ISSN 2186-5647

GPUによる粒子法シミュレーション及び物理ベースCG

日大生産工 (院)　○小原　俊介　　日大生産工　角田　和彦
日大生産工　　豊谷　純

1　はじめに

粒子法は、連続体を有限個の粒子によって表し、連続体の挙動を粒子の運動によって計算する方法である。各粒子は速度や圧力といった変数を保持しながら移動する。また、粒子法は格子法に比べ、比較的新しいシミュレーション手法であり、差分法や有限要素法のように格子で空間を覆う必要がないので、大変形を伴う解析に適している。　本研究では粒子法の一つである、MPS(Moving Particle　Semi-implicit) 法を用いて 3 次元水柱崩壊シミュレーションを行う[1)]。また、MPS 法の解析において、粒子数が多いほど解析時間は長くなるため、GPU を利用して並列計算を行う[4)]。そして、その計算結果をマーチングキューブ法により流体をポリゴン化[2)]した後、POV-Ray でレンダリングする[3)]ことで流体をリアルに表現することを目的としている。

2　MPS 法

MPS 法は流体解析や構造解析に用いられる代表的な粒子法である。勾配や発散といった微分演算子に対応する粒子間モデルを用意し、これらの微分演算子に適用することによって離散化する。重み関数 w を導入し、粒子間相互作用モデルにはこの重み関数を利用する[1)]。本研究では解の安定化を図るために式 (1) の従来の重み関数から式 (2) の重み関数を用いている。式 (1) の重み関数では粒子間距離が近づきすぎた場合に、無限大になり、急激に密度が高まる可能性があるため、それを防ぐために式 (2) を採用している。

$$w(r) = \begin{cases} \dfrac{r_e}{r} - 1 & (0 \le r < r_e) \\ 0 & (r_e \le r) \end{cases} \tag{1}$$

$$w(r) = \begin{cases} \log \dfrac{r_e}{r} & (0 \le r < r_e) \\ 0 & (r_e \le r) \end{cases} \tag{2}$$

図 2 の r は粒子間距離、r_e は影響半径である。

式 (3) は勾配モデル、式 (4) はラプラシアンモデルである。式 (5) はラプラシアンモデルにおいて、統計的な分散の増加を解析解と一致させるための係数である。

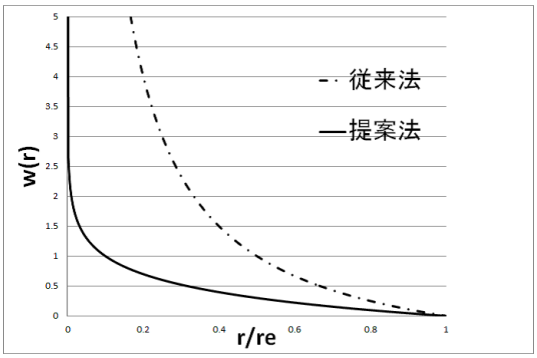


図 1. 重み関数

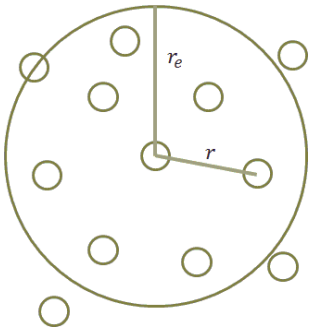


図 2 . 粒子間相互作用

$$\langle \nabla \phi \rangle_i = \frac{d}{n^0} \sum_{j \neq i} \left[\frac{\phi_j - \phi_i}{|r_j - r_i|^2} (r_j - r_i)\, w\left(|r_j - r_i|\right) \right] \tag{3}$$

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{\lambda n^0} \sum_{j \neq i} \left[(\phi_j - \phi_i) w\left(|r_j - r_i|\right) \right] \tag{4}$$

$$\lambda = \frac{\displaystyle\sum_{j \neq i} |r_j - r_i|^2 w(r_j - r_i)}{\displaystyle\sum_{j \neq i} w(|r_j - r_i|)} \tag{5}$$

