

青木尊之 (東京工業大学 学術国際情報センター 教授)

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 -



研究背景と目的

流体と構造物が共存する現象は実際に非常に多く、さまざまな分野で流体構造連成解析の重要性が広く認識されている。粒子法は複雑形状や流体構造連成計算を容易に行うことができる利点を持っているが、粒子法の並列計算には不均一な粒子分布が発生するためにプロセス間の負荷分散が著しく低下するなど大規模計算が難しい問題がある。複数の球形粒子を連結させ、多数の球形粒子で表現する複雑形状の構造物とSPH(Smoothed Particle Hydrodynamics)を用いた流体計算との大規模流体・構造連成計算を、演算性能、メモリバンド幅や電力効率の点で有利なGPU (Graphics Processing Unit)を用いて高効率に実行するためのアルゴリズムの開発と実問題の大規模シミュレーションを行う。

研究計画

紛体シミュレーションのためのDEM (Discrete Element Method)と流体シミュレーションのためのSPH (Smoothed Particle Hydrodynamics)法に対し、GPUスパコンにおいて大規模シミュレーションを可能にするための計算手法の開発を行う。

- 複数の球形粒子を剛体連結させた非球形DEMのGPU計算の効率化と通信負荷の低減手法を開発する。
- 第一段階として、スライスグリッド法による動的負荷分散を流体構造連成の大規模GPU計算にも適用し、演算や通信コストの問題を定量的に評価する。
- 第二段階は空間充填曲線による動的負荷分散について検討し、細分化の段数の制御、データ構造の管理コスト、領域間通信のコスト、リーフのマイグレーションのコスト等を検証する。
- 1000GPU以上の大規模アプリケーションを実行し、高い並列実行性能を達成できる動的負荷分散法を開発する。
- 分割領域の境界周辺リーフと内側リーフに分け、境界周辺リーフを先に計算した後内部リーフの計算とノード間通信をオーバーラップさせ、その有効性を検証する。

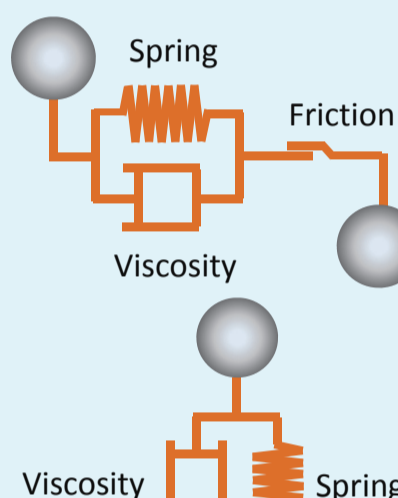
研究グループ

代表者	青木 尊之	(東京工業大学)
副代表者	西浦 泰介	(海洋研究開発機構)
課題参加者	都築 裕理	(東京工業大学)
課題参加者	渡辺 勢也	(東京工業大学)
課題参加者	森口 周二	(東北大学)
課題参加者	今井 陽介	(東北大学)
課題参加者	井元 佑介	(九州大学)

非球形粒子による個別要素法

■ Discrete Element Method

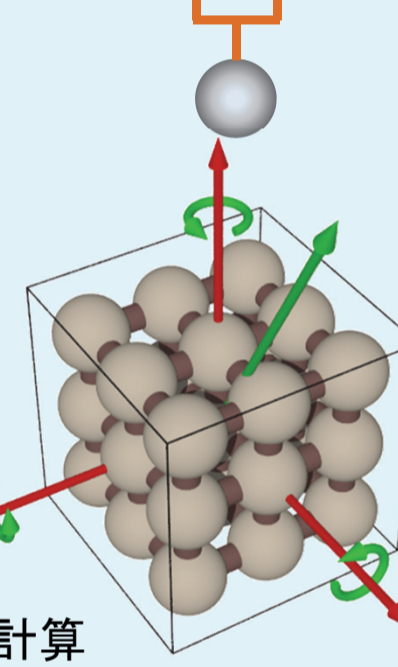
粉体を有限の大きさを持つ粒子で表現し、周囲の接触している粒子から力を受けて個々の粒子が運動する計算手法である。粒子間の接触相互作用はバネとダンパでモデル化され、接線方向に作用する力は粒子間の摩擦に相当する。



非球形粒子モデルの導入

球形粒子モデルでは、粉体同士が噛みあうような複雑な相互作用を表現できない。複数の球形粒子を剛体連結させて、非球形粒子をモデル化する。以下のことから球形粒子に比べて計算コストが高い。

