尊之 (東京工業大学)

**概要** DEM と SPH を使った流体構造連成計算について、動的負荷分散を適用することにより GPU スパコン上で大規模粒子シミュレーションを実行することを可能にした。木構造に基づいて計算領域を細分化し、空間充填曲線により分割領域内の粒子数の均一化を図った。物体は流体粒子と同じサイズの球形粒子を剛体連結することで表現し、同じ手続きで相互作用を計算する。GPU 内の粒子番号のソートにより相互作用計算を高速化し、物体にかかる力とトルクの総和計算の実装を検討した。大規模計算で並列性能が低下するスライス・グリッド法に基づいた動的領域分割に対し、空間充填曲線としてヒルベルト曲線、モートン曲線、ペアノ曲線に基づいた領域分割を検証し、大幅な性能向上を達成した。東京工業大学の TSUBAME2.5 の 256 個の GPU を用い、総粒子数 1.1 億個、1 万個の複雑形状をした瓦礫 (物体構成粒子数 210 万個) が浮遊する津波シミュレーションを流体・構造連成計算として実行した。

### 1. 共同研究に関する情報

#### (1) 共同研究を実施した拠点名

東京工業大学・学術国際情報センター

#### (2) 共同研究分野

- 超大規模数値計算系応用分野
- 超大規模データ処理系応用分野
- 超大容量ネットワーク技術分野
- 超大規模情報システム関連研究分野

#### (3) 参加研究者の役割分担

青木尊之 (東工大) DEM 計算、SPH コードの開発と動的負荷分散アルゴリズムの検討、全体総括

西浦泰介 (海洋研究開発機構) DEM 相互作用の最適化、SPH との連成計算

今井陽介 (東北大) SPH 法の計算手法・精度とアルゴリズム

森口周二 (東北大) DEM 計算の土質力学への適用性の検討とクォータニオンの導入

都築怜理 (東工大) DEM, SPH コード開発と、TSUBAME 2.5 上での大規模シミュレーションの実施、ポスト処理

渡辺 勢也 (東工大) 非球形 DEM 計算の実装

井元 佑介 (九州大) 粒子法 (SPH) による計算手法の改良

### 2. 研究の目的と意義

土質力学、斜面災害、各種の化学工学、錠剤製造等のさまざまなプロセスにおいて、粒子法を用いたシミュレーションの重要性は広く認識されている。

流体や粉体のシミュレーションにおいて粒子法を用いる利点は、ラグランジュ的に時間積分し空間格子を用いないため、流体や粉体の空間的な局在や時間的に粒子の空間分布の変化にも問題なく対応できる点にある。また、複雑な境界条件も粒子直径に近い空間精度で容易に取り扱うことができる。一方、粒子位置で流体方程式を解くため、1 個の粒子を計算するために周囲の 100 個近い (SPH 法の場合) 粒子の位置や速度を参照する必要があるため、浮動小数点演算量およびメモリアクセス量ともに多い。また、粒子位置がランダムであるため、メモリアクセスの効率も悪い。さらに計算精度がせいぜい 2 次であることと、粒子分布が少ない領域での計算精度の低下は深刻な問題である。

高いレイノルズ数の流体シミュレーションでは、物体近傍に薄い境界層が発達し、そこを解像するためには高い空間解像度が局所的に要求される。格子法であれば適合格子法により解決することができるが、粒子法の場合は空間解像度を場所に応じて適合させることは非常に困難である。また、高いレイノルズ数の流れに特有の乱流に対しても、LES (Large Eddy Simulation) のモデルを適切に導入することのできる計算精度を有していないという欠点がある。

粒子法が余り高い計算精度を有していないことは、逆に物体との相互作用に対しても低精度の計算を許容することができ、格子法では容易に扱えない流体構造連成計算を比較的容易に実現することができる利点がある。

本研究では、計算資源を豊富に利用できるスパコンで大規模な粒子法計算を実現し、粒子法の計算精度の低さを補い、格子法では適用困難な問題に対して粒子法の有効性を示すことを目的とする。計算負荷の高い粒子法の大規模計算を行うには、① 演算ア

クセラレータとして GPU を用い、② ノードに分散した複数 GPU を効率的に使うために動的負荷分散を行う必要がある。それらにより実用問題に対する大規模粒子法の有効性を示すことを目的とする。

平成 26 年度の研究目的はスライス・グリッド法を動的領域分割法へ導入し、粒子法の一つである個別要素法による粉体の大規模シミュレーションを実現することであった。さらに、個別要素法と計算アルゴリズムが酷似している SPH(Smoothed Particle Hydrodynamics) にも動的負荷分散を適用し、より応用範囲の広い流体解析でも大規模粒子法計算を行うことを目的としていた。

【今年度の研究目的】平成 27 年度は、粒子法の利点である流体構造連成問題に動的負荷分散法を適用し、平成 26 年度に開発した個別要素法による複数の球形粒子で構成される複雑形状の物体の運動と SPH 法による流体シミュレーションを連結させ、動的負荷分散による大規模な流体構造連成シミュレーションを実現する方法を示すことを目的とする。

各計算領域（ノード）間の計算負荷を均一にするために、木構造に基づいた空間充填曲線による領域分割を導入することにより、スライス・グリッド法で領域分割する際に問題となる高アスペクト比を大きく改善させる。一方、分割領域内の粒子数を一定に保つためには、領域再分割の際にリーフのマイグレーションが発生する。この通信コストが大きいいため、マイグレーションの頻度と負荷バランスのトレードオフが生じ、本研究ではマイグレーションの頻度と負荷不均一性を調べる。これらの手法を導入することで、流体構造連成計算として非常にチャレンジングなテーマであるサスペンション・フローの大規模シミュレーションを実現する。

