

再構築ポリゴン壁を用いた SPHの壁境界計算手法における探索範囲について

笠 晃一^{1,a)}

概要: 3次元コンピュータグラフィックスにおける流体の表現では、粒子法の一つであるSPHによる物理シミュレーションを使用することが多い。SPHでは壁境界も粒子で表現するが、壁粒子は離散的に配置されるため流体粒子にいくつかの不自然な挙動が見られる。そこで、本研究では壁境界をポリゴンで表現し、その情報を離散境界に持たせることにした。そして、流体粒子が離散境界に接近したとき、その近傍のみポリゴン壁を再構築するのである。ただし、この手法では流体粒子が離散境界を探索するときの範囲、すなわち探索半径が自明ではない。本研究では離散境界の配置を6種類に分類することにより探索半径を求めているが、ここではその導出過程を中心に報告する。なお、本研究の手法を用いることにより、流体粒子の不自然な挙動を解消することができた。

1. はじめに

流体力学における物理シミュレーションの一種に粒子法というものがある。これは粒子を使用して流体を表現するものであるが質量保存が容易に実現できるため、3次元コンピュータグラフィックスにおける流体の表現で頻繁に利用されている。現在までに開発されている粒子法として、SPH (Smoothed Particle Hydrodynamics) 法 [1] と MPS (Moving Particle Semi-implicit) 法 [2] がある。ただし、MPS に比べ SPH の方が計算コストが低いので、コンピュータグラフィックスで使用されるのはほとんど SPH であり、本研究でも SPH を用いている。粒子法では固体との境界を形成する壁も粒子で表現する。しかし壁粒子は離散的に配置されるため離散化誤差が発生して、斜面を流体が滑らかに流れないという現象が見られるし、壁から受ける力を正確に計算できないといった不具合も生じる。

この問題を解決する方法の1つとして、ポリゴンメッシュの使用が考えられる。これだと流体粒子からポリゴンメッシュまでの距離を用いて壁から受ける力を計算することになるので、壁粒子を使用したときの問題を避けることが可能である。しかし、この手法は距離をいかにして計算するかという問題を新たに発生させることになる。流体粒子からポリゴンメッシュまでの距離を計算する方法の一つに距離関数の利用があり、距離関数を高精度に求める手法

も開発されているが計算コストが高く、ポリゴンメッシュが移動した場合の再計算の負荷が大きくなる。我々はポリゴンメッシュの代わりに離散境界を用い、これにポリゴンメッシュの情報を持たせるという手法 [3] を新たに開発した。こうすることにより、ポリゴンメッシュの情報を高速に引き出し、流体粒子からポリゴンメッシュまでの距離を精度よく求めることが可能になる。また、この手法は壁境界が移動した場合でも、離散境界が持つ情報を更新するだけでよく、再計算のコストは低い。ただし、離散境界の配置は様々に変化するので、流体粒子が離散境界を探索するときの範囲、すなわち探索半径は自明ではない。本報告ではこの探索半径を求める過程を中心に議論する。

2. 従来の研究

壁境界を表現するのに粒子ではなくポリゴンメッシュを使用することを提案したのは原田ら [4] である。彼らは流体粒子からポリゴンメッシュまでの距離を計算する方法として距離関数を利用しているが、距離関数を高精度に求める手法 [5], [6], [7] は計算コストが高いため、離散境界を用いて距離関数を計算する手法 [8] を提案している。すなわちポリゴンメッシュを元にして離散境界を作成し、各格子点に対し離散境界までの最小距離を計算することにより距離関数を求めている。この手法は、従来の手法に比べ高速であるが、境界面を離散化するため離散化誤差の混入は避けられず、滑らかな斜面を流体が階段状に流れるなどの不具合が生ずる。

¹ 福岡工業大学情報工学部
Faculty of Information Engineering, Fukuoka Institute of
Technology

^{a)} ryu@fit.ac.jp

3. SPH について

流体力学における支配方程式を数値的に解く場合、空間を格子で分割して差分法を使用する方法と、粒子を使用する方法がある。粒子を使用する方法は粒子法と呼ばれ、SPH も粒子法の一つである。

