情報処理学会論文誌 Vol.58 No.7 1323–1334 (July 2017)

Slaveパーティクルを用いた弾塑性体シミュレーション

斉田 智也[1] 藤澤 誠[1,a)] 三河 正彦[1,b)]

受付日 2016年10月5日, 採録日 2017年4月10日

概要：本研究では，粒子法における弾塑性体の安定したシミュレーション手法を提案する．提案手法では，物質を表現する粒子（Master パーティクル）とは異なる粒子（Slave パーティクル）を用いて，変形勾配とそれによって生じる力の計算点をずらすことにより，弾塑性変形における計算の安定化を実現する．さらに，Master パーティクルに追従する形で Slave パーティクルを移流させ，追加・削除することで，塑性変形による分離・結合などの変形にも対応させた．最終的に，提案手法を用いることで完全な弾性体から塑性体までさまざまな物体をコンピュータグラフィクスで表現できることを実験で示す．

キーワード：物理シミュレーション，パーティクル法，弾塑性体

Particle-based Elastoplastic Simulation with Slave Particles

Tomoya Saida[1] Makoto Fujisawa[1,a)] Masahiko Mikawa[1,b)]

Received: October 5, 2016, Accepted: April 10, 2017

Abstract: We propose a particle-based stable simulation method for elastoplastic materials. Our method uses slave particles in addition to the master particles used to represent the object Slave particles allow for stable calculation of the deformation gradient for elasto-plasticity because they are initially placed at positions different from those of master particles and prevent zero-energy mode in which the deformation energy become zero when the distribution of the computational points has locally symmetric property. Slave particles are advected by the motions of nearby master particles. Further, by adding and removing slave particles, it is possible to treat deformation with plastic splitting and/or merging. By comparison with experimental results, we show that the proposed method can simulate a wide variety of materials, from purely elastic to elastoplastic.

Keywords: physics simulation, particle-based method, elastoplastic

1. はじめに

CG アニメーションにおいて，粘土やチューインガム，クリームといった弾塑性体と呼ばれる物質を扱うことがある．これら弾塑性体は，変形の度合いによって弾性変形から塑性変形へ変化するという特徴がある．ここで，弾性変形とは変形後に力を加えない状態にすると完全に元の形状に戻る変形であり，ポアソン比やヤング率といった物性値によってその変形表現を制御することができる．一方，塑性変形では変形後に力を加えない状態にしても形状が元に戻らず変形した状態を保つものであり，一般的に変形量が降伏点と呼ばれる点を超えることで，弾性変形から塑性変形へ遷移する．

弾塑性体の運動計算を有限要素法などの数値計算手法を用いて解くときに，メッシュ構造を用いて離散化する方法がよく用いられる．メッシュを用いることで物体構造を安定して保持することができるが，変形や分離・結合によってその構造を再構成する必要があり，その計算に多くの時間が必要となる．一方で，このようなメッシュ再構成を必要としない方法としてパーティクル法がある．パーティクル法ではメッシュのような接続情報を持たない点の集合によって物体を離散化するため，メッシュ構造で問題となっていた再構成のための計算が必要なくなる．しかし，パーティクルのみを用いて計算を行った場合，変形の様子によってゼロエネルギーモードになりやすく計算が不安定に

[1] 筑波大学
University of Tsukuba, Tsukuba, Ibaraki 305–8550, Japan
[a)] fujis@slis.tsukuba.ac.jp
[b)] mikawa@slis.tsukuba.ac.jp

なるという問題点がある．

近年では，パーティクルのみを用いた場合の計算の不安定化を防ぐためにパーティクルとグリッドを組み合わせて計算する方法がさかんに研究されている．パーティクルでは不安定になってしまう変形計算を，グリッドで計算することで不安定になることを防ぐことができる．しかし，グリッドを用いることになるため，計算空間の大きさによって計算コストが増大してしまい，また空間の拡張性や実装のシンプルさといったパーティクル法の特徴も失われてしまう．

本研究ではグリッドの代わりに応力計算点として Slave パーティクルを用いることで，計算空間に制限のない安定した計算手法を提案する．提案手法は，物質を表現する粒子とは別に計算用の粒子を生成することでグリッドを用いるのと同じように計算点をずらすことができ，パーティクルのみでも安定して計算することを可能とし，さらにすべてパーティクルだけで計算することで，定義されたグリッド内だけに計算空間が制限されることがないという利点を持つ．また，たとえば，パーティクル 1 層だけで構成されるような薄い物体など，物質を表現するパーティクルのみを用いた従来手法では実現されてない幅広い弾塑性変形表現も可能となる．

2. 関連研究

2.1 弾塑性体に関する関連研究

弾塑性体に関する初期の研究としては Terzopoulos ら [20] の研究や Terzopoulos ら [21] の研究があり，現在まで弾塑性体についての研究はさかんに行われてきており，計算方法としてメッシュ法 [9] やパーティクル法 [12], [14], [16] が用いられてきた．

メッシュ法の場合は有限要素法を用いることで，幅広い表現を安定してシミュレーションすることができる．しかし，塑性変形による分離や結合が発生したときにメッシュ構造を変えなければならないという問題点があり，この問題に対してさまざまな手法が提案されている [1], [23], [24], [25]．パーティクル法を用いた場合，構造変化に対して特別な処理を必要としないが計算を局所的に行うために計算が不安定になりやすく，幅広い表現を安定して計算することは難しい [5], [11]．

Zhou ら [26] は変形計算を陰的に行うことで，パーティクル法のみで安定かつ幅広い弾性体表現を実現したが，パーティクルが平面上や線上に並んでできるような形状では不安定になり，計算可能な物体の形状に制限がある．さらに Stomakhin ら [17], [18] は MPM というパーティクル法とグリッド法を組み合わせた手法を用いることで，さまざまな表現が可能であることを示した．しかし，グリッドは計算空間全体に計算点を配置するため計算空間の大きさが制限されてしまうという問題点がある．そこで，グリッドの代わりに応力計算点を導入することで計算空間に制限されずに，安定かつさまざまな表現が可能な計算手法を提案する．

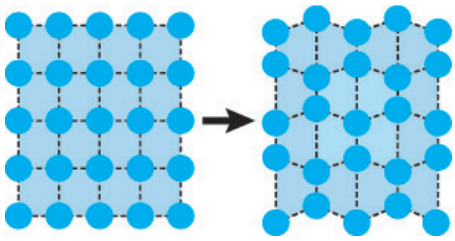


図 1 ゼロエネルギーモードの例．右図においてオブジェクトに変形が生じているにもかかわらず変形エネルギーはゼロになってしまう

Fig. 1 Example of zero energy mode. In right figure, the object is actually deforming but the deformation energy becomes zero.

2.2 応力計算点を用いた関連研究

変形する物体を数値解析するときに，**図 1** に示すように，計算点の位置関係によって変形しているにもかかわらず変形エネルギーがゼロとなってしまうゼロエネルギーモードという問題がある．たとえば，図 1 右では中心の計算点からすると周囲の点の分布が左右対称になっており，変形勾配がゼロと計算されてしまう．この問題に対して，Dyka ら [3], [4] はすでに定義した計算点とは別の位置に応力計算点を配置し，応力計算点の位置で変形エネルギーを計算することでゼロエネルギーモードを防ぐことができる手法を提案した．応力計算点を用いることで，1 次元物体および 2 次元物体について安定した計算が実現できることが分かっている [22], [27] が，3 次元の弾性体に対して適用した例は我々の調べた限りではない．He ら [8] はパーティクル法の一種である SPH 法に対して圧力計算点をずらすことでゼロエネルギーモードを抑制し，安定した計算が可能となるスタガードパーティクル法を提案した．彼らの方法は 3 次元流体を対象としたものであり，本研究で扱う弾塑性体とは異なる．また，パーティクル間に圧力計算点を配置するため，パーティクル 1 層で構成されるような薄い物体を扱うことも難しい．

