

分散メモリ型並列 MPS 陽解法による数十億粒子規模の津波シミュレーション

塩谷 隆二

東洋大学総合情報学部総合情報学科

越塚 誠一, 室谷 浩平

東京大学大学院工学系研究科システム創成学専攻

荻野 正雄

名古屋大学情報基盤センター大型計算支援環境研究部門

河合 浩志

諏訪東京理科大学システム工学部

1. はじめに

MPS 法[3]は, Moving Particle Simulation の略であり, 粒子法の代表的手法の一つである. 粒子法とは, 有限個の粒子(計算点)により連続体の支配方程式を離散化し, それらの粒子が保持する物理量の時間変化を Lagrange 的に追跡する数値計算手法である. 粒子法は計算点としての粒子を移動させるため, 格子法に比べて, 自由表面や大変形などの動きの激しい物理現象を解くのに優れている.

しかしながら, 粒子法による流体一般の解析では影響半径内に含まれる計算点のデータのみ参照し, その参照関係が時間発展に伴い変化してしまう. この性質が分散メモリ計算機環境での並列化を困難なものとしてきた. 代表的な並列化法には, スライスグリッドと呼ばれる計算領域内に1軸をとり, この1軸に直交するように計算領域をスライスする方法がある. この方法は, 計算速度やメモリ効率の向上に貢献したが, 大規模並列計算には対応できなかった. そのため, 粒子法計算は, 共有メモリ計算機での並列化が主に行われてきた. 本研究では, 大規模な粒子系シミュレーションのための分散メモリ型並列 MPS 陽解法を開発した[5].

今回, 我々が開発した手法を用いて, 東日本大震災を模擬した石巻市市街地への津波遡上解析を行った. 図1は, 石巻市街地を2つのタンクが津波によって流される解析を3.9億粒子によって可視化した結果である. 我々がターゲットとしている津波解析では, 数千万~数十億粒子の解析を, 実時間数分から数十分の長時間解析を行うため, 高速な大規模並列計算が必要となった.

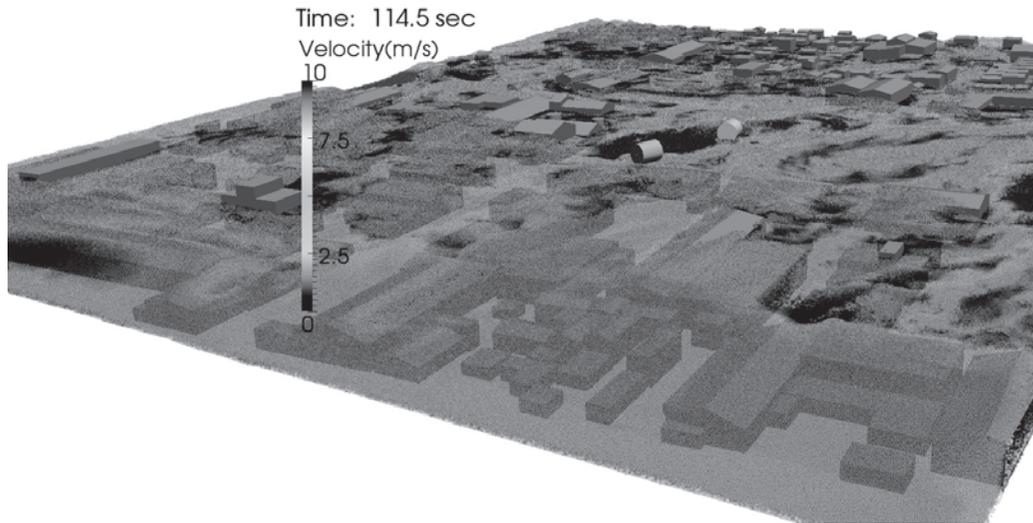


図 1. 直径 9m の 2 つのタンクが津波によって流される様子を 3.9 億粒子によって可視化した結果

2. 分散メモリ並列 MPS 陽解法アルゴリズム

本章では、我々が開発した分散メモリ並列 MPS 陽解法アルゴリズムを説明する。図 2 の様に、最初に、全計算領域のバウンディングボックスを定め、このバウンディングボックスを立方体のバケットによって切り分ける。MPS 法では、ある粒子が 1 タイムステップで影響を及ぼす範囲が決められているので、バケットの一辺の長さを、この影響範囲以上とする。全ての粒子は、このバケットに登録される。このバケットを用いて、各計算ノードに含まれる粒子数が等しくなるように、バケットベースの領域分割を行う。

領域分割には、ParMETIS ライブラリ [2] を用いた。ParMETIS ライブラリはグラフ分割を行うライブラリであるので、バケットをグラフの頂点、バケット間の面をグラフの辺とした双対グラフを作成し、その双対グラフを分割する。粒子数を等分配したいので、バケット内の粒子数をグラフの頂点の重みとする。

