

SPH における壁境界計算手法の改良

原田 隆 宏[†] 越 塚 誠 一[†]

本論文では流体の運動をシミュレーションするための粒子法の 1 つである SPH における壁境界計算モデルの改良手法を提案する。一般的に粒子法では壁境界を粒子として表して計算するが、本手法では壁境界を粒子で表現せず、壁境界にポリゴンモデルを用いる。本手法では壁粒子を用いないため、粒子解像度を高くすることがなく壁境界を正確に表すことができる。粒子法においては壁粒子の占める粒子数が総粒子数に占める割合が大きいが、本手法では壁粒子を生成する必要がないため、壁粒子を用いて同数の流体粒子を用いた計算をした場合と比較すると大幅に粒子数を削減することができる。また本論文では計算粒子群から生成した壁に触れている流体の表面を滑らかに描画する手法も提案する。

An Improvement of Wall Boundary Calculation Model in Smoothed Particle Hydrodynamics

TAKAHIRO HARADA[†] and SEIICHI KOSHIZUKA[†]

In this paper, we propose an improved calculation model of wall boundary computation in Smoothed Particle Hydrodynamics, a particle method for computer simulation of fluid motion. Although particle methods calculate a wall boundary with particle representation generally, our method uses polygons themselves as a wall boundary. Since our method does not need wall particles, the wall boundary can be represented precisely without increase of particle resolution. The proportion of the wall particle number in the total particle number is high in particle methods which use wall particles as a wall boundary. Because our method does not need wall particles, it can reduce the total particle number drastically. We also propose a visualization method which can make smooth fluid surface contacting with a wall boundary.

1. 序 論

流体は我々の身近に多く存在している。流体の挙動は複雑であるため、コンピュータグラフィックスで流体のアニメーションを作成するとき、流体の支配方程式を解き流体挙動を計算することが行われている。流体の解法は大きく分類してオイラー解法とラグランジュ解法が存在する。オイラー解法は空間に固定された計算格子を用いる。一方ラグランジュ解法は格子を用いず、互いに接続されていない粒子を用いて計算を行う。粒子を用いて計算する手法は粒子法と呼ばれている。この手法は粒子自体が流体を示すので自由表面の扱いが容易であり、また粒子自体が移流するため、移流項の計算における数値拡散が起らないという特徴がある。流体を計算する粒子法は Moving Particle Semi-implicit (MPS) Method¹⁾ と Smoothed Particle Hydrodynamics (SPH)⁵⁾ がある。MPS は

非圧縮流体を解くことが可能であり、工学分野において広く用いられている。また SPH は圧縮流体を解き、MPS に比べ計算コストが低く、コンピュータグラフィックスにおいて広く用いられている。圧縮性を弱くすることで、SPH ではほぼ非圧縮な流体を取り扱うこともできる。粒子法では流体は粒子として表され計算され、また流体以外の壁境界も一般的に粒子として表される。しかし壁粒子を用いると境界の形状を正確に表すことができないなどの問題がある。

本論文では SPH における粒子を用いない壁境界計算モデルを提案する。本手法は壁境界に粒子ではなくポリゴンを用いる。壁粒子を用いないため、壁粒子を生成する必要がなく、壁境界を正確に表すことができる。粒子法においては壁粒子の占める粒子数が流体の粒子数を含めた全粒子数に占める割合が大きいが、本手法を用いるとその壁粒子を生成する必要がないため、壁粒子を用いる場合と比べると同じ計算を行うときの粒子数を大幅に削減することが可能である。そのため、同数の流体粒子を用いた計算をしたときと比較すると、本手法を用いた場合は計算時間も削減できる。

[†] 東京大学

The University of Tokyo

また提案する壁境界計算モデルでは壁境界との計算を簡単な数回の補間計算のみによって求めることができるので、全体の計算コストに占める壁境界の計算時間は小さい。

粒子法では計算粒子から流体の表面を抽出してレンダリングするが、その際なめらかな表面を得ることが難しい。自由表面だけでなく、壁に触れている流体表面もなめらかにすることが難しい。本論文では壁に触れている流体表面をなめらかにする手法も提案する。

2. 関連研究

コンピュータグラフィックスにおいて、計算格子を用いるオイラー解法を用いた自由表面流れの解法では、レベルセット法が界面追跡手法として多く用いられている⁵⁾。しかしオイラー解法を用いると移流計算の際に数値拡散が起きるため、質量損失が起こってしまう。自由表面流れでの質量損失は流体が消えたように見えるため、映像制作などでは大きな問題となる。この問題点を克服するためレベルセット法をラグランジュ粒子と組み合わせたパーティクルレベルセット法が開発された⁴⁾。それ以外にも流体と剛体の相互作用²⁾、粘弾性流体⁶⁾、薄膜との相互作用⁷⁾、混相流¹⁴⁾、2次元計算と3次元計算のカップリング⁹⁾なども研究されている。構造格子を用いると壁境界の計算精度が下がるため Octree を用いたり¹³⁾、非構造格子を用いた研究も行われている¹⁰⁾。

またラグランジュ解法である粒子法に関して MPS¹¹⁾ は計算力学の分野において多く研究されており、剛体との相互作用²⁰⁾、混相流⁸⁾などの研究も行われている。コンピュータグラフィックスの分野においては Premoze らが MPS を用いた計算を行った¹⁹⁾。SPH は天文学の分野で開発された手法であったが、流体計算にも用いられており、弾性体とのカップリング¹⁷⁾や混相流¹⁸⁾の計算も行われている。粒子法は粗い計算解像度での計算でも質量が保存されるのでリアルタイムの計算に向いている¹⁶⁾。