本論文では，3 次元空間において応力計算点として物質を表現するパーティクルとは別のパーティクルを配置し，2 種類のパーティクルを用いて変形に関する計算を行う弾塑性体シミュレーション手法を提案する．応力計算点を元の形状の外側にも配置し，元のパーティクルに従って移流させることで，薄い物体を含むさまざまな形状に対応させる．また，応力計算点の移流にともない，大きな変形に対して追加・削除を行うことで，分離や結合を含むさまざまな変形の表現を可能とする．

3. 提案手法

図 2 左に示したように計算安定化のためにパーティクルとグリッドを使う方法として，PIC（Particle-In-Cell）[7]，FLIP（FLuid-Implicit Particle）[2]，MPM（Material Point Method）[19] などがある．提案手法ではこれらとは異なり，図 2 右に示したように 2 種類のパーティクルを用いることで計算空間の拡張性と安定性を両立させる．

以下，論文中では物質を構成するパーティクルを Master パーティクル，応力計算点用のパーティクルを Slave パーティクルと呼ぶ．また，Master パーティクルの要素については添え字 m を用いて X_m と表記し，Slave パーティクルの要素についても添え字 s を用いて X_s と表記する．たとえば，$a_{i,m}$ は Master パーティクル i に関するパラメータを示し，$a_{i,m} = \sum_{j,s} b_{j,s}$ のような表記の場合，Master パーティクル i を中心とした有効半径 h 内の近傍 Slave パーティクル j についての計算であることを示す．

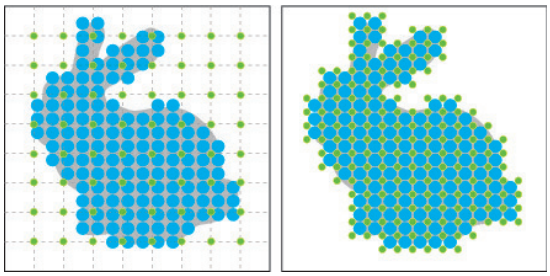


図 2 既存手法は左図のようにパーティクル（青点）とグリッド（緑点）を用いる．提案手法では代わりに右図のように Master パーティクル（青点）と Slave パーティクル（緑点）を用いる

Fig. 2 Previous methods use particle (blue) and grid (green) to realize stable simulation (*left*). Our method uses master (blue) and slave (green) particles (*right*).

シミュレーション全体の流れを図 3 に示す．以下，図 3 中に示した各計算ステップについて説明する．

1. **Slave パーティクルの速度と質量の計算**

Slave パーティクルは応力計算のために座標値を持つ必要があるが，Slave パーティクル自身は速度と質量をタイムステップを超えて保持しない．その代わりに Slave パーティクルの質量 $m_{i,s}$ と速度 $\boldsymbol{v}_{i,s}$ をタイムステップの最初に，近傍の Master パーティクルの質量 $m_{i,m}$ と速度 $\boldsymbol{v}_{i,m}$ から以下の式で計算する．

$$m_{i,s} = \sum_{j,m} m_{j,m} w(|\boldsymbol{x}_{i,s} - \boldsymbol{x}_{j,m}|, h) \tag{1}$$

$$\boldsymbol{v}_{i,s} = \frac{1}{m_{i,s}} \sum_{j,m} \boldsymbol{v}_{j,m} w(|\boldsymbol{x}_{i,s} - \boldsymbol{x}_{j,m}|, h) \tag{2}$$

ここで，$w(|\boldsymbol{x}_{i,s} - \boldsymbol{x}_{j,m}|, h)$ は Master パーティクル i と Slave パーティクル j の間の重みである．重み w はパーティクル間の距離 r と有効半径 h に関する関数であり，本論文では以下の式を用いる．

$$w(r,h) = \frac{3}{2\pi h^3} \begin{cases} \frac{1}{2}\left(\frac{r}{h}\right)^3 - \left(\frac{r}{h}\right)^2 + \frac{2}{3} & 0 \le \frac{r}{h} \le 1 \\ -\frac{1}{6}\left(\frac{r}{h}\right)^3 + \left(\frac{r}{h}\right)^2 - 2\frac{r}{h} + \frac{4}{3} & 1 \le \frac{r}{h} \le 2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

式 (3) は粒子法の一種である SPH 法でも用いられているカーネル関数である [13]．

2. **Master パーティクル体積の計算**

変形による力の算出のために Master パーティクルの

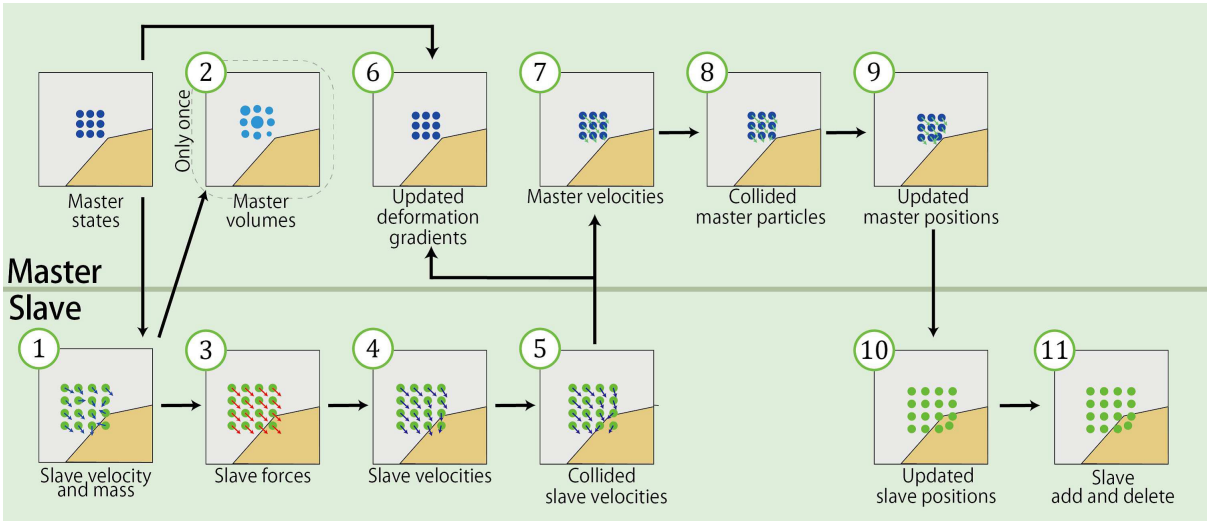


図 3 提案手法の概要．上段は Master パーティクル，下段は Slave パーティクルに関する処理である

Fig. 3 An overview of our method. The top row represent the procedure for the master particles and the bottom row is for the slave particles.

体積が必要となる．体積計算のためにまず式 (1) を用いて Slave パーティクルの質量 $m_{j,s}$ を推定し，$m_{j,s}$ から Master パーティクルの密度 $\rho^0_{i,m}$ を以下の式で計算する．

$$\rho^0_{i,m} = \sum_{j,s} m^0_{j,s} w(|\boldsymbol{x}_{i,m} - \boldsymbol{x}_{j,s}|, h) \tag{4}$$

密度から Master パーティクルの体積 $V^0_{i,m}$ を以下の式で計算する．

$$V^0_{i,m} = m_{i,m}/\rho^0_{i,m} \tag{5}$$

Master パーティクルの初期体積はシミュレーション中変化しないと考え，この処理は最初のステップでのみ実行される．

3. **Slave パーティクルにかかる力の計算**
近傍 Master パーティクルの変形勾配 $\mathbf{F}$ から Slave パーティクルにかかる力 $\boldsymbol{f}_{i,s}$ を計算する．この処理の詳細は 3.1 節で詳しく説明する．
4. **Slave パーティクル速度の計算**
求めた力 $\boldsymbol{f}_{i,s}$ から Slave パーティクルの速度 $\boldsymbol{v}^{n+1}_{i,s}$ を半陰的に計算し求める．この処理の詳細は 3.2 節で詳しく説明する．
5. **Slave パーティクルの衝突計算**
Slave パーティクルと障害物との衝突計算を行う．このとき更新されるのは速度のみであり，Slave パーティクル自体は障害物内に入ることができる．これはすべての Master パーティクルにおいて近傍にある Slave パーティクルの数が等しくなるようにするためである．
6. **Master パーティクルの変形勾配の更新**
各 Master パーティクルの変形勾配 $\mathbf{F}_{i,m}$ を以下のように更新する．

$$\mathbf{F}^{n+1}_{i,m} = (\mathbf{I} + \Delta t \nabla \boldsymbol{v}^{n+1}_{i,m})\mathbf{F}^n_{i,m} \tag{6}$$

