

# 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辺と頂点の扱いについて

笠 晃<sup>1,a)</sup>

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、その近傍でのみポリゴン壁を再構築するのである。これまでの研究で1個の平面に対する処理は確立されているが、ポリゴンが複数個あるときの辺と頂点の扱いは未解決のままであった。この問題に対しある程度の解決を見たので、ここでは辺と頂点の処理を中心に報告する。本研究の手法を用いることにより、ポリゴンメッシュの境界上でも流体粒子が自然な動きをすることが確認できた。

## Processing Edges and Vertices in Wall Boundary Calculation Model of SPH using Reconstructed Polygon Wall

RYU KOICHI<sup>1,a)</sup>

### 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、これまではポリゴンメッシュがただ1個のポリゴンを持つときのみを扱っており、ポリゴンメッシュが複数個のポリゴンを持つ

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

a) ryu@fit.ac.jp

ときに生ずる、辺や頂点における処理の問題は未解決のままであった。この辺と頂点の処理についてある程度の見通しがついたので、開発した手法について本報告で述べる。

## 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。

## 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、 $\rho$ ,  $\mathbf{v}$ ,  $p$ ,  $\mu$  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  $\Omega$  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 ( $\bullet$ ) 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  $\rho_i$  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、 $\rho_0$  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、 $\alpha_i$  の関数と

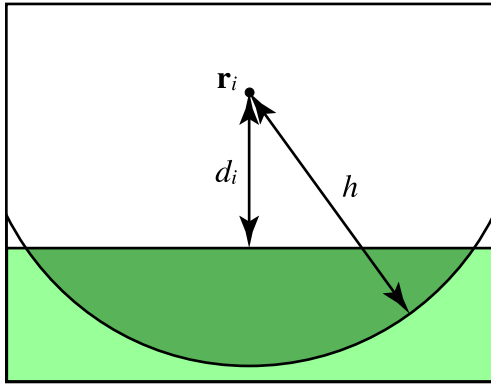


図 1 壁の積分体積

Fig. 1 Integral volume of the wall.

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、これは次のようになる。

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

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。

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

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。

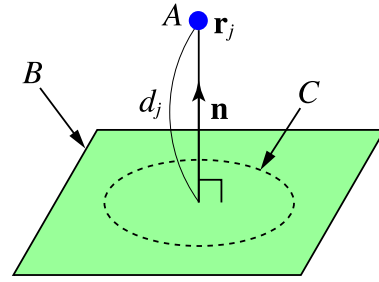


図 2 離散境界の情報と平面の再構築

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

#### 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  $\langle \mathbf{n}_j, \mathbf{r}_j, d_j \rangle$  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。

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

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  $\langle \mathbf{n}_j, \mathbf{r}_j, d_j \rangle$  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 辺と頂点の処理について

先に述べたように、ポリゴンメッシュをボクセル化し、各ボクセルの中心に離散境界の点を作成する。そして、この点にボクセルを横切る平面の情報を持たせるとというのが、今までの手法であった。すなわち、これまでの研究では各ボクセルがポリゴンの辺や頂点を含まない場合のみを扱ってきた。ポリゴンの辺や頂点を含むボクセルにおいて、ポリゴンメッシュをどのように近似するかに関しては様々な手法が考えられるが、ここでは平面で近似することにした。このようにすることにより、ボクセルがポリゴンの面のみを含む場合との統一性が保持されるためである。以下では、ボクセルがポリゴンの辺を含む場合と頂点を含む場合に分けて、近似手法について議論する。

### 5.4 平面による辺の近似

図3に示すように、あるボクセルの内部において2つの面  $A$  と  $B$  が辺  $l$  を共有しているものとする。このとき面  $C$  によって辺  $l$  を近似することにする。図4を用いて面  $C$  を厳密に定義しよう。これは図3を辺に平行な方向から見た図である。図4において、 $l_A$  は面  $A$  と面  $C$  の交線を表しており、 $l_B$  は面  $B$  と面  $C$  の交線を表している。このとき面  $C$  は交線  $l_A$  と交線  $l_B$  が辺  $l$  に平行になるように作成される。さらに、交線  $l_A$  と辺  $l$  との距離、および交線  $l_B$  と辺  $l$  との距離がともに  $ka$  になるようにする。ただし、 $a$  はボクセルの1辺の長さであり、 $k$  は比例定数である。 $k$  は通常1より小さい値をとる。なお、図4(a)は2つの面  $A$  と  $B$  がなす角度  $\theta$  が  $180^\circ$  以下の場合であるが、 $\theta$  が  $180^\circ$  より大きい場合の面  $C$  の位置を図4(b)に示す。面  $C$  は必ず元の面  $A$  と  $B$  に交差するように作成される。