粒子法では一般的に壁境界も粒子として表されて計算される。Müller らはラグランジュメッシュで弾性体を表し、メッシュと流体粒子を相互作用させる手法を開発した¹⁷⁾。この手法はメッシュ表面に粒子を発生させて粒子間相互作用を計算し、メッシュに運動量を伝達した。この手法では毎タイムステップごとにポリゴン上に仮想粒子を発生させなければならず、またポリゴンの大きさによって生成される粒子の密度が変化してしまい、壁付近において粒子数密度一定の条件を必ずしも満たすとは限らないという問題点がある。

3. Smoothed Particle Hydrodynamics

3.1 支配方程式

非圧縮流体の支配方程式は質量保存を表す連続の式と、運動量保存の式である。

$$\frac{D\rho}{Dt} = 0 \quad (1)$$

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

ρ 、 \mathbf{U} 、 P 、 ν はそれぞれ流体の密度、速度、圧力、動粘性係数であり、 \mathbf{g} は重力加速度である。

3.2 離散化

粒子法の1つである SPH は流体を粒子群として近似する。位置 \mathbf{x} での物理量 $\phi(\mathbf{x})$ は近傍粒子 j の持つ物理量 ϕ_j の重み付け和として定義される。

$$\phi(\mathbf{x}) = \sum_j m_j \frac{\phi_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j) \quad (3)$$

m_j 、 ρ_j 、 \mathbf{x}_j はそれぞれ粒子 j の質量、密度、位置であり、 W は重み関数である。流体の密度は式 (3) を用いて

$$\begin{aligned} \rho(\mathbf{x}) &= \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j) \\ &= \sum_j m_j W(\mathbf{x} - \mathbf{x}_j) \end{aligned} \quad (4)$$

と定義される。流体の圧力は状態方程式より以下のように計算される。

$$p - p_0 = k(\rho - \rho_0) \quad (5)$$

p_0 、 ρ_0 はそれぞれ基準となる圧力と密度である。

流体の運動量保存式を粒子を用いてモデル化するには、圧力項の勾配モデルと粘性項のラブラシアンモデルが必要になる。粒子 i に働く圧力項 \mathbf{F}_i^{press} と粘性項 \mathbf{F}_i^{vis} はそれぞれ以下のように計算される。

$$\mathbf{F}_i^{press} = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{press}(\mathbf{r}_{ij}) \quad (6)$$

$$\mathbf{F}_i^{vis} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla W_{vis}(\mathbf{r}_{ij}) \quad (7)$$

ここで $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ であり、 \mathbf{r}_i 、 \mathbf{r}_j はそれぞれ粒子 i 、 j の位置ベクトルである。

本研究では重み関数 W は Müller らの用いた重み関数 W を用いた¹⁶⁾。圧力項と粘性項、それ以外において用いる重み関数はそれぞれ

$$\nabla W_{press}(\mathbf{r}) = \frac{45}{\pi d^6} (d - |\mathbf{r}|)^3 \frac{\mathbf{r}}{|\mathbf{r}|} \quad (8)$$

$$\nabla W_{vis}(\mathbf{r}) = \frac{45}{\pi d^6} (d - |\mathbf{r}|) \quad (9)$$

$$W(\mathbf{r}) = \frac{315}{64\pi d^9} (d^2 - |\mathbf{r}|^2)^3 \quad (10)$$

を用い、これらはいずれも影響半径 d 以外では 0 にする。

4. 壁重み関数

本章では壁粒子を用いない壁境界計算モデルについて述べていく。SPH で支配方程式を解くときに壁粒子は流体の密度、粘性項、圧力項に関係する。そこで壁粒子を用いない壁境界計算モデルではこれらを新しくモデル化しなければならない。以下これらのモデル化を順に示していく。

4.1 密度

流体の密度は式 (4) のように表されるが、粒子 i の近傍に壁粒子が存在すると仮定して、流体粒子の寄与と壁粒子の寄与を分離する。

$$\begin{aligned} \rho_i(\mathbf{r}_i) &= \sum_j m_j W(\mathbf{r}_{ij}) \\ &= \sum_{j \in fluid} m_j W(\mathbf{r}_{ij}) + \sum_{j \in wall} m_j W(\mathbf{r}_{ij}) \end{aligned} \quad (11)$$

右辺第 1 項が流体粒子からの寄与、第 2 項が壁粒子からの寄与である。壁粒子の質量 m_j はすべて一定であり、重み関数 W では距離の絶対値のみが用いられるので、図 1 のように壁粒子が粒子 i から壁におろした垂線に垂直に均一に並んでいると仮定すると、粒子 i から壁までの距離 $|\mathbf{r}_{iw}|$ が決まれば、粒子 i の近傍に存在する壁粒子の配置は一意に決まる。つまり壁粒子の寄与は以下のように粒子 i から壁までの距離 $|\mathbf{r}_{iw}|$ の関数とすることができる。

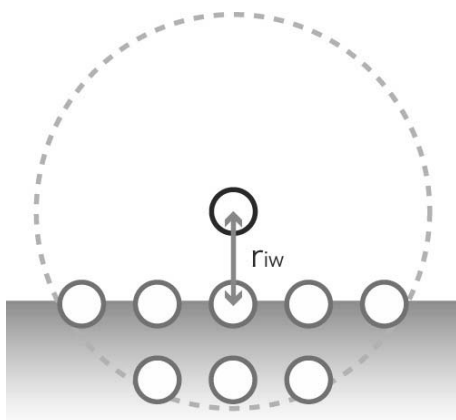


図 1 影響半径内における壁粒子の分布

Fig. 1 Distribution of wall particles in an effective radius.