### 3. 当拠点公募型共同研究として実施した意義

GPU スパコンにおいて、ステンシル計算の大規模実用アプリケーションで実績のある研究代表者が、地盤工学およびバイオメカニクス分野で DEM 法および SPH 法の計算手法の研究実績を持つ研究者らと共同研究することができ、粒子法の計算精度や計算アルゴリズムの知見を取り入れた実問題の大規模粒子計算のアプリケーション開発をすることができる。

### 4. 前年度までに得られた研究成果の概要

GPU スパコンにおいて、DEM 法や SPH 法の近接相互作用に基づいた大規模粒子シミュレーションを可能にする計算手法の開発を行った。スライス・グリッド法を用いて GPU 間の動的な負荷分散をそれぞれに適用し、GPU スパコン Tsubame 2.5 を用いた 1~20 億粒子の粒子計算に対するスケールリングを検証した。粉体および流体の実問題に対して本提案手法を適用し、DEM による粉体計算ではこれまで実現できなかった現実の砂粒子サイズを用いた 3 次元バンカーショット計算を実行する

ことができた。SPH による計算では、複雑形状の物体を含む 1200 万粒子を用いた流体シミュレーションを実現し、本手法の適用性の高さを示した。また、スライス・グリッド法による動的負荷分散の問題点を明らかにし、木構造に基づいた領域細分化に対して空間充填曲線を用いる動的負荷分散のプロトタイプの実装を行った。

## 5. 今年度の研究成果の詳細

### 5.1 流体と構造物の相互作用

図 1 に示すように、計算領域内で系を構成する流体、壁、物体の全てを粒子により表現する。物体は流体粒子と同じ大きさの球形粒子を剛体連結してモデル化する。3 種類の粒子はいずれも同じデータ構造を持ち、これらの粒子間の相互作用力を計算し時間積分することで流体構造連成計算を行う。

#### (1) 流体粒子間の相互作用

SPH 法のカーネル半径内の粒子から受ける相互作用を計算し、粒子の速度や位置、圧力や密度を更新する。

#### (2) 流体粒子-物体構成粒子・壁粒子間の相互作用

流体粒子計算の中で壁粒子は静止した境界条件として、物体構成粒子は移動境界条件として扱う。

#### (3) 物体構成粒子-流体粒子間の相互作用

物体構成粒子は、その位置での圧力およびせん断応力を流体粒子から計算し、物体の重心に対して垂直方向と水平方向に分解する。

#### (4) 物体構成粒子-物体構成粒子間の相互作用

複雑形状をした物体間の衝突について、接触している物体構成粒子間で DEM の衝突モデルによる反発と摩擦力を計算する。これも物体の重心に対して垂直方向と水平方向に分解する。

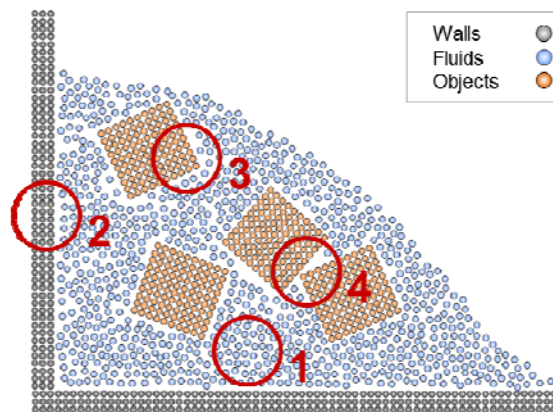


図 1 粒子法による流体構造連成計算の相互作用

(1)と(2)の流体粒子の時間発展は 2 次精度の予測子-修正子法を用いる。(3)と(4)の合力で物体にかかる力とトルクが求められ、物体の並進運動と回転運動が時間積分により求められる。 $i$  番目の物体構成粒子の受ける力を  $f_i$  とし、各粒子の剛体重心からの相対座標を  $\vec{r}_i$  とすると、物体の質量  $M$  に対する物体の並進速度  $v$ 、角速度  $\omega$ 、角運動量  $L$  は以下のように

求められる。

$$M \frac{dv}{dt} = \sum_{i \in \text{solid}} f_i$$

$$\frac{dL}{dt} = \sum_{i \in \text{solid}} \tilde{r}_i \times f_i$$

$$\omega = I^{-1}L$$

$I^{-1}$  は各時刻の慣性行列の逆行列であり、初期の慣性行列の逆行列から回転行列により形式的に求められる。また、物体の姿勢の管理にはクォータニオンを導入する。

### 5.2 Linked-list を用いた流体粒子の相互作用計算におけるソートの有効性評価

DEM や SPH などの近接相互作用に基づく粒子法では、全粒子との相互作用計算を行うことは非効率的である。計算領域を空間格子（セル）に分割し、自身と隣接するセルに属する粒子とのみ相互作用計算を行う方法（セル分割法）を用いる。SPH 法のカーネル半径は粒子間隔の 3~4 倍程度であり、セルの一辺の長さをカーネル半径以上にする必要がある。最大で 100 個程度の粒子が 1 セルに格納される可能性がある。静的に全セル用のメモリを確保し、すべての粒子の粒子番号をそれぞれの粒子が属するセルに登録する実装ではしばしばメモリ不足を引き起こすため、各セルには 1 つの粒子の番号のみを登録し、同一セル内の粒子は自分以外の粒子番号を数珠つなぎに保持する Linked-list を GPU 上で導入する。

粒子法では、計算が進むにつれて粒子が移動するためにカーネル半径内の粒子のメモリ・アドレスはランダムになる。CUDA では一度に 16 連続アドレスの配列データを読み出すが、同一セル内の粒子がメモリ上に連続に存在していないためにメモリ読み出し回数が増え、メモリアクセスの効率が大幅に低下する。そこで、図 2 に示すように粒子データをセル番号順にソートする。

