Vol.2015-CG-160 No.4
2015/8/29

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,a)]　藤澤 誠[1,b)]　三河 正彦[1,c)]

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る．さらに追加・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できる．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．

1. はじめに

CG アニメーションでは，粘土やチューインガム，パン生地といった弾塑性体と呼ばれる物質を対象とすることが多い．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．物性値によって弾性変形の表現が変化し，降伏点によって弾性変形から塑性変形への遷移する変形量が決まる．

弾塑性体の運動計算を有限要素法等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．メッシュ構造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・結合によってメッシュ構造を再構成する必要がある．一方で，パーティクルを用いて計算する方法では，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盛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．提案手法では，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が可能になる．これにより，物質を表現する粒子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を可能にする．

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 [7] の研究や Terzopoulos と Fleischer[8] の研究があり，現在まで弾塑性体についての研究は盛んに行われてきている．弾塑性体シミュレーションでは，メッシュ法 [10] やパーティクル法 [12] を用いた研究が多く行われてきた．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，変形によって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 [3][16].

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 [4][9]．Zhou ら [17] 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，計算可能な物体の形状に制限がある．さらに Stomakhin ら [1][2] 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

1 筑波大学
University of Tsukuba
a) saida@slis.tsukuba.ac.jp
b) fujis@slis.tsukuba.ac.jp
c) mikawa@slis.tsukuba.ac.jp

Vol.2015-CG-160 No.4
2015/8/29

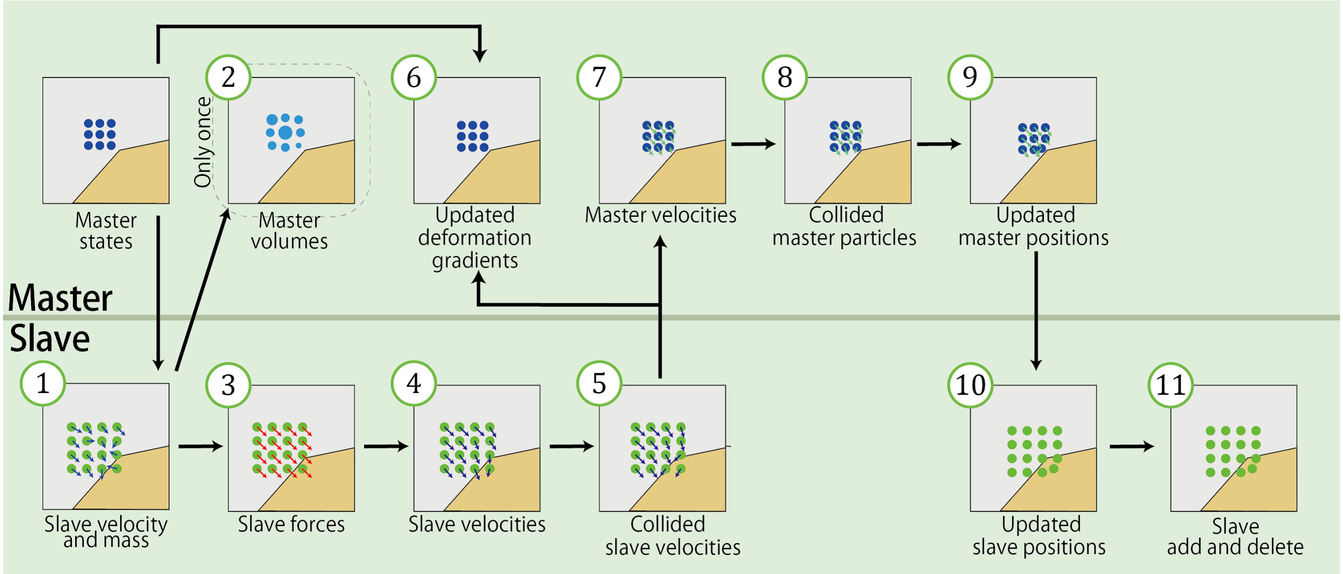


図 1　シミュレーションの流れ

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．

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，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．この問題に対して，Dyka ら [5][6] 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わかっている [14][15] 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．

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．3 次元空間における応力計算用パーティクルの移動や大変形に対しての追加・削除を行うことで，さまざまな変形の表現を可能とする．

3.　提案手法

本研究では，Stomakhin ら [2] の提案した計算手法に基づいて弾塑性変形の計算を行う．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 パーティクルとよぶ．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．計算式において $a_{i_m} = \sum_{j_s} b_{j_s}$ のような表記の場合，Master パーティクル i と有効半径 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．

