

SPH法による高速な2D-3Dハイブリッド流体シミュレーション

Fast 2D-3D Combined Fluid Simulation by SPH

仲田拓也 † 藤澤誠 † 三河正彦 †
Takuya NAKADA †, Makoto FUJISAWA † and Masahiko MIKAWA †

† 筑波大学 † University of Tsukuba

1 はじめに

近年、自然現象をコンピューターグラフィックスで再現するために物理シミュレーションが用いられている。特にゲームなどのインタラクティブなアプリケーションでは、高精細かつリアルタイムな計算速度が求められているが、流体シミュレーションは計算コストが高く、特に川や湖や海などの大規模な流体シミュレーションをリアルタイムで計算することは難しい。このような流体シミュレーションを実現するための方法の一つに Shallow Water モデルによる流体シミュレーションがある。この方法は3次元流体の運動計算を2次元の流体表面の運動計算だけで近似することで、計算コストを減らした方法であり高速に計算が可能である。しかし、流体表面の上下運動しか扱えないため、しぶきや大きな波が作り出すトンネル形状など、水が覆いかぶさるような3次元的な挙動を表現できないということが問題となる。

本論文では2次元のパーティクル法を用いた Shallow Water モデルの波のシミュレーションに対して、3次元的な動きをパーティクル法の一つである SPH (Smoothed Particle Hydrodynamics) 法により追加することで、より高精細で高速な流体シミュレーションを実現する方法を提案する。提案手法では Shallow Water モデルにより、大規模なシーンでも高速に計算可能であり、かつ、従来手法では実現できなかった3次元的な挙動を3次元の SPH 法と組み合わせることで実現する。

2 関連研究

リアルタイム流体シミュレーションを実現するために、さまざまな手法が提案されている。計算コストを減らすために、3次元の流体シミュレーションの代わりに、水面を表す2次元高さ場を用いる方法がある。Kass ら [9] は波動方程式により高さ場を動かすことでリアルタイムシミュレーションを実現した。しかしこの手法は運動量が考慮されていないため、物理的な表現とはなっていない。Layton ら [10] はナビエ・ストークス方程式に基づく速度場を用いて、2次元のグリッド法により Shallow Water Equation (以下 SWE) を解く方法を提案した。Solenthaler ら [16] はパーティクルを用いて SWE を解くシミュレーション手法を提案した。しかし、Shallow Water モデルは水面のシミュレーション

であるため、流体が覆いかぶさるような3次元的な挙動を表現することができない。Chentanez ら [3] はグリッド法による Shallow Water モデルに対して、しぶきなどの3次元的な挙動がおこる箇所を検出し、その箇所にパーティクルを生成することにより、高精細なリアルタイム流体シミュレーションを実現した。また Thürey ら [18] は波が覆いかぶさる部分を検出し、その箇所にメッシュを生成、そのメッシュを移流させることで波が覆いかぶさる3次元的な挙動を実現した。この方法では波の覆いかぶさり綺麗に表現できるが、波が垂直に上がったときに起こるしぶきなどは表現できない。Chentanez ら [4] は Shallow Water モデルに3次元のグリッド法を組み合わせることにより高精細なしぶき表現を実現した。3次元グリッド法シミュレーションにより様々な現象を表現できるようになったものの、シーンの大きさがグリッドに制限されるため、大規模なシーンの一部に3次元的な挙動が発生するような場合に有効な方法である。一方で水面全体に多くのしぶきなどが発生する場合はグリッドを大きくするか、ユーザが手動で各箇所にグリッドを設定する必要がある。本論文では Solenthaler ら [16] の2D パーティクル法による SWE と一般的な3D パーティクル法を組み合わせることで、シミュレーション空間の拡張性が高く、かつ、波の覆いかぶさり、しぶきなどの3次元的な挙動を実現可能な手法を提案する。水面と3次元的な挙動の両方がパーティクルで表現されるため、運動計算やレンダリングの際に親和性が高く、違和感のないシミュレーションが可能となる。また3D パーティクルの運動量を保存するように、2D パーティクルとの相互作用を計算することで現実的な挙動を実現している。

