粒子法におけるサブパーティクルスケールの乱流を考慮した液体表面生成手法

A Fluid Surface Reconstruction Method using SPH with Sub-Particle-Scale Turbulence

三村豪 † 藤澤誠 ‡ 天野敏之 †† 宮崎純 ‡‡ 加藤博一 ‡‡

Go Mimura† Makoto Fujisawa‡ Toshiyuki Amano†† Jun Miyazaki‡‡ and Hirokazu Kato‡‡

† 任天堂株式会社 † Nintendo Co., Ltd. ‡ 筑波大学 ‡ University of Tsukuba

†† 山形大学 †† Yamagata University ‡‡ 奈良先端科学技術大学院大学 ‡‡ Nara Institute of Science and Technology

E-mail: †fujis@slis.tsukuba.ac.jp, ‡g5roro@gmail.com, ††amano@yz.yamagata-u.ac.jp, ‡‡{miyazaki,kato}@naist.jp

1 はじめに

水，煙，炎などの流体はCGとして表現されることが多く，それらを含むリアリティの高いアニメーションを生成するために流体シミュレーションがよく用いられる．流れは層流と乱流の2種類に分けることができる．層流は乱れがほとんどなく，規則正しく運動する流れである．この流れをシミュレーションするための計算コストは乱流に比べて小さい．このため，これまでCGの分野で研究されてきたのは層流がほとんどである．逆に，乱流は乱れが多く，不規則な流れであり，自然現象には乱流が多く含まれる．例えば，川や滝，船のあとにできる後流などである．そのため，流体のCGアニメーションを生成するために，乱流を表現するということは重要な課題である．近年では，乱流を含む気体の流れを再現する研究もあるが，乱流を含む液体の流れを高品質にリアルタイムに表現した研究はない．これは，乱流を表現するスケールの乱れを捉えるための計算コストが大きいためである．そこで，本論文では，粒子法の一種であるSPHで平均流を計算し，パーティクルをサブパーティクルに分割することにより，粒子法によるシミュレーションだけでは再現できない乱れを含んだパーティクルの動き，表面を生成する手法を提案する．また，その実装はGPUでの並列処理に適したものであり，高速に実行できる．

2 関連研究

Stam[12] はセミラグランジュ移流法により，グリッド法で流れを安定して解く方法を開発した．これにより，グリッドを用いた流体計算がCG分野で広く用いられるようになった．しかし，この手法は移流による数値拡散という潜在的な問題点をかかえている．数値拡散は乱流エネルギーの散逸を起こし，乱流場を捉えるのを難しくする．この数値拡散を抑えるために，BFFCC[3] やQUICK[7]，CIP法[4] といった高次の補間法を移流に用いる手法が提案されている．また，Fedkiwら [1] は数値拡散で失われる渦を検出し，再注入する手法を提案し，Hongら [2] はこれをSPH法へ導入した．

一方，乱流モデルを用いて乱れを構成する渦を手続き的に生成する手法として，Stam[13] はKolgomorovの理論に基づき乱流を手続き的に生成する方法を開発した．また，渦や流れのエネルギーを計算することで，従来の流体シミュレーションに乱流を付加することも可能である．Selleら [11] は渦パーティクル法を提案し，グリッド上では数値拡散により失われる渦を再現した．Pfaffら [10] は渦パーティクルの散布場所を壁面境界層のせん断流れから計算することで固体とのインタラクションにより発生する乱れを計算した．Kimら [5] は，Kolgomorovの理論を用いて，粗いグリッドでの流体シミュレーションにウェーブレットノイズによる渦を組み合わせることで，高解像度の煙の乱流アニメーションを生成した．Pfaff[9] らはk-εモデルにより異方性を考慮した乱れを再現し，リアルタイムに煙のアニメーションを生成した．Narainら [8] は，液体の流れにお

ける乱れをエネルギーモデルによりシミュレーションした．しかし，多くの計算時間を必要とした．藤澤ら [15] は，粒子法においてウェーブレット解析を用いて求めた乱流応力を外力項に加えることで，高速に流れに乱れを加えている．ただし，この手法ではパーティクススケール以下の乱れは再現できていない．我々はパーティクルをサブパーティクルへと分割することで，パーティクルスケール以下の小さな乱れも再現する手法を提案する．