ただし、d と n^0 はそれぞれの次元数、重み関数で表される粒子密度の一定値である。

Simulation by Particle Method on GPU and Physics-based Computer Graphics

Shunsuke OBARA, Kazuhiko KAKUDA, and Jun TOYOTANI

3　マーチングキューブ法

マーチングキューブ (Marching　cubes) 法は、コンピュータグラフィックスのアルゴリズムである。スカラーデータで与えられた等方向 3 次元ボクセルデータを、ポリゴンデータに変換するアルゴリズムである。特定のアルゴリズムで 1,0 に変換されたボクセルデータを対象とし、8 点からなる立方体を考えると、結果的に 8 つの頂点に 0 か 1 の数字をもった立方体が形成される。組み合わせは 2 の 8 乗の 256 通りが考えられる。しかし回転や反転を無視すると図 3 に示すように 15 種類となる。この原理を用いてライブラリ化した処理をすることで変換の高速化を図ることができる。各キューブを順番に処理し等値面でつなぐことでポリゴンデータに変換する[2)]。

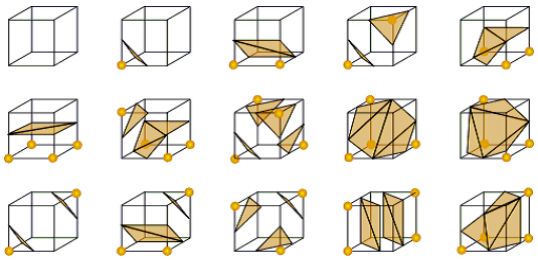


図 3 . マーチングキューブ法

4　近傍探索アルゴリズム

粒子法では、一つの粒子に対して、他のすべての粒子との距離を測る必要がある。その操作を近傍探索といい、MPS 法において最も時間がかかる処理である。従来では図 4 の左図のように一つの粒子に対して他の全ての粒子を探索する方法を用いる。しかし、N^2 回の計算を行うため膨大な計算量になってしまう。そこで、本研究では図 4 の右図のように、モデルをグリッドに区切り、周りのグリッドのみに対して近傍探索を行う方法を使用する。この方法を使用することにより、近傍探索の処理を減らすことができる。この方法は粒子数が増えるほど、従来の方法よりも効果的である。

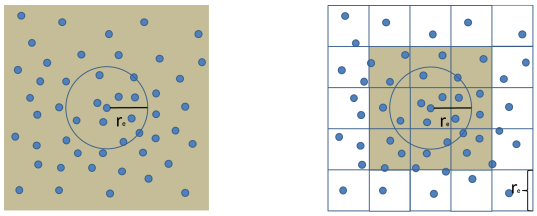


図 4 . 近傍探索アルゴリズム

5　CUDA による MPS 法アルゴリズム

本研究では GPU により並列計算を行う。並列計算を行うことで、計算に時間のかかる近傍探索や圧力ポアソン方程式の計算を高速で演算することができる。今回使った GPU を図 5 に示し、本研究の環境を表 1 に示す。

図 5 .GTX　690

表 1. システム環境

CPU	Intel Core i7, 3770K, 3.50GHz
Memory	DDR3 PC3-12800 16GB
OS	Ubuntu 12.04 64bit
gcc	version 4.4.7
GPU	Gefoce GTX 690 Bus interface PCI-e 3.0 x16 Memory Bandwidth 192.3 GB/sec Global Memory 2GB SM , CUDA Core 8 , 1536
CUDA	Driver and Tool kit and SDK 4.2

本研究で用いたアルゴリズムを図 6 に示す。本研究のアルゴリズムでは計算量の多い近傍探索と圧力ポアソン方程式の箇所を CUDA プログラムで解析した。圧力ポアソン方程式の解法には並列性の高い CG 法を用いて解いている。