ここで，I は単位行列，Δt はタイムステップ幅，$\nabla \boldsymbol{v}^{n+1}_{i,m}$ は速度勾配テンソルで以下のように計算する．

$$\nabla \boldsymbol{v}^{n+1}_{i,m} = \sum_{j,s} (\boldsymbol{v}^n_{i,m} - \boldsymbol{v}^{n+1}_{j,s})(\boldsymbol{x}^n_{i,m} - \boldsymbol{x}^n_{j,s})^T w(|\boldsymbol{x}_{i,m} - \boldsymbol{x}_{j,s}|, h). \tag{7}$$

式 (7) は文献 [27] で示された変形勾配テンソルの近似式を時間微分し，正規化項の代わりにカーネル関数を使って粒子法で近似した式である．ただし，文献 [27] と異なり塑性変形も扱うため，初期位置の代わりに n ステップ目の位置を用いた．式 (6) の変形勾配 $\mathbf{F}$ から塑性変形部分を取り出すことで弾性変形から塑性変形への変化を表現する．この処理の詳細は 3.3 節で説明する．

7. **Master パーティクルの速度を計算**
Master パーティクルの新しい速度 $\boldsymbol{v}^{n+1}_{i,m}$ は以下で計算される．

$$\boldsymbol{v}^{n+1}_{i,m} = (1-\alpha)\boldsymbol{v}^n_{i,m} + \alpha \hat{\boldsymbol{v}}_{i,m} \tag{8}$$

ここで，$\alpha \in [0,1]$ はユーザ指定の定数であり，本研究ではすべての例において $\alpha = 0.9$ を用いている．$\hat{\boldsymbol{v}}_{i,m}$ は 近傍の Slave パーティクルの速度 $\boldsymbol{v}^{n+1}_{j,s}$ の最小自乗補間によって以下の式で計算される．

$$\hat{\boldsymbol{v}}_{i,m} = \arg\min_{\hat{\boldsymbol{v}}_{i,m}} \sum_{j,s} w(|\boldsymbol{x}_{i,m} - \boldsymbol{x}_{j,s}|, h) \|\hat{\boldsymbol{v}}_{i,m} - \boldsymbol{v}^{n+1}_{j,s}\|^2 \tag{9}$$

8. **Master パーティクルの衝突計算**
Master パーティクルと障害物との衝突計算を行う．ステップ 5 の Slave パーティクルと異なり，位置の更新も行う．
9. **Master パーティクルの位置更新**
前ステップで更新された速度 $\boldsymbol{v}^{n+1}_{i,m}$ から以下の式により Master パーティクルの位置を更新する．

$$\boldsymbol{x}^{n+1}_{i,m} = \boldsymbol{x}^n_{i,m} + \Delta t \boldsymbol{v}^{n+1}_{i,m} \tag{10}$$

10. **Slave パーティクルの移流**
Master パーティクルの動きにあわせて，Slave パーティクルを移流させる．この処理の詳細は 4.1 節で説明する．
11. **Slave パーティクルの追加と削除**
大きな変形によって，分離や結合などが生じた場合は Slave パーティクルの追加もしくは削除を行う．追加と削除のルールについての詳細はそれぞれ 4.2 節と 4.3 節で説明する．

3.1 Slave パーティクルにかかる力の計算

Dyka ら [3], [4] が提案したように，Master パーティクルの位置ではなく Slave パーティクルの位置で変形のための力を計算を行うことで，計算の安定化が可能になる．Slave パーティクルにかかる力 $\boldsymbol{f}_{i,s}$ は以下の式で計算される．

$$\boldsymbol{f}_{i,s} = -\sum_{j,m} V^n_{j,m} \sigma_{j,m} \nabla w(|\boldsymbol{x}_{i,s} - \boldsymbol{x}_{j,m}|, h) \tag{11}$$

ここで，$V_{j,m}$ は Master パーティクル j の n ステップ目における体積であり，$\sigma_{j,m}$ はコーシー応力である．$V_{j,m}$ と $\sigma_{j,m}$ は Master パーティクル j の変形勾配 $\mathbf{F}$ の弾性変形部分 $\mathbf{F}_{E_{j,m}}$ と塑性変形部分 $\mathbf{F}_{P_{j,m}}$ から以下の式で計算する．

$$V_{j,m} = J\, V^0_{j,m} \tag{12}$$

$$\sigma_{j,m} = \frac{1}{J} \frac{\partial \Psi}{\partial \mathbf{F}_E} (\mathbf{F}_E)^T \tag{13}$$

ここで，$J = \det(\mathbf{F}_E \mathbf{F}_P)$ であり，$\Psi(\mathbf{F}_E, \mathbf{F}_P)$ は [17] で定

義されている弾塑性エネルギー密度関数であり，ポアソン比 ν とヤング率 E から計算されるラメ定数 μ, λ を含む関数となっている．このポアソン比 ν とヤング率 E を設定することで物性を変化させることができる．具体的にはポアソン比を大きく設定すると力が加わったときに力と垂直な方向に変形しやすくなり，ヤング率を大きくすると力に対して変形しにくい硬い物体となる．弾性変形勾配 $\mathbf{F}_E$ と塑性変形勾配 $\mathbf{F}_P$ は単位行列で初期化しておき，3.3 節に示す方法で更新する．

3.2 Slave パーティクル速度の更新

式 (11) で計算した力 $\boldsymbol{f}_{i,s}$ から，Slave パーティクルの仮速度 $\boldsymbol{v}^{\star}_{i,s}$ を計算する．

$$\boldsymbol{v}^{\star}_{i,s} = \boldsymbol{v}^{n}_{i,s} + \Delta t m^{-1}_{i,s} \boldsymbol{f}_{i,s} \tag{14}$$

式 (14) は $\partial \boldsymbol{v}/\partial t = \boldsymbol{f}/m$ を単純に離散化した式であり，完全に陽的な計算となっている．安定化のために [17] と同じ方法を用い，以下の線形システムを解くことで半陰的に速度を更新する．

$$\sum_{j,s} \left(\mathbf{I}\delta(i_s, j_s) + \beta \Delta t^2 m^{-1}_{i,s} \frac{\partial^2 \Phi}{\partial \boldsymbol{x}_{i,s} \partial \boldsymbol{x}_{j,s}} \right) \boldsymbol{v}^{n+1}_{j,s} = \boldsymbol{v}^{\star}_{i,s} \tag{15}$$

ここで，$\beta \in [0, 1]$ は定数であり，$\delta(i_s, j_s)$ と Φ は，

$$\delta(i_s, j_s) = \begin{cases} 1 & i_s = j_s \\ 0 & i_s \neq j_s \end{cases} \tag{16}$$

$$\Phi = \Sigma_{j,m} V^0_{j,m} \Psi(\mathbf{F}_{E_{j,m}}, \mathbf{F}_{P_{j,m}}) \tag{17}$$

となる．式 (15) を共役勾配法を用いて解くことで次のステップにおける Slave パーティクルの速度 $\boldsymbol{v}^{n+1}_{i,s}$ を計算する．

3.3 変形勾配の更新

前節で求めた Slave パーティクルの速度 $\boldsymbol{v}^{n+1}_{i,s}$ から，Master パーティクルの変形勾配 $\mathbf{F}^{n+1}$ を計算する．まず，近傍 Slave パーティクルから速度勾配 $\nabla \boldsymbol{v}^{n+1}_{m}$ を式 (7) を用いて計算する．速度勾配を式 (6) に代入することで変形勾配が計算される．変形勾配 $\mathbf{F}^{n+1}$ には弾性変形勾配と塑性変形勾配の両方が含まれている．次のステップでそれぞれの成分を用いるために，変形勾配を降伏点で分離させる [17]．