各計算ノード間の計算の整合性をとるために、各領域の境界からバケット 1 つ分だけ領域を広げ、その広げられた領域に所属する粒子を各計算ノードに割り当てる。このバケット 1 つ分だけ広げた領域内の粒子は、隣接間通信によって整合性を保たれる。

図 3 は、分散メモリ型並列 MPS 陽解法の計算フローチャートである。分散メモリ型並列 MPS 陽解法では、1 タイムステップ内で、3 回の通信が行われる。タイムステップが進み、各計算ノード間の粒子数のバランスが著しく崩れた時には、再度 ParMETIS ライブラリを用いて各計算ノード間の粒子数が均等になるように領域再分割を行う。

我々が開発を進めている分散メモリ型並列 MPS 陽解法は、ヘテロジニアス計算機を想定して、図 4 のような、2 階層の領域分割を行っている。第 1 階層では、計算ノード間の粒子数バランスをとるために第 1 レベルの領域分割を行い、第 2 階層では計算ノード内での thread 間の粒子数バランスをとるために第 2 レベルの領域分割を行っている。将来、第 2 階層に対応する部分に、複数の GPU や MIC が対応できるように開発を進めている。

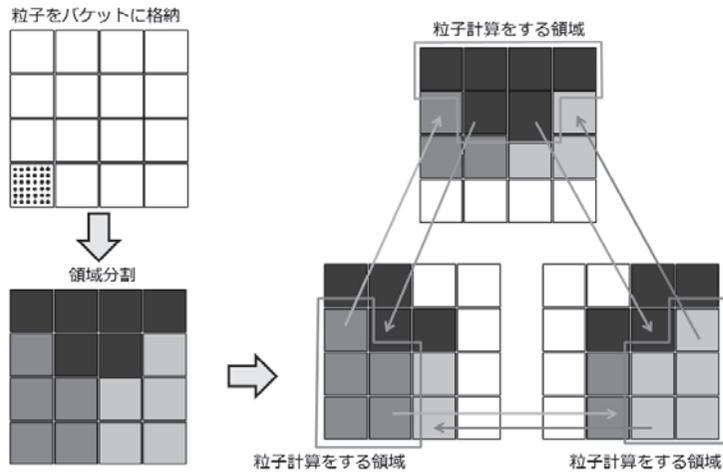


図 2. バケットベースの領域分割

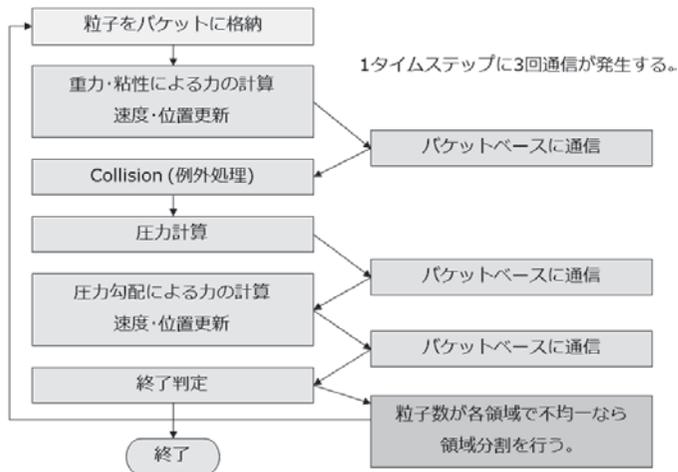


図 3. 分散メモリ型並列 MPS 陽解法の計算フローチャート

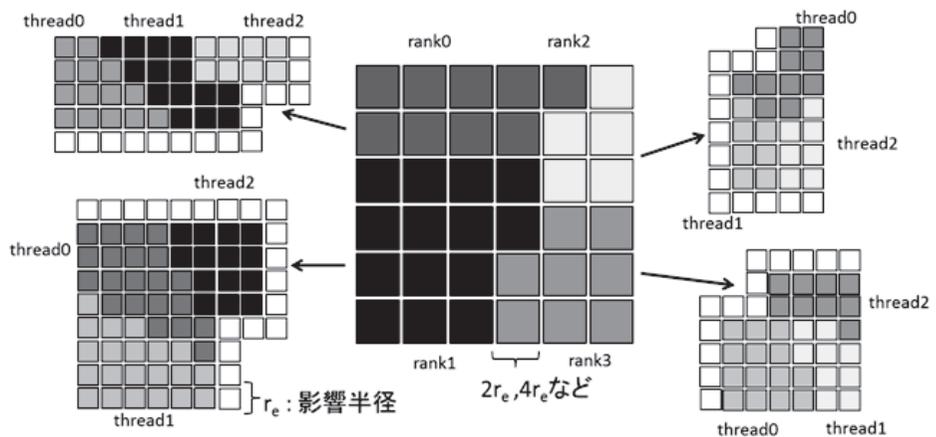


図 4. 2 階層の領域分割

図5を用いて領域再分割の様子を説明する。図5は各計算ノードが担当した粒子数の推移を示したグラフである。この解析で用いた計算ノード数は4であり、粒子数は19,136である。本解析では領域再分割は粒子数 5,500 を超える計算ノードが発生すると行われるものとしていいる。最初の領域分割は時刻 0.2 秒に行われている。