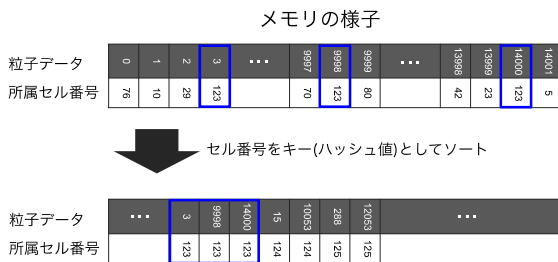


図 2 データのソート

Linked-list 法におけるセル番号によるソートは、5.1 に記述した 4 種類の相互作用計算における流体粒子の時間発展((1)と(2))において約 10 倍以上に計算を高速化し、その有効性を確認した。

### 5.3 物体にかかる力とトルクの計算

複数の球形粒子の剛体連結で構成される物体の時間発展は、物体構成粒子が流体および他の物体から

受ける力とトルクを総和し、各物体に対して時間積分が行われる。計算領域内に全部で  $N_{\text{particle}}$  個の粒子があり、その内の  $N_{\text{obj}}$  個が物体構成粒子であるとすると、物体の時間積分を行う疑似コードは以下のように書ける。

```

Algorithm
for k = 0,1,2... to N_obj do
  for j = 0,1,2... to N_particle do
    if particle[j] ∈ object[k] then
      object[k].force += particle.force[j]
      object[k].torque += particle.torque[j]
    end if
  end for
  update object[k].quaternion
  update object[k].velocity
end for
    
```

条件分岐は、粒子  $j$  が物体  $k$  の構成粒子であるかどうかを判定している。GPU 計算では、(A) 物体数についての外側のループをスレッド並列化する場合と、(B) 領域内の粒子数についての内側のループをスレッド並列化する場合の 2 通りの実装が考えられる。(A) を実装する場合、外側のループに CUDA の

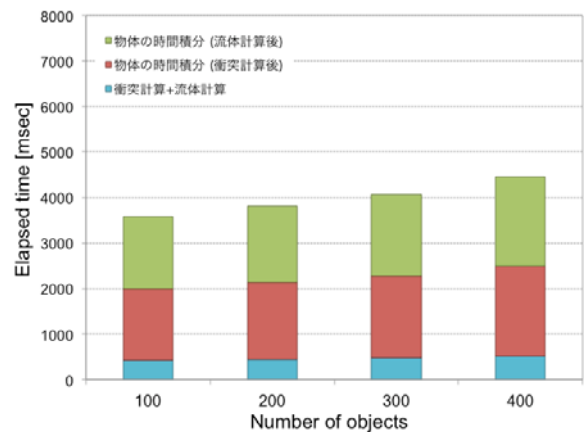


図 3 実装 (A) を用いた場合の計算時間の内訳

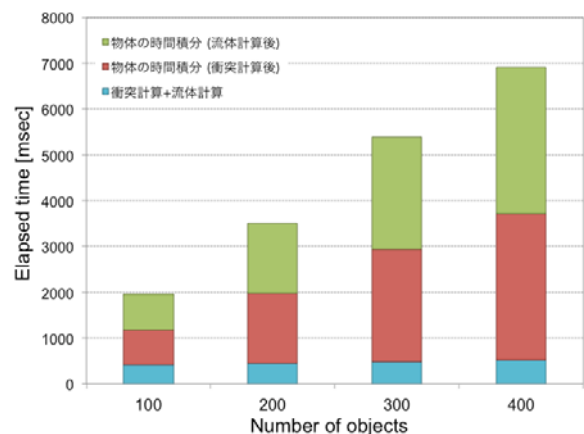


図 4 実装 (B) を用いた場合の計算時間の内訳

スレッドを並列に割り当て、各スレッドが内側のループの処理を行うように実装する。(B) を実装する場合には予め領域内の粒子数分の配列を用意し、GPU のスレッド並列で受ける力を書き込む。外側のループで  $k$  番目の物体の総和計算に Thrust ライブラリ

の inclusive scan を用いる。

表 1 流体計算の実行性能

計算項目	単精度	倍精度
予測子:	102.6 (2.59)	88.1 (6.72)
粒子粘性:	94.3 (2.38)	82.5 (6.29)
粒子圧力:	104.7 (2.65)	122.1 (9.32)
座標の修正子:	67.4 (1.70)	91.1 (6.95)
圧力補間:	90.6 (2.29)	106.8 (8.15)
速度の修正子:	67.2 (1.70)	91.7 (7.00)

単位: GFlops (括弧内は理論ピーク性能に対する割合(%))

総粒子数 140 万個を用い、計算領域内に 1 個当たり 1000 個の物体構成粒子からなる立方体を 100, 200, 300, 400 個と数を変えて配置して検証を行った。(A) と (B) のそれぞれの実装に対して、1 ステップ当たりの計算時間を測定した結果をそれぞれ図 3 と図 4 に示す。GPU 上で物体番号について並列化を行う実装 (A) は物体数が増えても緩やかに計算時間が増加するのに対し、物体数について並列化されていない実装 (B) を用いた場合は物体数に比例して計算時間が増加する。物体数が 300 個より多い場合は実装 (A) を用いる方がよいことが分かる。

図 3 や図 4 における流体計算の所要時間(水色の部分)について、その内訳を表 1 に示す。NVIDIA Tesla K20X におけるピーク性能比は、単精度演算で 2.65%、倍精度演算で 9.32% となっている。ランダムなメモリアクセスが実行性能を低下させているが、さらに高速化を進める余地があることが分かる。

### 5.4 空間充填曲線による動的負荷分散

平成 26 年度に行ったスライス・グリッド法による領域分割の欠点である分割領域のアスペクト比が高いことと、分割された領域が多数の隣接領域と接続する問題を解決するために、木構造に基づいて領域を細分化し、空間充填曲線を用いて動的負荷分散を行った。重力多体問題で良く使われる Kd-tree また