$$\rho_i(\mathbf{r}_i) = \sum_{j \in fluid} m_j W(\mathbf{r}_{ij}) + Z_{wall}^{rho}(|\mathbf{r}_{iw}|) \quad (12)$$

この $Z_{wall}^{rho}(|\mathbf{r}_{iw}|)$ を密度の壁重み関数と呼ぶ。

4.2 粘性項

密度と同様に粘性項も流体粒子による寄与と壁粒子による寄与に分解する。粒子 i に働く粘性力 \mathbf{F}_i^{vis} は、流体粒子から受ける粘性力 $\mathbf{F}_{i,fluid}^{vis}$ と壁から受ける粘性力 $\mathbf{F}_{i,wall}^{vis}$ に分けることができる。

$$\begin{aligned} \mathbf{F}_i^{vis} &= \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla W_{vis}(\mathbf{r}_{ij}) \\ &= \mathbf{F}_{i,fluid}^{vis} + \mathbf{F}_{i,wall}^{vis}. \end{aligned} \quad (13)$$

ここで壁粒子の速度を 0 とすると壁から受ける粘性力 $\mathbf{F}_{i,wall}^{vis}$ は

$$\begin{aligned} \mathbf{F}_{i,wall}^{vis} &= \mu \sum_{j \in wall} m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla W_{vis}(\mathbf{r}_{ij}) \\ &= -\mu \sum_{j \in wall} m_j \frac{\mathbf{v}_i}{\rho_j} \nabla W_{vis}(\mathbf{r}_{ij}) \\ &= -\mu \mathbf{v}_i \sum_{j \in wall} m_j \frac{1}{\rho_j} \nabla W_{vis}(\mathbf{r}_{ij}) \end{aligned} \quad (14)$$

と計算される。

我々が取り扱っている流体は非圧縮流体である。SPH では連続の式を解いていないため、完全な非圧縮を再現することができないが、疑似非圧縮を再現することができる。すなわち流体内部の粒子の密度はほぼ一定である。壁粒子を用いた場合は壁粒子の密度もほぼ一定になる。また重み関数 ∇W_{vis} では距離の絶対値のみ用いられる。そこで式 (14) において、壁粒子の密度 ρ_j が一定だと仮定すると、すべての壁粒子の $m_j \frac{1}{\rho_j}$ が定数になる。すなわち粘性項における壁粒子の寄与は以下のように粒子 i から壁までの距離 $|\mathbf{r}_{iw}|$ の関数とすることができる。

$$\mathbf{F}_i^{vis} = -\mu \mathbf{v}_i Z_{wall}^{vis}(|\mathbf{r}_{iw}|) \quad (15)$$

この $Z_{wall}^{vis}(|\mathbf{r}_{iw}|)$ を粘性項の壁重み関数と呼ぶ。

4.3 圧力項

非圧縮流体を粒子法で解くときには、圧力項は粒子数密度を一定にするように働く力となる。そのため壁が流体粒子に及ぼす圧力による力 \mathbf{F}_i^{press} は壁と流体粒子の距離を初期粒子間距離 d に保とうとする力にする。粒子 i が壁から $|\mathbf{r}_{iw}|$ の距離に存在するとき、圧力による力 \mathbf{F}_i^{press} は粒子を $d - |\mathbf{r}_{iw}|$ の距離だけ $\mathbf{n}(\mathbf{r}_i)$ の方向に押し戻す。つまり圧力による力 \mathbf{F}_i^{press} は

$$\begin{aligned} \mathbf{F}_i^{press} &= m_i \frac{\Delta \mathbf{x}_i}{dt^2} \\ &= m_i \frac{(d - |\mathbf{r}_{iw}|) \mathbf{n}(\mathbf{r}_i)}{dt^2} \end{aligned} \quad (16)$$

とモデル化することができる．ここで $\mathbf{n}(\mathbf{r}_i)$ は粒子 i の位置 \mathbf{r}_i における距離関数の勾配ベクトルであり，このベクトルは粒子 i に最も近い壁の法線ベクトルである．

4.4 壁重み関数の計算

式 (12) と式 (15) での壁重み関数の計算は粒子 i から壁までの距離 $|\mathbf{r}_{iw}|$ のみに依存する関数であるため，この距離が分かれば計算することができる．壁重み関数は粒子から壁までの距離にのみ依存する関数なので前計算しておき，流体計算を行うときには前計算しておいたデータを参照するだけでよい．壁重み関数は壁内部に存在する粒子位置で重み関数を積分しなければならないため，幾何学的には求めることはできない．そこで図 1 に示すように壁内部に粒子を配置し，それらによる重み関数を数値積分することで求める．前計算では粒子の影響半径内の数点で壁重み関数を計算しておく．そしてそれ以外の点においては計算された値を線形補間して近似値を求めた．