シミュレーション全体の流れを図 1 に示す．以下，図 1 中に示したの各計算ステップ 1〜11 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**
近傍にある Master パーティクルを用いて，Slave パーティクルの質量 m_{i_s} と速度 $\boldsymbol{v}_{i_s}$ を求める．

$$m_{i_s} = \sum_{j_m} m_{j_m} w_{i_s j_m} \tag{1}$$

$$\boldsymbol{v}_{i_s} = \frac{1}{m_{i_s}} \sum_{j_m} \boldsymbol{v}_{j_m} w_{i_s j_m} \tag{2}$$

このとき $w_{i_s j_m}$ は Master パーティクル i と Slave パーティクル j の距離についての重みである．

2) **Master パーティクル体積の計算**
Master パーティクルの初期体積 $V^0_{i_m}$ を計算する．

$$\rho^0_{i_m} = \sum_{j_s} m^0_{j_s} w_{i_m j_s} \tag{3}$$

$$V^0_{i_m} = m_{i_m}/\rho^0_{i_m} \tag{4}$$

$\rho^0_{i_m}$ は初期密度である．この計算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 (§3.1)

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（§3.2）

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**
変形勾配の更新は以下のように計算する．

$$\mathbf{F}_{i_m}^{n+1} = (\mathbf{I} + \Delta t \nabla \boldsymbol{v}_{i_m}^{n+1})\mathbf{F}_{i_m}^{n} \tag{5}$$

このとき，$\nabla \boldsymbol{v}_{i_m}^{n+1}$ は速度勾配テンソルで以下のように計算する．

$$\nabla \boldsymbol{v}_{i_m}^{n+1} = \sum_{j_s} (\boldsymbol{v}_{i_m}^{n} - \boldsymbol{v}_{j_s}^{n+1})(\boldsymbol{x}_{i_m}^{n} - \boldsymbol{x}_{j_s}^{n})^T w_{i_m j_s} \tag{6}$$

弾性変形から塑性変形への変化について詳細は §3.3 で説明する．

7) **Master パーティクルの速度を計算**
近傍の Slave パーティクルの速度 $\boldsymbol{v}_{j_s}^{n+1}$ の最小自乗補間によって計算される速度 $\hat{\boldsymbol{v}}_{i_m}$ を用いて Master パーティクルの速度 $\boldsymbol{v}_{i_m}^{n+1}$ を計算する．

$$\hat{\boldsymbol{v}}_{i_m} = \arg\min_{\hat{\boldsymbol{v}}_{i_m}} \sum_{j_s} w_{i_m j_s} \|\hat{\boldsymbol{v}}_{i_m} - \boldsymbol{v}_{j_s}^{n+1}\|^2 \tag{7}$$

$$\boldsymbol{v}_{i_m}^{n+1} = (1-\alpha)\boldsymbol{v}_{i_m}^{n} + \alpha \hat{\boldsymbol{v}}_{i_m} \tag{8}$$

$\alpha \in [0, 1]$ は任意の定数である．

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．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**
衝突計算まで行って得られた速度 $\boldsymbol{v}_{i_m}^{n+1}$ から以下の式により位置を更新する．

$$\boldsymbol{x}_{i_m}^{n+1} = \boldsymbol{x}_{i_m}^{n} + \Delta t \boldsymbol{v}_{i_m}^{n+1} \tag{9}$$

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（§4.1）

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および削除を行う（§4.2,§4.3）

3.1 Slave パーティクルにかかる力の計算

Dyka ら [5][6] 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についての計算を行うことで，計算の安定化が可能になる．近傍 Master パーティクル j_m のもつ弾性変形勾配 $\mathbf{F}_{E_{j_m}}$ と塑性変形勾配 $\mathbf{F}_{P_{j_m}}$ から Slave パーティクルにかかる力 $\boldsymbol{f}_{i_s}$ は以下の式 (10) で計算される．

$$\boldsymbol{f}_{i_s} = -\sum_{j_m} V_{j_m}^{n} \sigma_{j_m} \nabla w_{i_s j_m} \tag{10}$$

このとき，Master パーティクルの n ステップ目における体積 V_m と，コーシー応力 σ_{j_m} は，

$$V_{j_m} = \det(\mathbf{F}_E \mathbf{F}_P) V_m^0 \tag{11}$$

$$\sigma_{j_m} = \frac{1}{J} \frac{\partial \Psi}{\partial \mathbf{F}_E} (\mathbf{F}_E)^T \tag{12}$$