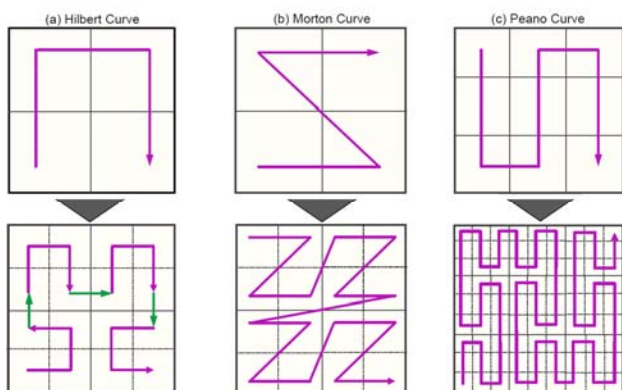


図 5 本研究で用いた 3 種類の空間充填曲線

は ORB (Orthogonal Recursive Bisection) は分割数が 2 のべき乗となるため、実用的な問題に適さない。

4 分木 (3 次元では 8 分木) で計算領域を再帰的に分割し、分割領域の中の粒子数が指定した数以下に

なるまで細分化を繰り返す。分割領域を空間充填曲線で辿り、線上の粒子数の和が均等になるように曲線を分割すると空間も分割される。本年度はヒルベルト曲線、モートン曲線、ペアノ曲線を用いた領域分割を検討し、比較を行った。ただし、ペアノ曲線の場合は空間を 9 分木 (3 次元では 27 分木) で分割する。

例として、現在の分割領域に対して粒子が図 5 左上のように分布している場合の領域再分割の手続きを示す。まず、図 6 右上のように各分割領域で粒子数テーブルを作成しマスターに集める。通信量を抑えるために粒子が存在するセル(黄色の部分)のみを GPU 上でパッキングして通信する。マスターでは、集めた粒子データから図 5 左下のように計算領域全体の粒子数テーブルを作成する。領域番号を格納するための粒子数リストと同じ大きさのテーブルを用意し、各リーフに対して新しい分割領域の番号を割り振りながら、同時に空間格子において各リーフが占める部分の格子に領域番号を格納して図 5 右下のような新しい領域番号テーブルを作成する。

複数 GPU を用いた粒子計算では、時間発展により粒子が領域外に移動する。GPU のデバイス・メモリに置かれた領域分割テーブルを参照して移動先の判定を行う。本計算では、ツリーの構築はホスト (CPU) 側で行い、領域番号テーブルもホスト側に確保されている。領域分割完了後、予め領域番号テーブルをデバイス側にコピーしておく。この際、CPU-GPU 間の通信量を抑えるため、領域分割前のテーブルと更新後の領域分割テーブルと比較して更新されたリーフ領域部分に該当するセルのみを GPU に転送する。

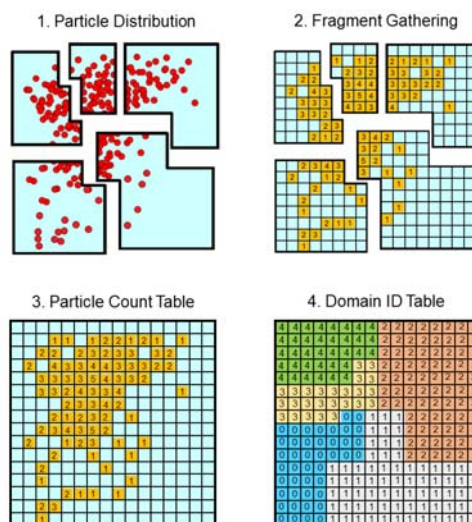


図 6 粒子数/領域番号テーブルを利用した領域分割の複数 GPU 実装

DEM や SPH などの粒子法の計算では、ステンシル計算と同じように隣接された分割領域の境界近傍粒子の情報も必要になる。各分割領域の裾領域(ハロー)を隣接する分割領域にコピーする必要がある。複数 GPU を使う場合の領域分割の実装は、粒子数、領域番号、再分割前の領域番号テーブルという 3 つの

整数型の空間格子を利用するため、設定可能な細分化の最大の深さがメモリ容量の制約を受ける。図 7 はそれぞれの曲線において、最大の深さとテーブル生成に必要なメモリ容量の総量の関係を示す。粒子の物理情報を格納する配列なども考慮すると、メモリ容量が 6GB の GPU (NVIDIA Tesla K20X) の場合は、ヒルベルト曲線とモートン曲線の場合に 9 階層、ペアノ曲線の場合に 6 階層程度が上限だとわかる。

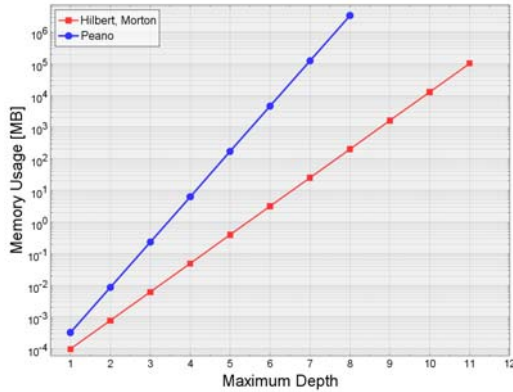


図 7 粒子数/領域番号テーブルに要するメモリ消費量

### 5.5 ヒルベルト、モートン、ペアノ空間充填曲線による領域分割とスライス・グリッド法の比較

ダム崩壊問題に対し並列性能を検証した。鉛直方向と水平な方向に対してスライス・グリッド法および 2 次元空間充填曲線を用いた 2 次元動的領域分割を比較した。1 台の GPU あたりに 441, 535 個の粒子を割り当て、GPU の台数に比例させて粒子を敷き詰める水平方向の面積を拡大して弱スケールリングを実行した結果を図 8 に示す。時間刻み幅を  $5.0 \times 10^{-4}$  とし、初期の 2,000 ステップに対して計算時間の測定を行い、平均の 1 ステップあたりの計算時間を求め性能値を算出している。