壁重み関数を前計算しておくことで，粒子から壁までの距離に対応する値を参照するだけで壁の寄与を計算できるようになったが，粒子から壁までの距離の計算は壁境界を構成するすべてのポリゴンまでの距離計算を行い，それらのうちの最小値を求める必要があるため，ポリゴン数が増加すればするほど計算コストが高くなる．そこで距離関数を用いると，この計算も前計算で構築したデータを参照するだけで済むようになる．距離関数とは計算領域に境界が定義されたときに，その境界までの最短距離を表す関数である．つまり粒子 i における壁までの距離 $|\mathbf{r}_{iw}|$ は距離関数である．流体計算において壁境界が不動ならば壁の距離関数は変化しないため，壁の距離関数もあらかじめ計算しておくことができる．これによってそれぞれの粒子の計算の際に，壁までの距離を計算する必要はなく，計算された距離関数を参照するだけで距離 $|\mathbf{r}_{iw}|$ を求めることができる．距離関数の計算手法については次章で述べる．

圧力項の計算では粒子 i から壁までの距離以外に，粒子 i に最も近い壁の法線ベクトルが必要となる．ある点において距離関数の勾配を取り正規化したベクトルは，その点から最近点の境界への方向を示すベクトルである．そしてこのベクトルはその最近点での境界の法線ベクトルとなっている．壁境界が変化しないた

め，計算領域内のある位置から最も近い壁の法線ベクトルも不変である．そのため，この法線ベクトルもあらかじめ計算しておき，流体計算時には値を参照するだけで求めることができる．

5. 距離関数の計算

距離関数の計算は計算領域内の点から境界までの最近点を計算し，そこまでの距離を求める．境界上の最近点は 3 次元の問題だと面か辺か点のいずれかになる¹⁾．境界を構成するポリゴン数が多い場合，距離関数の計算コストは特に高くなる，本研究では距離関数の近似計算を行った．

5.1 手 法

前述のように計算領域内の点での境界上の最近点は境界の面か辺か点のいずれかになるが，境界を構成するポリゴンが十分細かく境界から距離が近い点では最近点は面上にあることが多い．本研究に必要な距離関数は境界から粒子の影響半径内の領域のみであるため，最近点はすべて面上に存在すると近似して距離関数を求めた．

まずある 1 ポリゴンによる距離関数を考える．このポリゴンを最近点とする計算領域内の点はこのポリゴンの近辺にしか存在しないため，ポリゴンを内包する立方体を生成する．そしてこの立方体内部でポリゴンまでの距離を計算する．隣り合ったポリゴンでも同様の処理を行うと距離計算を行う立方体が重複する部分が出てくる．距離関数は境界上の最近点までの距離なので，重複した部分においては距離の比較を行い，小さい値を保持する．この処理をすべてのポリゴンに対して行うことで距離関数を計算できる．

この距離関数計算手法は Graphics Processing Unit (GPU) を用いて高速化することが可能である．まず 3 次元格子にあたるテクスチャを用意し，その上にポリゴンを内包する立方体をレンダリングしていく．まずピクセルシェーダで距離関数を計算し，この値を出力する．そして最小値を求める処理は深度テストを用いて行うことができる．このように求めた距離関数を Z 軸で 8 枚にスライスしたものを図 2 に示す．距離関数の値によってコンター表示している．

計算された距離関数は 3 次元格子点上の値として格納した．本研究では 128^3 の 3 次元格子を用いた．そして流体粒子の座標における距離関数の評価は，その粒子を囲む 8 格子点での距離関数から線形補間することで求めた．本手法を用いた距離関数の計算時間を表 1 に示す．約 20 万ポリゴンで構成されているドラゴンポリゴンモデルでも本手法では 1 秒未満で距離関

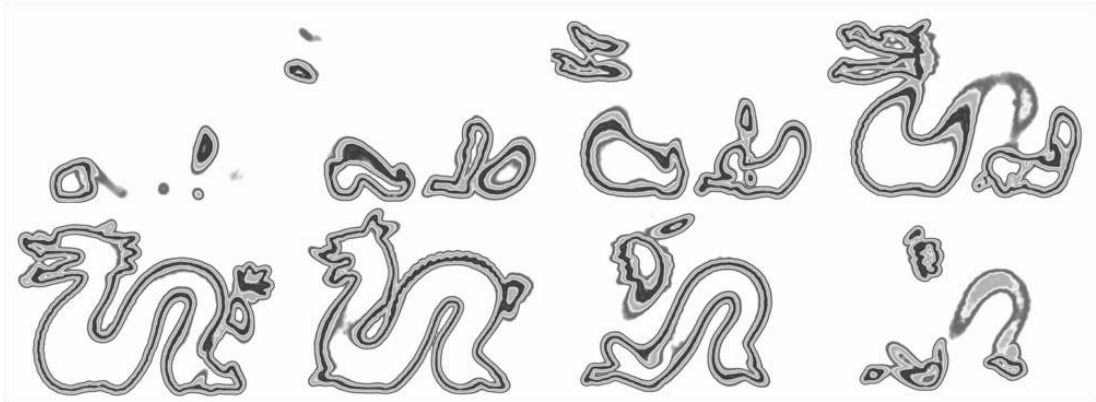


図2 ドラゴンポリゴンモデルから計算された距離関数

Fig. 2 Distance function calculated with dragon polygon model as a boundary.

表1 3種類のポリゴンモデルのポリゴン数と距離関数の計算時間
Table 1 Number of polygon models and computation time of distance function (in milliseconds).

Model	Number of polygon	Time
Gargoyle	500,000	2,172
Dragon	202,520	906
Buddha	32,328	297

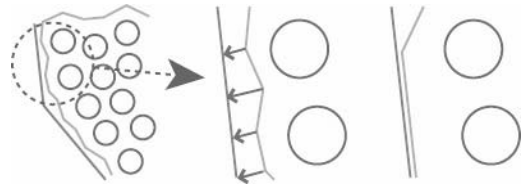


図3 表面フィッティングの手順

Fig. 3 Procedures of surface fitting.