3次元空間での流体シミュレーション手法として、オイラー的手法の一つであるグリッド法 [2] とラグランジュ的手法の一つであるパーティクル法についての研究が盛んに行われている。パーティクル法の中でも SPH 法 [6] は実装の容易さ、計算空間の拡張性などの理由から、コンピューターグラフィックス分野で非常に良く用いられている [14, 8]。SPH 法を用いた流体シミュレーションにおいて、流体の質量保存はパーティクル数が増えない限り満たされる。しかしパーティクル同士が重なり合うことで、流体が圧縮してしまうという問題点がある。この問題を解決するために WCSPH 法 [1] や PCISPH 法 [17] といった非圧縮性を保つ手法が提案されている。Macklin ら [12] は初期密度を保つ

ような拘束条件から、パーティクルの位置を直接修正することにより高い非圧縮性を保つ手法を提案した。この手法は非圧縮性を保ちつつインタラクティブな速度を実現可能であるため、本論文では3D パーティクルの運動計算にこの手法を用いる。

3 提案手法

本論文では Shallow Water モデルを用いて2次元のパーティクル法により波のシミュレーションを行う。この2D シミュレーションでは、砕け波やしぶきなど3次元的な動きは再現できない。そこで2D シミュレーション結果をもとに3次元 SPH 法のためのパーティクルの生成および削除を行い、その運動計算を行うことでこれらの現象を追加する。最後に両シミュレーションで用いたパーティクルをあわせてレンダリングする。これらを各タイムステップ毎に実行する。以降の節では2次元 Shallow Water シミュレーションに用いるパーティクルを2D パーティクル、3次元 SPH 法に用いるものを3D パーティクルとして処理の詳細を説明していく。

3.1 2D パーティクルによる

Shallow Water シミュレーション

SWE によるシミュレーションでは、シミュレーション空間をグリッド分割し、グリッドごとに定義された高さ場を、速度場に従って変化させることでシミュレーションを行う [10]。SWE はナビエ・ストークス方程式に基づいた式であり、水面の運動を表している式である。SWE を解くことで、流体の速度場を更新し、その速度をもとに高さ場を変化させ、シミュレーションを行う。SWE は流体の質量と運動量の保存を表しており以下のように記述できる。

$$\frac{\partial h}{\partial t} = -\nabla \cdot (h\mathbf{u}) \quad (1)$$

$$\frac{D\mathbf{u}}{Dt} = -g\nabla(h + H(\mathbf{x})) + \mathbf{a}_{ext} \quad (2)$$

ここで h は水深、 g は重力、 $H(\mathbf{x})$ は位置 \mathbf{x} における地形(水底)の高さ、 \mathbf{a}_{ext} は外力を表す。

本論文ではしぶきなどの3次元的な挙動を3D パーティクルで表現するため、パーティクルを用いた SPH 法による Shallow Water シミュレーション [16] を用いる。2次元の SPH 法により Shallow Water Equation (以下 SWE) を解くことで全体的な動きを求める。SPH 法による SWE シミュレーションでは、グリッド毎に定義される高さ場の代わりに、図1のように各パーティクルに高さを定義し、以下の式を解くことで各パーティクルの高さを更新する。

$$h_i = \frac{\rho_i^{2D}}{\rho_0} \quad (3)$$

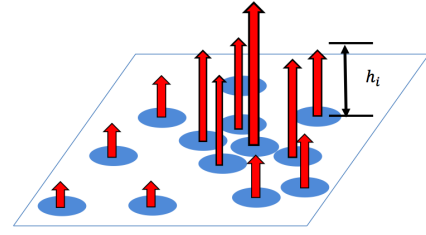


図 1: パーティクルに対する高さ定義

h_i , ρ_i^{2D} はそれぞれパーティクル i の水底からの高さと密度、 ρ_0 は水の初期密度である。重力は y 軸(鉛直方向)に対して加わると仮定し、2D パーティクルは xz 平面上に配置され、 xz 平面内での2次元的な動きから高さを求める。パーティクル密度 ρ_i^{2D} は2次元の SPH 法における密度計算式を用いて以下のように計算する。

$$\rho_i^{2D} = \sum_{j \in N} m_j^{2D} W_{\text{poly6}}(\mathbf{x}_j - \mathbf{x}_i, l^{2D}) \quad (4)$$