PE0 から PE1 へ粒子が流れ込むにつれ、PE0 の粒子が減り、PE1 の粒子が増えていっている。PE1 の粒子数が 5,500 を超えた時刻が 0.2 秒であり領域再分割が行われ、粒子が各計算ノードに均等に割り当てられている。図5から、時刻 0.2 秒, 0.2896 秒, 0.5368 秒, 0.9296 秒で領域再分割が行われていることが分かる。

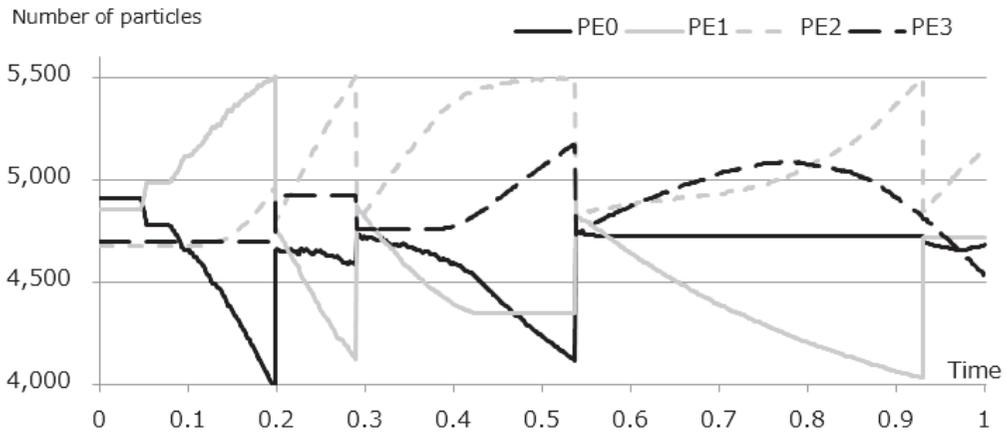


図5. 各計算ノードが担当する粒子数の推移

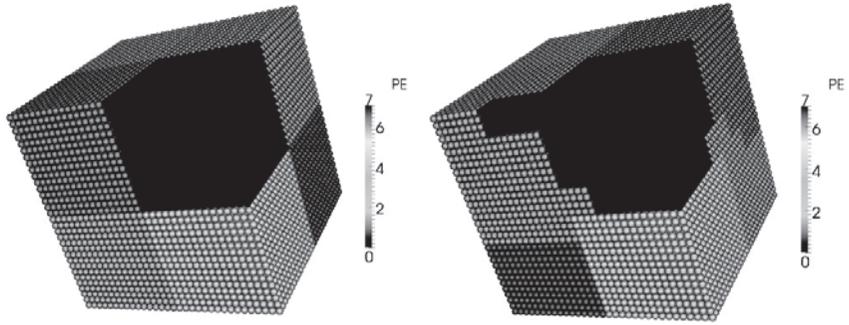
3. 並列性能の測定

本研究で開発を進めている分散メモリ並列MPS陽解法では、

- ① 領域分割による動的負荷分散
- ② 毎ステップ行う隣接間通信のための通信テーブルの作成、
- ③ 計算ノード内チューニング

の3つに切り分けて性能向上を目指している。①に関しては現在の所、ParMETISを用いることで、再領域分割が行われた時には粒子数が5%程度の差で粒子が再分配されている(5%という値は変更できる)。①に関しては今後引き続き検討を重ねていくべき課題である。③に関しては研究チーム内で汎用的なライブラリ作成を行っており、今回は開発版を用いて測定を行った。②に関してはほぼ完成しており、今回の測定は②の完成度を見る目的が高い。

並列性能を測定するに際して、FX10の6次元メッシュトラスインターコネクットの特性を生かす為、図6(a)の様な粒子モデルを作成した。もし、ParMETISを用いて領域分割を行うと、図6(b)のような領域分割が行われ、格子状の領域に粒子が割り当てられることはない。今回は、理想的な状態での隣接間通信の性能を測定するのが目的であるので、図6(a)のような最も理想的な状態の粒子配置で測定を行った。計算ノードの格子配置に合わせて、表1のような並列プロセスの3次元形状指定を行い、粒子モデルの領域分割も表1に対応するように作成した。