3.1 非圧縮性流体の支配方程式

水などの非圧縮性流体に対する支配方程式は連続の式とナビエ・ストークス方程式であり、これらは次のように記述される。

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{g} \quad (2)$$

ただし、 ρ , \mathbf{v} , p , μ は、それぞれ流体の密度、速度、圧力、粘性率を表しており、 \mathbf{g} は重力加速度である。

3.2 カーネル近似と粒子近似

SPH の定式化は 2 段階で実行される。最初のステップは場の関数のカーネル近似である。場の関数を $A(\mathbf{r})$ で表すことにすると、 $A(\mathbf{r})$ のカーネル近似は次のように記述される。

$$\langle A(\mathbf{r}) \rangle = \int_{\Omega} A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' \quad (3)$$

ここに、 $W(\mathbf{r}, h)$ は平滑化カーネル関数、カーネル関数あるいは重み関数などと呼ばれるものである。 h は一般にカーネル半径と呼ばれ、カーネル関数はこの半径を持つ球の外側で 0 となる性質を持っている。したがって、積分領域 Ω は点 \mathbf{r} を中心とする半径 h の球である。

定式化の次のステップは粒子近似である。すなわち、カーネル近似された場の関数を離散化し、有限個の粒子によって近似する。各粒子の位置を \mathbf{r}_i 、質量を m_i とし、さらに粒子の持つ関数の値を A_i とするとき、空間内の点 \mathbf{r}_i における平滑化された関数の値は次の式で与えられる

$$\langle A_i \rangle = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4)$$

上の式において和は \mathbf{r}_i から h 以内の距離にある粒子に対してのみ取ればよく、いわゆる近傍粒子探索を利用することが可能である。なお、以下では平滑化された関数の括弧 (●) は省いて記述することにする。

3.3 ナビエ・ストークス方程式の SPH による近似

式 (4) を用いれば、 \mathbf{r}_i における密度 ρ_i は次の式で計算される。

$$\rho_i = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (5)$$

さらに、式 (2) の右辺の第 1 項と第 2 項はそれぞれ次のように離散化される。ただし、 i と j が対称になるように若干の補正がなされている。

$$\mathbf{f}_i^p \equiv -\nabla p(\mathbf{r}_i) = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (6)$$

$$\mathbf{f}_i^v \equiv \mu \nabla^2 \mathbf{v}(\mathbf{r}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (7)$$

なお、一般に式 (6) は圧力項と呼ばれ、式 (7) は粘性項と呼ばれている。カーネル関数は物理量や項によって最適なものを使用するのが一般的であり、本研究では Müller ら [9] のものを用いた。

4. 壁重み関数

粒子が壁から受ける力を壁からの距離の関数として求めるために、密度などの計算を流体粒子からの寄与分と壁からの寄与分とに分割する。そして、流体粒子からの寄与分の計算には SPH の粒子近似を用いるが、壁からの寄与分の計算にはカーネル近似のみを用い、粒子近似を使用しないことにする。これにより、壁からの寄与分を解析的に求めることが可能になる。なお、ここでは紙面の都合で圧力項の計算に関する議論を省略している。

4.1 密度の計算

点 \mathbf{r}_i における密度は式 (5) によって計算されるが、ここではこれを流体粒子からの寄与分と壁からの寄与分とに分割し、次のように変形する。

$$\rho_i = \sum_{j \in \text{fluid}} m_j W_\rho(\mathbf{r}_i - \mathbf{r}_j, h) + \rho_0 \int_{\Omega_w} W_\rho(\mathbf{r}_i - \mathbf{r}', h) d\mathbf{r}' \quad (8)$$

右辺の第 2 項が壁からの寄与分であり、 ρ_0 は壁の密度を表している。右辺第 2 項は解析的に計算できて、 α_i の関数となる。ただし、 $\alpha_i \equiv d_i/h$ であり、 d_i は点 \mathbf{r}_i から壁までの距離である。この様子を図 1 に示す。そして右辺第 2 項を $F_\rho^w(\alpha_i)$ で表すことにすると、これは次のようなる。

$$F_\rho^w(\alpha_i) = \frac{\rho_0}{256} (1 - \alpha_i)^5 \times (35\alpha_i^4 + 175\alpha_i^3 + 345\alpha_i^2 + 325\alpha_i + 128) \quad (9)$$