数を計算できている。

6. レンダリング

6.1 概要

粒子法では流体を粒子群と離散化し、流体の挙動を計算するが、粒子群を球としてレンダリングしたのでは流体には見えない。流体に見えるようにレンダリングするには、これらの粒子群から流体の表面を抽出する必要がある。具体的には流体粒子それぞれに濃度分布を与え、すべての粒子の濃度分布の足し合わせを行うことで陰関数曲面を構築することができる。しかしこのように構築された流体表面はスムーズにはならない。壁境界に接している粒子から抽出される表面は壁の形状に沿うはずであるが、計算された陰関数曲面は必ずしも壁の形状に沿っているわけではない。本章では壁に接している粒子から抽出される表面を壁の形状に沿わせる手法を提案する。

6.2 表面抽出

まずは粒子から流体の表面を抽出する。座標 x_i に存在する流体粒子に関数 f という濃度球 (プロブ) を割り当てる。座標 x での濃度分布は以下のように与える。

$$\phi_i(x) = f(|x - x_i|) \tag{17}$$

そしてすべての計算粒子の濃度分布を足し合わせる

ことで陰関数曲面を計算する。

$$\Phi(x) = \sum_j \phi_j(x) \tag{18}$$

計算された $\Phi(x) = 0$ が流体の表面を示すとし、Marching Cubes¹²⁾ を用いて表面ポリゴンを生成する。

6.3 サーフェイスフィッティング

前述の手法で抽出された表面は壁に沿っているべきところでも壁に沿ってはいない。そこで抽出されたポリゴンを構成する頂点が壁からある一定距離以内に存在するならば、その頂点は壁に沿っているものだと考え、頂点を最も近い壁に接するように移動させる (図3)。

ここで頂点が壁からある一定距離以内に存在する場合は流体計算で用いた壁の距離関数 d を用いることで判定することができる。ここで一定距離を ϵ と定めると位置 x に存在する頂点が以下の条件を満たす場合に頂点を移動させる。

$$d(x) \leq \epsilon \tag{19}$$

この頂点から最も近い壁へのベクトルも流体計算で用いた壁の法線ベクトルを参照すればよい。位置 x に存在した頂点は以下の式で壁に接する位置 x' に移動させる。

$$x' = x - d(x)n(x) \tag{20}$$

そしてこの頂点をレンダリングするときは位置 x で

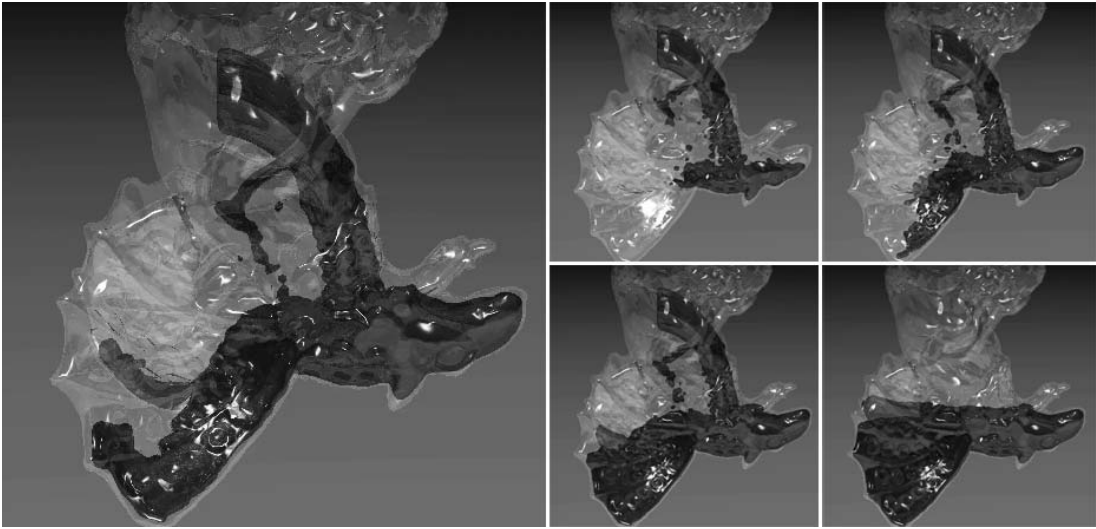


図 4 ガーゴイルポリゴンモデルを壁境界として用いた自由表面流れの計算結果

Fig. 4 A computation result of free surface flow. A gargyle polygon model was used as a wall boundary.

の勾配ベクトルの逆方向 $-n(x)$ を頂点の法線として用いる。

7. 結果と考察

本手法を Pentium 4 3.6 GHz の CPU, 3.0 GB の RAM を搭載した PC 上で C++ を用いて実装した。そして様々なポリゴンモデルを壁境界として用いて計算した。本論文ではいくつかの計算結果を示す。ここで示す計算結果は最大粒子数はいずれも 20,000 である。計算はすべて固定壁境界を用いており、この壁の距離関数と勾配ベクトルは前計算によって求めた。そのため提案している壁境界計算モデルは計算時には各粒子位置における距離関数と勾配ベクトルを計算された値から補間して求めた。また壁重み関数も前もって計算によって求めており、粒子位置での値も補間して求めるため、壁境界の計算コストは低い。粒子法では壁境界を正確に表すためには粒子間距離を小さくする必要があるが、そうすると全粒子数が膨大になってしまう。オイラー解法では壁境界での計算解像度を上げるためにアダプティブグリッドが用いられているが、粒子法ではまだそのような手法は開発されていない。しかし本手法を用いることで壁境界は粒子の計算解像度によらず正確に表すことができる。