(a) 格子状の領域分割を行った場合 (b) ParMETIS を用いて領域分割を行った場合
 図 6. 各計算ノードに配置された粒子

表 1. 並列プロセスの 3 次元形状指定

| | | | | | | | | | | | | | | | |
|------------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| tofu_x_dim | 2 | 3 | 4 | 5 | 6 | 5 | 6 | 8 | 5 | 6 | 10 | 6 | 10 | 12 | 20 |
| tofu_y_dim | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 6 | 6 | 6 | 12 | 12 | 12 | 15 |
| tofu_z_dim | 2 | 5 | 4 | 5 | 5 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 16 |
| nodes | 12 | 45 | 48 | 75 | 90 | 150 | 180 | 240 | 300 | 360 | 600 | 720 | 1200 | 1440 | 4800 |

並列性能の測定は、影響半径が粒子直径の 2 倍 (r2) と 4 倍 (r4) の 2 ケースで測定を行った。
 図 7 は、314, 572, 800 粒子のストロングスケーリングを 12 ノードから 4, 800 ノードで測定を行
 った結果である。

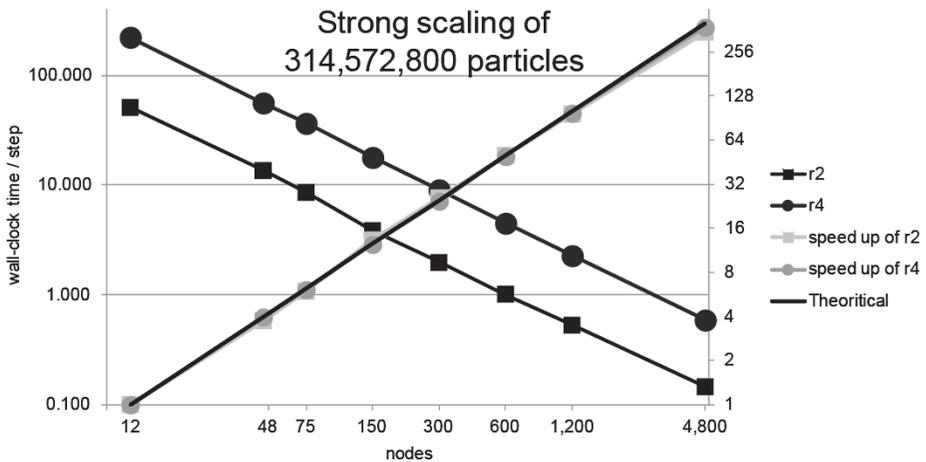


図 7. 314, 572, 800 粒子のストロングスケーリング

図 8 は、2, 516, 582, 400 粒子のストロングスケーリングを 60 ノードから 4, 800 ノードで測定
 を行った結果である。

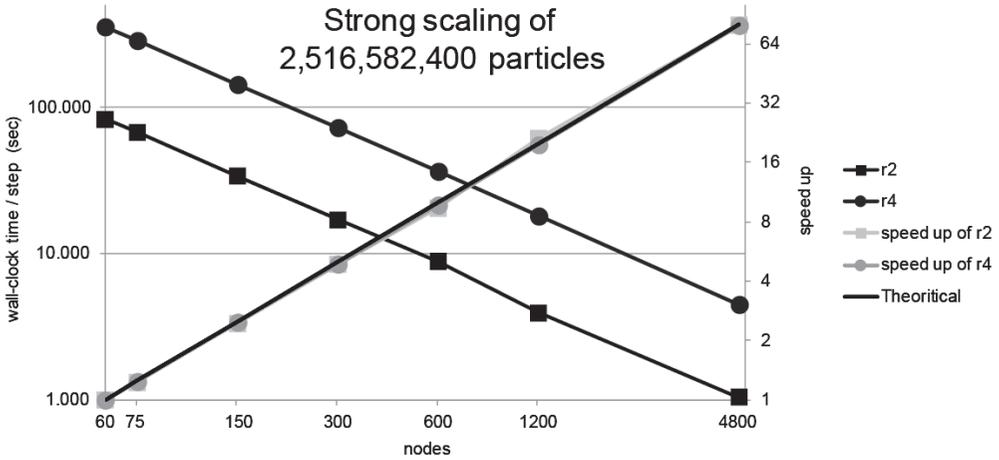


図 8. 2,516,582,400 粒子のストロングスケーリング

図 9 は、1 ノード当り 65,536 粒子のウィークスケーリングを 12 ノードから 4,800 ノードで測定を行った結果である。

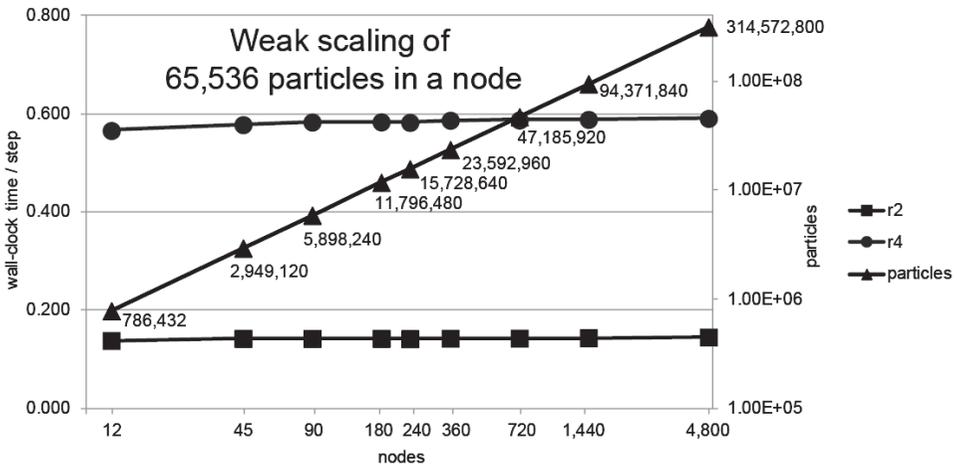


図 9. 1 ノード当り 65,536 粒子のウィークスケーリング

図 10 は、1 ノード当り 4,194,304 粒子のウィークスケーリングを 12 ノードから 4,800 ノードで測定を行った結果である。

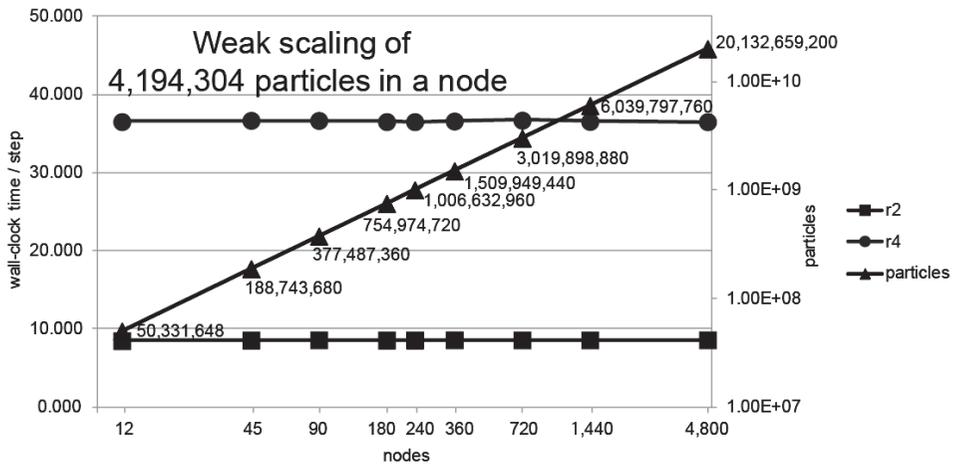


図 10. 1 ノード当たり 4,194,304 粒子のウィークスケーリング