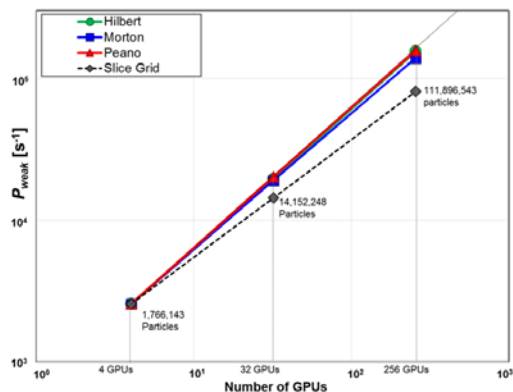


図 8 スライス・グリッド法と 2 次元空間充填曲線による領域分割の GPU 計算の強スケールリング

並列化効率を比較するために始点である 4 GPU の性能値を 2 次元ペアノ曲線の場合に規格化して示している。256 GPU を 4 GPU の測定値 (2 次元ペアノ曲線) から外挿した理想性能と比べ、2 次元ペアノ曲線を用いた場合に 96%、2 次元ヒルベルト曲線を用いた

場合に 95%、2 次元モートン曲線の場合に 85% の並列化効率を達成した。一方、スライス・グリッド法を用いた場合は 49% であり、空間充填曲線を用いた場合に大幅に性能向上できることが分かった。モートン曲線の接続数がヒルベルト曲線およびペアノ曲線に対して増えているのは、モートン曲線は空間的に離れた領域にジャンプしてしまうのに対し、後者は自身の曲線とは交差しないために領域が局所化する点が有利に働くと考えられる。

表 2 小領域間の最大接続数の変化

	Hilbert	Morton	Peano	Slice-grid
4GPU	3	3	3	4
32GPU	9	11	8	10
256GPU	9	14	9	34

分割された小領域間の通信がスケールリングを悪くしていることは明らかであり、その 97% 以上がハロー領域の通信時間である。小領域間の最大接続数を表 2 に示す。空間充填曲線を用いた場合、分割数が増えても 32 GPU と 256 GPU とで最大接続数が変わらないが、スライス・グリッド法の場合は分割数の増加とともに最大接続数も増え続けるため、通信時間が増加すると考えられる。

計算条件を固定して GPU 数を増やす強スケールリング (図 9) においても、空間充填曲線を用いることによりスライス・グリッド法より GPU 数が増加するにつれて並列化効率を大幅に向上させることができた。しかし、通信時間が支配的になるためにスケールリングが低下する以前に、既に理想的な勾配から低下しており、計算と通信のオーバーラップの導入による性能改善の余地がある。

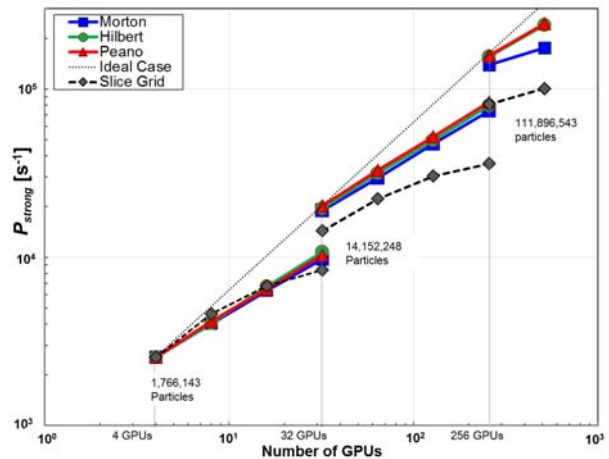


図 9 スライス・グリッド法と 2 次元空間充填曲線による領域分割の GPU 計算の強スケールリング

空間充填曲線による領域分割を用いた場合の計算負荷の不均一については、各セル内の粒子数が所定の数以下になるまで再帰的にセルを分割することにより制御することが可能である (図 10)。ペアノ曲線は 27 分木を用い局所的に細かい分割を行うために細分化の階層が浅く、均等な粒子数になるような領域分割を容易にしている。

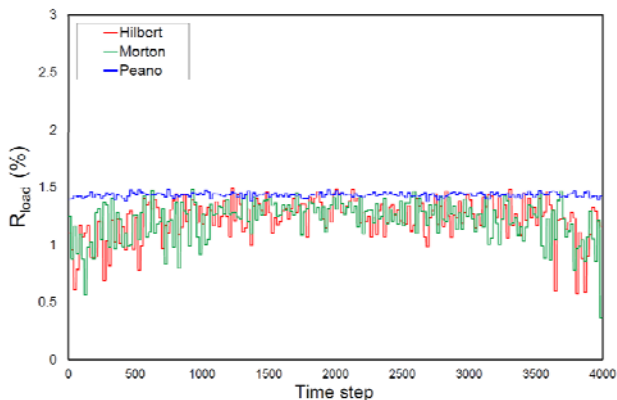


図 10 分割領域間の負荷の不均一

111, 896, 543 粒子, 256 GPU による流体計算を実行した際の、領域の再分割 1 回あたりの計算時間の内訳を図 11 に示す。10 ステップに 1 回の頻度で領域の再分割を実行した場合、リーフ・マイグレーションを含めても 1 ステップあたりに換算すると最大で 275 msec となる。1 ステップの計算時間が約 800 msec であることを考えると、100 ステップ以上に 1 回領域の再分割を行う場合には、オーバーヘッドはかなり小さいことが分かる。