と計算する．ここで，$\Psi(\mathbf{F}_E, \mathbf{F}_P)$ は [2] で定義されている弾塑性エネルギー密度関数である．また，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 に示す方法で更新する．

3.2 Slave パーティクルの速度

式 (10) で計算した力 $\boldsymbol{f}_{i_s}$ から，仮速度 $\boldsymbol{v}_{i_s}^{\star}$ を計算する．

$$\boldsymbol{v}_{i_s}^{\star} = \boldsymbol{v}_{i_s}^{n} + \Delta t m_{i_s}^{-1} \boldsymbol{f}_{i_s} \tag{13}$$

求めた仮速度から以下の線形システム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 $\boldsymbol{v}_{i_s}^{n+1}$ を計算する．

$$\sum_{j_s} \left(\mathbf{I}\delta_{i_s j_s} + \beta \Delta t^2 m_{i_s}^{-1} \frac{\partial^2 \Phi}{\partial \boldsymbol{x}_{i_s} \partial \boldsymbol{x}_{j_s}} \right) \boldsymbol{v}_{j_s}^{n+1} = \boldsymbol{v}_{i_s}^{\star} \tag{14}$$

3.3 変形勾配の更新

式 (14) で求めた Slave パーティクルの速度 $\boldsymbol{v}_{i_s}^{n+1}$ から，Master パーティクルの変形勾配を計算する．

まず，近傍 Slave パーティクルから速度勾配 $\nabla \boldsymbol{v}_m^{n+1}$ を計算する．

$$\nabla \boldsymbol{v}_{i_m}^{n+1} = \sum_{j_s} (\boldsymbol{v}_{i_m}^{n} - \boldsymbol{v}_{j_s}^{n+1})(\nabla w_{i_m j_s})^T \tag{15}$$

変形勾配は速度勾配を用いて以下のように更新される．

$$\mathbf{F}^{n+1} = \left(\mathbf{I} + \Delta t \nabla \boldsymbol{v}_m^{n+1}\right) \mathbf{F}^{n} \tag{16}$$

この変形勾配 $\mathbf{F}^{n+1}$ 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ため，これを降伏点で分離させる．

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

$$\hat{\mathbf{F}}_E^{n+1} = \left(\mathbf{I} + \Delta t \nabla \boldsymbol{v}_m^{n+1}\right) \mathbf{F}_E^{n} \tag{17}$$

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 $\hat{\mathbf{F}}_E^{n+1}$ を特異値分解（$\hat{\mathbf{F}}_E^{n+1} = \mathbf{U}\hat{\mathbf{\Sigma}}\mathbf{V}^T$）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．

$$\mathbf{\Sigma} = \text{clamp}\left(\hat{\mathbf{\Sigma}},\ [1-\theta_c, 1+\theta_s]\right) \tag{18}$$

式 (18) 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

$$\mathbf{F}_E^{n+1} = \mathbf{U\Sigma V}^T \tag{19}$$

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．

$$\begin{aligned}\mathbf{F}_P^{n+1} &= (\mathbf{F}_E^{n+1})^{-1}\mathbf{F}^{n+1} \\ &= (\mathbf{U\Sigma V}^T)^{-1}\hat{\mathbf{F}}_E^{n+1}\mathbf{F}_P^n \\ &= \mathbf{V\Sigma}^{-1}\mathbf{U}^T\hat{\mathbf{F}}_E^{n+1}\mathbf{F}_P^n\end{aligned} \tag{20}$$

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の Slave パーティクルの数が等しくなるように 3 次元空間に配置している．そのため，毎ステップ Slave パーティクルすべてを再生成すると計算時間が膨大になる．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

文献 [1][2] で用いられている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 $\boldsymbol{v}_m^{n+1}$ を用いる．近傍 Master パーティクルの速度から

$$\boldsymbol{v}_{i_s}^{n+1} = \frac{1}{m_{i_s}}\sum_{j_m} m_{j_m}\boldsymbol{v}_{j_m}^{n+1} w_{i_s j_m} \tag{21}$$

のように，Slave パーティクルの速度を計算する．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により，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 2 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．

