

## 計算格子を用いたSPHシミュレーションの精度向上に関する考察

前田和広<sup>†1</sup> 奥山祐市<sup>†1</sup> 黒田研一<sup>†1</sup>

本論文では、多数の単精度FPUが実装されているデバイスを用いて、SPHにおける流体シミュレーションの計算高速化を行う場合を考慮し、計算格子の導入と部分的な倍精度演算を用いることにより演算精度を向上させた。提案手法をGPU上に実装し、ダム崩壊問題におけるシミュレーションの経過時間と、先端位置の平均二乗誤差の比較を行い評価した。その結果、提案手法が計算速度及び演算精度の点において、倍精度と単精度の中間に位置していることがわかった。

### Precision Improvement in SPH Simulation Using a Computational Grid

KAZUHIRO MAEDA,<sup>†1</sup> YUICHI OKUYAMA<sup>†1</sup>  
and KENICHI KURODA<sup>†1</sup>

In this paper, we work on the precision improvement of the floating point arithmetic in the SPH simulation by introducing computational grid and the partial doubleprecision arithmetic in order to improve calculation speed using an accelerator with many single-precision FPUs. We reveal the efficiency of proposed method by using the GPU to evaluate an elapsed time and the MSE of precision. The result shows that the proposed method can get a trade-off between the double-precision arithmetic and the single-precision arithmetic with acceptable computational speed.

### 1. 緒言

粒子法は流体力学におけるラグランジュ的解析手法の一つである。この手法では多数の粒子で流体を表現し、流体の挙動を粒子の動きによってシミュレートする。また、この手法は液体等の自由界面の解析、構造物と流体との相互作用、大変形を伴う現象の解析に適している。粒子法の実現方法として、主にSmoothed Particle Hydrodynamics (SPH)<sup>1)</sup>とMoving Particles Semi-implicit (MPS)<sup>2)</sup>がある。実際に映像製作や、災害分析を行う際に、これらの手法が適用される。一方、汎用演算装置で計算を行うには、時間を要することが既に知られている<sup>3)</sup>。

近年、計算高速化の分野において、画像描画を目的としたグラフィクスプロセッシングユニット (GPU) を用いて汎用計算を行う GPGPU が注目されている<sup>4)</sup>。GPU は汎用演算装置に比べ、多くの整数演算器や浮動小数点演算器 (FPU) を持つており、並列計算に適している。GPU を効率的に利用することにより、汎用演算装置に比べ高速な数値計算を可能とする。さらに、GPU は計算速度における優位性だけでなく、対コスト性能においても優れている。実際に GPU を用いることにより、SPH 法による流体シミュレーションを高速化できることがわかっている<sup>5)</sup>。

一般的に、流体シミュレーションをコンピュータで行う場合、数値表現として浮動小数点形式が用いられる。市販されている GPU に実装されている FPU の多くは単精度 FPU であり、倍精度 FPU はそれに比べて少ない。単精度 FPU を用いた場合、解析対象の振る舞いによっては正しい結果が得られない場合があるが、倍精度 FPU は単精度 FPU に比べて計算速度の点において劣る。つまり、計算速度と演算精度の間にトレードオフが存在する。

本論文では、多くの単精度 FPU が実装されているデバイスを用いて、SPH における流体シミュレーションの計算高速化を行う場合に、演算精度を向上させる手法の提案とその考察を行う。提案手法では、SPH の座標系に計算格子の概念を導入することにより、浮動小数点演算の桁落ちを防ぐ。加えて一部の物理パラメータを保持する変数や浮動小数点演算を倍精度で行うことにより、計算格子の座標を用いた計算時の丸め誤差を防ぐ。最後に提案手法の評価を行うために、ダム崩壊問題におけるシミュレーションの経過時間及び粒子の先端位置との比較を、GPU を用いて行う。

本論文の構成は後述のとおりである。まず第2章で、SPH の概要を述べる。第3章では SPH を用いた流体シミュレーションの実装と、浮動小数点演算における問題を述べる。第4章では提案手法について述べる。そして、第6章では提案手法の評価を述べる。最後に第