表 2 は、図 7 と図 8 のストロングスケーリングの並列化率を $\frac{T_{12}/T_{4800}}{4800/12}$ と $\frac{T_{60}/T_{4800}}{4800/60}$ で評価した結果である。

表 2. 図 4 と図 5 のストロングスケーリングの並列化率

| | | radius of influence: 2 x particle diameter | | radius of influence: 4 x particle diameter | |
|---------------|-------|---|----------------|---|----------------|
| particles | nodes | Time(sec) | Strong Scaling | Time(sec) | Strong Scaling |
| 314,572,800 | 12 | 51.02895 | 0.8756 | 223.00009 | 0.9453 |
| | 4,800 | 0.14570 | | 0.58977 | |
| 2,516,582,400 | 60 | 82.90577 | 0.9933 | 357.10971 | 0.9899 |
| | 4,800 | 1.04332 | | 4.50962 | |

表 3 は、図 9 と図 10 のウィークスケーリングの並列化率を T_{12}/T_{4800} と評価した結果を評価した結果である。本アルゴリズムでは、表面粒子よりも内部粒子の方が演算量が多いので、Flops でウィークスケーリングを評価してしまうと、並列性能がスーパーリニアになってしまうことがある。そのため今回は、計算時間を用いて並列化率を評価した。

表 3. 図 6 と図 7 のウィークスケーリングの並列化率

| | | radius of influence: 2 x particle diameter | | radius of influence: 4 x particle diameter | |
|----------------|-------|---|--------------|---|--------------|
| particles | nodes | Time(sec) | Weak Scaling | Time(sec) | Weak Scaling |
| 786,432 | 12 | 0.13836 | 0.9496 | 0.56627 | 0.9602 |
| 314,572,800 | 4,800 | 0.14570 | | 0.58977 | |
| 50,331,648 | 12 | 8.52191 | 0.9865 | 36.55244 | 0.9995 |
| 20,132,659,200 | 4,800 | 8.63830 | | 36.57051 | |