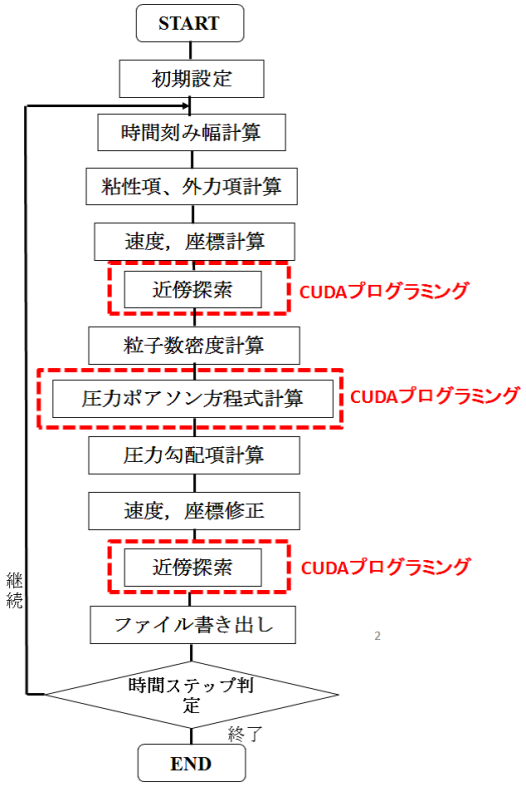


図 6 .CUDA による MPS アルゴリズム

6　物理ベース CG 化

ここでは、物理ベース CG 化への手順を図 7 に示し、①〜⑦について簡潔に記述する。
①エクセル等で粒子データを作成し、grid ファイルを作る。data ファイルには解析条件を書き込む。
②MPS プログラムで解析を行い解析する。
③解析結果の prof ファイルができる。
④マーチングキューブプログラムでは解析から得られた粒子データをポリゴンデータに変換し、そのデータを POV-Ray 形式に出力する。
⑤pov ファイルが作成される。
⑥POV-ray でレンダリングする。
⑦画像ファイルが作成される。

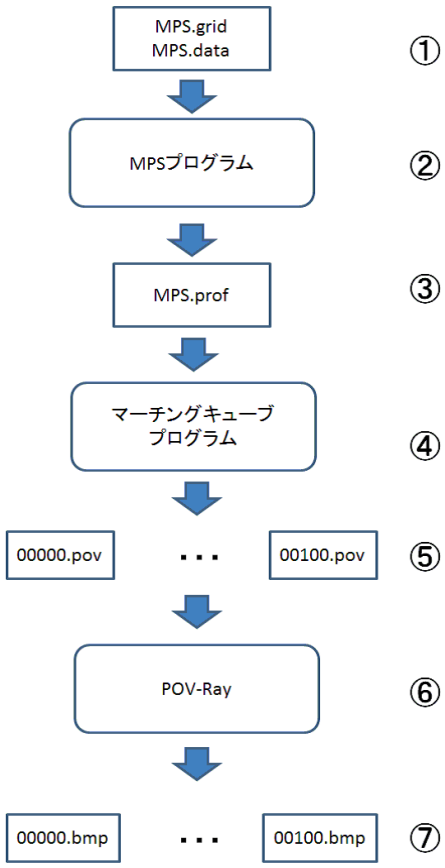


図 7 . 物理ベース CG 化

7　数値解析例

水柱崩壊シミュレーションに用いたモデルは粒子総数 89168 である。モデル中央の直方体の柱が流体の粒子の固まりである。このモデルは時間が経過すると流体の粒子が重力の影響を受けて崩壊する様子を表 2 の計算条件を用いて数値計算し、可視化する。

図 9 は上から初期状態の 0 秒, 水柱が崩壊している様子の 0.25 秒, 反対の壁の衝突している様子の 0.5 秒の粒子挙動である。図 9 (a) は CPU で解析した結果、図 9 (b) は GPU で GRID 分割法を使用した結果である。どちらも、結果はよく一致する結果となった。