これが壁重み関数 [4] と呼ばれるものである。この壁重み関数を用いれば、式 (8) は次のように単純化される。

$$\rho_i = \sum_{j \in \text{fluid}} m_j W(\mathbf{r}_i - \mathbf{r}_j, h) + F_\rho^w(\alpha_i) \quad (10)$$

4.2 粘性項の計算

密度と同様、粘性項の計算も流体粒子からの寄与分 $\mathbf{f}_{i,\text{fluid}}^v$ と壁からの寄与分 $\mathbf{f}_{i,\text{wall}}^v$ に分割する。

$$\mathbf{f}_i^v = \mathbf{f}_{i,\text{fluid}}^v + \mathbf{f}_{i,\text{wall}}^v \quad (11)$$

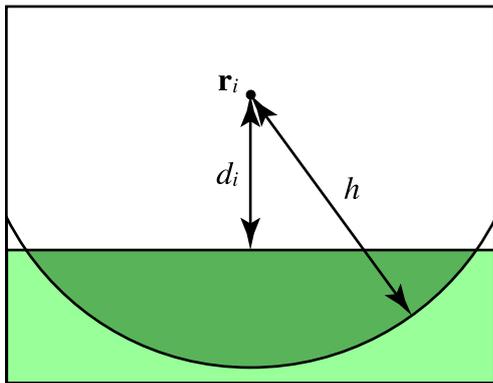


図 1 壁の積分体積

Fig. 1 Integral volume of the wall.

そして、前節と同様にして壁からの寄与分が次のように求まる。

$$F_v^w(\alpha_i) = \frac{15\mu}{2h^2}(1 + \alpha_i)(1 - \alpha_i)^3 \quad (12)$$

$$\mathbf{f}_{i,\text{wall}}^v = (\hat{\mathbf{v}} - \mathbf{v}_i)F_v^w(\alpha_i) \quad (13)$$

ただし、 $F_v^w(\alpha_i)$ は粘性項に対する壁重み関数であり、 \mathbf{v}_i は流体粒子の速度、 $\hat{\mathbf{v}}$ は流体粒子の近傍の壁の速度である。

5. ポリゴン壁の再構築

ここでは、離散境界を用いてポリゴン壁を再構築する方法について述べる。ポリゴン壁が再構築できたならば、これを使用して壁重み関数を計算することが可能となり、結果としてナビエストークス方程式の密度、粘性項および圧力項の値が求まる。これらの値はポリゴン壁を使用して計算したものであるため精度が高く、平らな斜面で流体粒子が滑らかに流れなかったり階段状に流れたりする問題も発生しない。

5.1 離散境界の作成

離散境界とは、ポリゴンメッシュの境界面を点によって離散化したものであり、ポリゴンメッシュをボクセル化し、各ボクセルの中心に点を作成することにより構築される。ポリゴンメッシュのボクセル化は、空間を格子で分割し、ポリゴンメッシュと交差する格子セルをボクセルとすることにより実行される。

5.2 離散境界による壁の再構築

ポリゴンを離散境界に変換する際、離散境界に元のポリゴンの情報を持たせることが可能である。図2においてAは離散境界を構成する点の1つを表し、Bはポリゴン平面を表している。ここで、平面Bの単位法線ベクトルを \mathbf{n}_j とし、点Aの位置ベクトルを \mathbf{r}_j 、点Aの平面Bに関する符号付き距離を d_j で表すことにし、三つ組 $(\mathbf{n}_j, \mathbf{r}_j, d_j)$ を点Aの持つ情報として記憶しておく。すると、これらの情報よりポリゴン平面Bの方程式が次のように得られる。

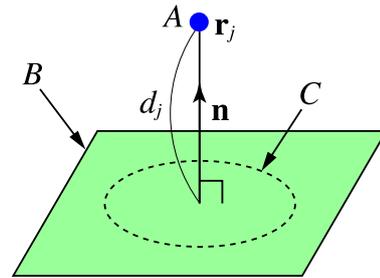


図 2 離散境界の情報と平面の再構築

Fig. 2 Information about the discrete boundary and reconstruction of the plane.

