

粒子法陽解法におけるデータレイアウトを考慮した CUDA 実装

CUDA Implementation Considering Data-Layout for MPS Method

富岡 我空†
Gaku Tomioka

吉田 明正†
Akimasa Yoshida

1 はじめに

粒子法 [1][2] は、流体力学シミュレーションに広く用いられる数値解析手法である。大規模なシミュレーションを計算するためには、膨大な計算時間を必要としており、並列処理による高速化が期待されている [3]。本稿では、非圧縮性流体を解析する粒子法 (MPS 法) の陽解法を対象とし、その高速化を目指す。粒子法陽解法は、圧力勾配項などを陽的な計算により代入操作のみで求めるため CUDA 実装が容易にできる。粒子法陽解法の高速度化について多くの研究 [4][5] が行われてきたが、GPU 上でデータ再配置を伴う高速化手法は提案されていない。そこで、本稿では AoS 形式で格納されるバケット内粒子の空間データを、動的に再配置することにより、GPU 上での実行時間の短縮を目指す。性能評価では、NVIDIA Tesla K80 上にて 141,216 粒子の 3 次元水柱崩壊のシミュレーションを行い、提案手法の有効性を確認する。

2 粒子法陽解法の CUDA 実装

本章では粒子法陽解法のアルゴリズムと CUDA 実装手法について述べる。

2.1 粒子法陽解法

粒子法の MPS 法における陽解法の計算には、式 (1) の流体の支配方程式であるナビエーストックス方程式 (運動量保存則) と式 (2) の圧力と密度の関係式を用いる。

$$\frac{DP}{Dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mu + g \quad (1)$$

$$\frac{\partial P}{\partial \rho} = c^2 \quad (2)$$

式 (1) の P は圧力、 ρ は密度、 μ は流体の速度、 ν は動粘性係数、 g は重力加速度、式 (2) の c は音速である。粒子法のアルゴリズムでは、タイムステップ毎に式 (1) の重力項、粘性項、圧力勾配項を計算する。ここで、各タイムステップの処理について述べる。

- (1) 粒子をバケットに格納する。
- (2) 粘性項と重力項から仮の加速度を求める。
- (3) 仮の加速度から仮の位置と速度を求める。
- (4) 剛体衝突の運動量を計算する。
- (5) 仮の圧力を求める。
- (6) 圧力勾配項から加速どの修正量を求める。
- (7) 加速度の修正量から更新後の位置と速度を求める。
- (8) タイムステップ後の圧力を求める。

† 明治大学大学院先端数理科学研究科ネットワークデザイン専攻
Network Design Program, Graduate School of Advanced
Mathematical Sciences, Meiji University



図 1 AoS データレイアウト。

各処理において粒子間のデータ依存がないため、粒子計算において粒子数分の並列性が利用できる。手順 (2), (4), (5), (6), (8) において圧力勾配項を計算する際、各粒子は影響半径内にある粒子と相互作用を計算するために近傍粒子探索を行う。本稿では、バケットと呼ばれる立方体の空間で粒子の位置情報を管理するバケット法を採用する。バケットの一边を影響半径にすることで近傍粒子探索を行う際に、粒子の所属するバケットを中心として x, y, z 方向にそれぞれ ± 1 バケット分の 27 バケット内に存在する粒子との相互作用を計算するだけでよく計算量が削減される。

2.2 CUDA 処理系

CUDA (Compute Unified Device Architecture) とは、NVIDIA が開発した GPU 向けの総合開発環境であり、CUDA を用いることで GPU の並列処理能力を活用して高速化することができる。各タイムステップの処理は、8 つの CUDA カーネルとして実装されている。粒子法では粒子計算を各粒子ごとに並列化するため、GPU に搭載される多数のスレッドに粒子を対応させて並列処理する。粒子が近傍粒子との相互作用を計算する際にメモリアクセスを効率化するため、近傍粒子探索におけるデータ再配置を適用する。

3 GPU 上でのデータ再配置を伴う CUDA 実装

GPU 上のグローバルメモリへのアクセスを高速化するため、タイムステップ毎にバケット内粒子のデータレイアウト変換を行い、近傍粒子探索におけるメモリアクセス時間の軽減を目指す。

3.1 AoS 形式によるデータレイアウト

本稿で用いた AoS (Array of Structures) データレイアウトを図 1 に示す。AoS 形式はベクトル x, y, z の並びが繰り返されるデータレイアウト形式である。粒子の相互作用を計算する際に、読み出すデータの順に並んでいるためアクセスは効率よく行える。

3.2 バケット法におけるデータ再配置の適用

バケット内の粒子番号は線形リストにより管理されているが、通常、バケット内の粒子番号は非連続であるため、粒子番号順に配置された粒子データのメモリアクセ