ここで N は有効半径 l^{2D} 内の近傍 2D パーティクル集合、 m_j^{2D} は 2D パーティクル j の質量、 $W(\mathbf{x}_j - \mathbf{x}_i, l^{2D})$ は重み関数である。式 (3) により式 (2) を書き換えることで、以下のようなパーティクル法のための SWE が得られる。

$$\frac{\partial \mathbf{u}_{xz}^{2D}}{\partial t} = -\frac{g}{\rho^{3D}} \nabla \rho_i^{2D} - g\nabla H(\mathbf{x}) + \nu \nabla^2 \mathbf{u}_{xz}^{2D} + \mathbf{a}_{ext} \quad (5)$$

\mathbf{u}_{xz}^{2D} は 2D パーティクルの xz 方向の速度、 $\nu (= \mu/\rho)$ は動粘性係数である。右辺第 3 項は安定化のために加えた粘性項 [11] である。式 (5) により 2D パーティクルの速度 \mathbf{u}_{xz}^{2D} と位置 \mathbf{x}_{xz}^{2D} をタイムステップ幅 Δt ごとに更新する。また $x_y^{2D} = h_i + H(\mathbf{x}_{xz}^{2D})$ とする。SPH 法による SWE シミュレーションでは、パーティクルの位置が変化することにより、密度が変わり、密度の変化を式 (3) で高さに変換することで水面のシミュレーションが行える。このとき、2D パーティクル運動の安定化のために最大速度を、波動方程式から求めた長波における水の伝播速度 $\sqrt{gh_i}$ に制限する。

SPH 法ではシミュレーション境界や固体境界において近傍パーティクルが少なくなり、境界付近の粒子の密度が小さくなってしまふ。結果として密度が小さい粒子に他の粒子が集まり圧縮してしまうという問題点がある。そこでシミュレーション境界と固体内部に物理量の計算に用いる境界パーティクルを配置することで、境界付近で粒子の密度が小さくなることを防ぐ。さらにシミュレーション境界に流体パーティクルが接触した際に速度を 0 とすることで固体境界とのインタラクションを行う。

3.2 3D パーティクルの生成と運動計算

Shallow Water モデルでは波の覆い被さりやしぶきなどの3次元的な現象を表現することができない。このような

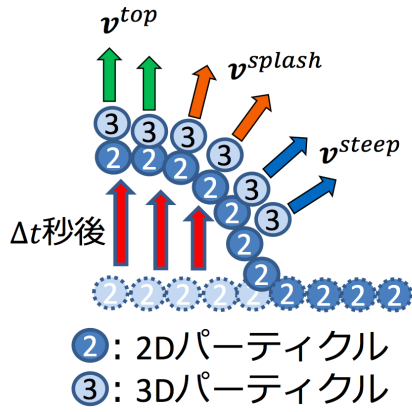


図 2: 3D パーティクルの分類と初速度の方向

現象を表現するために 3D パーティクルを生成する。これらの現象は、2D パーティクルの水深 h_i が大きく増加する、つまり水面が上昇する際に起こると考え、2D パーティクルが以下の条件式を満たすとき、3D パーティクルを生成する。