3　シミュレーション手法

全体的なシミュレーションの流れは以下のようになる．

1. SPH により粘性項，圧力項，外力項を計算し，速度，位置を更新
2. ウェーブレット解析により乱流エネルギースペクトル分布を算出し，乱流応力を計算
3. パーティクルの位置と速度を更新
4. 各パーティクルに属するサブパーティクルの位置を更新
5. マーチングキューブ法により表面生成

この章の残りの部分でこれらの各ステップについて詳しく述べる．(3.1 節において 1,2,3 について，3.2 節において 4 について，最後に 3.3 節において 5 について述べる)

3.1　SPH 法による流れの計算

流体の流れを計算するためにパーティクル法の一種である SPH を用いる．支配方程式である非圧縮のナビエ・ストークス方程式は以下である．

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nu \nabla^2 \boldsymbol{u} - \frac{1}{\rho}\nabla p + \boldsymbol{f} \tag{2}$$

ここで，t は時間，$\boldsymbol{u}$ は流体速度，μ は粘性係数，ρ は流体の密度，p は圧力，$\boldsymbol{f}$ は外力である．式 (1) は質量保存則である．実際の流体では厳密には非圧縮でないため，右辺はゼロにはならないが，一般的に流速が音速に比べて十分小さい場合は非圧縮とみなしてゼロとする．本論文では，液体のシミュレーションにラグランジュ的方法である SPH を用いる．SPH はパーティクル自体が液体を表しているため，パーティクル質量が変化しないかぎり質量保存性が常に保持され (質量保存式である式 (1) を解く必要がない)，

パーティクルスケールでの乱流を再現するために，[15] と同様に式 (2) の外力項 $\boldsymbol{f}$ に乱流応力を加える．乱流はさまざまな大きさの渦の集合として考えることができ，細かな渦に分裂したり (forward scattering)，合流して大きな渦を形成する (back scattering)．このようにして，異なるスケールの渦間のエネルギー遷移により流れが変化する．波数 k が 2 倍になった場合の乱流エネルギー $\hat{e}$ は Kolmogorov の理論により，以下のようになる [5].

$$\hat{e}(2k) = \hat{e}(k)2^{-\frac{5}{3}} \tag{3}$$

[15] ではウェーブレット解析とウェーブレットノイズにより，パーティクルスケール (以下，PS) での乱流を再現している．本論文でもこの手法を用いて PS での乱流を再現する．さらに，PS での乱流エネルギーを式 (3) に代入することで，サブパーティクルスケール (以下，SPS) でのエネルギーを算出する．

3.2　サブパーティクル分割

乱流は様々な大きさの渦の集合と考えることができる．そのため，乱流を完全に表現したいならば，すべての大きさの渦を再現しなければならない．しかし，SPH において再現できる渦の大きさはパーティクルの大きさに依存し，小さな渦をシミュレートするためにはより小さいパーティクルが必要となる．これはシミュレーション全体のパーティクル数を増大させ，同時に計算時間を爆発的に増大させる結果となる．Laan ら [14] はスクリーンスペースメッシュに対してノイズ関数を適用することで，パーティクル以外のフレームワークを用いた SPS での乱れの表現を可能とする手法を提案した．我々はより物理的なアプローチをとる．3.1 節で述べたように乱流では異なるスケールの渦間でのエネルギー遷移が発生する．このエネルギー遷移の中でも forward scattering に注目し，ウェーブレット解析で算出した PS での乱流エネルギーに基づき，SPH のパーティクルを分割し，それを元のパーティクルを中心として回転させることでより細かな渦を再現する．この分割したパーティクルをサブパーティクルと呼ぶ．渦パーティクル法 [11] と似たものであるが，サブパーティクルは単に元のパーティクルの周りを回転するだけであり，計算コストは非常に小さく，また，並列化にも適している．