まず，すべて弾性変形であると仮定して，仮の弾性変形勾配を

$$\hat{\mathbf{F}}^{n+1}_{E} = \left(\mathbf{I} + \Delta t \nabla \boldsymbol{v}^{n+1}_{m} \right) \mathbf{F}^{n}_{E} \tag{18}$$

と計算する．実際の物質ではある一定までの変形は弾性変形によって元の形状に戻るが，降伏点と呼ばれる点以上の変形を行うと塑性変形になり元の形状に完全には戻らなくなる．これを再現するために，仮の弾性変形勾配から降伏点を超えた分を塑性変形勾配にわたす．仮の弾性変形勾配 $\hat{\mathbf{F}}^{n+1}_{E}$ を特異値分解（$\hat{\mathbf{F}}^{n+1}_{E} = \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^T$）したとき，得られた行列 $\hat{\mathbf{\Sigma}}$ の対角要素には変形量を示す値が並んでいる．この対角要素の値が，ユーザが指定した伸びに対する降伏点 θ_s と，圧縮に対する降伏点 θ_c の間の値になるように制限する．

$$\mathbf{\Sigma} = \mathrm{clamp}\left(\hat{\mathbf{\Sigma}},\ [1 - \theta_c, 1 + \theta_s] \right) \tag{19}$$

式 (19) によって弾性変形部分が取り出されたので，各変形勾配を更新していく．弾性変形勾配は特異値分解を元に戻すだけなので

$$\mathbf{F}^{n+1}_{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{20}$$

と計算する．塑性変形勾配は，全体の変形勾配が $\mathbf{F} = \mathbf{F}_E\mathbf{F}_P$ であること用いて以下のように計算する．

$$\mathbf{F}^{n+1}_{P} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\hat{\mathbf{F}}^{n+1}_{E}\mathbf{F}^{n}_{P} \tag{21}$$

このようにして更新された変形勾配 $\mathbf{F}_E, \mathbf{F}_P$ を用いて，次ステップで §3.1 で示したように Slave パーティクルにかかる力の計算をする．

4. Slave パーティクル

Slave パーティクルは応力計算点であり，これまで提案されてきた手法では物質を構成するパーティクルの間を埋めるように配置されていた．提案手法では，すべての Master パーティクルについて有効半径内に Slave パーティクルが均一に分布するように配置する．グリッドを用いた場合はは空間全体に応力計算点を配置することになるが，同様のことを Slave パーティクルでも行うと空間の大きさに計算速度が比例し，計算空間が制限されてしまう．また，Master パーティクル周囲に毎ステップ再配置すると変形によって Master パーティクルの分布が均一でなくなったときに，多くの近傍パーティクル探索処理が発生し，計算コストが大きくなってしまう．そこで Slave パーティクルを Master パーティクルの動きに合わせて移流させ，部分的に Slave パーティクルの追加・削除を行う．

4.1 Slave パーティクルの移流

グリッド法では空間全体に計算点を配置することになるが，Slave パーティクルの場合は Master パーティクルの動きに合わせて移流させることで必要な箇所だけに計算点を配置できる．Slave パーティクルの移動には式 (8) で求めた速度 $\boldsymbol{v}^{n+1}_{i,m}$ を用いる．近傍 Master パーティクルの速度から以下の式で Slave パーティクルの速度を計算する．

$$\boldsymbol{v}^{n+1}_{i,s} = \frac{1}{m_{i,s}} \sum_{j,m} m_{j,m} \boldsymbol{v}^{n+1}_{j,m} w(|\boldsymbol{x}_{i,s} - \boldsymbol{x}_{j,m}|, h) \tag{22}$$

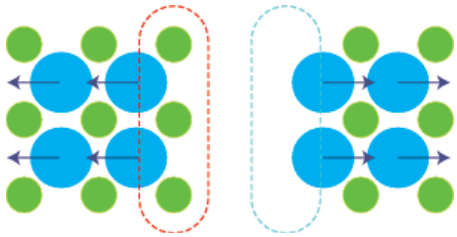


図 4 分裂した際に，左側の Master パーティクルに赤枠で囲まれた Slave パーティクルが追従し，青枠部分で Slave パーティクルの不足が発生する

Fig. 4 The slave particles surrounded by a red frame on the left side follow the master particle in case of splitting, and then there is a lack of slave particles in the blue-framed part.

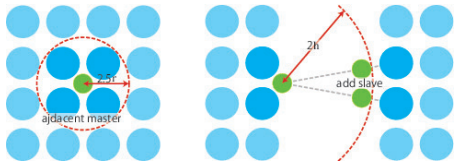


図 5 隣接する Master パーティクルの位置（左）と隣接する Master パーティクルが有効半径 $2h$ よりも離れたとき Master パーティクルに近い位置に Slave パーティクルを追加する（右）

Fig. 5 Position of the adjacent master particle (*left*). When the formerly adjacent master particle moves beyond a distance of more than the effective radius $2h$, a slave particle is newly added between the departed master particle and the distant slave particle (*right*).

計算した速度から $\boldsymbol{x}_s^{n+1} = \boldsymbol{x}_s^n + \Delta t \boldsymbol{v}_s^{n+1}$ により位置を計算する．

4.2 Slave パーティクルの追加

塑性変形時に物体が分離したときや大きな変形が発生したときに，Slave パーティクルがほかの部分よりも不足してしまい，結果として計算が不安定になる．Slave パーティクル不足を防ぐために Slave パーティクルの追加を行う．

まず分裂時について考える．**図 4** に示すように，隣接する Master パーティクルが離れることで分裂が発生し，間にある Slave パーティクルがどちらか一方のみに追従する形で移動する．このとき，Slave パーティクルが追従しなかった方の Master パーティクルで Slave パーティクルの不足が生じる．

Slave パーティクルの不足を補うために，Master パーティクル半径 r を使って，まず初期状態において半径 γr 内に含まれている Master パーティクルを隣接してる Master パーティクルとして記憶しておく（本論文の例では $\gamma = 2.5$ を用いている）．シミュレーションステップを進める中で，隣接していた Master パーティクルが有効半径 $2h$ よりも離れたときに，Slave パーティクルと離れた Master パーティクルとの間に新たに Slave パーティクルを追加する（**図 5**）．ここで h は式 (1) で用いられているものと同じであり，$2h$ は式 (3) の重み関数の有効半径である．新たに追加される Slave パーティクルの位置 $\boldsymbol{x}_s^{new}$ の位置は以下の式で求める．

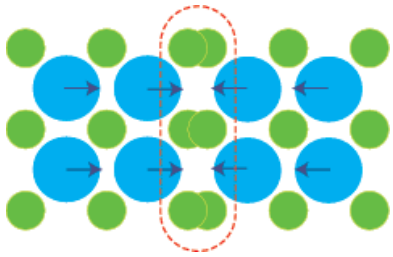


図 6 赤枠で囲まれた部分の Slave パーティクルが過密になっている

Fig. 6 Slave particles surrounded by a red frame has become overcrowded.

$$\boldsymbol{x}_s^{new} = \frac{n\boldsymbol{x}_s + m\boldsymbol{x}_m}{m+n} \tag{23}$$

ここで，$\boldsymbol{x}_m, \boldsymbol{x}_s$ は離れてしまった Master パーティクルの Slave パーティクルのそれぞれの位置であり，m, n は任意の定数である（本論文では $m = 2, n = 1$ を用いている）．

Slave パーティクルの追加を距離だけで判断した場合，やわらかい弾性体のように大きな変形をともなう物体のシミュレーション時にも余分に生成されてしまう．これを防ぐために追加する位置における Slave パーティクルの密度が，初期状態における最大 Slave パーティクル密度よりも高い場合は追加しないという条件も加える．

4.3 Slave パーティクルの削除

Slave パーティクルは近傍 Master パーティクルに追従させて移動させるが，物体が結合する場合，**図 6** のように Master パーティクルどうしが近づくため，間にある Slave パーティクルが過剰となる．これを防ぐために各 Slave パーティクルについて，有効半径 $2h$ 内の Slave パーティクルとの距離が $0.15h$ よりも小さくなったとき片方の Slave パーティクルを削除する．