$$(\mathbf{r} - \mathbf{r}_j) \cdot \mathbf{n}_j + d_j = 0 \quad (14)$$

これは、点Aの持つ三つ組情報からポリゴン平面を局所的に再構築可能なことを意味している。図2のCが局所的に再構築されたポリゴン平面または壁を表している。

ナビエストークス方程式に関係する壁重み関数は、この再構築されたポリゴン壁を用いて計算する。すなわち、壁重み関数はすべて流体粒子から壁までの距離 d_i によって計算されるが、ここでは離散境界のうち流体粒子に最も近い点を求め、この点の三つ組情報 $(\mathbf{n}_j, \mathbf{r}_j, d_j)$ を用いて次の式により d_i を計算することにする。

$$d_i = (\mathbf{r}_i - \mathbf{r}_j) \cdot \mathbf{n}_j + d_j \quad (15)$$

ただし、 \mathbf{r}_i は流体粒子の位置座標である。なお、ポリゴン壁が移動や回転をしたときは離散境界の持つ情報のうち \mathbf{n}_j と \mathbf{r}_j を再計算する必要がある。しかしながら、これは行列を掛ける計算であり、しかも、距離関数の計算が3次元空間内の格子点に対して実行しなければならないのに対し、離散境界はポリゴンオブジェクトの表面付近にしか存在しないので高速な計算が可能である。また、ポリゴン壁が移動や回転をしたとき、実際に更新されるのは \mathbf{r}_j のみである。 \mathbf{n}_j の方は必要になったときのみ再計算され、更新される。

5.3 離散境界探索の範囲

流体粒子に関する壁重み関数を計算するとき、離散境界を構成する点のうち流体粒子に最も近いものを求めなければならないが、流体粒子に関しどの範囲を探索すれば十分であろうか。ここで考慮すべきなのが、流体粒子から境界壁までの距離が h 以上のとき壁関数がゼロになるという事実である。したがって、壁境界から h 以内の距離にある流体粒子から離散境界までの最短距離の最大値を l_{\max} とすれば、各流体粒子に対し半径 l_{\max} の球の内部を探索すればよいことになる。ただし、 l_{\max} は離散境界を構成する点の配置パターンによって変化するので、配置パターン c に対する l_{\max} を l_{\max}^c と表記することにする。なお、あらゆる場合をコンピュータを用いて計算したところ、点の配置

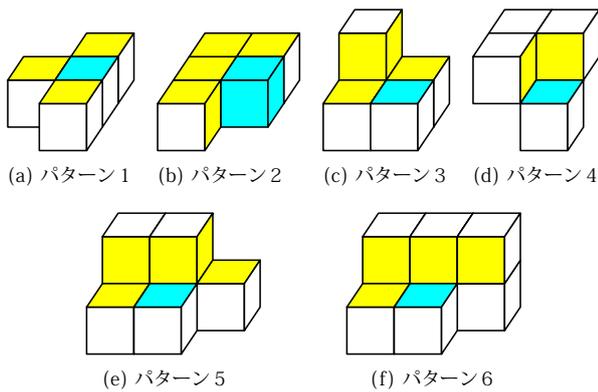


図 3 離散境界の点の配置パターン

Fig. 3 Configuration patterns of points in discrete boundary.

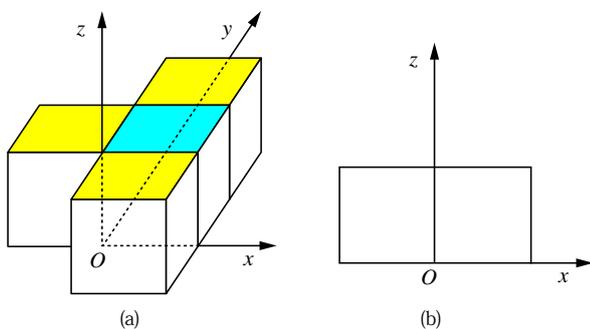


図 4 デカルト座標の配置

Fig. 4 Placement of Cartesian coordinates in the pattern 1.