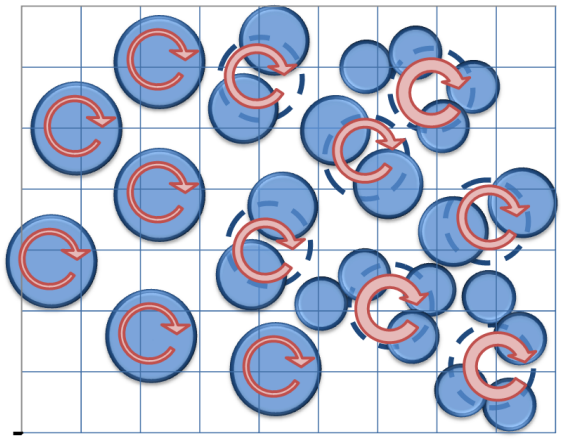


図 1: パーティクルからサブパーティクルへの分割

式 (3) に基づき乱流エネルギー遷移を計算するために，SPH のパーティクル p_i^0 は図 1 に示したように，2 つのサブパーティクル $p_i^{1,j}$, $j = 0, 1$ に分割できるものとする．また，サブパーティクル p_i^L についてもさらに 2 つのサブパーティクル p_i^{L+1} に再帰的に分割できるものとする．つまり，SPH での各パーティクルを設定したレベルまでそれぞれ 2^j 個に分割する．ここで，分割前のサブパーティクルを親，分割後の 2 つのサブパーティクルを子と呼ぶ．これにより，乱流理論における forward scattering を表現する．レベル L まで分割すると，1 つのパーティクルが 2^L 個のサブパーティクルへと分割されることとなる．このように分割することで，他のパーティクルと干渉することなしにサブパーティクルを生成できるため，並列処理に適している．あるサブパーティクル p^L とそれをさらに 2 つに分割したサブパーティクル p^{L+1} の位置関係を図 2 に示す．$Vc_i^{L,j}$ はパーティクル $p_i^{L,j}$ からその子 $p_i^{L+1,2j}$ への単位ベクトルである $(j = 0, 1, ..., 2^L - 1)$．p_i^L に対する $p_i^{L+1,j}$ の相対的な位置は $\pm(r_L \boldsymbol{c}_i^{L,j})$ となる．$p_i^{L+1,j}$ の半径 r_L と質量 m_L は

$$r_L = 2^{-L/3} r_0 \tag{4}$$

$$m_L = 2^{-L} m_0 \tag{5}$$

である．ここで，r_0，m_0 は SPH でのパーティクルの半径，質量である．サブパーティクルの質量の総和が元 (レベル 0) のパーティクルの質量となる．

子への単位ベクトル $Vc_i^{L,j}$ は，各サブパーティクルの回転速度が Kolmogorov の理論を満たすよう親の位置を通るベクトル $\boldsymbol{a}_i^{L,j}$ $(\perp \boldsymbol{c}_i^{L,j})$ を軸として回転させる．角速度 ω

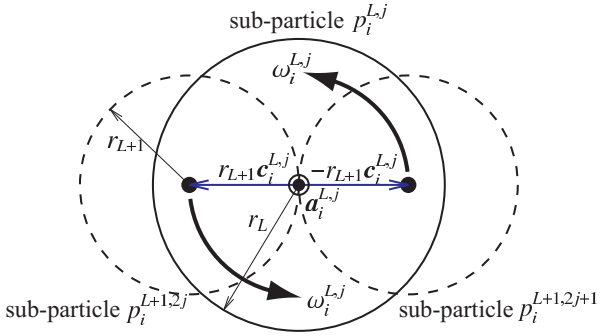


図 2: サブパーティクル p^L と p^{L+1} の位置関係

は，

$$|\hat{\boldsymbol{u}}(\frac{1}{2r_{L+1}}, \boldsymbol{x}_i)| = \sqrt{2\hat{e}(\frac{1}{2r_{L+1}}, \boldsymbol{x}_i)}$$

$$= \sqrt{2\hat{e}(\frac{1}{2r_0}, \boldsymbol{x}_i)(2^{-\frac{5}{9}})^{L_i}}, \tag{6}$$

$$\omega_i^{L,j} = \frac{|\hat{\boldsymbol{u}}(\frac{1}{2r_{L+1}}, \boldsymbol{x}_i)|\Delta t}{r_{L+1}}. \tag{7}$$