5. 実験結果

提案手法の有効性を確かめるためにさまざまな特性を持つ弾塑性体の振舞いをシミュレーションした結果を示す．各シーンのパラメータを**表 1** に，平均計算時間を**表 2** に示す．すべての結果は Intel Corei7-4930K 3.40 GHz を搭載した PC 上で実行し，パーティクルからの表面メッシュの生成には OpenVDB [15] を，レンダリングには Mitsuba [10] を用いた．すべての結果を含む動画は以下の URL から取得できる．
http://slis.tsukuba.ac.jp/pbcglab/files/ipsj_slave.mp4

Slave パーティクルの有無によるシミュレーションの数値的安定性を比較した結果を**図 7** に示す．図 7 における Master と初期 Slave パーティクルの数はそれぞれ 12,582, 19,260 である．$\Delta t = 0.5$ [msec] の場合，Slave パーティクルなしですべての計算を Master パーティクルで行った結

果（図 7 左）では 40 フレーム目でシミュレーションが発散している．一方，Slave パーティクルありの結果（図 7 右）では同じタイムステップ幅でも安定してシミュレーションができている．

図 8 は図 7 と同じシーンでタイムステップ幅を変えて実行した結果である．$\Delta t = 0.5$ [msec] では安定かつ自然な変形が実現できているが，$\Delta t = 0.7$ [msec] では計算は発散しないものの図中に赤枠で示したような不自然な変形が観測された．この部分はバニー形状の耳が変形により他の部分と結合する部分であり，結合により Master パーティクル周囲の Slave パーティクル分布が不均一になったことが原因ではないかと考えられる．

図 9 は硬さのパラメータを変えて実行した結果である．変形が分かりやすいように立方体形状の物体を地面に落下させている．物体を構成する Master，Slave パーティクルはそれぞれ 4,096，6,647 とし，降伏点を設定しないことでゴムのような完全弾性変形を再現している．図 9 上段はポアソン比を小さく，ヤング率を大きく設定することで変形しにくい硬い物体が，下段は逆にポアソン比を大きく，ヤング率を小さくしたため柔らかい物体の挙動が再現されて

表 1 各シーンのパラメータ
Table 1 Parameters for each scene.

シーン	ポアソン比	ヤング率 [Pa]	降伏点 θ_s	降伏点 θ_c
図 7，8，14	0.35	1.2×10^{6}	0.1	0.8
図 9（上）	0.3	6.9×10^{7}	None	None
図 9（下）	0.4	1.0×10^{6}	None	None
図 10（左）	0.2	1.4×10^{5}	4.5×10^{-3}	1.5×10^{-2}
図 10（右）	0.2	1.4×10^{5}	7.5×10^{-4}	2.5×10^{-3}
図 11（上）	0.35	5.0×10^{7}	None	None
図 11（下）	0.25	5.0×10^{5}	0.05	0.5
図 12	0.4	2.0×10^{7}	7.5×10^{-4}	2.5×10^{-3}
図 13（上）	0.25	1.2×10^{6}	0.1	0.75
図 13（下）	0.45	2.0×10^{5}	0.05	0.2

表 2 計算時間
Table 2 Computational times.

シーン	タイムステップ幅 [sec]	計算時間 [sec/frame]
図 9（上）	0.001	0.9030
図 9（下）	0.001	0.9679
図 10（左）	0.001	0.6275
図 10（右）	0.001	0.8744
図 11（上）	0.0005	2.3703
図 11（下）	0.0005	2.9802
図 12	0.0005	1.0914
図 13（上）	0.0005	7.0282
図 13（下）	0.0005	4.6485
図 14	0.0005	8.7657

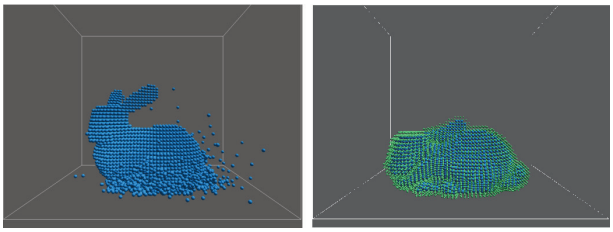


図 7 Slave パーティクルの有無によるシミュレーションの数値的安定性の比較．左が Slave なし，右は Slave あり
Fig. 7 Comparison of the numerical stability of the simulations with and without slave particles. *Left:* the method without slave particles. *Right:* the method with slave particles.

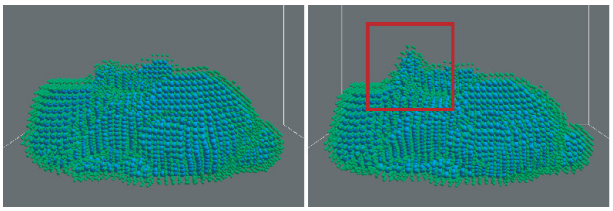


図 8 Slave パーティクルありでタイムステップ幅を変えた結果．左は $\Delta t = 0.5$ [msec]，右は $\Delta t = 0.7$ [msec] でシミュレーションを行った
Fig. 8 Simulation of different time steps with slave particles. *Left:* $\Delta t = 0.5$ [msec]. *Right:* $\Delta t = 0.7$ [msec].

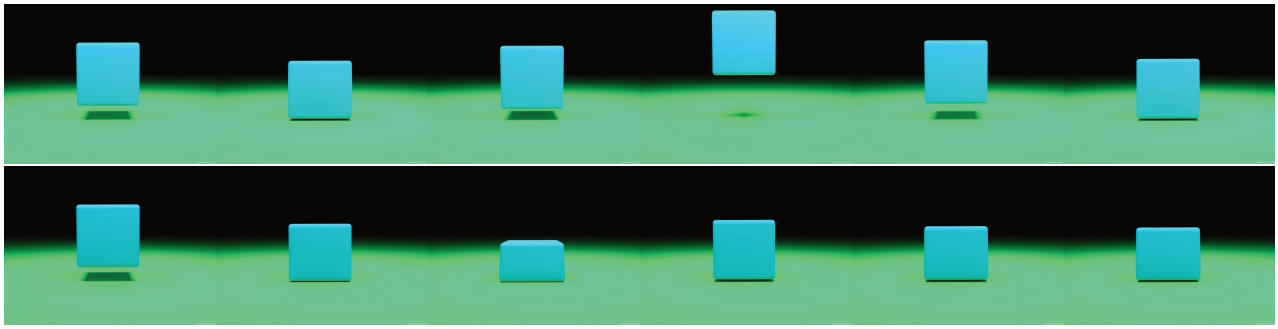


図 9 立方体を地面に落下させるシーンの結果．提案手法では上段のような固い弾性体から下段のような柔らかい弾性体まで扱うことができる
Fig. 9 Comparison of materials with different stiffness settings applied to the cube falling on the ground. Our algorithm is able to simulate objects with materials ranging from highly elastic (top row) to only slightly elastic (bottom row).

いる（具体的な値は表 1 参照）．

弾性体だけでなく降伏点の設定を変えることで**図 10** に示したように異なる性質を持つ塑性体もシミュレーション可能である．図 10 は図 9 と同じく立方体を落下させた結果であり，物体を構成する Master，Slave パーティクルはそれぞれ 4,096，6,647 である．図 10 左では降伏点を高く設定しており，大きな変形でもある程度元の形状に戻ろうとする挙動を示す．図 10 右では降伏点を低く設定しており，変形の多くの部分が塑性変形となり，元の形状から大きく異なる形状に変形したままになっている．

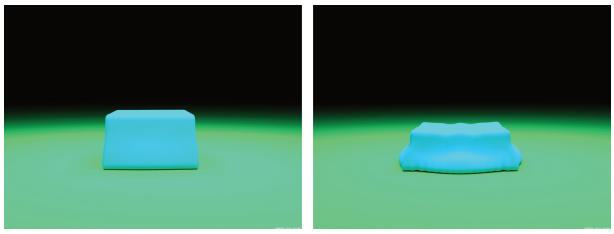


図 10 異なる塑性変形パラメータを持つ物体のシミュレーション．左は塑性変形が小さく，右は塑性変形が大きい物体

Fig. 10 Simulation of different plastic materials. *Left:* moderately plastic. *Right:* highly plastic.