$$\frac{h_i - h_i^{prev}}{\Delta t} > \gamma_{3D} \quad (6)$$

h_i^{prev} は前ステップの 2D パーティクルの水深、 γ_{3D} は 3D パーティクル生成閾値である。式 (6) の条件を満たしたパーティクルを、近傍パーティクル j の最大高低差 $h_{diff} = \max(h_j) - \min(h_j)$ に応じて分類分けする。ここで $\max(h_j)$ は近傍 2D パーティクルの最大の高さ、 $\min(h_j)$ は近傍 2D パーティクルの最小の高さである。図 2 に示すように、3D パーティクルを h_{diff} の値により Top, Splash, Steep パーティクルに分類する。 $h_{diff} \leq \gamma_{top}$ の時 Top パーティクルを生成、 $\gamma_{top} < h_{diff} < \gamma_{steep}$ の場合 Splash パーティクルを生成、 $\gamma_{steep} \leq h_{diff}$ である時 Steep パーティクルを生成する。ここで $0 \leq \gamma_{top} \leq \gamma_{steep}$ である。

3D パーティクルの初期位置は分類によらず以下とする。

$$\mathbf{x}_{xz}^{3D} = \mathbf{x}_{xz}^{2D} \quad (7)$$

$$x_y^{3D} = (1 - \alpha_{height})h_i^{2D} + \alpha_{height} \max(h_j) \quad (8)$$

α_{height} は 3D パーティクルの生成高さを調整する係数である。本論文では、 $\alpha_{height} = 0 \sim 0.3$ を用いている。

それぞれのパーティクルの説明と初速度の計算方法を以下に示す。

- Top パーティクル

h_{diff} が小さい時、タイムステップの間に高さが大きく変化したものの、近傍の水面形状が平らになっている、つまり鉛直方向に上がった波の先端部分であると考え、このとき波の先端からしぶきが鉛直上方向に飛び出すとして、3D パーティクルの速度を以下のように計算する。

$$\mathbf{u}_{xz}^{top} = 0, \quad u_y^{top} = k_{top} \frac{h_i - h_i^{prev}}{\Delta t}$$

ここで k_{top} は Top パーティクルの速度係数である。

- Steep パーティクル

h_{diff} が大きい時、近傍の水面形状が斜めになっており、波の側面部分であると考えられる。この場合は覆い被さるような波を表現するために 3D パーティクルに斜めに飛び出す速度を以下のように設定する。

$$\mathbf{u}_{xz}^{steep} = \frac{a_{xz}^{2D}}{|a_{xz}^{2D}|} \sqrt{gh_i}, \quad u_y^{steep} = k_{steep} \frac{h_i - h_i^{prev}}{\Delta t}$$

ここで k_{top} は Top パーティクルの速度係数、 a_{xz}^{2D} は 2D パーティクルの xz 方向の加速度のベクトルである。 $\sqrt{gh_i}$ は波動方程式から求めた、長波における水の伝播速度である。

- Splash パーティクル

Splash パーティクルは Top パーティクルと Steep パーティクルの間にあり、両パーティクルの速度を線形に補間することで、初速度を計算する

$$\mathbf{u}_{xyz}^{3D} = \alpha_{splash} \mathbf{u}_{xyz}^{steep} + (1 - \alpha_{splash}) \mathbf{u}_{xyz}^{top}$$

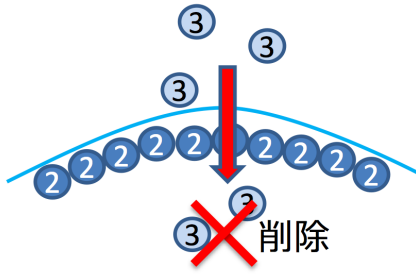
ここで α_{splash} は Splash パーティクルの補間係数であり、本論文では $\alpha_{splash} = h_{diff} - \gamma_{top} / \gamma_{steep} - \gamma_{top}$ とする。

3D パーティクルの運動には Position Based Fluids[12] を用いる。この手法はナビエ・ストークス方程式の圧力項を計算する代わりに、位置を繰り返し修正することで高速かつ高い非圧縮性を保つ位置ベース手法 [13] である。高い非圧縮性により、パーティクル間隔が保たれ、3D パーティクルと 2D パーティクルを合わせてレンダリングする際に違和感のないメッシュ形成が可能となる。