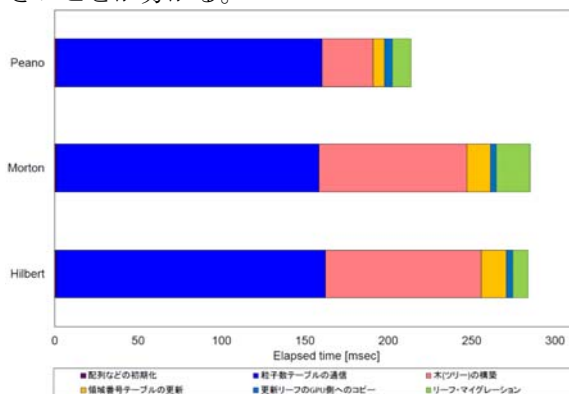


図 11 領域の再分割に要する計算時間

スライス・グリッド法による 3 次元分割は極めて煩雑であり現実的ではない。一方、空間充填曲線を用いる場合には 3 次元曲線を用いることにより 3 次元空間分割を行うことができる。256 分割した場合の各時刻における隣接小領域間の最大接続数を曲線ごとに比較した結果を図 12 に示す。モートン曲線を用

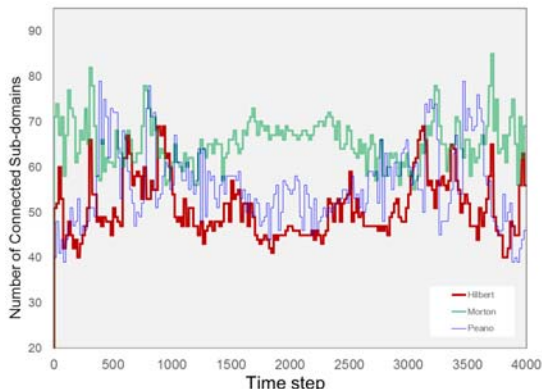


図 12 各曲線における最大接続数の変化

いた場合、最大接続数は 55 から 85 の間を変動しており、3 種類の曲線の中で一番接続数が多い。ペアノ曲線では 39 から 79 の間を変動しており、各時刻における最大接続数の幅が広い。しかし、2 次元分割の表 2 と比べると 3 次元分割は接続数が圧倒的に多く、並列性能も大幅に低下することが分かった。

ヒルベルト曲線、モートン曲線、ペアノ曲線を使って、典型的なベンチマークであるダム崩壊問題で動的負荷分散を検証した。図 13 に示すように、均一な負荷になるように領域分割を行うことは可能であるが、領域の形状は複雑になる。3 次元領域分割がどうしても必要な場合は、ハロー領域の粒子情報の通信に要するコストの低減を検討する必要がある。

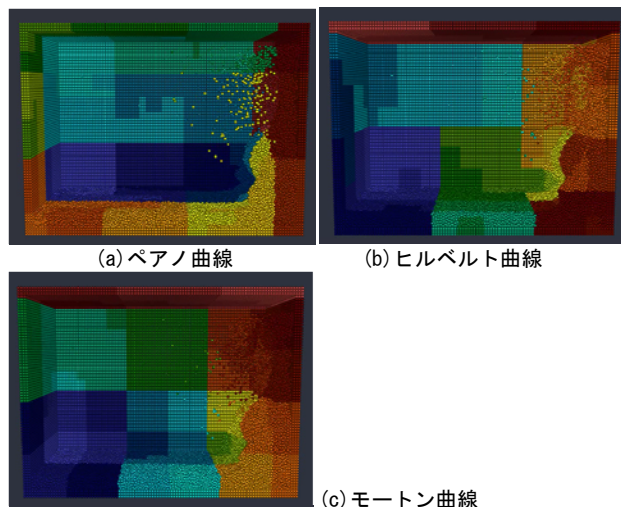


図 13 ダム崩壊問題における空間充填曲線による動的領域分割

### 5.6 TSUBAME2.5 における大規模計算

東京工業大学・学術国際情報センターの GPU スパコン TSUBAME2.5 の複数 GPU を用いた粒子法によるサスペンション・フローの大規模計算を行った。計算領域の縦×横×高さを 180m×160m×20m とし、深さ 2.0m の静止した水を張り、そこに合計 10,368 個の浮遊する物体を配置する。全部で 117,561,285 個 (流体粒子が 93,887,932 個、壁粒子が 21,535,585 個、物体構成粒子の総数が 2,137,768 個) の粒子を用いた。各物体は図 14 に示すように CAD データから距離関数に基づいて作成した 19 個~472 個の球形粒子で構成されている。津波が押し寄せてくる状況を想定し、左側に高さ 10 m の水柱を設定している。

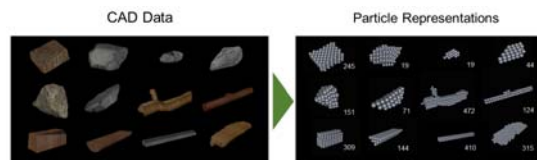


図 14 瓦礫の CAD データと粒子群データの変換

256 個の GPU を使い、時間刻み幅を  $5.0 \times 10^{-4}$  sec として 20,000 ステップを 100 時間かけて計算した。物理時間で 10 秒経過後のスナップショットを図 15 に示す。1 億 1,750 万個の粒子のうち壁粒子以外の流