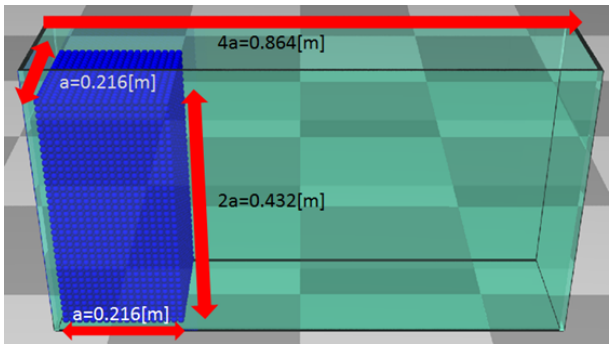
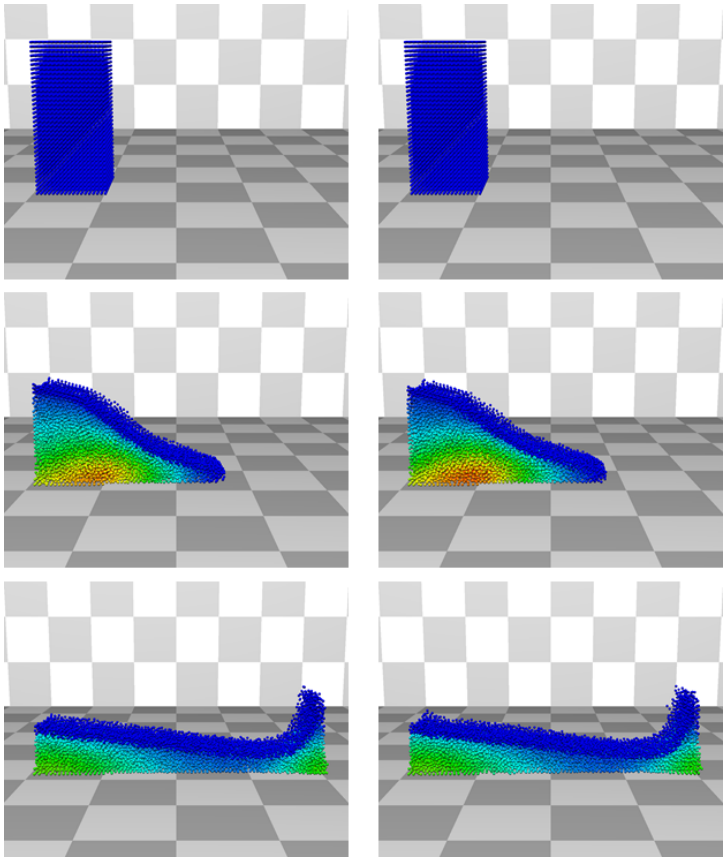


図 8 . 水柱崩壊モデル

表 2 . 計算条件

総粒子数	89168
粒子間距離	$0.012m$
密度	$1000kg^2/s$
重力加速度	$9.8m/s^2$
動粘性係数	$10 \times 10^{-6}m^2/s$



(a)CPU　　(b)GPU+grid

図 9 . 解析結果 (0s,0.25s,0.5s)

図１０に解析時間を示す。0.5s 間の解析を行った結果、CPU で解析した場合には 34189.46 秒 (9.49 時間) かかったが、GPU で解析すると 4659.06 秒 (1.29 時間) かかり、GPU ＋ GIRD 分割法で解析すると、3674.59 秒 (1.02 時間) かかった。この結果から、GPU での解析も GRID 分割法も精度が変わらず解析時間を短縮するのに有効であることが分かった。

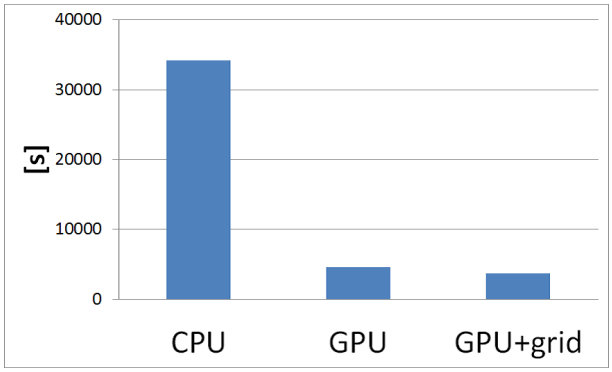


図１０. 計算時間の比較

表３. 計算時間

0.5 秒間における解析	CPU	GPU	GPU +Grid
解析時間	34189.46[s]	4659.06[s]	3674.59[s]