複雑な形状で弾性変形，塑性変形パラメータを変えた場合の例を**図 11** に示す．このシーンで用いたアルマジロ形状はそれぞれ 11,161，20,242 の Master，Slave パーティクルで構成されている．提案手法によって複雑な形状でも弾塑性変形の振舞いを再現できることが確認できた．

弾塑性体と障害物のインタラクションの例を**図 12** と**図 13** に示す．図 12 では 1 層の Master パーティクルで構成された薄いシート状の物体をクロス形状の障害物に落下させた結果である．Master，Slave パーティクル数はそれぞれ 7,569，31,680 である．従来の手法では扱えなかったような薄い形状でも提案手法では扱うことができ，さらに物体の分離や結合もシミュレーションできている．図 13 は異なる塑性変形パラメータを持つバニー形状を棒状の障害物に落下させた結果である．バニー形状はそれぞれ 17,263 と 25,411 の Master，Slave パーティクルで構成される．体積を持つ物体が大きく分離するような変形も扱えており，パラメータを変えることで分離後も元の形状を保つような変形も可能である．図 12 下段の結果においては，棒状の

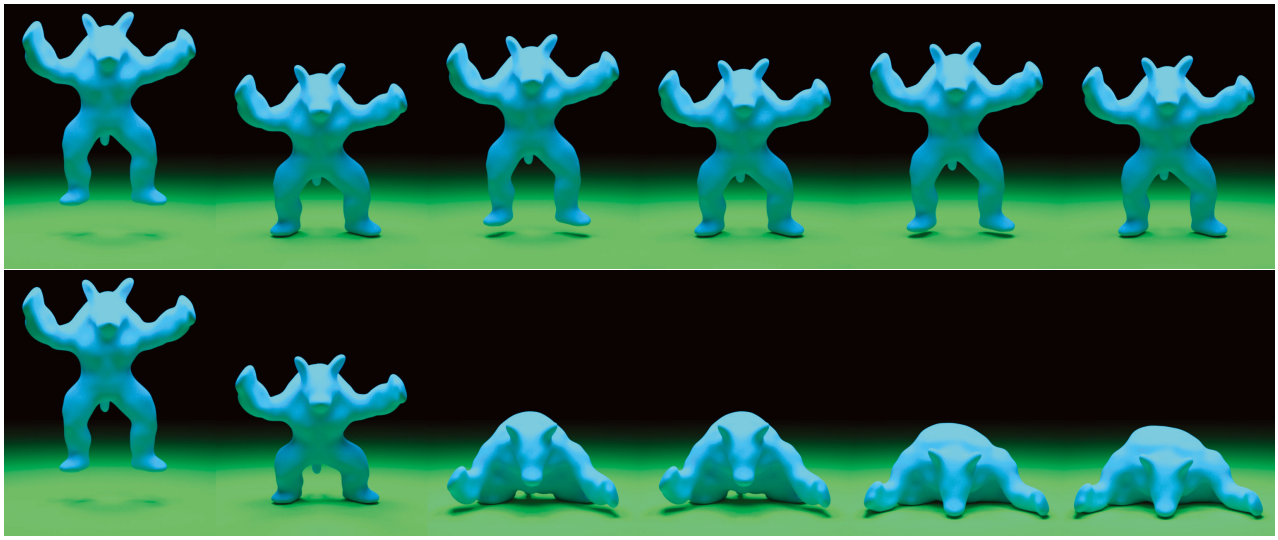


図 11 アルマジロ形状の弾性変形（上段）と弾塑性変形（下段）

Fig. 11 Simulation of elastic and elastoplastic materials using armadillo model. *Top:* elastic. *Bottom:* elastoplastic.

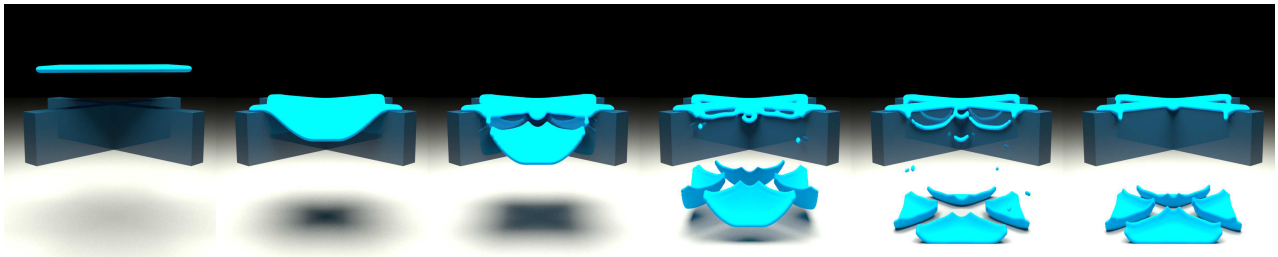


図 12 提案手法では 1 層のパーティクルで構成される非常に薄い物体を扱うこともできる

Fig. 12 Our method is able to treat thin objects composed of one-layer of master particles.

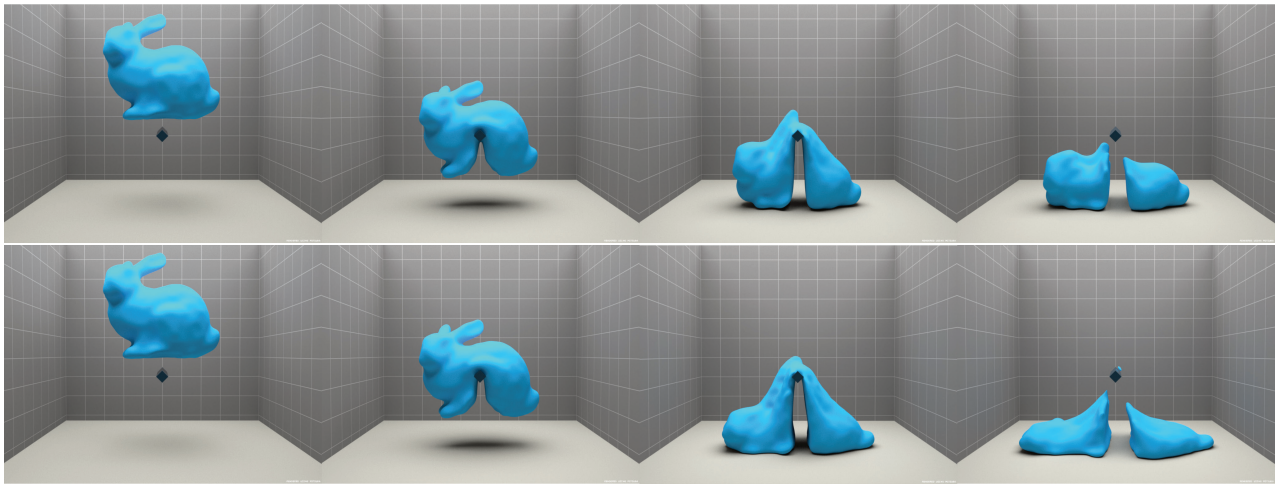


図 13 棒状の障害物に異なるパラメータを与えたバニー形状のオブジェクトを落下させた場合の比較．上段は塑性変形が小さい場合で衝突して分離した後も元の形状をある程度残している．下段は塑性変形が大きく衝突後元の形状が完全に崩れてしまっている

Fig. 13 Comparison of different plastic materials applied to the bunny model falling onto the obstacle. Low plastic bunny still have shape after collision (top row), and high plastic bunny model does not have shape after collision (bottom row).