図 4 はガーゴイルモデル内部に流体を流入させた結果を示す。計算結果から表面を抽出しサーフェスフィッティングを行って構築したメッシュを透明にレンダリングした結果である。このモデルは 2 枚の薄い羽を持っている。このような細かい部分も滑らかな

壁境界として計算できている。図 5 と図 6 はドラゴンモデル内部に流体を流入させた結果である。これらでも抽出された表面にサーフェスフィッティングを行った。サーフェスフィッティングの結果がよく分かるように流体は不透明な材質を適用した。サーフェスフィッティングにより、流体の表面が壁境界の形に沿っていることがよく分かる。図 7 にブダモデル内部に流体を流入させた結果を示す。このモデルは粒子径よりも小さな構造を持っている。そのような部分には流体粒子は流入することはできない。これらのモデルを壁境界として用いたときの計算時間を表 2 に示す。

壁境界形状が複雑になればなるほど壁境界の粒子を生成することも難しくなってくる。壁粒子を用いる手法では壁に接している流体の粒子数密度を正確に計算するため、その粒子の影響半径内には壁粒子を配置しなければならない。これにより大抵の場合壁粒子を数層とらなければならない。

ここで単純な計算体系を用いて計算時間を測定し比較した。20 × 50 × 20 粒子で構成されている水柱と内部に 60 × 50 × 20 粒子入れることができる水槽を用いた。この水柱を構成する粒子数は 20,000 であり、壁に 3 層粒子をとると水槽を構成する壁粒子の数は 36,096 である。すると本手法で計算に用いる粒子数は 20,000 であるのに比べ、壁粒子を用いる場合は 56,096 粒子を用いなければならない。全粒子数の 64% を壁粒子が占めている。20,000 粒子を用いた計算時間は 1 タイムステップあたり 133.4 ミリ秒であったのに比べ、56,096

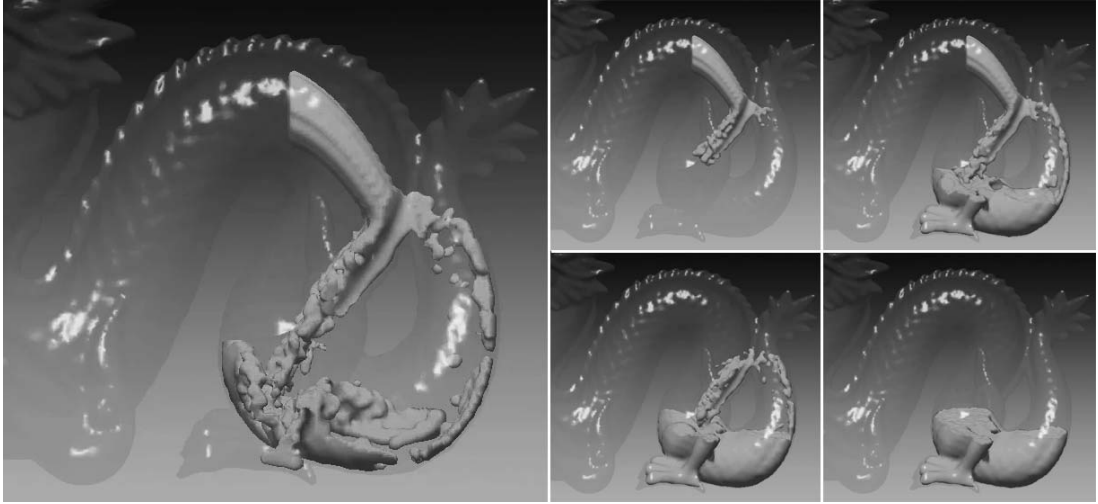


図 5 ドラゴンポリゴンモデルを壁境界として用いた自由表面流れの計算結果

Fig. 5 A computation result of free surface flow. A dragon polygon model was used as a wall boundary.

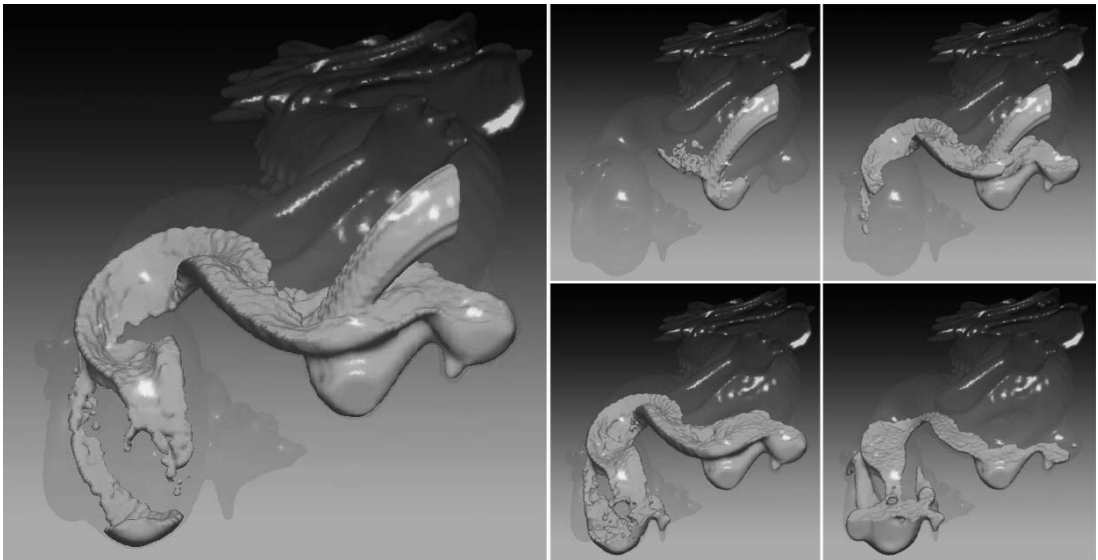


図 6 ドラゴンポリゴンモデルを壁境界として用いた自由表面流れの計算結果

Fig. 6 A computation result of free surface flow. A dragon polygon model was used as a wall boundary.