<sup>†1</sup> 会津大学コンピュータ理工学部

The School of Computer Science and Engineering, The University of Aizu

5章で結論を述べる。

## 2. Smoothed Particle Hydrodynamics

### 2.1 Smoothed Particle Hydrodynamics の概要

SPHは天体物理学の問題を解くために、Lucyによって発明された<sup>6)</sup>。SPHでは粒子は解析対象のオブジェクト形状に合わせて生成される。また、各粒子の状態は支配方程式によって決定される。ラグランジュ形式の支配方程式は以下のとおりである。

- 連続の式

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \vec{v} \quad (1)$$

- 運動量の式

$$\frac{D\vec{v}}{Dt} = -\frac{\nabla P}{\rho} + \vec{F} \quad (2)$$

これらの方程式にはいくつかのパラメータが含まれる。 $\rho$ 、 $\vec{v}$ 、 $P$  及び  $\vec{F}$  はそれぞれ密度、速度、圧力、外力である。本論文では外力として、重力と固定する物体からの力を考慮する。加えて、運動方程式は SPH によって近似されるので、連続の式は常に満たされる。

SPH の計算では、一つの粒子は表 1 に示される、いくつかのパラメータを保持する。

表 1 粒子が持つパラメータ

Table 1 Parameters of one particle

位置	$\vec{r}$	密度	$\rho$
速度	$\vec{v}$	圧力	$P$
加速度	$\vec{a}$	質量	$m$
平滑距離	$h$		

SPH では、位置  $\vec{r}_i$  における、物理量  $f(\vec{r}_i)$  及び物理量の勾配  $\nabla f(\vec{r}_i)$  は、近傍粒子との重み付け係数の和により表す。物理量  $f(\vec{r}_i)$  は式 (3) により表現される。また、物理量の勾配は、式 (4) により表現される。

- 物理量

$$\langle f(\vec{r}_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f(\vec{r}_j) W(\vec{r}_i - \vec{r}_j, h_{ij}) \quad (3)$$

- 物理量の勾配

$$\langle \nabla f(\vec{r}_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} f(\vec{r}_j) \nabla W(\vec{r}_i - \vec{r}_j, h_{ij}) \quad (4)$$

$h_{ij} = \frac{1}{2}(h_i + h_j)$  であり、それはカーネル関数のサイズを表し、平滑距離と呼ぶ。 $N$  は問題領域の粒子  $i$  における近傍粒子の数である。また、 $\vec{r}_j$  は、近傍粒子の位置を表す。カーネル関数  $W(\vec{r}, h)$  は、ある粒子  $i$  に影響を与える近傍粒子の、重み付け係数を求める関数である。カーネル関数には様々な実装が存在するが、本論文では下記のスプライン型のカーネル関数を採用する。

- スプライン型のカーネル関数

$$W(\vec{r}, h) = Cn \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & 0 \leq q < 1 \\ \frac{1}{4}(2-q)^3 & 1 \leq q < 2 \\ 0 & 2 \leq q < +\infty \end{cases} \quad (5)$$

$q$  は  $\frac{|\vec{r}|}{h}$  と等しい。スプライン型のカーネル関数は  $r = 0$  付近をピークに持ち、 $2h$  程度の広がりを持つ関数である。定数  $Cn$  は  $\int W(\vec{r}, h) d\vec{r} = 1$  を満たす。 $Cn$  は 1 次元及び 2 次元、3 次元空間においてはそれぞれ  $\frac{2}{3h}$ ,  $\frac{10}{7\pi h^2}$ ,  $\frac{1}{\pi h^3}$  を満たす。

### 2.2 密度の近似

SPH では粒子の分布を決めるため、密度の近似は重要である。本論文では、密度の近似手法として summation density の式を用いる。この式では、粒子間の距離を求めた上で、カーネル関数によって影響が及ぶ粒子を選択し、密度  $\rho_i$  の計算を行う。

$$\rho_i = \sum_{j=1}^N m_j W(\vec{r}_i - \vec{r}_j, h_{ij}) \quad (6)$$

### 2.3 運動方程式の近似

SPH によって近似された運動方程式は以下のとおりである。

$$\frac{d\vec{v}_i}{dt} = -\sum_{j=1}^N \left( \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W(\vec{r}_i - \vec{r}_j, h_{ij}) \quad (7)$$

式 (3) と (4) を用いて運動方程式 (2) を近似するとき、方程式 (7) が導かれる。粒子間の相互作用をこの式により求める。式 (7) の  $\Pi_{ij}$  は人工粘性項であり、粒子の浸透による数値