一方でナビエ・ストークス方程式の圧力項及び質量保存式を位置ベース法で置き換える形になるため、ナビエ・ストークス方程式に基づく SWE とは異なる近似に基づくことになってしまう。質量保存式 (SWE では式 (1)) に関しては 2D, 3D とともにパーティクル法で近似しているため問題ないと考えられる。また、SWE における圧力項は圧力 p に関して $p = g\rho h$ とし、流体の圧力が均一になろうとする力を高さが均一となろうとする力で近似している [10]。 h を y 軸方向の位置と考えれば、位置ベース法と同様であるとも考えられるが、これによる影響に関しては、グリッド法ではあるが高さ方向に伸ばしたグリッドセルを用いることで 2D, 3D 両者とも同じ近似を用いた方法 [5] などと比較し、検討する必要がある。

3.3 3D パーティクルの削除

xz 平面上において、3D パーティクルの y 座標値が 2D パーティクルの近傍半径 l^{2D} 以内に存在する全近傍 2D パー



- ② : 2Dパーティクル
- ③ : 3Dパーティクル

図 3: 3D パーティクルの削除条件

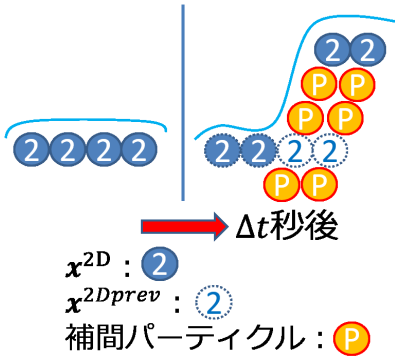
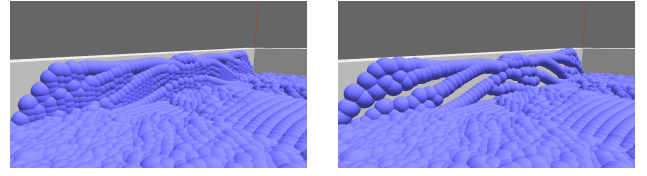


図 4: 補間パーティクルの配置例

パーティクルの y 座標値よりも低い場合、3D パーティクルの削除を行う (図 3)。削除の際に近傍半径 l^{2D} 以内に存在する近傍 2D パーティクルに対して、3D パーティクルの xz 方向の運動量と y 方向の運動量を保存するように 2D パーティクルの速度を以下の式 (9) で更新する。

$$\mathbf{u}_{xz}^{2D} = \frac{m_i^{2D} \mathbf{u}_{xz}^{2D} + \frac{m_i^{3D}}{N_{neigh}} \left(\mathbf{u}_{xz}^{3D} + |u_y^{3D}| \frac{\mathbf{r}_{xz}}{|\mathbf{r}_{xz}|} \right)}{m_i^{3D}} \quad (9)$$

N_{neigh} は近傍 2D パーティクルの個数、 \mathbf{r}_{xz} は xz 平面における 3D パーティクルから近傍 2D パーティクルへの相対位置を表している。 $m_i^{2D} \mathbf{u}_{xz}^{2D}$ は元の 2D パーティクルの運動量であり、これに 3D パーティクルの運動量を加える。3D パーティクルの運動量 $m_i^{3D} \mathbf{u}^{3D}$ を近傍 2D パーティクルに均等に分散させるために、質量 m_i^{3D} を近傍 2D パーティクルの個数 N_{neigh} で割っている。3D パーティクルの xz 方向の運動量 ($m_i^{3D} \mathbf{u}_{xz}^{3D} / N_{neigh}$) は、2D パーティクルの xz 方向の運動量としている。3D パーティクルの y 方向の運動量は 2D パーティクルを押しつける力として、2D パーティクルとの相対位置方向に変換している。



補間あり

補間なし

図 5: パーティクル補間

3.4 レンダリング

本論文では Screen Space Meshes (以下 SSM) [15] を用いてメッシュ生成を行う。この手法は 2D 空間でメッシュを生成するので計算コストが低く、さらに提案手法ではシーン中の大部分において、表面のみにしかパーティクルがないためこの方法を用いる。