さて、面  $A$  と面  $B$  の方程式がそれぞれ次の式 (16) と式 (17) で与えられているとする。

$$\mathbf{r} \cdot \mathbf{n}_A = h_A \quad (16)$$

$$\mathbf{r} \cdot \mathbf{n}_B = h_B \quad (17)$$

ただし、 $\mathbf{n}_A$  は平面  $A$  の単位法線ベクトルであり、 $\mathbf{n}_B$  は平面  $B$  の単位法線ベクトルである。すると、求めるべき面  $C$  の方程式が、次のようにして求められる。

$$\mathbf{r} \cdot \mathbf{n}_C = h_C \quad (18)$$

ここに、 $\mathbf{n}_C$  および  $h_C$  は次の式により計算される。

$$\mathbf{n}_C = p(\mathbf{n}_A + \mathbf{n}_B) \quad (19)$$

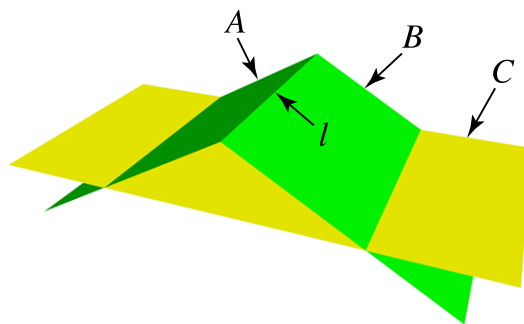


図3 辺を含むボクセルにおける近似の概略  
Fig. 3 Outline of approximation in a voxel that includes an edge.

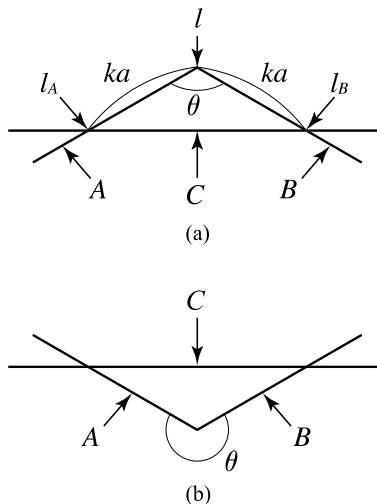


図4 辺を近似する平面  
Fig. 4 A plane which approximates an edge.

$$h_C = \begin{cases} p(h_A + h_B) - ka\sqrt{1-q^2} & \theta \leq 180^\circ \\ p(h_A + h_B) + ka\sqrt{1-q^2} & \theta > 180^\circ \end{cases} \quad (20)$$

ただし、 $p$  と  $q$  は次のように与えられる。

$$p = \frac{1}{\|\mathbf{n}_A + \mathbf{n}_B\|} \quad (21)$$

$$q = \mathbf{n}_A \cdot \mathbf{n}_B \quad (22)$$

### 5.5 平面による頂点の近似

あるボクセルの内部に頂点  $P$  が存在して、 $N$  個の面  $F_1 \sim F_N$  が頂点  $P$  を共有しているものとする。ただし、面  $F_i$  と面  $F_{i+1}$  ( $i = 1 \sim N-1$ ) および面  $F_N$  と面  $F_1$  が辺を共有するように添え字が与えられている。また、辺を共有する隣り合った面のなす二面角がすべて  $180^\circ$  以下のとき、頂点  $P$  は凸性質を持つとすることにし、逆に、これらの二面角がすべて  $180^\circ$  以上のとき、凹性質を持つとすることにする。そして、まずは凸性質を持つ頂点のみを考察しよう。図5に  $N = 4$  の場合の例を示す。このような頂点と面を近似するのに、たとえば図5の面  $G$  を使用することにする。近似面の厳密な定義を以下で与えよう。

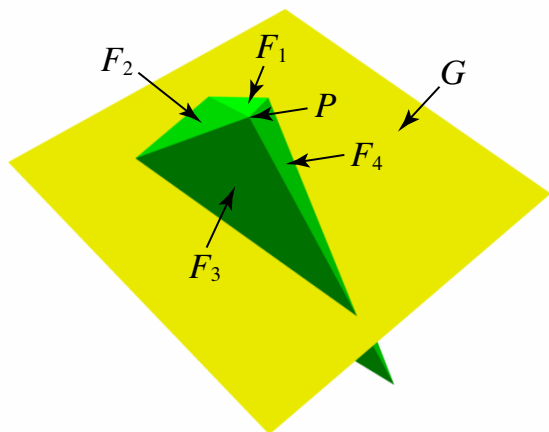


図 5 辺を含むボクセルにおける近似の概略

Fig. 5 Outline of approximation in a voxel that includes a vertex.