パターンは基本的に 6 個あることが確認された。これらのパターンを図 3 に示す。図内に描画されている立方体は、離散境界を作成したときに利用したボクセルである。ただし、着色された面は、この面を介して隣接するボクセルが存在しないことを意味している。さらに、着色された面のうちシアン色の面は、この面を持つボクセル内の点に現在注目しているということを表している。

さて、各配置パターンに対して l_{\max} を求める必要があるが、ここでは最も単純な図 3 のパターン 1 について計算してみよう。まず、次の 2 つの定理を用意する。

定理 1 点 P と多面体 H があるとする。 H に属する任意の点を Q とするとき、線分 PQ の長さが最大ならば、 Q は H の頂点である。

定理 2 有限個の平面に囲まれた凸領域があるとき、この領域の閉包は凸多面体である。そこで、ある点 P からこの多面体の任意の点 Q までの距離の最大値を考える。領域を囲む面の 1 つが移動すると一般に線分 PQ の長さの最大値は変化するが、それは移動の開始時または終了時に最大となる。

点の配置パターンが決定しても、ボクセル内を通過する境界面の位置によって l_{\max} は変化する。したがって、境界

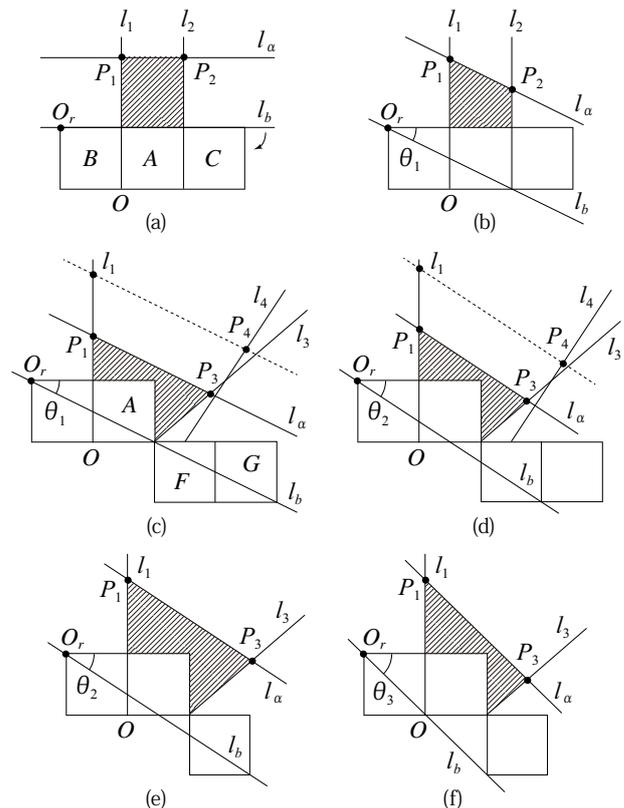


図 5 配置パターン 1 における境界面の回転

Fig. 5 Rotation of a boundary plane in the configuration pattern 1.

面の位置を変化させて、その中から最大となる l_{\max} を求める必要がある。そこで、配置パターン 1 の中に図 4(a) に示すようなデカルト座標を配置する。そして、配置パターン 1 を平面 $y = 0$ で切断したときの断面図を使用して l_{\max} を求めることにする。この断面図を図 4(b) に示す。さて、図 5(a) は図 4(b) に境界面を配置したものであるが、直線 l_b が壁境界と平面 $y = 0$ の交線を表している。この図では、境界面の配置を z 軸と直交する場合に限定し、その中で l_{\max} が最大となる位置に境界面を設定している。より正確に表現すれば、直線 l_b は、点 O_r を通り x 軸に平行な直線に対し、点 O_r を中心に無限小の角度だけ矢印の方向に回転したものである。境界面をこのように配置すると、図に示すように、現在注目しているボクセル A の左右にそれぞれボクセル B とボクセル C が作成される。また、図には記入されていないが、ボクセル A の前後 (y 軸方向) にもそれぞれボクセル D とボクセル E が作成される。離散境界はボクセルの中心に作成されるが、ボクセル X ($X = A, B, C, D, E$) に対する離散境界の点を P_X で表すことにすると、流体粒子の中で点 P_A によって処理される可能性のあるものは P_A に最も近い粒子であるので、点 P_A と点 P_X の垂直二等分面 ($X = B, C, D, E$) によって囲まれた領域にある流体粒子がこれに相当する。なお、図の直線 l_1 は点 P_A と点 P_B の垂直二等分面が平面 $y = 0$ と