粒子を用いた場合は 356.4 ミリ秒であった．このように総粒子数の増加は計算時間の増加に直結するため，本手法を用いることで計算時間の削減が可能であった．図 4，図 6，図 7 のモデルでそれぞれ壁粒子を生成したときの総粒子数における壁粒子の割合を表 3 に示す．ここでは入力としたポリゴンモデルが複雑であり，壁粒子を複数層生成するのが困難であったため，1 層だけ生成した．ポリゴンモデルの変換には Dong らの手法を用いた³⁾．数層の壁粒子を生成する場合には 1

層だけの場合より粒子数が多くなる．1 層だけの壁粒子を生成した場合でも壁粒子数は総粒子数の約半分を占めていた．

本手法では壁の平均曲率が 0 であると仮定しているが，本研究の結果から分かるように大きい平均曲率を持つ壁境界でも十分良い結果が得られている．これは大部分の壁の平均曲率に比べ平均粒子間距離が小さく，粒子の影響半径内の壁の平均曲率がほぼ 0 と近似できているからであると考えられる．そのため平均粒

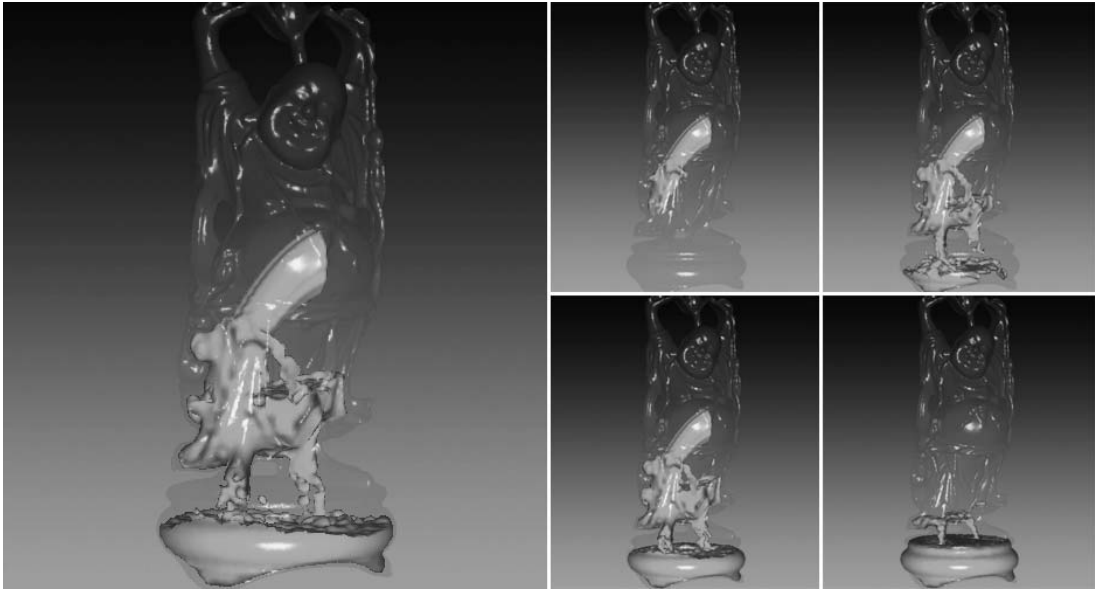


図 7 ブッダポリゴンモデルを壁境界として用いた自由表面流れの計算結果

Fig. 7 A computation result of free surface flow. A Buddha polygon model was used as a wall boundary.

表 2 総粒子数と 1 タイムステップにかかった計算時間

Table 2 Total particle number of fluid and computation time for one timestep (in milliseconds).

Model	Fluid	Time
Gargoyle	20,000	309.4
Dragon	20,000	281.2
Buddha	20,000	296.8

表 3 1 層の壁粒子数と総粒子数に占める壁粒子の割合

Table 3 Wall particle number (1 layer) and the ratio of wall particle number in all particle number.

Model	Wall	Total	Ratio
Gargoyle	26,688	46,688	0.572
Dragon	18,582	38,582	0.482
Buddha	12,084	32,084	0.377

子間距離が大きくなってしまふとこの近似が難しくなると考えられる。

8. 結 論

本論文では SPH における壁粒子を用いずにポリゴンモデルを用いる壁境界計算手法を開発した。本手法では壁境界条件を粒子ではなく壁重み関数と距離関数で表す。壁粒子を用いる場合と総粒子数を比較し、本手法を用いることで総粒子数の削減ができることを示した。また粒子から抽出した物体に接している流体表面を滑らかに描画するサーフェイスフィッティングも開発した。本研究での壁境界計算モデルは壁の平均曲