SSM を用いた場合、2D パーティクルの動きの差が大きい場合に 2D パーティクルが疎になり、メッシュに穴がいてしまう。そこでレンダリングの際に図 4 に示すようにパーティクルを補間する。補間パーティクルは $\gamma_{ghost} < |\mathbf{x}_{xyz}^{2D} - \mathbf{x}_{xyz}^{2Dprev}|$ の条件を満たすとき生成する。 $\mathbf{x}_{xyz}^{2Dprev}$ は前ステップの 2D パーティクルの位置、 γ_{ghost} は補間パーティクル生成閾値である。補間パーティクルは前ステップの 2D パーティクルの位置と現在の 2D パーティクルの位置を結ぶ直線上に配置する。図 5 にパーティクル補間の有無によるシミュレーション結果の違いを示す。

4 実験結果と考察

本論文を CPU 上で実装し、5 つのシーンで実験を行った。実行環境は CPU: Intel Core i5-3470 3.20GHz, GPU: NVIDIA GeForce GTX 680 である。シミュレーション、SSM はすべて CPU で実行され、近傍パーティクル探索にはハッシュを用いた空間分割法 [7] を用いた。スクリーンサイズは $1,280 \times 720$ 、タイムステップ幅 $\Delta t = 0.005$ としている。

図 6 に実行結果のスクリーンショットを、各シーンにおけるパラメータを表 1、計算時間を表 2 に示す。3D パーティクル数が多くなるほど計算コストが高くなっている事が分かる。図 6(a),(b) に示すように壁で水が跳ね返る際に水が覆い被さる現象、中央付近でぶつかる際にしぶきがある現象が再現できており、3D パーティクルの生成の際の初期位置、初速度の場合分けが有効であると考えられる。一方で 2D パーティクルが急激に集まった際に密度が大幅に上昇し、現実と比べて大きな水柱が生じる現象が確認された。この問題点の対策としては、2D パーティクルに対して、高さ変化の上限を設定することが有効であると考えられる。図 6(c),(d) ではそれぞれ垂直方向と斜め方向の速度を持つ水滴が落下し、それぞれしぶきがあがっている様子が確認でき、垂直方向、水平方向の運動量保存による効

表 1: シーンごとのパラメータ

	シーン 1	シーン 2	シーン 3	シーン 4	シーン 5
3D パーティクル高さ係数 α_{height}	0.3	0.3	0.3	0	0.3
3D パーティクル生成閾値 γ_{3D}	3	5	1.5	3	4
Top パーティクル生成閾値 γ_{top}	0.1	0.13	0.15	0.07	0.1
Steep パーティクル生成閾値 γ_{steep}	0.15	0.15	0.2	0.15	0.2
Top パーティクル速度係数 k_{top}	0.3	0.3	0.6	0.5	0.1
Steep パーティクル速度係数 k_{steep}	0.2	0.1	0.2	0.5	0

表 2: シーンごとの実行結果

Scene	2d partices	Max. 3d particles	Avg. time(fps)	Max. time(fps)
シーン 1	13,700	24,774	1.82	0.92
シーン 2	11,200	12,817	4.29	1.87
シーン 3	10,800	8,000	4.31	2.59
シーン 4	10,800	8,000	4.72	2.70
シーン 5	24,600	4,264	5.67	3.62

果が見て取れる。図 6(e) では複雑な地形においても、3D パーティクルによりしぶきがあがることで、ある程度の現実的な振る舞いが観測できた。しかし複雑な地形においては、3D パーティクルの挙動が不自然なものとなる場合がある。これは、様々な水の流れが合わさることで、適切な初速度が設定できていないことが原因であると考えられる。