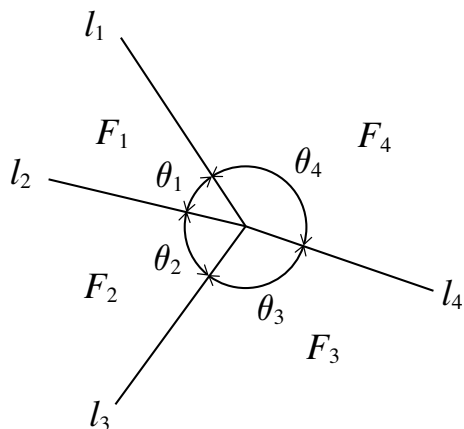


図 6 共有される辺および各面が占める角度

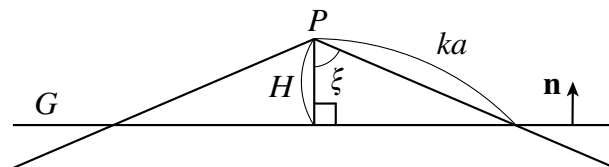
Fig. 6 Shared edges, and angles occupied by each of faces.

まず、近似面の単位法線ベクトル  $\mathbf{n}$  を求めることから始める。これは面  $F_1 \sim F_N$  の単位法線ベクトルの角度重み付き平均から求められる。1 から  $N-1$  までの  $i$  について、面  $F_i$  と面  $F_{i+1}$  が共有している辺を  $l_{i+1}$  とし、さらに面  $F_N$  と面  $F_1$  が共有している辺を  $l_1$  とする。  $N=4$  の場合の例を図 6 に示す。そして、1 から  $N-1$  までの  $i$  について、辺  $l_i$  と辺  $l_{i+1}$  のなす角を  $\theta_i$  とし、さらに辺  $l_N$  と辺  $l_1$  のなす角を  $\theta_N$  とする。一方、面  $F_i$  のそれぞれについて単位法線ベクトルを  $\mathbf{n}_i$  とする。このとき、近似面の単位法線ベクトル  $\mathbf{n}$  は次の式によって求められる。

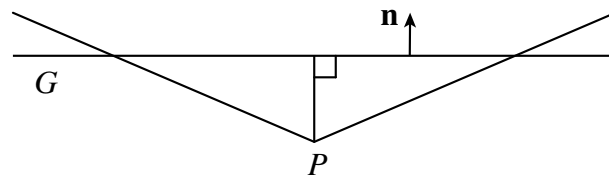
$$\tilde{\mathbf{n}} = \sum_{i=1}^N \theta_i \mathbf{n}_i \quad (23)$$

$$\mathbf{n} = \frac{\tilde{\mathbf{n}}}{\|\tilde{\mathbf{n}}\|} \quad (24)$$

次に、近似面の位置を決定する。これは、まず  $F_1 \sim F_N$  によってできる立体角を求め、それと等しい立体角を持つ円錐面を考え、円錐面上の点で頂点から距離  $ka$  の点を通る面を求めることで決定される。3つの辺  $l_i, l_j, l_k$  に対し、



(a)



(b)

図 7 近似に使用される円錐と近似面の位置

Fig. 7 The cone used in approximation and the position of a approximating face.

この中の2つの辺同士がなす角をそれぞれ  $\gamma_1, \gamma_2, \gamma_3$  とすれば、3つの辺によってできる立体角  $\tilde{\Omega}$  は次の L'Huilier の定理によって求めることが可能である。

$$\tilde{\Omega} = 4 \tan^{-1} \sqrt{\tan \frac{s}{2} \tan \frac{s-\gamma_1}{2} \tan \frac{s-\gamma_2}{2} \tan \frac{s-\gamma_3}{2}} \quad (25)$$

$$s = \frac{\gamma_1 + \gamma_2 + \gamma_3}{2} \quad (26)$$

そこで、3つの辺  $l_i, l_j, l_k$  によってできる立体角を  $\omega(l_i, l_j, l_k)$  で表すことにすれば、  $F_1 \sim F_N$  によってできる立体角  $\Omega$  は次の式によって求められる。

$$\Omega = \sum_{i=2}^{N-1} \omega(l_1, l_i, l_{i+1}) \quad (27)$$

近似に使用される円錐を図 7(a) に示す。これは円錐を回転軸に垂直な方向から見た図であり、近似面  $G$  が追加されている。さて、円錐の頂角を  $2\xi$  とすると、円錐の側面が切り取る空間の立体角が簡単な計算により  $2\pi(1 - \cos \xi)$  と求められる。これが  $F_1 \sim F_N$  によってできる立体角  $\Omega$  に等しいのだから、次の式が成り立つ。