ウェーブレット解析で求めた各パーティクルの乱流エネルギーから，そのサブパーティクルのスケールでの乱流エネルギーを求めることで，サブパーティクルでの速度を決定する．ベクトル $\boldsymbol{a}_i^{L,j}$ を軸として角速度 $\omega_i^{L,j}$ で回転させる．ただし，三次元の場合，その回転軸は一意に決まらない．そこで，軸であるベクトル $\boldsymbol{a}_i^{L,j}$ を単位ベクトル $\boldsymbol{c}_i^{L,j}$ と垂直の状態を保ったまま，$\boldsymbol{c}_i^{L,j}$ を軸として毎ステップ角度 $\theta_i^{L,j}$ 回転させる．

$$\theta_i^{L,j} = \alpha\phi\omega_i^{L,j} \tag{8}$$

ϕ は $[-1.0, 1.0]$ の乱数，α は回転軸のぶれの大きさを決定する 0 以上の定数であり，ユーザーが設定する．以上の計算により，任意に設定したレベルまでパーティクルを分割することができる．

3.3 サブパーティクルを考慮した表面生成

表面の生成には Marching Cube 法 [6] を用いる．本研究では，レベルを考慮したカーネル関数 $W_{sub}(\boldsymbol{x}, L)$ を導入し，以下の関数に関して等値面を生成する．

$$\phi(\boldsymbol{x}) = \sum_i^N W_{sub}(\boldsymbol{x}_i, L_i) \tag{9}$$

W_{sub} はエネルギースペクトル値に基づき，用いるサブパーティクルを変化させるカーネルである．基準となるエネルギースペクトル値 e_{cri} を定数として設定し，以下の式を満

たすようなレベル L_i を求める．

$$e_{cri} = \hat{e}(\frac{1}{2r_L}, \boldsymbol{x})(2^{-\frac{5}{9}})^{L_i} \tag{10}$$

ここで，$\hat{e}(s, \boldsymbol{x})$ は位置 $\boldsymbol{x}$ でのスケール s でのエネルギースペクトルである．この式は，Kolmogorov の理論に従い，レベル 0 のパーティクルからどのレベルのサブパーティクルまでエネルギー遷移 (forward scattering) すれば，基準のエネルギースペクトルとなるかを計算する．式 (10) より使用するサブパーティクルレベル L_i を算出する．L_i の値により W_{sub} は以下のように決定される．ここで，分割するレベルの最大値を L_{max} とする．

$$W_{sub}(\boldsymbol{x}_i, L_i) =$$

$$\begin{cases} W(\boldsymbol{x}_i, r_0) & L_i \leq 0 \\ \sum_{j=0}^{2^{L_{max}}} W(\boldsymbol{x}_i^{L_{max},j}, r_{L_{max}}) & L_i \geq L_{max} \\ (L_i^{up} - L_i) \sum_{j=0}^{2^{L_i^{down}}} W(\boldsymbol{x}_i^{L_i^{down},j}, r_{L_i^{down}}) & \\ \quad +(L_i - L_i^{down}) \sum_{j=0}^{2^{L_i^{up}}} W(\boldsymbol{x}_i^{L_i^{up},j}, r_{L_i^{up}}) & \text{otherwise} \end{cases} \tag{11}$$

ここで，$L_i^{down} = \lfloor L_i \rfloor, L_i^{up} = \lfloor L_i \rfloor + 1$ である．エネルギースペクトルが大きい，つまり乱れが発生しやすい場所では大きなレベルまでサブパーティクル分割され細かい乱れを再現し，一方，エネルギースペクトルが小さい，つまり乱れが発生しにくい場所ではサブパーティクル分割のレベルは小さく，細かい乱れは現れにくくなる．陰関数 ϕ に対して，Marching Cube 法 [6] で流体表面を生成する．

4 結果

提案手法を実装した結果を示す．実行環境は，CPU:Corei7 2.93GHz, GPU:GeForce GTX580 である．手法のほとんどの部分は NVIDIA CUDA を用いて GPU 上で実装した．