ストリングスケーリングとウィークスケーリングの両方で、②に関する「毎ステップ行う隣接間通信のための通信テーブルの作成」の部分は非常に良くスケールしている。全体通信が発生せず、隣接間通信のみ行われる部分であるので当然の結果であるが、隣接間通信に関する部分は完成しているといえる。

次に、③計算ノード内チューニングに関する考察を行う。表4と表5は影響半径が粒子直径の4倍(r_4)のケースの1ノード当たり65,536粒子と4,194,304粒子のときの主要項の計算コストの内訳である。「その他」は、粒子のボクセルへの格納、通信テーブル作成、隣接間通信、袖領域の更新などである。また、4800ノード(20,132,659,200粒子)の解析で、1次キャッシュミス割合6%、2次キャッシュミス割合0.07%であった。一方で、SIMD率は0%であり、翻訳時情報をもみても「s」表記されSIMD化されていない。Intel compilerによる翻訳時情報でも同様にベクトル化されていないと表記されている。現在、FX10において、SIMD命令率を上げるチューニングを行っている。2013年度は計算ノード内チューニングを中心に行う予定であり、今回測定した値が来年度の成果の基準とする予定である。

表4. 1ノード当たり65,536粒子のときの主要項の計算コストの内訳

| | 12 nodes | | 4,800 nodes | |
|-----------|-----------|---------------|-------------|---------------|
| | Time(sec) | Flops/Peak(%) | Time(sec) | Flops/Peak(%) |
| 総計算時間 | 0.5663 | 7.1017 | 0.5898 | 7.2464 |
| 粘性項の計算 | 0.1496 | 11.2142 | 0.1593 | 10.9926 |
| Collision | 0.0525 | 7.4737 | 0.0597 | 7.0855 |
| 圧力の計算 | 0.0874 | 7.2776 | 0.0956 | 7.0558 |
| 圧力勾配の計算 | 0.1854 | 7.0868 | 0.2013 | 6.9299 |
| その他 | 0.0914 | | 0.0738 | |

表5. 1ノード当たり4,194,304粒子のときの主要項の計算コストの内訳

| | 12 nodes | | 4,800 nodes | |
|-----------|-----------|---------------|-------------|---------------|
| | Time(sec) | Flops/Peak(%) | Time(sec) | Flops/Peak(%) |
| 総計算時間 | 36.5522 | 7.4262 | 36.5705 | 7.5352 |
| 粘性項の計算 | 10.1949 | 11.1396 | 10.3859 | 11.1255 |
| Collision | 3.6774 | 7.4224 | 3.7717 | 7.4188 |
| 圧力の計算 | 6.0264 | 7.2425 | 6.1484 | 7.2508 |
| 圧力勾配の計算 | 12.7904 | 7.0555 | 13.0461 | 7.0665 |
| その他 | 3.8632 | | 3.2184 | |

4. 東日本大震災を模擬した石巻実地形津波解析

本研究では、東京大学情報基盤センターFX10 を用いて、石巻市街地に津波が侵入し、剛体でモデル化した直径9mの2つのタンクが地上構造物に衝突を繰り返しながら漂流する解析を行った[1][4]。本解析では、図11の石巻沿岸(4.0km×3.5km)と図14の石巻市街地(400m×550m)の2つのモデルを用いてズーム解析を行った。

地表データは石巻市から5m標高データを提供してもらい、地上構造物はGoogle Earthの建築物3Dmodelから震災前の町並みを再現した。入力境界条件は東北大学津波研究所の今井健太郎助教・越村俊一教授(東北大学)および国際航業株式会社から東日本大震災時に押し寄せた津波の波高データを提供してもらった。

まず、粒子直径1mの石巻沿岸の津波解析を行った。モデルの下側に波高データを入力境界条件として設定し、実時間800秒の解析に、東京大学情報基盤センターFX10の96ノードを用いて約7日間かかった。最大2.6億粒子の解析となった。図11は、粒子ベースの半透明可視化を行った結果である[6]。元々海であった部分は半透明で薄く可視化し、津波が地上に遡上している部分は半透明で濃く可視化した結果である。

牡鹿半島の北西に位置する石巻市は、東日本大震災の際には、南東から北西に向って津波が押し寄せた。これは牡鹿半島の南を通って津波が押し寄せたからである。しかしながら、図11の四角の枠内の市街地では、南西から北西に向って津波が遡上している。これは、石巻湾の防波堤に津波が誘導されたり、旧北上川を津波が遡上して溢れてきたりした結果である。この様に、石巻沿岸部の津波解析を、3次元の詳細な地形データを元に解析することによって、市街地に押し寄せる津波の方向を正確に再現することができる。