- 計算に必要な粒子数の増加
- クォータニオンによる回転管理
- 重心に作用する合力とトルクの総和計算

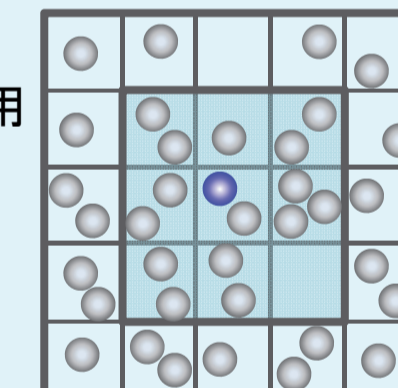


GPUにおける高速化手法

■ 近傍粒子探索リストの利用

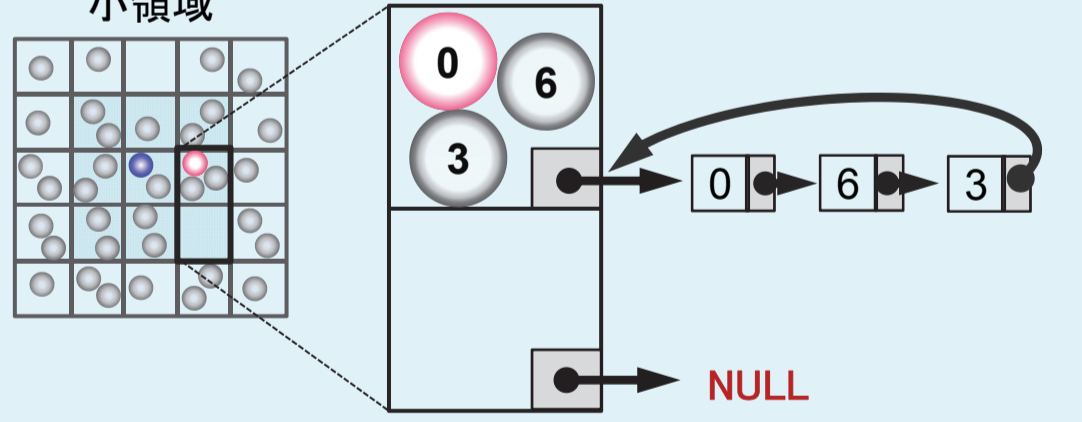
自身と隣接セル内の粒子とのみ相互作用
⇒ O(N²)からO(N)へと計算量を削減

最大収容数を想定して静的メモリを確保
⇒ **メモリ消費量が著しく増加**



リンクリスト法によるメモリ使用量の削減

空間格子には粒子を一つだけ登録し、セル内で粒子同士がリンクリスト構造を生成



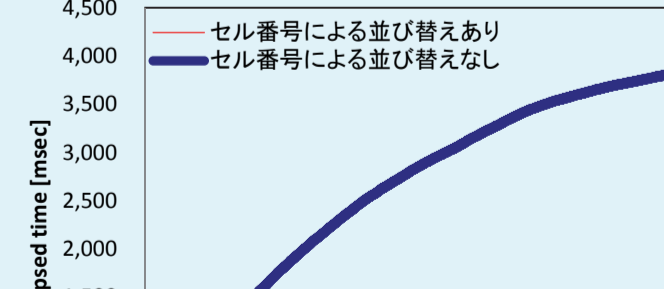
メモリ消費量を大幅に削減 (DEMなら1/8, SPHなら1/64)