$$\Omega = 2\pi(1 - \cos \xi) \quad (28)$$

図 7(a) に示すように、円錐の回転軸と近似面の単位法線ベクトルが平行になるように近似面を作成する。ただし、頂点  $P$  が近似面の正領域にあるようにする。頂点  $P$  が凹性質を持つ頂点のときも、これまでとほぼ同様な議論が成り立つが、図 7(b) に示すように、頂点  $P$  が近似面の負領域にあるようにする。頂点  $P$  が凸性質を持つと凹性質を持つと、円錐と近似面の交線は円であり、頂点  $P$  から円までの距離が  $ka$  になるようにする。そうすると、頂点  $P$  から近似面までの距離  $H$  は図 7(a) から次のように求まる。

$$H = ka \cos \xi \quad (29)$$

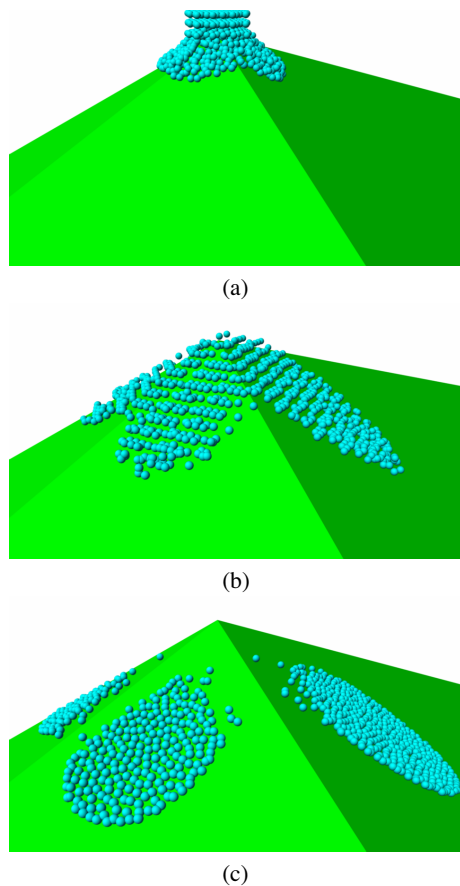


図 8 ポリゴンメッシュを用いたシミュレーション  
(a) 開始直後の状態 (b) 粒子壁 (c) 再構築壁

Fig. 8 Simulations using a polygon mesh. (a) the state immediately after starting (b) particle wall (c) reconstructed wall

よって、式 (28) と式 (29) より  $H$  を求める式が得られる。

$$H = ka \left( 1 - \frac{\Omega}{2\pi} \right) \quad (30)$$

頂点  $P$  の座標は、面  $F_1 \sim F_N$  から 3 つの面を選択し、それらの面の方程式を連立して解くことによって得られるが、これを  $\mathbf{r}_0$  とする。そうすると、式 (24) と式 (30) で得られる  $\mathbf{n}$  と  $H$  を用いて、近似面  $G$  の方程式は次のように与えられる。

$$\begin{cases} \mathbf{r} \cdot \mathbf{n} = \mathbf{r}_0 \cdot \mathbf{n} - H & \text{頂点が凸性質を持つとき} \\ \mathbf{r} \cdot \mathbf{n} = \mathbf{r}_0 \cdot \mathbf{n} + H & \text{頂点が凹性質を持つとき} \end{cases} \quad (31)$$

## 6. 数値シミュレーション

山型のポリゴンメッシュを用いて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  $1000 \text{ kg/m}^3$  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  $9.8 \text{ m/s}^2$  を用いた。

シミュレーションでは、4 個のポリゴンが 1 つの頂点を共有するポリゴンメッシュを作成した。大きさは  $44 \text{ cm} \times 28 \text{ cm} \times 8 \text{ cm}$  程度である。そして、その頂点のほぼ真上から流体粒子を発生させ、頂点付近に衝突するように

落下させた。この様子を図 8(a) に示す。同図 (b) は粒子壁を使用したときの  $0.83 \text{ sec}$  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。これに対し、同図 (c) は再構築されたポリゴン壁を用いた場合の  $0.68 \text{ sec}$  後の流体粒子の様子を表している。粒子が頂点や辺によって分割され、斜面に沿って滑らかに流れ落ちていくのが見てとれる。また、頂点や辺の付近における流体粒子の動きに不自然な点は見られなかった。

## 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、ポリゴンメッシュの辺と頂点における処理を中心に論じたが、どちらの場合も平面によって近似され、単一平面の場合の自然な拡張となった。特に頂点の場合は頂点を共有するポリゴンに関する立体角を求め、これと同じ立体角を持つ円錐面を使用して近似平面を決定した。なお、実際に数値シミュレーションを実施したが、境界として粒子壁を使用した場合、流体粒子が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。また、再構築したポリゴン壁の場合も頂点や辺の付近で流体粒子の不自然な動きは観測されず、辺と頂点の近似がうまく機能していることが推測された。

## 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).