振動を防ぐために導入される。人工粘性項  $\Pi_{ij}$  は以下のように表される。

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{c}_{ij} \phi_{ij} + \beta \phi_{ij}^2}{\bar{\rho}_{ij}} & \vec{v}_{ij} \cdot \vec{r}_{ij} < 0 \\ 0 & \vec{v}_{ij} \cdot \vec{r}_{ij} \geq 0 \end{cases} \quad (8)$$

ただし、

$$\phi_{ij} = \frac{h_{ij} \vec{v}_{ij} \cdot \vec{r}_{ij}}{|\vec{r}_{ij}|^2 + \varphi^2} \quad (9)$$

$$\bar{\rho}_{ij} = \frac{1}{2}(\rho_i - \rho_j) \quad (10)$$

$$\bar{c}_{ij} = \frac{1}{2}(c_i - c_j) \quad (11)$$

$$\vec{v}_{ij} = \vec{v}_i - \vec{v}_j \quad (12)$$

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j \quad (13)$$

$\varphi$  は 2 つの粒子が近づくときに数値発散を防ぐために挿入される値である。 $\alpha$  及び  $\beta$  は 1.0 程度の設定する値であり、それぞれ体積による粘性、相互浸透を防ぐ働きをする。人工粘性項  $\Pi_{ij}$  は圧力に関する項に挿入される。

## 2.4 状態方程式

圧力  $P$  以下の状態方程式によって求める。

$$P_i = B \left( \left( \frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right) \quad (14)$$

$\gamma = 7$  は液体シミュレーションにおいてよく使われる値である<sup>8)</sup>。 $\rho_0$  は参照密度である。 $B$  は問題依存の値であり、密度の最大変化に応じて設定される。

## 3. SPH の実装

### 3.1 SPH のフローチャート

SPH のフローチャートを図 1 に示す。シミュレーション時間  $t$  が  $t_{end}$  に達するまで、“Nearest neighboring particle searching” から “Output” までの処理を繰り返し実行する。フローチャート内の各処理の詳細を以下に述べる。

- Initialization

シミュレーション対象の形状に応じて粒子を生成し、各粒子のパラメータの値を初期化する。

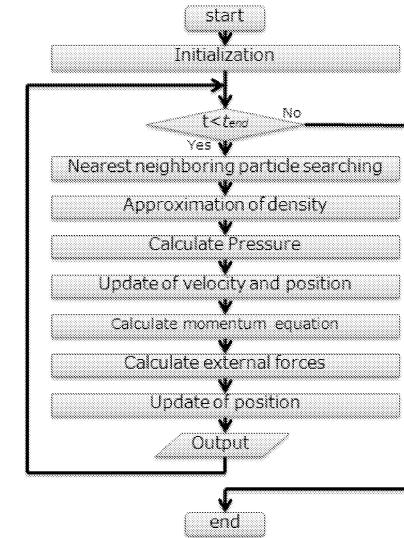


図 1 SPH シミュレーションのフローチャート  
Fig. 1 Flow chart of SPH simulation

- Nearest neighboring particle searching  
SPH では粒子同士の接続情報を持たないので、ループ毎に接続情報を求める必要がある。ここでは後述の NNPS アルゴリズムを用いて探索を行う。
- Approximation of density  
全ての粒子の密度は summation density (6) の式を用いて計算する。この式は近傍粒子を用いて求める。
- The equation of state  
全ての粒子の圧力は状態方程式 (14) を用いて計算する。本論文では水の挙動をシミュレーションする。
- Update of velocity and position (1/2)  
本論文では速度及び位置の時間積分に蛙飛び法<sup>7)</sup>を適用する。その為、前半部分と後半部分に計算を分割し、ここでは前半部分の計算を行う。
- Calculate momentum equation

全ての粒子の運動量は式(7)によって計算する。この式も同様に近傍粒子を用いて求める。

- Calculate external force

本論文では外力として重力及び位置を固定している粒子からの力を扱う。粒子が境界粒子に近づく場合、斥力が発生する。

- Update of velocity and position (2/2)