セル番号をキーとした粒子ソートによる高速化

✓ 同一セル内の粒子がメモリ上に連続する様子をソートを実行し、GPUにおけるコアレッシングを実現



✓ 140万個の粒子を用いたSPH法によるダム崩壊を、単体GPU上で実行した場合の計算時間の比較

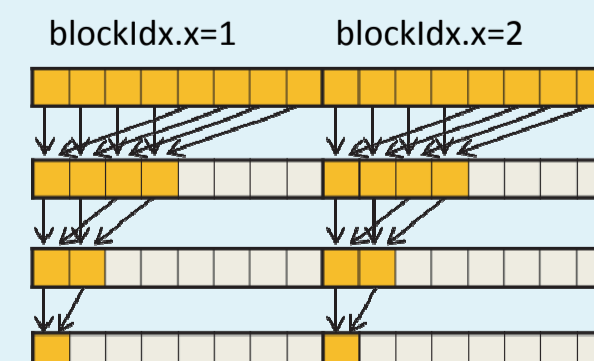


約 8.6 倍に高速化

カとトルクのリダクションによる総和計算

✓ 1つの非球形粒子の総和計算を1blockが担当

✓ シェアードメモリを用いたリダクションで総和計算



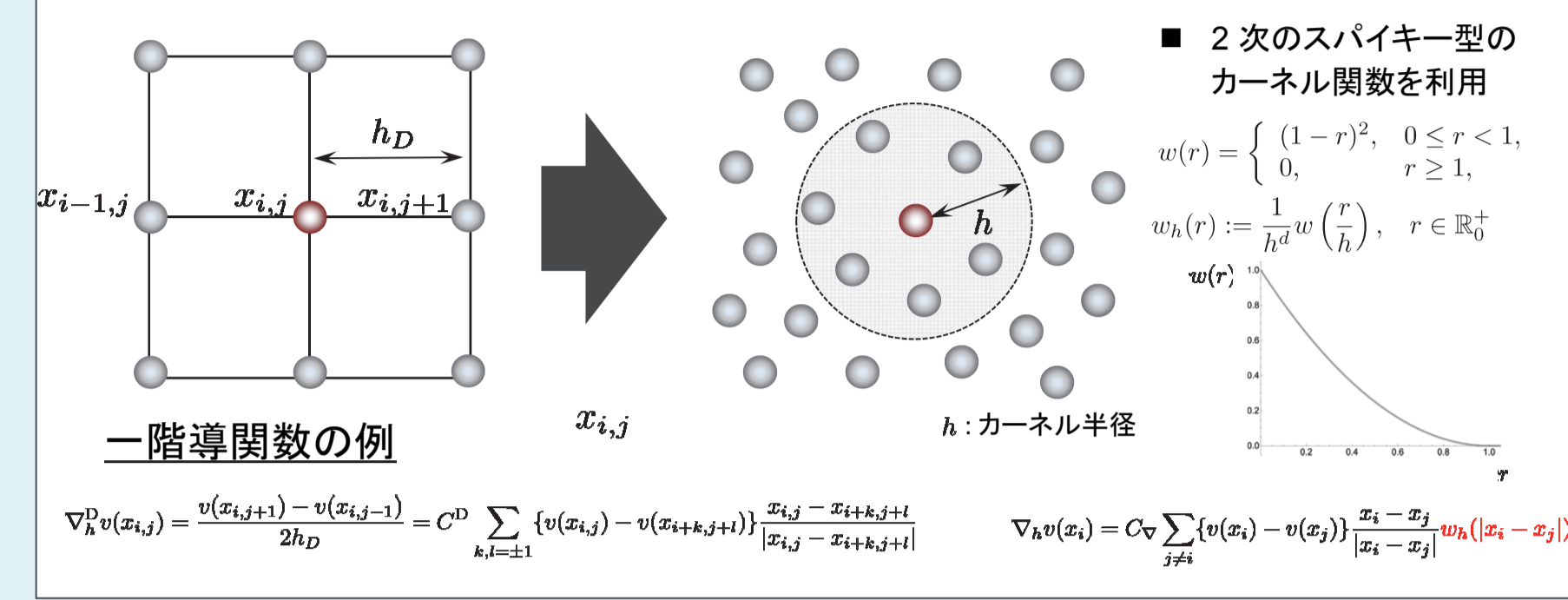
2048個の非球形粒子の総和計算でバンド幅測定



構成粒子数が多いほどリダクションの効果大

改良型粒子法(井元、田上 2014)