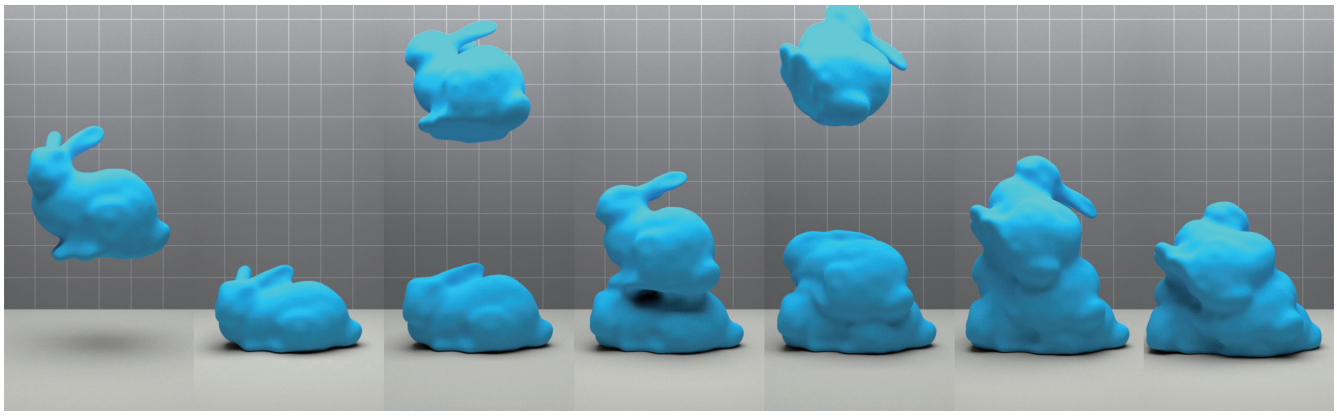


図 14 弾塑性物体が結合する場合の例．左から右へフレームが進む際に，新しいバニー形状のオブジェクトを上部に追加している

Fig. 14 Our method can simulate elastoplastic objects merging each other. From the left to right, adding new bunny model above old bunny model, and new bunny model merge old one.

障害物上に残った物体が消えるような挙動が見られたが，これは表面メッシュを生成する際の関数場の閾値を形状を滑らかにするために大きく設定したためであり，計算に用いたパーティクルが消えたものではない．分離だけではなく物体どうしの結合も扱えることを示す結果を**図 14** に示す．各バニー形状を構成するパーティクル数はそれぞれ 12,582，19,260 である．このように弾塑性体を次々と落下させ，物体どうしが結合するようなシーンも提案手法では扱えることが実験により確かめられた．

最後に提案手法がパーティクルのみを用いたことによる計算空間の広げやすさを確かめるための実験の結果を**図 15** に示す．図 15 のシーンではバニー形状の弾塑性体をシミュレーションが進むにつれて次々と追加している．各バニー形状は図 13 のものと同じパーティクル数で構成されている．オブジェクトが存在する計算空間の広さは図 13 と比較して，水平および奥行き方向にそれぞれ 4 倍となっている．このシーンにおける計算時間と Master，Slave パーティクル数の変化を**図 16** に示す．図 16 から分かるように，提案手法の計算速度はあくまでパーティクル数に依存し，計算空間の広さには依存していないことが分かる．なお，図 16 において 8,600 フレームおよび 10,800 フレーム付近で計算時間が大きく上昇しているのは，その付近でオブジェクトが床と衝突し，変形が始まったことで処理が増えたことが原因である．本論文では近傍パーティクルの探索にグリッド構造を用いたため，計算空間の大きさに依存して必要なメモリ量が増える可能性はある．しか

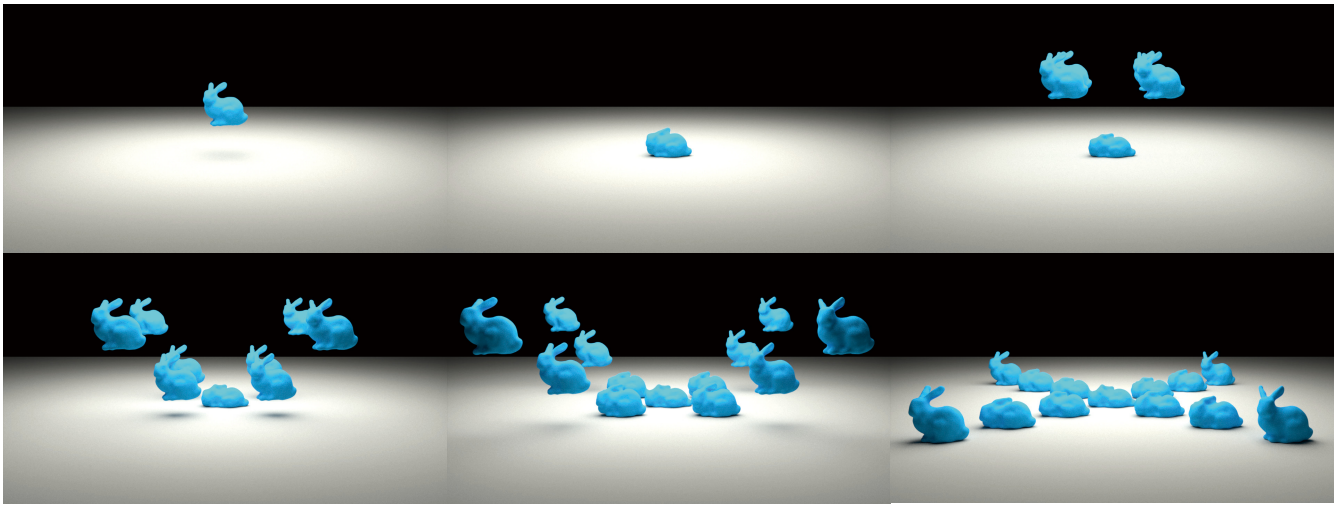


図 15 広い計算空間内にバニー形状の弾塑性物体を多数追加した場合の例
Fig. 15 Simulation of many elastoplastic materials using bunny model in a large simulation space.

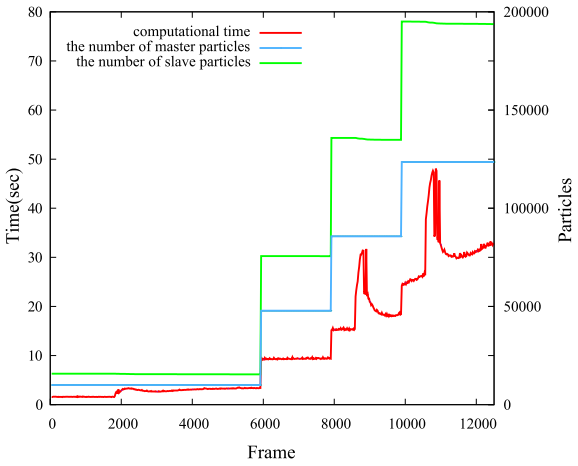


図 16 図 15 の計算時間と Master/Slave パーティクル数
Fig. 16 Computational time of Fig. 15 with the number of master/slave particles.

し，提案手法を実装する際に近傍探索に Green [6] のハッシュを用いた方法を CPU 上で実装して用いたため，グリッド数に依存して増えるデータは少なく，必要なメモリ量もほぼパーティクル数に比例している．

6. まとめと今後の課題

本論文ではパーティクル法を用いた弾塑性体シミュレーションに対して Slave パーティクルを用いることで，高速かつ安定した弾塑性体表現のシミュレーション手法を実現した．Slave パーティクルを Master パーティクルの動きに合わせて移流させることで，計算空間に制限されない安定した計算を，また，変形による物体の分裂・結合表現を Slave パーティクルの追加・削除を行うことで可能にした．

今後の課題としては，Slave パーティクルの追加・削除の条件および方法についての検証がある．図 5 に示したように，分離では Slave パーティクルが 1 つだった箇所に 2 つ追加している．Slave パーティクルの数が少ないほど計算コストがかからないので，追加する Slave パーティクルの数が 1 つでも動作するようなら 1 つにする必要がある．削除については，閾値よりも近づいた Slave パーティクルの片方を削除するだけでなく，近づいた Slave パーティクルの組の間に再配置するなどの処理が考えられる．また，計算速度の向上も今後の課題である．式 (15) の線形システムを解く部分が全体の計算時間の約半分を占めており，この部分の並列化やソルバーの変更による収束の高速化も行っていきたい．さらに薄い物体を垂直に配置して落とした場合の座屈現象の再現についても検証していきたい．

参考文献

[1] Bargteil, A.W., Wojtan, C., Hodgins, J.K. and Turk, G.: A finite element method for animating large viscoplastic flow, *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, p.16, ACM (online), DOI: http://doi.acm.org/10.1145/1275808.1276397 (2007).