5 まとめと今後の課題

本論文では2次元のパーティクル法を用いた Shallow Water モデルによる波のシミュレーション結果に対して、3次元のパーティクルを追加することで、3次元的な挙動を表現できるシミュレーション手法を提案した。3D パーティクルを発生時の状況により3種類に分類し、それぞれの運動を決定することで様々な表現を可能にした。そして3D パーティクルの運動量を保存することで、現実的な振る舞いを可能にした。実験では、様々な状況において本論文が有効であることを示した。

今後の課題としてはメッシュ形成の問題があげられる。本論文ではSSMを用いているが、パーティクルが疎である箇所ではパーティクル補間を用いてもまだ穴があいてしまうという問題がある。この問題を解決するための単純な方法は、補間パーティクルを増やすことである。しかし、適切な位置にパーティクルを増やさなければ、本来と異なる水面が形成されてしまうので、生成位置をどのように求めるかを考える必要がある。しぶきの生成条件に対する考察も不十分であり検討する必要がある。本論文では、閾値による生成、場合分けを行っているが、不自然な挙動のものがあつた。この問題を解決するために、閾値を動的に変更

するなどの対策が必要となる。また、3D パーティクルが削除する際は運動量保存を考慮したが、逆に3D パーティクル生成時の運動量保存や生成/削除時の質量保存は考えられていない。特に質量保存が考えられないため、3D パーティクルがどんなに追加されても2D パーティクルの全体的な水面の高さが変化せず、水を注いで水面が上昇するようなシーンには対応できない。この問題を解決するために3D パーティクル生成/削除時に2D パーティクルとの間で質量交換を行う必要がある。最後にGPUを用いた並列計算により高速化を行い、リアルタイムでのシミュレーションを実現したい。

参考文献

- [1] Markus Becker and Matthias Teschner. Weakly compressible sph for free surface flows. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 209–217, 2007.
- [2] R. Bridson. *Fluid Simulation for Computer Graphics*. A K Petters, 2008.
- [3] N. Chentanez and M. Müller. Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2010.