- 有限差分法の一般化としてとらえる新しい粒子法 -



空間充填曲線による領域分割

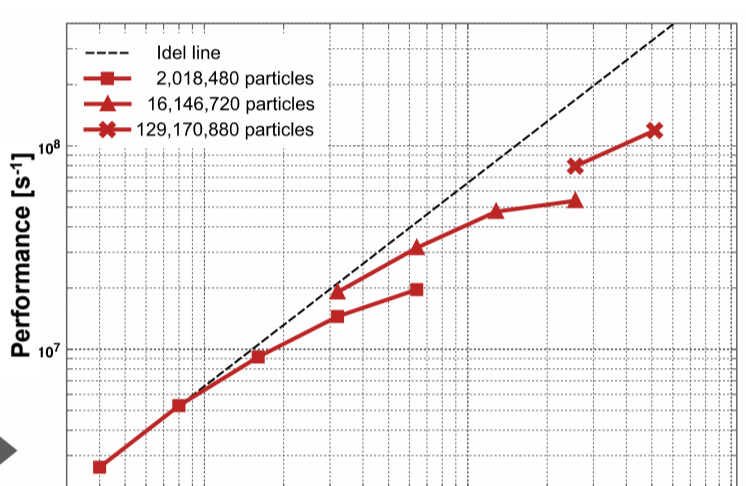
スライスグリッド法の問題点

x方向に粒子数が一定になるように領域を分割し、分割された各小領域をy方向に独立に分割する。

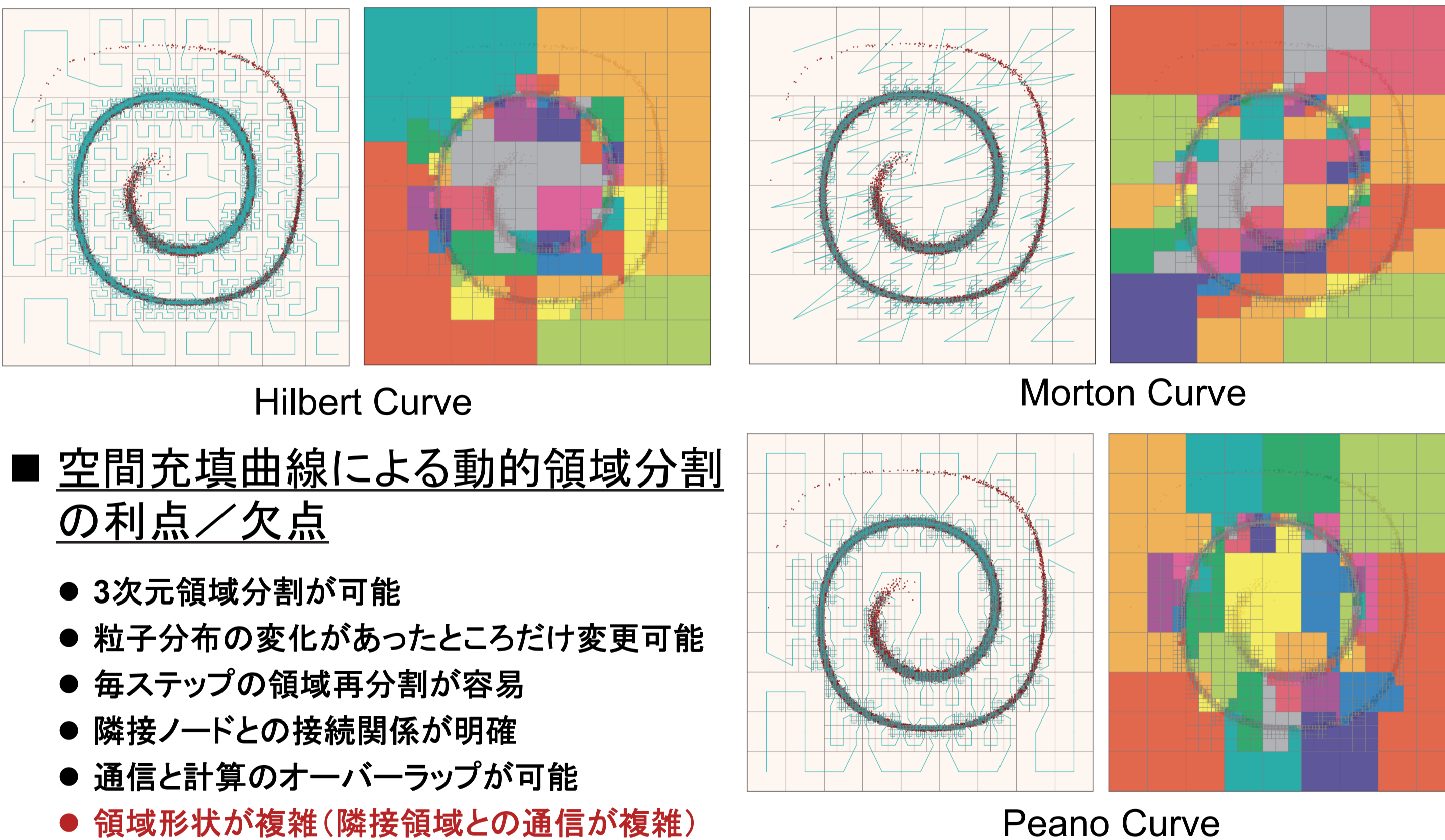
- 2次元領域分割までが限界(分割手続きが複雑)
- 一つの分割領域が複数の分割領域と接触
- 分割領域が高アスペクト比(粒子が分割領域境界を横切り易い:ノード間データ通信量の増加)

スライスグリッド法による動的負荷分散によるDEM計算は、弱スケールリングですら512GPUで50%程度まで並列化性能が低下。

DEMIに対する強・弱スケールリング



計算領域を8分木または27分木で再帰的に分割し、全てのリーフの中の粒子数が所定の数以下になるまで細分化を続ける。計算領域内のリーフを空間充填曲線でたどり、通過したリーフ内の粒子数の積算により領域を分割。



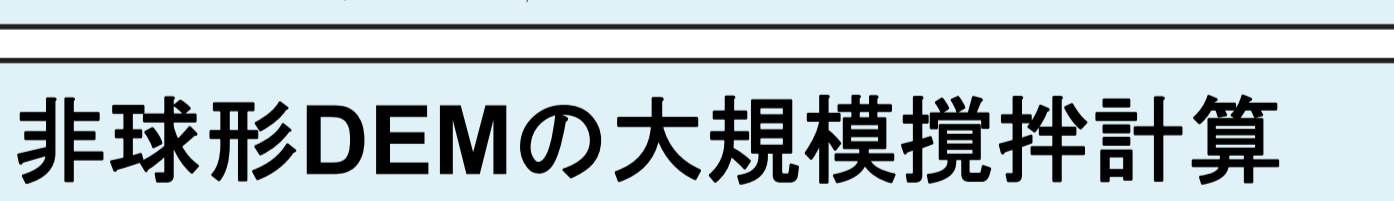
空間充填曲線による動的領域分割の利点/欠点

- 3次元領域分割が可能
- 粒子分布の変化があったところだけ変更可能
- 毎ステップの領域再分割が容易
- 隣接ノードとの接続関係が明確
- 通信と計算のオーバーラップが可能
- 領域形状が複雑(隣接領域との通信が複雑)
- リーフの隣接ノードへのマイグレーションの負荷が不明