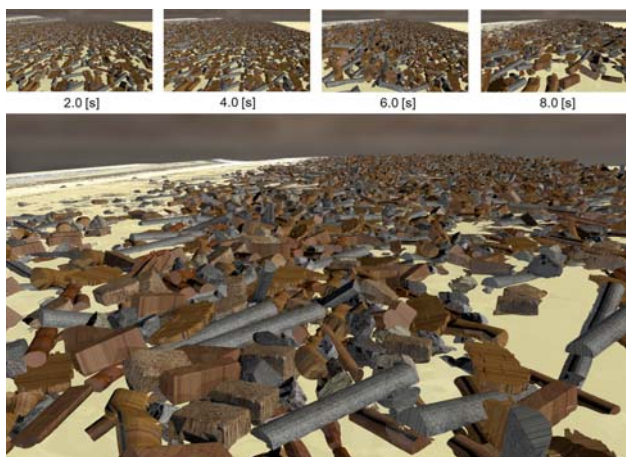


図 15 10,368 個の瓦礫を含む 1 億 1,750 万粒子による大規模津波シミュレーション

体粒子と物体構成粒子を所属領域ごとに 17 種類の色で塗り分けて示した結果を図 16 に示す。ヒルベルト空間充填曲線を用いた 3 次元の動的領域分割を 50 ステップ毎に実行している。

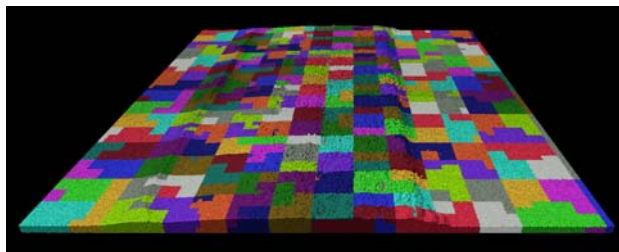


図 16 ヒルベルト曲線に基づいた 256 領域分割

主要な計算項目の 1 ステップ当たりの計算時間の内訳を図 17 に示す。水色の部分が流体の計算時間であり、橙色はそのためのハロー領域の通信である。橙色部分の合計が水色部分の 4 倍近くあり、物体を含まない流体計算部分だけでも通信に律速されていることが分かる。

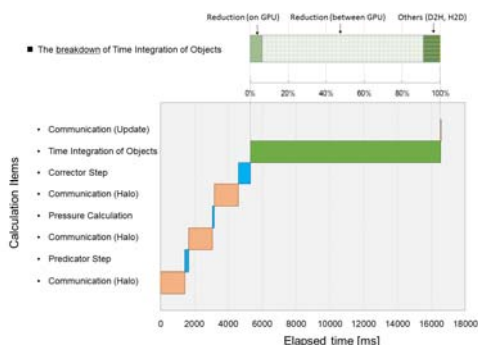


図 17 単位ステップの主要項目の計算時間の内訳

緑色部分は物体の時間積分に要する計算時間であり、全体の 67% を占めている。図 17 の上部には物体の時間積分の内訳を示している。各物体の構成粒子の保持する座標、力、トルクについての GPU 上での Reduction と、それらを GPU 間で Reduction させるのにかかる時間を合わせると物体の時間積分全体の 90% を占めていることが明らかになった。

## 6. 今年度の進捗状況と今後の展望

配分して頂いた当初の計算資源 20 口 (1 口=3,000 ノード・時間) に対し、拠点内の調整により最終的に 31 口の資源配分を頂き全て有効に使用させて頂いた。研究計画の通りに空間充填曲線を用いた動的負荷分散を導入した粒子法による大規模流体構造連成計算の実アプリケーションを実行した。スライス・グリッド法による領域分割に対し、空間充填曲線を用いることにより並列計算のスケールアップが大幅に向上することを示すことができた。一方、当初から予定していた粒子法計算における通信と計算のオーバーラップは本共同研究の中では導入の検討段階までしか到達できなかった点は残念である。

より効率的な大規模計算を行うには、各物体を構成する粒子に対するノードを超えた力とトルクの総和計算 (Reduction 操作) の並列化及び効率的な物体管理法を導入する必要がある。また、個別要素法による物体衝突計算は時間刻みが流体計算より小さい場合が多く、サブステップの時間積分を行うことを検討している。その場合には、流体計算と個別要素計算に負荷バランスが単純な粒子数の均等分割では十分に達成されないことも明らかになった。

平成 27 年度の後半に検討した、物体については個別要素法で計算し、流体については格子法 (格子ボルツマン法) と連成させる計算については、全てを粒子法で計算するよりも精度が高く計算効率も良くなる見通しが得られ、本研究とは別課題として平成 28 年度共同研究の課題申請に至っている。

## 7. 研究成果リスト

### (1) 学術論文

- [1-1] C. Feichtinger, J. Habich, H. Koestler, U. Reude, T. Aoki: Performance modeling and analysis of heterogeneous lattice Boltzmann simulations on CPU-GPU clusters, *Parallel Computing*, Volume 46, June, 2015
- [1-2] 渡辺 勢也, 青木 尊之, 都築 怜理, 下川辺 隆史: 接触による粒子間相互作用の GPU 計算での近傍探索手法, *情報処理学会論文誌 コンピューティングシステム (ACSS2)*, Vol.8, No.4, pp.50-60, 2015
- [1-3] 渡辺 勢也, 青木 尊之, 都築 怜理: GPU による非球形粒子を用いた DEM の大規模シミュレーション, *粉体工学会誌* Vol.52 No.12, pp.730-735, 2015
- [1-4] 渡辺 勢也, 青木 尊之, 都築 怜理: GPU を用いた個別要素法による粉体シミュレーションに対するメモリ使用量を抑えた高速化手法, *日本計算工学会論文集* **[accepted]**

### (3) 国際会議発表

- [3-1] S. Tsuzuki, T. Aoki: Large-scale SPH Simulations using Dynamic Load Balance on a GPU Supercomputer, *Computational Methods in Marine*