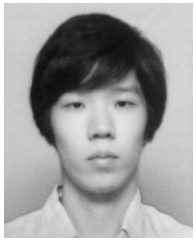
[2] Brackbill, J., Kothe, D. and Ruppel, H.: FLIP: A low-dissipation, particle-in-cell method for fluid flow, *Computer Physics Communications*, Vol.48, No.1, pp.25–38 (online), DOI: http://dx.doi.org/10.1016/0010-4655(88)90020-3 (1988).

[3] Dyka, C.T., Randles, P.W. and Ingel, R.P.: Stress Points for Tension Instability in SPH, *International Journal for Numerical Methods in Engineering*, Vol.40, pp.2325–2341 (online), DOI: 10.1002/(SICI)1097-0207(19970715)40:13<2325::AID-NME161>3.0.CO;2-8 (1997).

[4] Dyka, C. and Ingel, R.: An approach for tension instability in smoothed particle hydrodynamics (SPH), *Computers & Structures*, Vol.57, No.4, pp.573–580 (online), DOI: http://dx.doi.org/10.1016/0045-7949(95)00059-P (1995).

[5] Gerszewski, D., Bhattacharya, H. and Bargteil, A.W.: A point-based method for animating elastoplastic solids,

SCA '09: Proc. 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.133–138, ACM (online), DOI: http://doi.acm.org/10.1145/1599470.1599488 (2009).
[6] Green, S.: CUDA Particles, Technical Report, NVIDIA Whitepaper (2008).
[7] Harlow, F. and Welch, J.: Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Physics of Fluids*, Vol.8, pp.2182–2189 (online), DOI: http://dx.doi.org/10.1063/1.1761178 (1965).
[8] He, X., Liu, N., Wang, G., Zhang, F., Li, S., Shao, S. and Wang, H.: Staggered Meshless Solid-fluid Coupling, *ACM Trans. Graphics*, Vol.31, No.6, pp.149:1–149:12 (online), DOI: 10.1145/2366145.2366168 (2012).
[9] Irving, G., Teran, J. and Fedkiw, R.: Invertible finite elements for robust simulation of large deformation, *SCA '04: Proc. 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association, pp.131–140 (online), DOI: http://doi.acm.org/10.1145/1028523.1028541 (2004).
[10] Jakob, W.: Mitsuba Physically Based Renderer (2015), available from ⟨http://www.mitsuba-renderer.org⟩.
[11] Jones, B., Ward, S., Jallepalli, A., Perenia, J. and Bargteil, A.W.: Deformation Embedding for Point-based Elastoplastic Simulation, *ACM Trans. Graphics*, Vol.33, No.2, pp.21:1–21:9 (online), DOI: 10.1145/2560795 (2014).
[12] Macklin, M., Müller, M., Chentanez, N. and Kim, T.-Y.: Unified Particle Physics for Real-time Applications, *ACM Trans. Graphics*, Vol.33, No.4, pp.153:1–153:12 (online), DOI: 10.1145/2601097.2601152 (2014).
[13] Monaghan, J.J.: Smoothed particle hydrodynamics, *Reports on Progress in Physics*, Vol.68, No.8, p.1703 (online) (2005), available from ⟨http://stacks.iop.org/0034-4885/68/i=8/a=R01⟩.
[14] Müller, M., Keiser, R., Nealen, A., Pauly, M., Gross, M. and Alexa, M.: Point Based Animation of Elastic, Plastic and Melting Objects, *SCA2004* (2004).
[15] Museth, K.: VDB: High-resolution Sparse Volumes with Dynamic Topology, *ACM Trans. Graphics*, Vol.32, No.3, pp.27:1–27:22 (online), DOI: 10.1145/2487228.2487235 (2013).
[16] Solenthaler, B., Schläfli, J. and Pajarola, R.: A Unified Particle Model for Fluid-Solid Interactions, *Computer Animation and Virtual Worlds*, Vol.18, No.1, pp.69–82 (2007).
[17] Stomakhin, A., Schroeder, C., Chai, L., Teran, J. and Selle, A.: A material point method for snow simulation, *ACM Trans. Graphics*, Vol.32, No.4, pp.102:1–102:10 (online), DOI: 10.1145/2461912.2461948 (2013).
[18] Stomakhin, A., Schroeder, C., Jiang, C., Chai, L., Teran, J. and Selle, A.: Augmented MPM for Phase-change and Varied Materials, *ACM Trans. Graphics*, Vol.33, No.4, pp.138:1–138:11 (online), DOI: 10.1145/2601097.2601176 (2014).
[19] Sulsky, D., Zhou, S.-J. and Schreyer, H.L.: Application of a particle-in-cell method to solid mechanics, *Computer Physics Communications*, Vol.87, No.1-2, pp.236–252 (online), DOI: http://dx.doi.org/10.1016/0010-4655(94)00170-7 (1995).
[20] Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K.: Elastically deformable models, *SIGGRAPH '87: Proc. 14th Annual Conference on Computer Graphics and Interactive Techniques*, pp.205–214, ACM (online), DOI: http://doi.acm.org/10.1145/37401.37427 (1987).
[21] Terzopoulos, D. and Witkin, A.: Physically Based Models with Rigid and Deformable Components, *IEEE Comput. Graph. Appl.*, Vol.8, No.6, pp.41–51 (online), DOI: http://dx.doi.org/10.1109/38.20317 (1988).
[22] Vignjevic, R., Campbell, J. and Libersky, L.: A treatment of zero-energy modes in the smoothed particle hydrodynamics method, *Computer Methods in Applied Mechanics and Engineering*, Vol.184, No.1, pp.67–85 (online), DOI: http://dx.doi.org/10.1016/S0045-7825(99)00441-7 (2000).
[23] Wicke, M., Ritchie, D., Klingner, B.M., Burke, S., Shewchuk, J.R. and O'Brien, J.F.: Dynamic local remeshing for elastoplastic simulation, *SIGGRAPH '10: ACM SIGGRAPH 2010 papers*, pp.1–11, ACM (online), DOI: http://doi.acm.org/10.1145/1833349.1778786 (2010).
[24] Wojtan, C., Thürey, N., Gross, M. and Turk, G.: Deforming meshes that split and merge, *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pp.1–10, ACM (online), DOI: http://doi.acm.org/10.1145/1576246.1531382 (2009).
[25] Wojtan, C. and Turk, G.: Fast viscoelastic behavior with thin features, *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pp.1–8, ACM (online), DOI: http://doi.acm.org/10.1145/1399504.1360646 (2008).
[26] Zhou, Y., Lun, Z., Kalogerakis, E. and Wang, R.: Implicit Integration for Particle-based Simulation of Elasto-Plastic Solids, *Computer Graphics Forum*, Vol.32, No.7, pp.215–223 (online), DOI: 10.1111/cgf.12229 (2013).
[27] 邵　陽, 伊藤広貴, 柴田和也, 越塚誠一：Hamiltonian MPS 粒子法による Reissner-Mindlin シェルの解析モデル，日本計算工学会論文集，Vol.2012, pp.20120013–20120013 (2012).



斉田 智也

2014 年筑波大学情報学群情報メディア創成学類卒業．2016 年同大学大学院図書館情報メディア研究科修了．修士（情報学）．在学中，物理シミュレーションに関する研究に従事．現在，株式会社スクウェア・エニックスに勤務．

藤澤 誠　（正会員）

2005 年静岡大学大学院理工学研究科修士課程修了．2008 年同大学大学院博士課程修了．同年奈良先端科学技術大学院大学情報科学研究科助教．2011 年筑波大学大学院図書館情報メディア研究科助教，現在に至る．博士（工学）．CG，物理シミュレーション等の研究に従事．画像電子学会，日本 VR 学会，ACM，IEEE CS 各会員．

三河 正彦

1994 年大阪大学大学院機械工学分野修士課程修了．同年 NTT アクセス網研究所入所．2003 年筑波大学図書館情報学系講師．2006 年同大学院図書館情報メディア研究科准教授，現在に至る．博士（工学）．日本ロボット学会，計測自動制御学会，日本知能情報ファジィ学会，IEEE 各会員．