図 3 は谷の中における流れをシミュレーションした結果である．パーティクル数は最大で 40,000，サブパーティクル最大分割レベル L_{max} は 3 とした．図 3(a) は SPH のみ，(b) は PS 乱流 [15]，(c) は提案手法による SPS 乱流を含めた結果である．(c) では (b) に比べて流れが激しいところでより細かな乱れが発生している．乱流を含む流体シミュレーションにかかった計算時間は約 15 ミリ秒/ステップである．また，Marching Cube 法によるメッシュ生成 (最大メッシュ数は約 80,000) には，(a),(b) について 15 ミリ秒/ステップ，SPS 乱流を含めた (c) については 120 ミリ秒/ステップかかった．PS 乱流を含めた (b) と (c) の間のシミュレーション時間にはほとんど変化がなく，計算時間の主な違いはメッシュ化のための陰関数 ϕ の計算に現れた．これは，SPS 乱流でのメッシュ化の際にはサブパーティクルを含んでおり，$L_{max} = 3$ の場合，最大でパーティクル数は 8 倍となるためである．

サブパーティクルの効果を検証するために，より単純な移動物体の後にできる後流をシミュレーションした結果を図 4 に示す．図 4(a) は SPH のみ，(b) は PS 乱流，(c) は提案手法による SPS 乱流，(d) は (b) のパーティクル数を 8 倍にした結果である．(a),(b),(c) におけるパーティクル数は約 23,000，(d) は約 184,000 である．サブパーティクル最大分割レベル L_{max} は 3 とし，(c) と (d) のレンダリングにおけるパーティクル数がほぼ同じとなるようにした．提案手法はパーティクル数を増やした場合と同じ程度の細かさの渦を再現できている．また，計算時間は (d) の 10 倍高速であった．しかし，(d) と比べて小さな渦が物体が離れた後も残っており，不自然な結果となった．これは，サブパーティクルにより forward scattering は再現されたものの，その逆のエネルギー遷移である backward scattering が考慮されていないことが原因と考えられる．PS 乱流では数値拡散によりこれらの現象が再現されていたのに対して，サブパーティクルでは原理上数値拡散が発生しない．また，粘性による渦エネルギーの拡散も考慮されていない．そのため，エネルギー遷移においてこれらの現象を取り入れる必要がある．

5 まとめと今後の課題

本論文では，サブパーティクル分割により，計算速度を抑えつつ，パーティクルスケール以下の乱れを再現可能な方法を提案した．提案手法は従来手法のパーティクルスケールでの乱流エネルギーからのエネルギー遷移によりより小さな渦が発生する現象を表現できた．しかしながら，発生した渦の生存時間が長く，流れが不自然に見える現象が観測された．これを解決するために，大きな渦へのエネルギー遷移および，粘性拡散を考慮する必要がある．また，従来手法に比べて高速化できているものの，まだ，リアルタイムで実行することは難しい．特に計算コストの大きい，サブパーティクルを含めた表面生成部分の高速化も今後の課題である．

(a) SPH のみ　(b) PS 乱流　(c) SPS 乱流 (提案手法)

図 3: 谷の中における流れ

参考文献

[1] R. Fedkiw, J. Stam, and H. Jensen. Visual simulation of smoke. In *Proc. SIGGRAPH 2001*, pp. 15–22, 2001.

[2] Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. Bubbles alive. In *Proc. SIGGRAPH 2008*, pp. 1–4, 2008.

[3] B.M. Kim, Y. Liu, I. Llamas, and J. Rossignac. Flowfixer: Using bfecc for fluid simulation. In *Eurographics Workshop on Natural Phenomena*, 2005.

[4] Doyub Kim, Oh young Song, and Hyeong-Seok Ko. A semi-lagrangian cip fluid solver without dimensional splitting. *Computer Graphics Forum (Proc. Eurographics)*, Vol. 27, No. 2, pp. 467–475, 2008.

[5] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. In *Proc. SIGGRAPH 2008*, pp. 1–6, 2008.

[6] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Computer Graphics (SIGGRAPH '87 Proceedings)*, Vol. 21, No. 4, pp. 163–169, 1987.

