

密度の異なる非圧縮性流れの運動シミュレーション

著者 宮本 健太郎

指導教員 下條 雅史

1. 目的

物理シミュレーションの1つに流体のシミュレーションが挙げられる。本研究では、大規模の変形を扱う流体シミュレーション法である粒子法の中でも MP(S)(Moving Particle (Semi-implicit))法を用いて、水とそこに配置した油など密度の異なる液体の運動をシミュレーションすることを目的とした。

2. 概要

2.1 粒子法

連続体を有限個の粒子の集合として表し、この粒子1つ1つに速度圧力といった変数を保持させ、連続体の挙動を粒子の運動によって計算する方法を粒子法という

2.2 MP(S) (Moving Particle (Semi-implicit))法

今回取り扱う MP(S)法は粒子法の1つであり、連続体の挙動を表す微分方程式の微分演算子に粒子間相互作用モデルを適用して離散化を行う。

2.3 非圧縮性流れ

非圧縮性流れは圧力や温度による密度の変化が無視でき、密度がほぼ一定である性質をもつ。今回はこの非圧縮性流れの1つである水を扱っている。非圧縮流れの支配方程式はナビエストークス方程式の他に以下のものがある。

$$\frac{dp}{dt} = 0$$

2.4 CUDA(Compute Unified Device Architecture)

CUDA は NVIDIA 社製 GPU 向けのプログラミング環境であり、C/C++言語を元に独自の拡張を行った専用の言語および対応するコンパイラといくつかの数値計算ライブラリから構成されている。適切に利用することで GPU の持つ性能を引き出すことができる。

2.4.1 グリッド、ブロック、スレッド

GPU コンピューティングでは CPU と GPU の両方を用いて計算する。CUDA ではスレッドを主体として並列計算が行われる。複数のスレッドの集まり

をブロック、複数のブロックの集まりをグリッドとして扱い、ブロックごとに GPU 内の演算部が搭載されており、その内部の CUDA コアがスレッドを処理する。

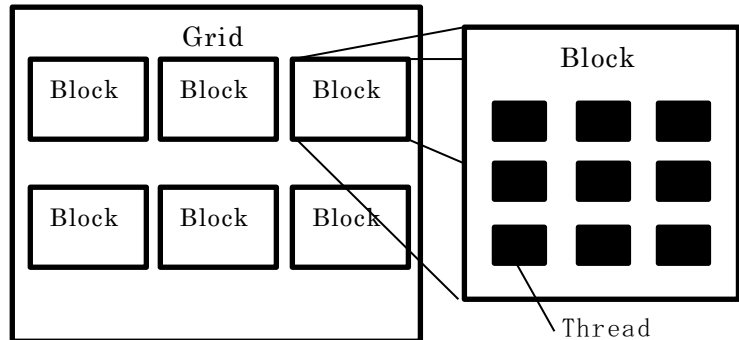


図1 グリッド、ブロック、スレッドの構造

2.4.2 プログラムの基本的流れ

CPU やメインメモリ側を「ホスト」、GPU やビデオメモリ側を「デバイス」と呼び、区別している。また、CUDA において CPU に実行させるプログラムを「ホストプログラム」と呼び、GPU に実行させるプログラムを「カーネル」と呼ぶ。ホストプログラムで扱うデータはメインメモリに、カーネルで扱うデータはデバイス側のメモリに保存され、やり取りされる。