蛙飛び法における後半部分の計算をここで行う。

- Output

必要に応じて粒子の各パラメータはここで出力する。

### 3.2 Nearest neighboring particle searching

SPHでは粒子同士の接続情報を持たないため、相互作用の計算を行うには近傍粒子の探索を毎回行う必要がある。Nearest Neighboring Particle Searching (NNPS) は、近傍粒子を探索するためのアルゴリズムである。SPHにおける探索の主な方法としては、全通り探索と、リンクリスト探索、ツリー探索があり、それぞれオーダーは  $O(N^2)$ ,  $O(N)$ ,  $O(N \log N)$  である<sup>9)</sup>。本論文では NNPS としてリンクリスト探索を用いる。

リンクリスト探索アルゴリズムは、平滑距離が一定の値である場合に使用することができる。図2に示すように、問題領域は仮想的なセルにより分割される。セルの長さは問題領域に適した値が設定される。本論文ではスプライン型のカーネル関数を採用しているので、セルの長さは  $2h$  を用いる。ある粒子  $i$  に影響を与える近傍粒子は、その粒子が属しているセル及び近接セルの、リンクリストから求めることができる。

### 3.3 SPHにおける浮動小数点の誤差

浮動小数点演算では、絶対値の等しい数値同士の減算において大幅な有効桁数の減少が発生する。これを桁落ち誤差と呼ぶ。桁落ち誤差は、絶対値が等しく符号の異なる数値同士の加算においても発生する。これら誤差は浮動小数点形式における、正規化の処理で発生する。浮動小数点形式では最上位ビットは常に0が存在しないように正規化されるが、それにより下位ビットにゼロが挿入されるからである。

## 4. SPHの精度向上手法

### 4.1 計算格子の概念

本論文では、桁落ち誤差を防ぐために SPH には本来存在しない2種類の座標系を導入する。一つは問題領域の原点から格子点への絶対座標系であり、もう一つは、格子点から粒子

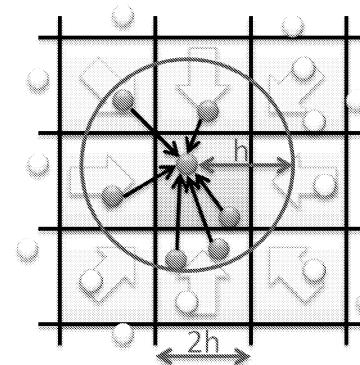


図2 リンクトリストを用いた2次元空間における近傍粒子探索

Fig. 2 Linked list algorithm for searching the nearest neighboring particles in two-dimensional space

の座標への相対座標系である。本論文では、これらの座標系を用いるために導入する格子点群を特に計算格子と呼ぶ。

### 4.2 計算格子を用いた精度向上

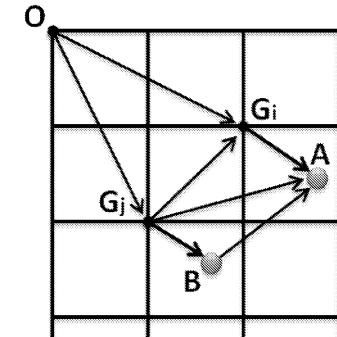


図3 計算格子を用いた絶対座標及び格子点からの相対座標

Fig. 3 Vector of grid values and relative values

式(6)及び(7)は、粒子間距離計算を含み、桁落ち誤差が発生する可能性がある。図3において、点 $G_i$ は点Aに対応する格子点であり、点 $G_j$ は点Bに対応する点である。点Oは問題領域の原点である。計算格子を導入しない粒子間距離計算において $\vec{BA}$ を求めるには、 $\vec{OA}$ から $\vec{OB}$ を引く。一方、計算格子を適用した場合の粒子間距離計算では、絶対座標系と相対座標系の計算をそれぞれ行う。まず、 $\vec{G_jG_i}$ または $\vec{G_iG_j}$ を、格子点 $G_i$ 及び格子点 $G_j$ から求める。実際には、原点から数えた格子点の番号の差をセルの長さで乗ずれば良い。その後、格子点に基づいた相対座標を用いて、粒子間距離を求めればよい。つまり、 $\vec{G_jA} - \vec{G_jB}$ または $\vec{G_iA} - \vec{G_iB}$ を計算すれば良い。これにより、粒子間距離 $\vec{BA}$ を求めることができる。