符号付距離関数による物体表現

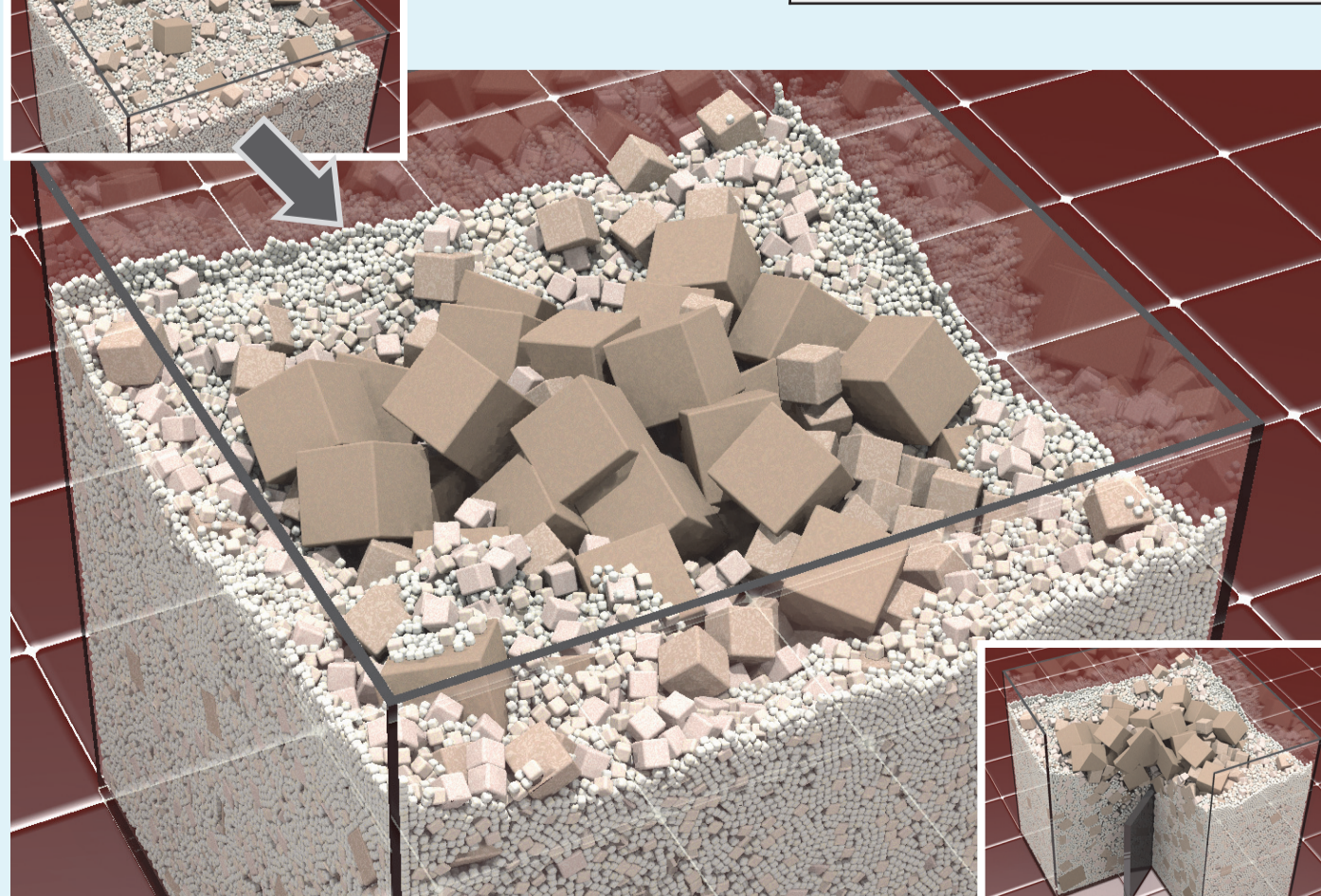
✓ 計算開始前にCADデータから符号付き距離場を生成し、格子データに格納し、計算時にはデータを参照するだけで距離判定を行う。

✓ 符号付距離関数を用いて、流体計算用の粒子群データも生成できる。



非球形DEMの大規模攪拌計算

- 計算粒子数 : 6,205,952 個
- 非球形粒子数 : 299,584 個
- ステップ数 : 800,000
- 計算時間 : 13.7 h



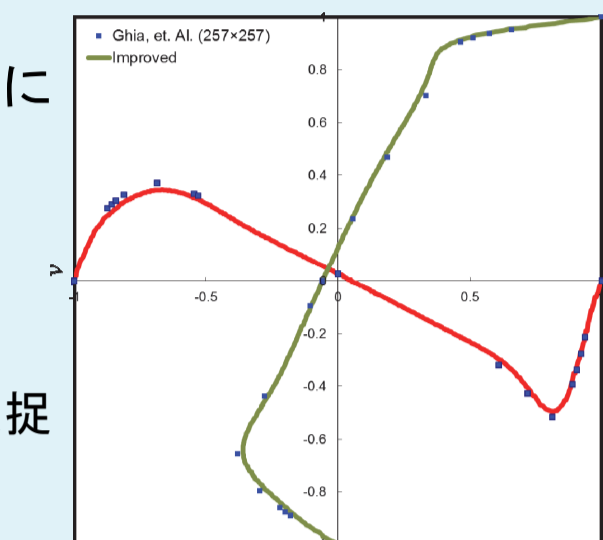
改良型粒子法の陽解法の場合の計算手順

1. 予測子を計算: $u_i^k = u_i^{k-1} + \Delta t u_i^{k-1}$, $x_i^k = x_i^{k-1} + \Delta t u_i^k$
2. Birch-Murnaghan's の方程式から計算される仮の圧力を計算:
$$p_i^k = \frac{C_0 \rho}{\gamma} \left\{ \left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right\}$$
3. 仮の圧力をもとに粒子座標を修正:
$$x_i^{k+1} = x_i^k - \frac{\Delta t}{\rho} \nabla_h p_i^k$$
4. 補間により圧力を再計算:
$$p_i^k = \sum_j w_{ij} p_j^k \left(\frac{x_i^k - x_j^k}{x_i^k - x_j^{k+1}} \right)$$
5. 修正子により流速を計算: $u_i^{k+1} = u_i^k - \frac{\Delta t}{\rho} \nabla_h p_i^k$