交差するときの交線を表しており、直線 l_2 は点 P_A と点 P_C の垂直二等分面が平面 $y = 0$ と交差するときの交線を表している。さらに、境界面から h 以内の距離にある流体粒子のみが壁の影響を受けるので、境界面から h だけ離れた平面を考え、この平面と平面 $y = 0$ の交線を直線 l_α で表す。ただし α はパラメータであり、ボクセルの1辺の長さを a とするとき、 $\alpha = h/a$ である。以上より、離散境界の点 P_A によって処理される流体粒子の存在範囲は図 5(a) の斜線を引いた部分になる。そして、この範囲にある点のうち点 P_A から最も離れた点は定理 1 により点 P_1 または点 P_2 ということになる。平面 $y = a$ 上にも P_A から最も離れている可能性のある点が存在するが、配置パターン 1 は平面 $y = a/2$ に関して対称であるので、これを考慮に入れる必要はない。そこで、線分 $\overline{P_A P_1}$ と $\overline{P_A P_2}$ の長さを計算すると次のようになる。

$$L(\overline{P_A P_1})/a = L(\overline{P_A P_2})/a = \sqrt{4\alpha^2 + 4\alpha + 3/2} \quad (16)$$

ただし、 L は線分の長さを返す関数である。

図 5(b) は、点 O_r を通り x 軸に平行な直線 l_b に対し、点 O_r を中心にして $\theta_1 - (\tan(\theta_1) = 1/2)$ だけ回転したときのものである。ここに、 $-$ は θ_1 から無限小だけ小さい角度であることを表す。定理 2 より l_{\max}^1 は回転中に生じないことが保証されているので、回転後の状態を検査すればよい。図 5(a) の場合と同様にして、次の式が得られる。

$$L(\overline{P_A P_1})/a = \sqrt{5\alpha^2 + 2/2} \quad (17)$$

$$L(\overline{P_A P_2})/a = \sqrt{5\alpha^2 - 2\sqrt{5}\alpha + 3/2} \quad (18)$$

さらに、図 5(c) は直線 l_b を $\theta_1 +$ だけ回転したときのものである。ただし、 $+$ は θ_1 から無限小だけ大きい角度であることを表している。このときは、ボクセル C が消失し、代わりにボクセル F とボクセル G を考慮に入れることが必要になる。すなわち、ボクセル F とボクセル G に対する離散境界の点をそれぞれ P_F と P_G で表せば、これらの点と点 P_A の垂直二等分面が点 P_A によって処理される流体粒子の存在範囲に関係してくる。この図において、点 P_A と点 P_F の垂直二等分面が平面 $y = 0$ と交差するときの交線を l_3 で表し、点 P_A と点 P_G の垂直二等分面が平面 $y = 0$ と交差するときの交線を l_4 で表している。直線 l_3 と直線 l_α の交点を P_3 とし、直線 l_4 と直線 l_α の交点を P_4 とするとき、 α が小さい間は P_A から最も離れた点の候補は P_3 であるが、 α が大きくなって点 P_3 が直線 l_3 と l_4 の交点を越えると P_A から最も離れた点の候補は P_4 となる。よって、この場合の最長距離の候補は次のようになる。

$$\begin{cases} L(\overline{P_A P_3})/a = \sqrt{40\alpha^2 + 27/6} & \alpha < 3/\sqrt{5} \\ L(\overline{P_A P_4})/a = \sqrt{100\alpha^2 - 20\sqrt{5}\alpha + 155/10} & \text{それ以外} \end{cases} \quad (19)$$

この他に $L(\overline{P_A P_1})$ も候補であるが、これは図 5(b) の場合と全く同一の式になる。

表 1 各配置パターンに対する l_{\max}^c の値

Table 1 Values of l_{\max}^c for each configuration pattern.

c	l_{\max}^c/a
1	$\begin{cases} \sqrt{4\alpha^2 + 4\alpha + 3/2} & \alpha < 1 + \sqrt{2} \\ \sqrt{8\alpha^2 - 4\sqrt{2}\alpha + 3/2} & \text{それ以外} \end{cases}$
2	$\sqrt{6\alpha^2 + 5/2}$
3	$\leq \sqrt{4\alpha^2 + 4\alpha + 3/2}$
4	$\begin{cases} \sqrt{11/2} & \alpha < \sqrt{2} \\ \sqrt{6\alpha^2 - 2\sqrt{2}\alpha + 3/2} & \text{それ以外} \end{cases}$
5	$\begin{cases} \sqrt{11/2} & \alpha < \sqrt{6/2} \\ \sqrt{48\alpha^2 + 27/6} & \sqrt{6/2} \leq \alpha < (3\sqrt{29} + 6\sqrt{13})/23 \\ \sqrt{29\alpha^2 - 2\sqrt{29}\alpha + 9/4} & \text{それ以外} \end{cases}$
6	$\sqrt{6\alpha^2 + 5/2}$