- Engineering (Marine 2015), Rome, Jun 15, 2015
- [3-2] T. Aoki, S. Tsuzuki, S. Watanabe: Large-Scale Suspension Flow Simulations Using a Particle Method on a GPU Supercomputer, 13th US National Congress on Computational Mechanics, San Diego, Jul 28, 2015
- [3-3] S. Watanabe, T. Aoki: Large-scale DEM Simulations using Non-spherical Elements on GPU, PARTICLES 2015, Barcelona, Sep 28, 2015
- [3-4] S. Tsuzuki, T. Aoki: Large-scale Particle-based Simulations for Granular and Fluid Dynamics using Dynamic Load Balance on a GPU supercomputer, PARTICLES 2015, Barcelona, Sep 29, 2015
- [3-5] Takayuki Aoki, Satori Tsuzuki, Yuta Hasegawa: Dynamic domain decomposition using space-filling curves for large-scale computational mechanics applications, the 3rd International Workshops on Advances in Computational Mechanics (IWACOM-III), Tokyo, October 13, 2015.
- [3-6] Takayuki Aoki: Beyond Peta-scale on Stencil and Particle-based GPU Applications, SPPEXA Annual Plenary Meeting 2016, Munich, Germany, January 28, 2016. **【Keynote Lecture】**
- [3-7] Takayuki Aoki: Large-scale physics-based simulations and scientific visualizations, The Visualisation, Big Data, Art and Science Festival 2016, Brisbane, Queensland, February 18, 2016 **【Keynote Lecture】**
- [3-8] Satori Tsuzuki and Takayuki Aoki: Large-scale Particle-based Simulations for Debris Flows using Dynamic Load Balance on a GPU Supercomputer, Violent Flows 2016, Osaka, March 10, 2016.
- (4) 国内会議発表
- [4-1] 渡辺勢也, 青木尊之, 都築怜理: 非球形粒子を用いた個別要素法による大規模粉体シミュレーション, 粉体工学会 2015 年度春期研究発表会, 神田駿河台, 2015 年 5 月 19 日
- [4-2] 都築怜理, 青木尊之: 動的領域分割を用いた流体構造連成によるサスペンション・フローの大規模 GPU 計算, 2015 年ハイパフォーマンスコンピューティングと計算科学シンポジウム, 東京, 2015 年 5 月 20 日
- [4-3] 都築怜理, 青木尊之: サスペンション・フローの超大規模シミュレーション, 粉体工学会 2015 年度春期研究発表会, 東京, 2015 年 5 月 20 日
- [4-4] 青木尊之: GPU スパコンによる大規模流体・粒子シミュレーション, 2015 年度 放電学会シンポジウム, 東京, 2015 年 6 月 5 日[招待講演]
- [4-5] 渡辺勢也, 青木尊之, 都築怜理, 下川辺隆史: GPU を用いた個別要素法計算における近傍探索手法の比較評価, 第 20 回計算工学講演会, つくば, 2015 年 6 月 8 日
- [4-6] 都築 怜理, 青木 尊之: GPU スパコンにおける動的負荷分散を用いた大規模流体構造連成シミュレーション, 第 20 回計算工学講演会, つくば, 2015 年 6 月 8 日
- [4-7] 渡辺勢也, 青木尊之, 都築怜理: 接触相互作用に基づく粒子法の GPU 計算におけるメモリアクセスの改善による高速化, 第 149 回ハイパフォーマンスコンピューティング研究発表会, 工学院大学, 新宿, 2015 年 6 月 26 日
- [4-8] 都築怜理, 青木尊之: GPU スパコンによる動的領域分割を用いたサスペンション・フローの大規模粒子法シミュレーション, 日本応用数理学会 2015 年度年会, 金沢, 2015 年 9 月 9 日
- [4-9] 渡辺勢也, 青木尊之, 都築怜理, GPU による実形状粉体モデルを用いた大規模 DEM シミュレーション, 第 28 回 計算力学講演会, 横浜国大, 2015 年 10 月 11 日
- [4-10] 渡辺勢也, 青木尊之, 長谷川雄太: LBM-DEM 固液二相流の GPU 計算, 第 29 回数値流体力学シンポジウム, 九州大学 筑紫キャンパス, 2015 年 12 月 15 日
- [4-11] 青木尊之: GPU スパコンを用いた格子ボルツマン法/有限体積法/粒子法による大規模流体シミュレーション, 日本機械学会 第 28 回バイオエンジニアリング講演会, 東京工業大学, 2016 年 1 月 10 日 **【特別講演】**
- [4-12] 青木尊之: 災害・ものづくり系シミュレーションとスパコンからのイノベーション, 平成 27 年度地球シミュレータ利用報告会, 東京, コクヨホール, 2016 年 3 月 11 日 **【特別講演】**
- (5) 受賞学術賞
- [5-1] 都築怜理: 若手講演フェロー賞, 日本機械学会・計算力学部門, 2015 年 5 月
- [5-2] 都築 怜理, 青木 尊之: 2015 年ハイパフォーマンスコンピューティングと計算科学シンポジウム最優秀論文賞, 情報処理学会, 2015 年 5 月 20 日
- [5-3] 都築 怜理, IEEE Computer Society Japan Chapter 優秀若手研究賞, 2015 年 5 月 20 日
- [5-4] 渡辺勢也: 粉体工学会・2015 年度春期研究発表会・B P 賞 (ベストプレゼンテーション賞), 2015 年 5 月 20 日
- [5-5] 都築 怜理, 青木 尊之: グラフィクスアワード・ビジュアルコンピューティング賞, 日本計算工学学会・第 20 回計算工学講演会, 2015 年 6 月 8 日
- [5-6] 渡辺勢也, 青木尊之, 長谷川雄太: ベスト CFD グラフィクスアワード動画部門 第 2 位, 第 29 回数値流体力学シンポジウム, 2015 年 12 月 16 日
- [5-7] Satori Tsuzuki: Best Student Paper Award, Violent Flows 2016, March 11, 2016