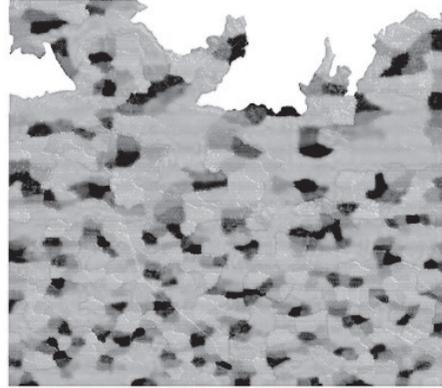


図11. 石巻沿岸の津波解析(時刻400秒)

図12は、ParMETISライブラリによって石巻沿岸モデルを2階層に領域分割した結果である。南側部分は水深が深いため、高さ方向の粒子数が多くなり領域分割された面積は少なめになっている。



(a) 第1階層目の領域分割



(b) 第2階層目の領域分割

図 12. 石巻沿岸モデルを2階層に領域分割した結果

図 13 は、図 11 の四角の枠上の波高と流速を取り出して作成した入力境界条件である。

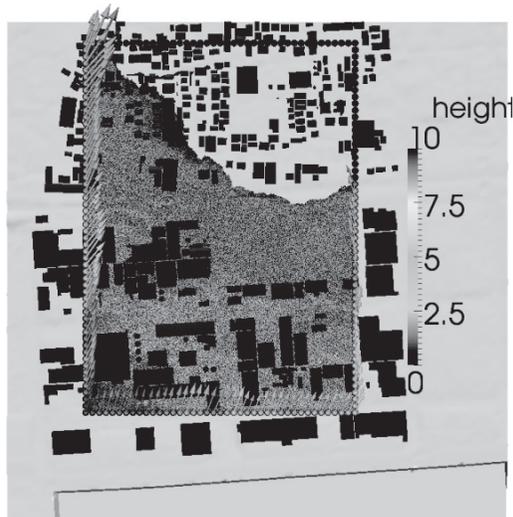
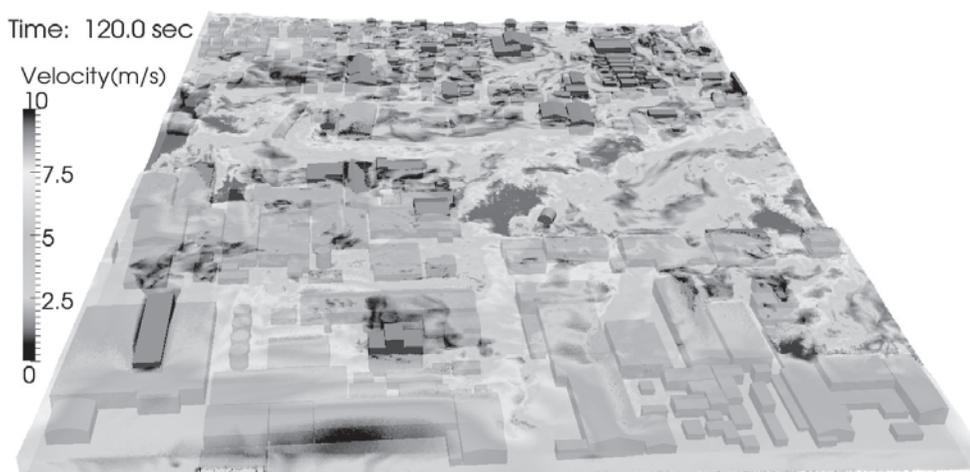


図 13. 石巻沿岸モデルの解析から求めた入力境界条件

この入力境界条件を用いて、図 14 のような、粒子直径 0.5m の石巻市街地に津波が侵入する解析を行った。実時間 300 秒の解析に、東京大学情報基盤センターの FX10 の 600 ノードを用いて約 12 時間かかった。最大 5000 万粒子、約 20 万タイムステップの解析となった。図 14(a) から、南北に走る大通りと東西を走る中央の県道に沿って津波が遡上し、地上構造物間に水流が入り込み高速な流れとなる縮流現象を再現することができた。図 14(b)からは、2つのタンクが流速の速い所に沿って流れていくのが確認できる。東日本大震災の際には、このタンクは、元あった場所から北東に向かって流れ、300m 先の広場の北の県道の中央分離帯にまで流れ着いた。本解析は、タンクは流れ過ぎて解析領域外まで行ってしまったが、タンクの流れる方向は再現できた。