率を 0 と仮定しているが、今後の課題として平均曲率を考慮したより正確な壁重み関数の計算手法の開発があげられる。

本研究では壁境界が固定された問題のみ取り扱ったが、これを拡張し剛体や弾性体との相互作用、薄膜構造物との相互作用へ応用していく。さらに本手法の壁境界計算の概念は非圧縮流体を解くことのできる粒子法である MPS にも適用することができる。今後は MPS への適用も行っていく。

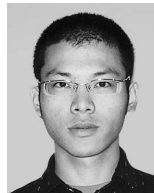
参 考 文 献

- 1) Baerentzen, J.A. and Aanaes, H.: Generating signed distance fields from triangle meshes, IMM Technical Report, 21 (2002).
- 2) Carlson, M., Mucha, P.J. and Turk, G.: Rigid fluid: Animating the interplay between rigid bodies and fluid, *ACM Trans. Graphics*, Vol.23, No.3, pp.377–384 (2004).
- 3) Dong, Z., Chen, W., Bao, H., Zhang, H. and Peng, Q.: Real-time voxelization for complex polygonal models, *Proc. Pacific Graphics*, pp.73–78 (2004).
- 4) Enright, D., Marschner, S. and Fedkiw, R.: Animation and rendering of complex water surfaces, *ACM Trans. Graphics*, Vol.21, No.3, pp.721–728 (2002).
- 5) Foster, N. and Fedkiw, R.: Practical animation of liquids, *Proc. Siggraph*, pp.15–22 (2001).

- 6) Goktekin, T., Bargteil, A.W. and O'Brien, J.F.: A method for animating viscoelastic fluids, *ACM Trans. Graphics*, Vol.23, No.3, pp.464–467 (2004).
- 7) Guendelman, E., Selle, A., Losasso, F. and Fedkiw, R.: Coupling water and smoke to thin deformable and rigid shells, *ACM Trans. Graphics*, Vol.24, No.3, pp.910–914 (2005).
- 8) Harada, T., Suzuki, Y., Koshizuka, S., Arakawa, T. and Shoji, S.: Simulation of droplet generation in micro flow using mps method, *JSME International Journal Series B*, Vol.49, No.3, pp.731–736 (2006).
- 9) Irving, G., Guendelman, E., Losasso, F. and Fedkiw, R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques, *ACM Trans. Graphics*, Vol.25, No.3, pp.812–819 (2006).
- 10) Klinger, B.M., Feldman, B.E., Chentanez, N. and O'Brien, J.F.: Fluid animation with dynamic meshes, *ACM Trans. Graphics*, Vol.25, No.3, pp.820–825 (2006).
- 11) Koshizuka, S. and Oka, Y.: Moving-particle semi-implicit method for fragmentation of incompressible flow, *Nucl. Sci. Eng.*, Vol.123, pp.421–434 (1996).
- 12) Lorenson, W.E. and Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm, *Proc. 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp.163–169 (1987).
- 13) Losasso, F., Gibou, F. and Fedkiw, R.: Simulating water and smoke with an octree data structure, *ACM Trans. Graphics*, Vol.23, No.3, pp.457–462 (2004).
- 14) Losasso, F., Shinar, T., Selle, A. and Fedkiw, R.: Multiple interacting liquids, *ACM Trans. Graphics*, Vol.25, No.3, pp.812–819 (2006).
- 15) Monaghan, J.J.: Smoothed particle hydrodynamics, *Annu. Rev. Astrophys.*, Vol.30, pp.543–574 (1992).
- 16) Müller, M., Charypar, D. and Gross, M.: Particle-based fluid simulation for interactive applications, *Proc. Siggraph Symposium on Computer Animation*, pp.154–159 (2003).
- 17) Muller, M., Schirm, S., Teschner, M., Heidelberger, B. and Gross, M.: Interaction of fluids with deformable solids, *Journal of Computer Animation and Virtual Worlds*, Vol.15, No.3, pp.159–171 (2004).
- 18) Müller, M., Solenthaler, B., Keiser, R. and Gross, M.: Particle-based fluid-fluid interaction, *Proc. Siggraph Symposium on Computer Animation*, pp.237–244 (2005).
- 19) Premoze, S., Tasdizen, T., Bigler, J., Lefohn, A. and Whitaker, R.T.: Particle-based simulation of fluids, *Computer Graphics Forum*, Vol.22, No.3, pp.401–410 (2003).
- 20) Shibata, K., Koshizuka, S., Oka, Y. and Tanizawa, K.: A three-dimensional numerical analysis code for shipping water on deck using a particle method, *Proc. ASME Heat Transfer/Fluid Engineering*, HT-FED04-56477 (2004).

(平成 18 年 8 月 30 日受付)

(平成 19 年 1 月 9 日採録)



原田 隆宏 (正会員)

昭和 56 年生 . 平成 18 年東京大学大学院工学系研究科システム量子工学専攻修士課程修了 . 同年東京大学大学院情報学環学際情報学府助手 . 平成 19 年 4 月助教 . 計算力学とコンピュータグラフィックスの研究に従事 . 日本機械学会 , ACM 各会員 .



越塚 誠一

昭和 37 年生 . 昭和 61 年東京大学大学院工学系研究科原子力工学専攻修士課程修了 . 同年東京大学工学部助手 . 平成 3 年博士 (工学) . 同年講師 . 平成 5 年助教授 . 平成 16 年教授 . 連続体の力学シミュレーションの研究に従事 . 特に粒子法の開発を行う . 平成 17 年に丸善より『粒子法』を出版 . 平成 18 年に日本学術振興会賞を受賞 . 日本原子力学会 , 日本機械学会 , 日本流体力学会 , 日本計算工学会 , 日本応用数理学会各会員 .