2次元キャピティ・フロー問題による精度検証 (Ghia らとの比較)

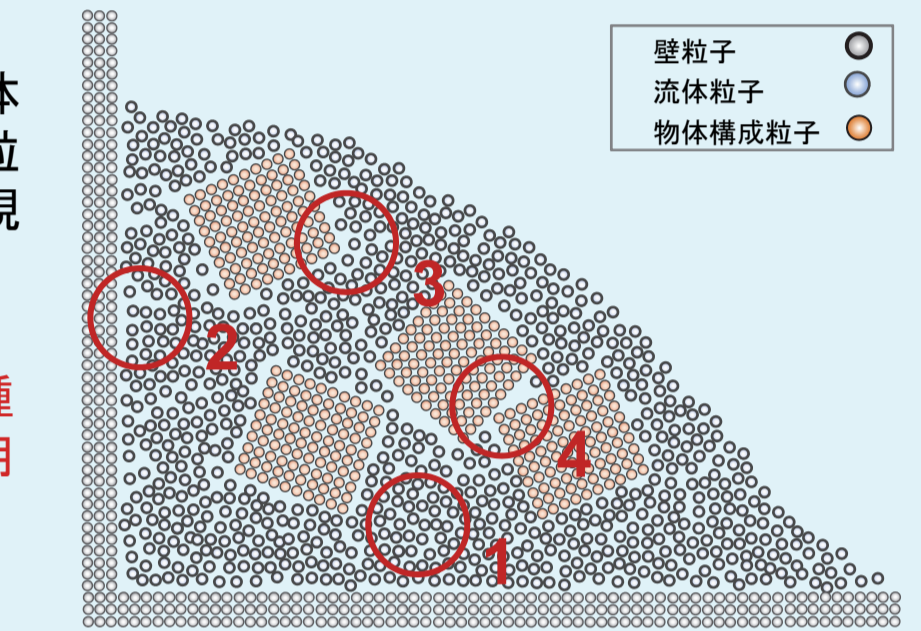
Re = 1,000

粒子法としては、渦を非常に良く捉えている。



粒子法による流体・構造連成計算手法

- ✓ 壁、流体、物体のすべてを粒子により表現する。
- ✓ 粒子間で4種類の相互作用計算を行う。

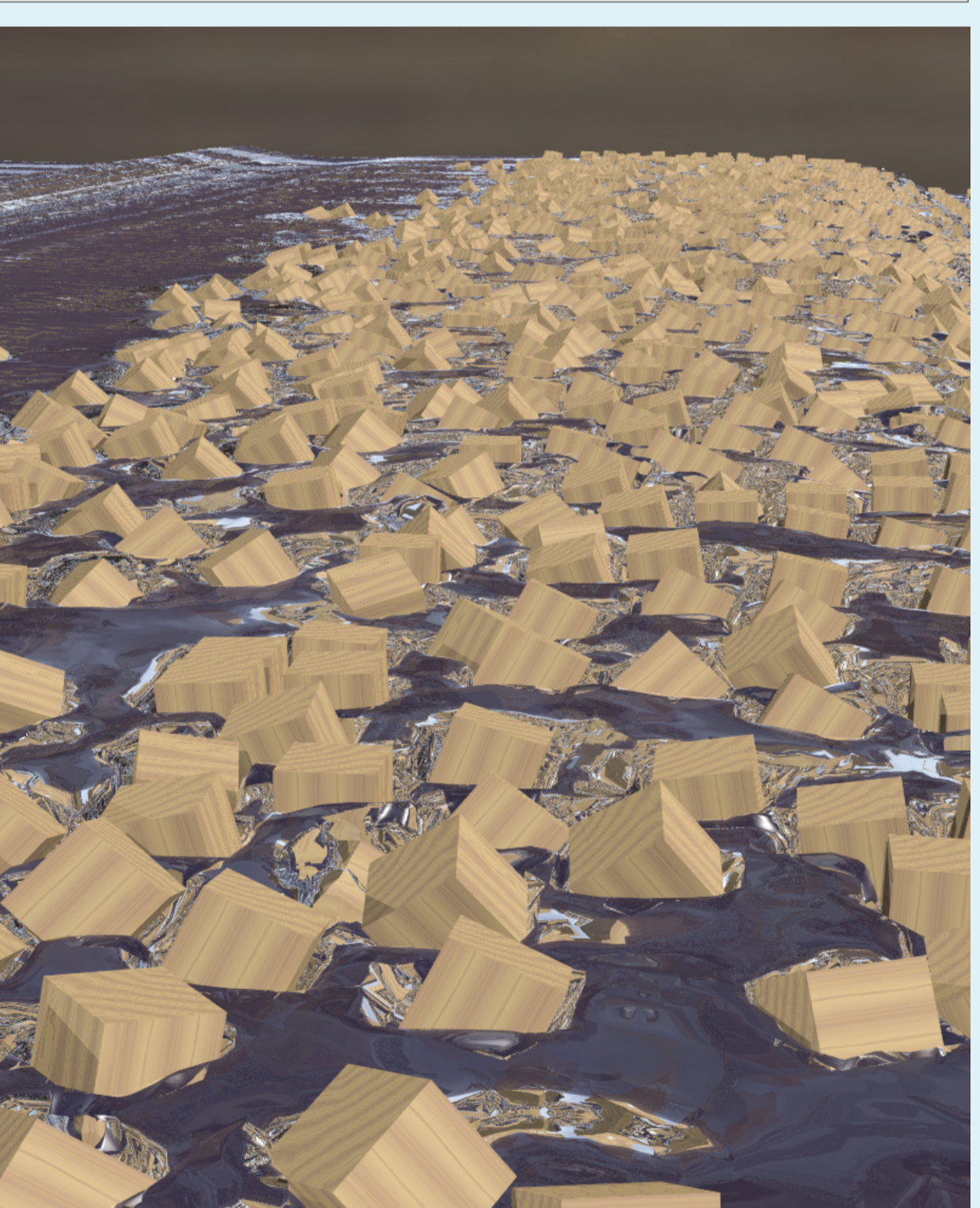


1. 流体粒子-流体粒子間 (流体計算)
改良型粒子法(上述参照)を用いる。
2. 流体粒子-壁粒子・物体構成粒子間 (境界条件)
壁粒子は静止した境界条件、物体構成粒子は移動境界条件として扱う。
3. 物体構成粒子-流体粒子間の相互作用計算
粒子 i への圧力勾配による流体作用力を与える:
$$f_i = \frac{v_i^s - v_i}{\Delta t}$$
 (v: 現在の速度 v^s: 仮の速度)
4. 物体構成粒子-物体構成粒子間の相互作用計算
個別要素法(DEM)による衝突力を与える。