(a) 時刻 30 秒での解析結果



(b) 時刻 120 秒での解析結果

図 14. 石巻市街地の津波解析

図 15 は、石巻市街地解析の各ステップの計算時間、第 1 レベルと第 2 レベルロードバランスの推移を示している。第 1 レベルのロードバランスは、各計算ノードが担当する（最大粒子数） \div （最小粒子数）であり、第 2 レベルのロードバランスは、各計算ノード内の各 thread が担当する（最大粒子数） \div （最小粒子数）の最大値である。第 1 レベルのロードバランスが 2.0 を超えた場合、全領域の領域再分割が行われ、第 2 レベルのロードバランスが 3.0 を超えた場合、全計算ノードで計算ノード内の領域再分割が行われる（第 1 層は領域再分割されない）。図 15 には、ファイル I0 と領域分割にかかる時間は省いてある。図 15 から第 1 層のロードバランスが支配的であるが、第 2 層のロードバランスも計算時間に影響を与えていることが確認でき、ロードバランスを取ることで各タイムステップの計算時間の上昇を抑えている様子が見取れる。

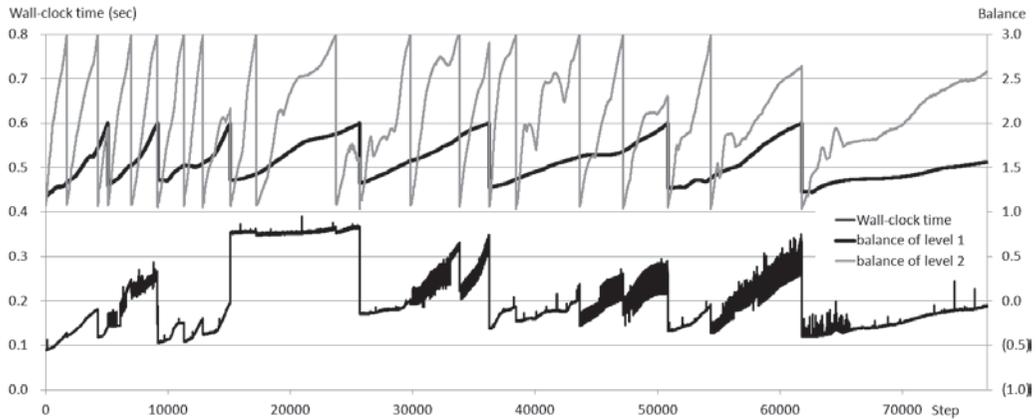


図 15. 1 タイムステップの計算時間と 2 階層のロードバランス

5. まとめと今後の課題

今回の最大の成果は、我々が開発を進めている分散メモリ並列 MPS 陽解法アルゴリズムの隣接間通信の部分の並列性能が高いことが確認できた。

東京大学情報基盤センター-FX10 を用いて 600 ノードを用いて、石巻市街地に津波が侵入し、タンク（剛体）が地上構造物に衝突を繰り返しながら漂流する解析に成功した。本解析では、剛体流体連成問題と剛体間接触解析を解いているが、今後、様々の物理モデルを導入する予定である。

本解析の計算時間は、図 9 の 1 ノード当り 65,536 粒子に相当する規模であり、理想的な場合からそれほどかけ離れていない性能が引き出せていることが確認できた。これまでは、このような解析を数日間から 1 週間程度かけて行ってきたが、今後は大規模並列計算を行うことで、短時間で結果が得られることが期待できる。

参考文献

- [1] M. Carlson, P. J. Mucha, G. Turk, Rigid Fluid : Animating the Interplay between Rigid Bodies and Fluid, SIGGRAPH 2004, ACM Trans. Graphics, 23, 377-384, 2004..
- [2] G. Karypis and V. Kumar, A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs, SIAM Journal on Scientific Computing, 20-1, pp. 359-392, 1999.
- [3] S. Koshizuka and Y. Oka, Moving-particle semi-implicit method for fragmentation of incompressible fluid, Nuclear Science and Engineering, 123, 421-434, 1996.
- [4] S. Koshizuka, A. Nobe, Y. Oka, Numerical Analysis of Breaking Waves Using the Moving Particle Semi-implicit Method, Int. J. Numer. Meth. Fluids, 26, 751-769, 1998.
- [5] 室谷浩平, 大地雅俊, 藤澤智光, 越塚誠一, 吉村忍, ParMETIS を用いた MPS 陽解法の分散メモリ型並列アルゴリズムの開発, 日本計算工学会論文集, No. 20120012, 2012.
- [6] S. Tanaka, K. Hasegawa, Y. Shimokubo, T. Kaneko, T. Kawamura, S. Nakata, S. Ojima, N. Sakamoto, H. Tanaka, and K. Koyamada, "Particle-Based Transparent Rendering of Implicit Surfaces and its Application to Fused Visualization", EuroVis 2012, Vienna (Austria), June, 5-8, 2012.