図 5(d) は、直線 l_b を $\theta_2 - (\tan(\theta_2) = 2/3)$ だけ回転したときのものである。他の条件は図 5(c) の場合と全く同一であるので、 P_A からの最長距離の候補として次の式を得ることができる。

$$L(\overline{P_A P_1})/a = \sqrt{52\alpha^2 - 4\sqrt{13}\alpha + 19/6} \quad (20)$$

$$\begin{cases} L(\overline{P_A P_3})/a = \sqrt{104\alpha^2 - 16\sqrt{13}\alpha + 83/10} & \alpha < 6/\sqrt{13} \\ L(\overline{P_A P_4})/a = \sqrt{65\alpha^2 - 20\sqrt{13}\alpha + 116/8} & \text{それ以外} \end{cases} \quad (21)$$

直線 l_b をさらにわずかに回転させ、回転角を $\theta_2 +$ にすると図 5(e) のようになる。このときボクセル G が消失するので、それに伴い直線 l_4 を考慮する必要がなくなる。すなわち、 α の値に関係なく直線 l_α と直線 l_3 の交点 P_3 のみを検査すればよいようになる。すなわち、 P_A からの最長距離の候補として次の式が得られる。

$$L(\overline{P_A P_3})/a = \sqrt{104\alpha^2 - 16\sqrt{13}\alpha + 83/10} \quad (22)$$

この他に $L(\overline{P_A P_1})$ も候補であるが、これは図 5(d) の場合と全く同一の式である。

最後に、直線 l_b を $\theta_3 - (\theta_3 = \pi/4)$ まで回転させ、図 5(f) を得る。この場合、 P_A からの最長距離の候補は次の 2 式である。

$$L(\overline{P_A P_1})/a = \sqrt{8\alpha^2 - 4\sqrt{2}\alpha + 3/2} \quad (23)$$

$$L(\overline{P_A P_3})/a = \sqrt{4\alpha^2 - 4\sqrt{2}\alpha + 5/2} \quad (24)$$

以上、式 (16) から式 (24) までのすべての候補の最大値を求めると、次のように l_{\max}^1 が得られる。

$$l_{\max}^1/a = \begin{cases} \sqrt{4\alpha^2 + 4\alpha + 3/2} & \alpha < 1 + \sqrt{2} \\ \sqrt{8\alpha^2 - 4\sqrt{2}\alpha + 3/2} & \text{それ以外} \end{cases} \quad (25)$$

図 3 の残りの配置パターンに対しても、上と同様に境界面を移動することで l_{\max}^c を求めることが可能であり、これらの式を表 1 に示す。さらに、これらをまとめることにより $1 \leq \alpha \leq 3$ のとき、近似的に次の式が成立する。

$$l_{\max}^c/a = \sqrt{6\alpha^2 + 5/2} \quad (26)$$

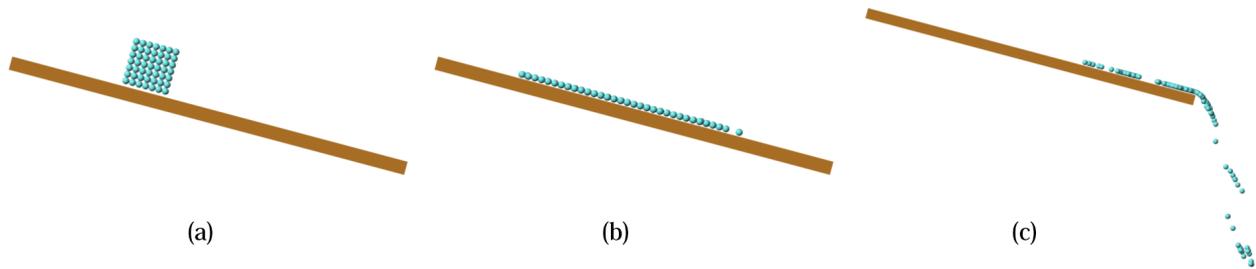


図 6 平らな斜面を用いたシミュレーション (a) 初期配置 (b) 3層粒子壁 (c) 再構築壁
Fig. 6 Simulations using a flat slop. (a)initial configuration (b)3 layered particle wall (c)reconstructed wall