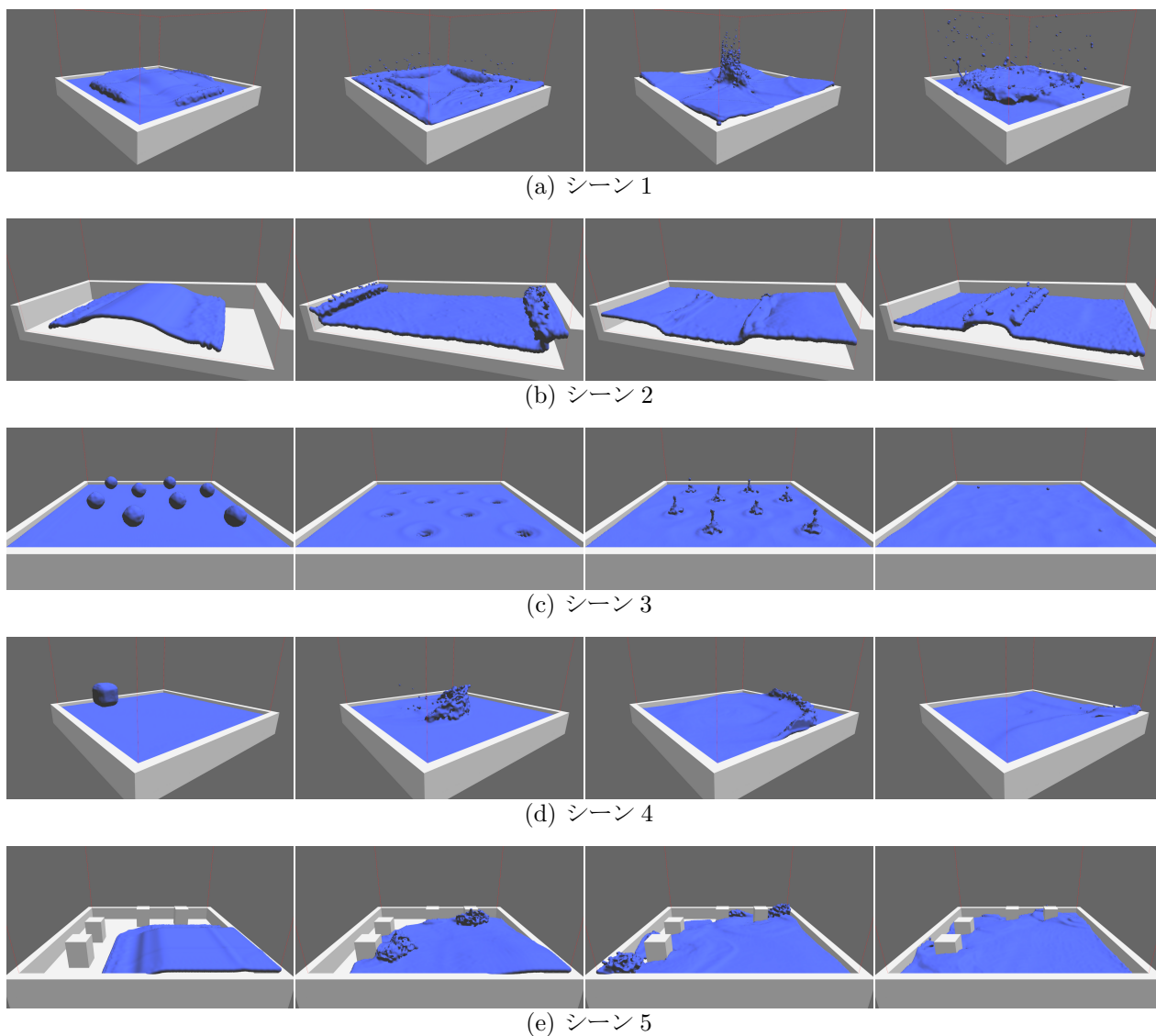


図 6: それぞれのシーンの実行結果

- [4] N. Chentanez, M. Müller, and T.-Y. Kim. Coupling 3d eulerian, heightfield and particle methods for interactive simulation of large scale liquid phenomena. In *Proceedings of the 2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2014.
- [5] Nuttapon Chentanez and Matthias Müller. Real-time eulerian water simulation using a restricted tall cell grid. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pp. 82:1–82:10, New York, NY, USA, 2011. ACM.
- [6] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, Vol. 181, pp. 375–389, 1977.
- [7] S. Green. Cuda particles. Technical report, NVIDIA Whitepaper, 2008.
- [8] Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. Sph fluids in computer graphics. In *Eurographics 2014 - State of the Art Reports*, pp. 21–42, 2014.
- [9] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *Comput. Graph. (Proc. of SIGGRAPH 90)*, Vol. 24, pp. 49–57, 1990.
- [10] A. T. Layton and M. Van De Panne. A numerically efficient and stable algorithm for animating water waves. *The Visual Computer*, Vol. 18, No. 1, pp. 41–53, 2002.
- [11] H. Lee and S. Han. Solving the shallow water equations using 2d sph particles for interactive applica-

tions. *The Visual Computer*, Vol. 26, No. 6-8, pp. 865–872, 2010.

- [12] M. Macklin and M. Müller. Position based fluids. *ACM Trans. Graph.*, Vol. 32, No. 4, pp. 104:1–104:12, 2013.
- [13] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, Vol. 18, No. 2, pp. 109–118, 2007.
- [14] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proc. ACM/Eurographics Symposium on Computer Animation 2003*, pp. 154–159, 2003.
- [15] Matthias Müller, Simon Schirm, and Stephan Duthaler. Screen space meshes. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 9–15, 2007.
- [16] B. Solenthaler, P. Bucher, N. Chentanez, M. Müller, and M. Gross. Sph based shallow water simulation. In *Proceedings of Virtual Reality Interactions and Physical Simulations (VRIPhys)*, pp. 39–46, 2011.
- [17] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible sph. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pp. 1–6, New York, NY, USA, 2009. ACM.
- [18] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross. Real-time breaking waves for shallow water simulations. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*, 2007.