データのデバイスへの受け渡しや、ホスト側での受け取りなどはすべて CPU が命令し、GPU は処理だけを行う。

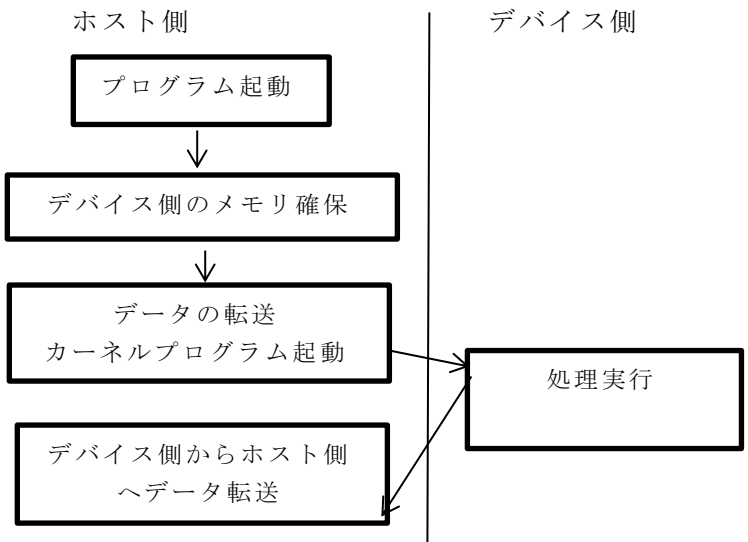


図 2 プログラムの大まかな流れ

### 3. 計算アルゴリズム

計算の流れを図 3 に示す。

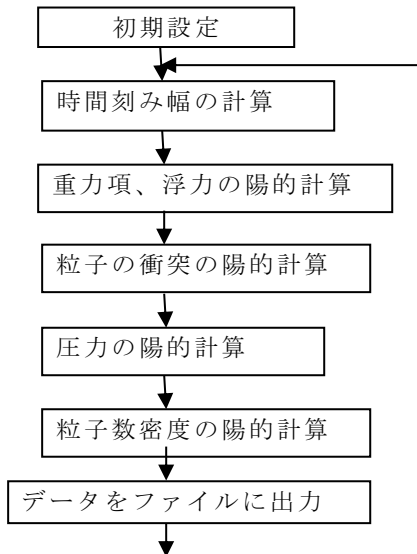


図.3 計算アルゴリズム

テキストファイルから粒子の座標やタイプ(水や油等)のデータを読み込み、メインループに入って数値が安定する最大の刻み幅を計算する。

次に、各種の計算を行う。この計算の際には、ナビエ-ストークス公式  $\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + \mathbf{g}$  を

基本公式として用いて行う。上式の右辺第 1 項は圧力勾配項、第 2 項は粘性項、第 3 項は重力項である。ここで、 $\mathbf{u}, t, \rho, P, \nu, \mathbf{g}$  は粒子の速度、時間、密度、圧力、粘性係数、重力加速度である。重力項や浮力、衝突、圧力の計算の際は、仮の粒子数密度を用いて、 $\Delta t$  秒後の粒子の速度、位置を求める。ここで求めた値は仮のものである。粒子数密度が一定量  $n^0$  になっていないため、 $n^0$  に修正する。

浮力は、 $g*(1-\rho_0/\rho_1)$  の式で与える。

$\rho_0, \rho_1$  は、水の密度、水より軽い液体の密度である。陽的計算を用いて、メモリの使用を抑え短い時間で計算を行う。

### 4. 実行結果

水より密度の小さい液体が 1 種類ある時と 2 種類ある時の 2 つでシミュレーションを行った。初期位置を図 4、図 6 のように配置し、以下の条件でシミュレーションを行った。

- ・水は粘性が小さいと考え、粘性項の作用は考えない

- ・粒子間距離を 0.008m とし水を 100 段×100 段配置した。密度の小さい液体は下段から 10 段ずつ配置し、水より軽い液体の密度は 1 種類の時密度 0.6(濃)、2 種類の時密度 0.6(濃)と密度 0.8(薄)とした。

- ・ステップ数はそれぞれ 8 万回とした。

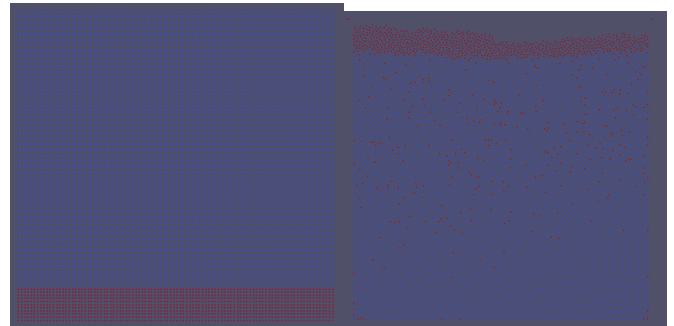


図.4 1 種類の時の初期位置

図.5 実行結果



図.6 2 種類の時の初期位置

図.7 実行結果

### 5. 結論

水より軽い液体の密度に比例してその粒子を軽くすると、その液体は水に閉じ込められて動かない。これは衝突計算ですべての粒子の質量を等しくすることで解決し、自然な結果が得られた。2 種類の密度が小さい液体を配置した時のシミュレーションは、最後に上部から密度の小さい順に層となり、自然なシミュレーションとなった。しかし、この状態に安定するまで多くの時間がかかり、不自然さが残った。

### 6. 参考文献

- ・ Jason Sanders , Edward Kandrot 著 「CUDA BY EXAMPLE 汎用 GPU プログラミング入門」
- ・ 越塚誠一著 「粒子法」
- ・ 越塚誠一、柴田和也、室谷浩平著「粒子法入門」