図１１、１２に粒子表現とマーチングキューブ法によってポリゴン化した表現を示す。図１１では粒子一つ一つを表示させているが、図１２ではポリゴン化された面を表示させているため、よりリアルな流体が表現されているのがわかる。

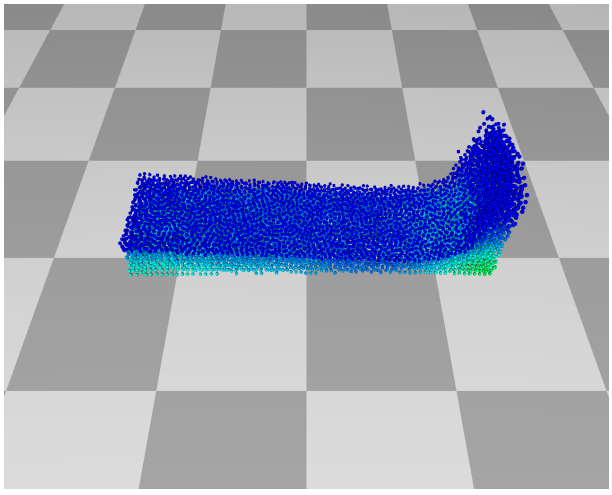


図１１. 粒子表現

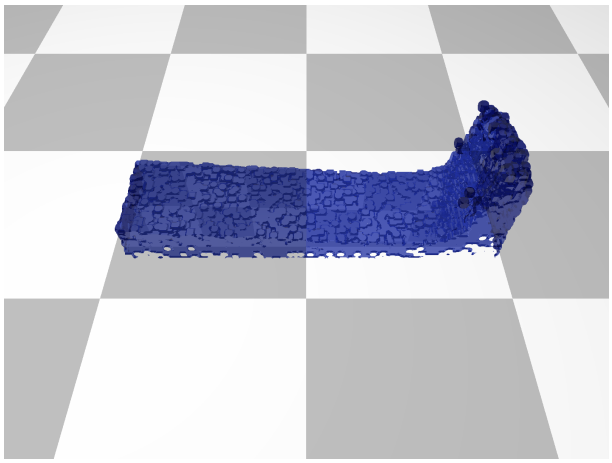


図１２. ポリゴン表現

8　おわりに

本研究では、GPU に重み関数を工夫した MPS 法により水柱崩壊シミュレーションを行い、その結果をマーチングキューブ法によってポリゴン化し、POV-Ray によってレンダリングすることで流体をリアルに表現することを目的にしてきた。その結果、以下のことが分かった。

- GPU を使用し、GRID 分割法を使用することにより演算時間を約 10 倍近く短縮することができた。
- GPU を使用した場合でも、解析結果はほぼ同じ結果を得ることができた。
- マーチングキューブ法を使うことにより、よりリアルな流体を表現することができた。

今後のテーマとしては、粒子数を増やしたモデルで解析し、よりリアルな表現ができるように改良したい。また、粒子数が増えるほど解析時間もかかるため GPU を効率よく使えるアルゴリズムを検討していきたい。

参考文献

1)S.Kosizuka and Y.Oka,”Moving-Particle Semi-implicit Menthod for Fragmentation of Incompressible Fluid”,NUCLEAR SCIENCE AND ENGINEERING,123,pp.421-434(1996)
2) 越塚誠一, 原田隆宏, 田中正幸, 近藤雅裕,「粒子法シミュレーション　物理ベース CG 入門」, 越塚誠一編, 培風館,pp1-65,(2008)
3)Kakuda,K.;Obara,S.;Toyotani,J.;Meguro,M.;Furuichi,M,”Fluid flow simulation using particle method and its physics-based computer graphics.”,CMES,vol.83,no.1,pp57-72(2012)
4)Kazuhiko Kakuda,Tsuyoki Nagashima,Shunsuke,Obara,”Particle-based Fluid Flow Simulation on GPGPU Using CUDA”,CMES,submit