#### 4.3 NNPSのデータ構造を用いた計算格子

本論文では、格子点からの粒子の相対座標をNNPSの処理内で行う。

NNPSアルゴリズムを特定のデバイス上に実装する場合、問題領域に応じた多次元配列をメモリ上に確保する。例えば図3では、問題領域は2次元であり、それぞれのセルは粒子との接続情報を持つので、NNPSのためにメモリ上に3次元配列を確保する。前で述べたとおり、問題領域はセルによって分割されるので、各セルの頂点座標は一意に表現することができる。例えば $i$ をセルのインデックス番号とすると、x軸方向の頂点座標は $2h * i$ で表すことができる。よって、セルのインデックス番号を用いることにより、計算格子の導入を行う。なお、ある粒子 $i$ の絶対座標及び相対座標は、その粒子が属するセルが変わることに再計算する必要がある。

#### 4.4 部分的倍精度演算

計算速度と考慮し、基本的に浮動小数点演算は単精度で行い、パラメータの保持も単精度で行う。しかしながら計算格子を適用した座標系では、粒子の座標は二つの単精度浮動小数点で保持されるので、精度は単精度より高い。よって、計算格子を適用した座標系を用いた浮動小数点演算は、部分的に倍精度で行う。さらに、速度のパラメータの変数を倍精度とする。

### 5. 評価

#### 5.1 評価環境

流体シミュレーションのプログラムはC++、OpenGL及びNVIDIA Compute Unified Device Architecture (CUDA)<sup>10)</sup>を用いて実装を行った。また、プログラムはCore i7 920、9.0GB RAMとGeForce 480 GTX GPUを搭載したコンピュータ上で実行した。

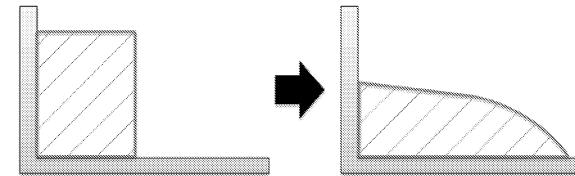


図4 ダム崩壊問題

Fig. 4 Dam break problem

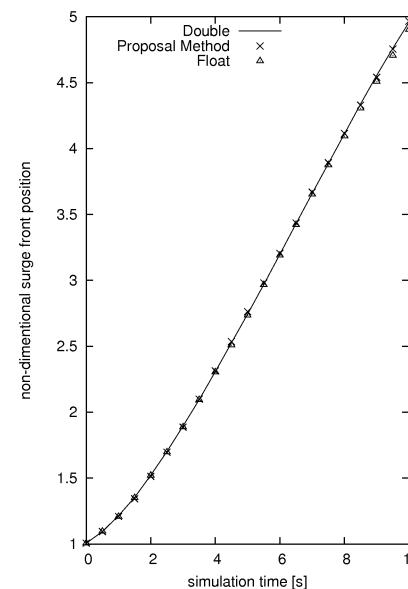


図5 粒子の先端位置

Fig. 5 Surge-front position

#### 5.2 ダム崩壊問題

本論文では提案手法の検証のために、ダム崩壊問題を用いた。ダム崩壊問題の概要を図4に示す。問題領域において水柱の崩壊時における、水平方向の広がりの挙動と、シミュレーションに要した経過時間の評価を行う。提案手法及び倍精度、単精度のそれぞれにおける、

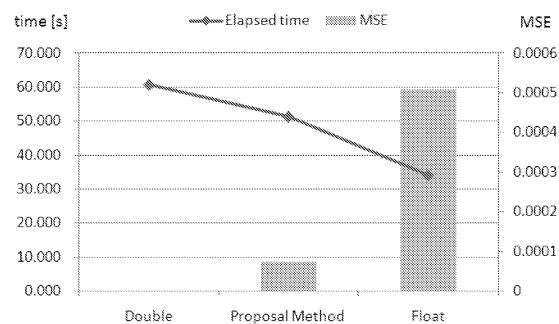


図 6 粒子の先端位置の MSE と経過時間のグラフ

Fig. 6 The graph of the MSE of the surge-front position, and an elapsed time

ダム崩壊問題を用いた流体シミュレーションを行った。シミュレーションに用いた粒子数は 2866 個で、水柱が接する境界には固定された粒子を用い、シミュレーション時間  $t_{end}$  は物理時間 100 秒とした。グラフとして図 5 に、各時間における粒子の先端位置の関係を示す。グラフに示されている粒子の先端位置は水柱の右端により正規化してある。加えて、シミュレーション時間  $t = 90$  のダム崩壊問題における粒子の分布を図 7 にそれぞれ示す。