[7] Jeroen Molemaker, Jonathan M. Cohen, Sanjit Patel, and Jun yong Noh. Low viscosity flow simulations for animation. In *Proc. ACM/Eurographics Symposium on Computer Animation 2008*, 2008.

[8] Rahul Narain, Jason Sewall, Mark Carlson, and Ming C. Lin. Fast animation of turbulence using energy transport and procedural synthesis. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, pp. 1–8, New York, NY, USA, 2008. ACM.

[9] Tobias Pfaff, Nils Thuerey, Jonathan Cohen, Sarah Tariq, and Markus Gross. Scalable fluid simulation using anisotropic turbulence particles. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, pp. 174:1–174:8, New York, NY, USA, 2010. ACM.

[10] Tobias Pfaff, Nils Thuerey, Andrew Selle, and Markus Gross. Synthetic turbulence using artificial boundary layers. In *Proc. SIGGRAPH Asia 2009*, pp. 1–10, New York, NY, USA, 2009. ACM.

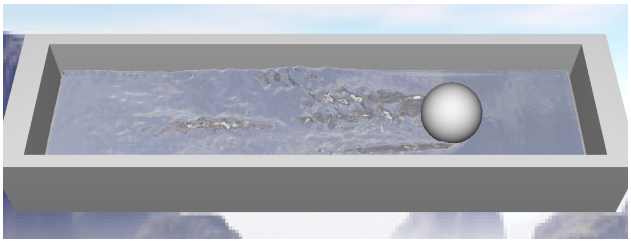
[11] Andrew Selle, Nick Rasmussen, and Ronald Fedkiw. A vortex particle method for smoke, water and explosions. In *Proc. SIGGRAPH 2005*, pp. 910–914, 2005.

[12] Jos Stam. Stable fluids. In *Proc. SIGGRAPH 1999*, pp. 121–128, 1999.

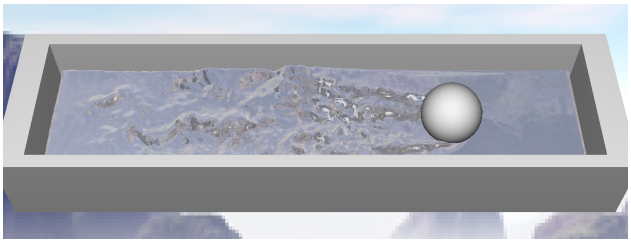
[13] Jos Stam and Eugene Fiume. Turbulent wind fields for gaseous phenomena. In *Proc. SIGGRAPH1993*, pp. 369–376, 1993.

[14] Wladimir J. van der Laan, Simon Green, and Miguel Sainz. Screen space fluid rendering with curvature flow. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, I3D '09, pp. 91–98, New York, NY, USA, 2009. ACM.

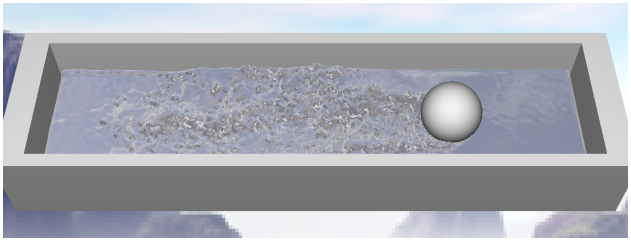
[15] 藤澤誠, 加藤博一. 粒子法とウェーブレット解析によるリアルタイム乱流シミュレーション. グラフィックスと CAD/Visual Computing 合同シンポジウム 2010 予稿集, 2010.



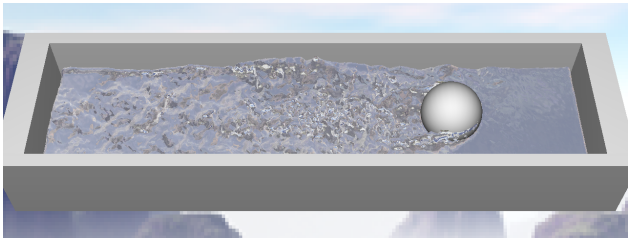
(a) SPH のみ



(b) PS 乱流



(c) SPS 乱流 (提案手法)



(d) PS 乱流 (パーティクル数 8 倍)

図 4: 移動固体の後流