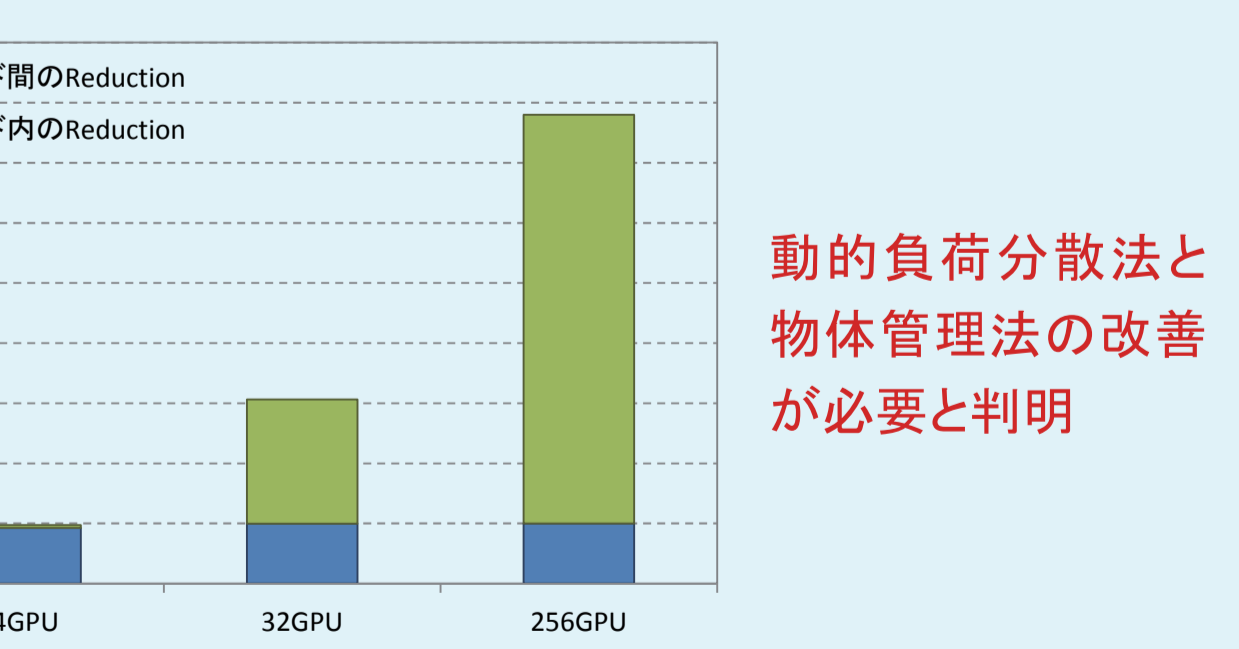
流体構造連成の大規模GPU計算

■ サスペンションフロー計算

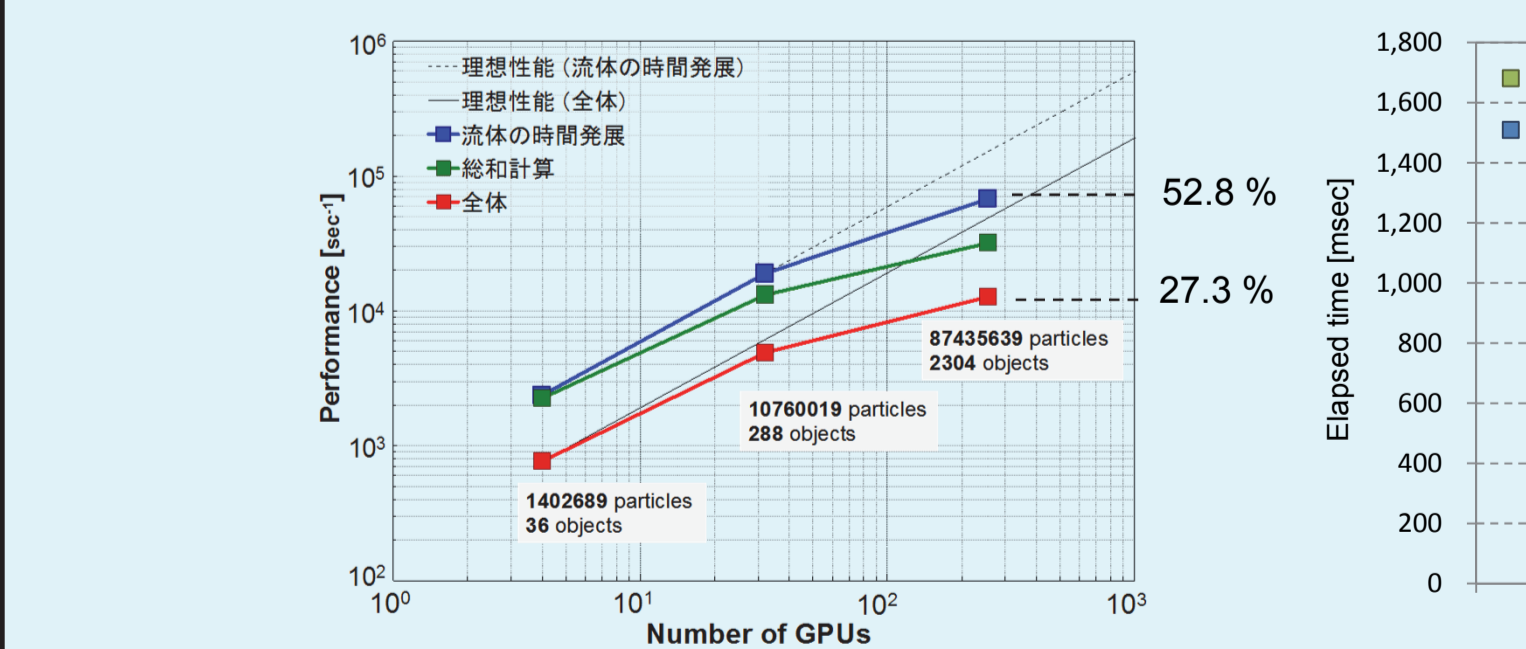
計算領域	144 m × 160 m × 60 m	粒子数	87,430,000
物体構成粒子数	1000 / 物体	物体数	2304 個
物理時間 (秒)	6.8	ステップ数	13600
GPU 数	256	計算時間	43.1 h



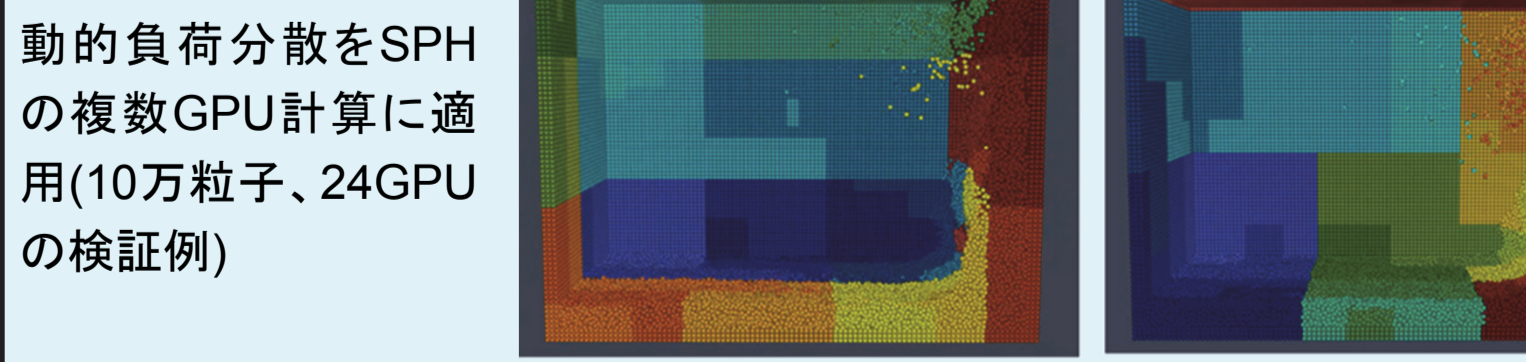
総和計算の内訳



2次元スライスグリッド法を用いた場合の弱スケールリング



空間充填曲線による動的負荷分散をSPHの複数GPU計算に適用(10万粒子、24GPUの検証例)



謝辞 本研究の一部は、科学研究費補助金・基盤研究(S) 課題番号26220002、日本学術振興会特別研究員奨励費・課題番号14J12004、科学技術振興機構CREST「ポストベタスケール高性能計算に資するシステムソフトウェア技術の創出」、革新的ハイパフォーマンス・コンピューティング・インフラから支援を受けた。また、本研究は東京工業大学・学術国際情報センターTSUBAME2.5を利用して計算を行ったものであり、記して謝意を表す。