表 2 粒子の先端位置の MSE と経過時間の値

Table 2 Value of the MSE of the surge-front position and and elapsed time

	倍精度	提案手法	単精度
経過時間	60.700	51.371	34.086
MSE	-	0.000073	0.000509

同様に物理時間 100 秒のシミュレーションに要した経過時間と、倍精度を基準とした平均二乗誤差 (MSE) を表 2 と図 6 に示す。提案手法に関する MSE は、単精度と比べ  $1/7$  となった。また経過時間は、倍精度と比較して約 15 %ほど速い結果となった。これらの結果より、提案手法が計算速度及び演算精度の点において、倍精度と単精度の間に位置していることがわかった。また、SPH シミュレーションを倍精度 FPU が遅いデバイス上で計算高速化する場合に、要求される演算精度に応じてトレードオフが可能であることを示した。

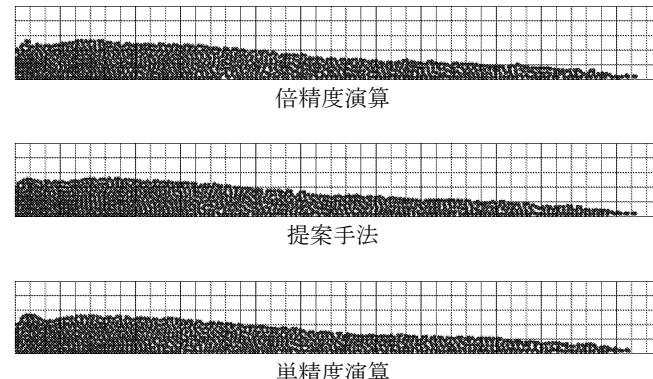


図 7 ダム崩壊問題における粒子の分布 ( $t=90.0$ )

Fig. 7 The particle distribution of the collapsed water in the dam break problem ( $t=90.0$ )

## 6. 結 論

SPH は液体等の自由界面の解析、構造物と流体との相互作用、大変形を伴う現象の解析に適しているが、汎用演算装置で計算を行うには、時間を要する。その為、多くの単精度 FPU を持った GPU を用いて SPH シミュレーションの高速化を試みた。しかしながら、単精度 FPU では正しい解が得られない可能性があり、また倍精度 FPU では単精度に比べ多くの計算時間を要求する。これらは計算速度と演算精度とのトレードオフになっている。本論文では、まず計算格子を導入することにより浮動小数点演算の桁落ち誤差を防ぐ。次に部分的倍精度演算を適用することにより、丸め誤差を防ぐ。最後に、提案手法の検証にダム崩壊問題を適用し、各条件の経過時間と粒子の先端位置の MSE の比較を行った。その結果、要求される計算速度と演算精度に応じてトレードオフが可能であることを示した。

## 参 考 文 献

- 1) Monaghan, J.J., "Smoothed Particle Hydrodynamics," Annu. Rev. Astron. Astrophys. , Vol. 30. (1992), pp. 543-74
- 2) S. Koshizuka, and Y. Oka, "Moving-Particle Semi-implicit Method for Fragmentation of Incompressible Fluid," Nucl. Sci. Eng., 123, pp. 421-434, 1996.
- 3) Joseph Ha, "Computational Resources Required for the SPH Simulation of Water

- Flooding a Stairway," Technical Report CMIS 05/27, May 9, 2005.
- 4) GPGPU.org, "GPGPU.org :: General-Purpose computation on Graphics Processing Units," <http://gpgpu.org/>
  - 5) Takahiro Harada, Seiichi Koshizuka, Yoichiro Kawaguchi, "Smoothed Particle Hydrodynamics on GPUs," Proc. of Computer Graphics International, pp. 63-70, 2007).
  - 6) L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," Astronomical Journal, vol. 82, pp. 1013-1024, Dec. 1977.
  - 7) 十河清, "非線形物理学 -カオス・ソリトン・パターン-," 裳華房, pp. 18, Oct. 2010.
  - 8) G. R. Liu and M. B. Liu, "Smoothed Particle Hydrodynamics," World Scientific Publishing Co. Pte. Ltd, 2003, pp 137.
  - 9) Ibid, pp 143-146
  - 10) NVIDIA, "CUDA Programming Guide for CUDA Toolkit 3.2,"  
<http://developer.nvidia.com/object/gpucomputing.html>, Nov. 9, 2010.