α は通常 3 以下であることが多いので、この式だけで十分であるが、 $3 < \alpha \leq 5$ のときは次の式を使用する必要がある。

$$l_{\max}/a = \sqrt{7}\alpha/2 \quad (27)$$

6. 数値シミュレーション

平らな斜面を用いた数値シミュレーションを実施した。シミュレーションにおいて、液体の密度は水と同じ 1000 kg/m^3 を使用し、表面張力はないものとした。また、重力加速度として 9.8 m/s^2 を用いた。実験に使用したコンピュータは Apple Mac Pro 2013 (クロック周波数 3.5 GHz) で、 32 GByte の主記憶を搭載している。

シミュレーションでは、 15 度の傾斜を持つ平らな斜面に立方体状の流体粒子を配置し、重力により崩壊させた後、斜面を滑り落とさせた。立方体の 1 辺の長さは 4 cm とした。シミュレーション結果を図 6 に示しているが、(a) が流体粒子の初期配置である。また、(b) は境界壁として 3 層粒子壁を用いた場合の 0.9 sec 後の流体粒子の様子を表しているが、粒子が斜面の途中で停滞しているのが観測された。これは、時間がさらに経過しても状況は変化しなかった。これに対し、(c) は再構築されたポリゴン壁を用いた場合の 0.9 sec 後の流体粒子の様子を表している。粒子が斜面に沿って滑らかに流れ落ちているのが見てとれる。

7. おわりに

離散境界を使用してポリゴン境界を再構築する手法について述べた。本報告では特に、流体粒子を基準にして離散境界を探索するときの球の半径を求める方法を中心に論じた。すなわち、離散境界の配置パターンを基本的な 6 個に分類し、それぞれについて探索半径を求め、それらを最終的に簡単な式へと縮約した。なお、実際に数値シミュレーションを実施したが、境界として 3 層粒子壁を使用した場合、流体粒子が平らな斜面の途中で停滞したのに対し、再構築されたポリゴン壁ではそのようなことは起こらないことが確認できた。

さらに、本手法を用いる利点として、壁境界が移動した場合の処理コストが小さいことが挙げられる。すなわち、壁境界が移動した場合でも離散境界の位置ベクトルに対して移動変換と回転変換を適用し、各離散境界が持つ平面の単位法線ベクトルに対して回転変換を適用するだけでよく、簡単な処理で対応可能である。

参考文献

- [1] Monaghan, J.: An introduction to SPH, *Computer Physics Communications*, Vol. 48, No. 1, pp. 89–96 (1988).
- [2] Koshizuka, S. and Oka, Y.: Moving-particle semi-implicit method for fragmentation of incompressible fluid, *Nuclear Science and Engineering*, Vol. 123, No. 3, pp. 421–434 (1996).
- [3] 笠 晃一: 再構築ポリゴン壁を用いた SPH の壁境界計算手法, 情報処理学会研究報告, Vol. 2014-CG-156(4), pp. 1–6 (2014).
- [4] 原田隆宏, 越塚誠一: SPH における壁境界計算手法の改良, 情報処理学会論文誌, Vol. 48, No. 4, pp. 1838–1846 (2007).
- [5] Mauch, S.: A Fast Algorithm for Computing the Closest Point and Distance Transform, *Technical Report, California Institute of Technology* (2000).
- [6] Sigg, C., Peikert, R. and Gross, M.: Signed Distance Transform using Graphics Hardware, *Proc. IEEE Visualization*, pp. 83–90 (2003).
- [7] Sud, A., Govindaraju, N., Gayle, R. and Manocha, D.: Interactive 3D Distance Field Computation using Linear Factorization, *Proc. ACM symposium on Interactive 3D graphics and games*, pp. 117–124 (2006).
- [8] 原田隆宏, 越塚誠一: 離散境界を用いた距離関数の構築手法, 情報処理学会論文誌, Vol. 48, No. 4, pp. 1820–1828 (2007).
- [9] Müller, M., Charypar, D. and Gross, M.: Particle-based Fluid Simulation for Interactive Applications, *Proc. SIG-GRAPH/Eurographics Symposium on Computer Animation*, pp. 154–159 (2003).