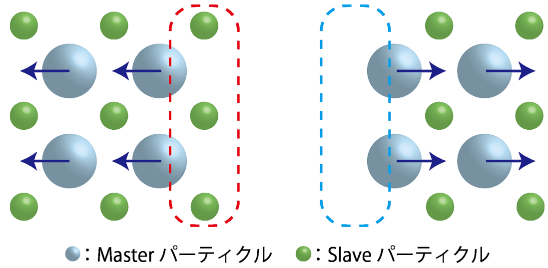


図 2 分裂した際に，左側の Master パーティクルに赤枠で囲まれた Slave パーティクルが追従し，青枠部分で Slave パーティクルの不足が発生する

本研究では，Master パーティクル半径 r を使って，初期状態において半径 $2.5r$ 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．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 3）．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を $\boldsymbol{x}_m, \boldsymbol{x}_s$ としたとき，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．

$$\boldsymbol{x}_s^{new} = \frac{n\boldsymbol{x}_s + m\boldsymbol{x}_m}{m+n} \tag{22}$$

m, n は任意の定数である．

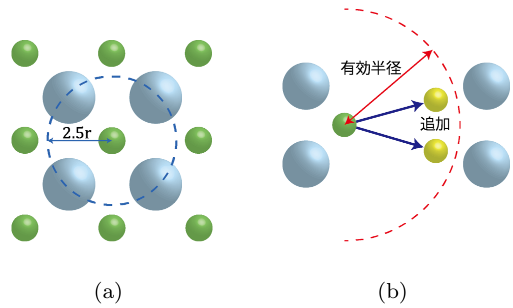
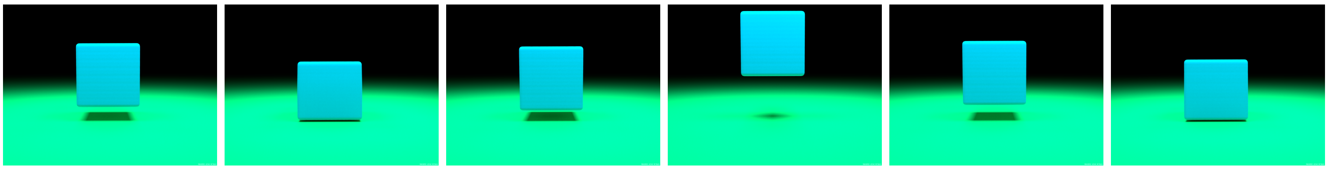


図 3 (a) 隣接する Master パーティクルの位置と (b) 隣接する Master パーティクルが有効半径 $2h$ よりも離れたとき Master パーティクルに近い位置に Slave パーティクルを追加する

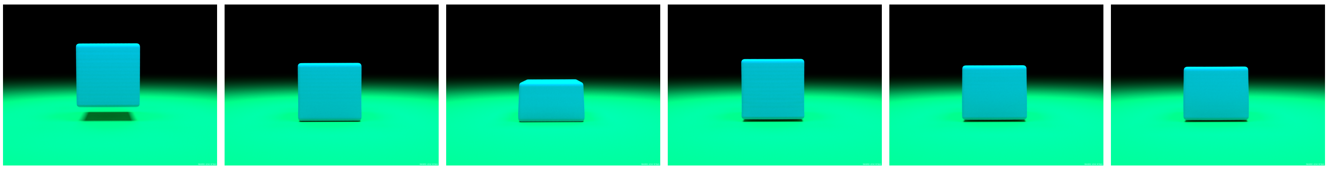
しかし距離だけで判断した場合，やわらかい弾性体のように大きな変形を伴う物体のシミュレーション時にも余分に生成されてしまう．なので，追加する位置における Slave パーティクルの密度が，初期状態における最大 Slave パーティクル密度よりも高い場合は追加しないという条件も加える．

表 1　シーンごとのパラメータおよびパーティクル数

シーン	タイムステップ幅 Δt	ポアソン比 ν	ヤング率 E[Pa]	降伏点 θ_s	降伏点 θ_c	Master	Slave
図 5(a)	0.001	0.3	6.9×10^7	なし	なし	4096	6647
図 5(b)	0.001	0.4	1.0×10^6	なし	なし		
図 6(a)	0.001	0.2	1.4×10^5	4.5×10^{-3}	1.5×10^{-2}		
図 6(b)	0.001	0.2	1.4×10^5	7.5×10^{-4}	2.5×10^{-3}		
図 7	0.0005	0.35	2.0×10^7	7.5×10^{-4}	2.5×10^{-3}	15120	21395
図 8	0.0005	0.4	2.0×10^7	7.5×10^{-4}	2.5×10^{-3}	5547	12320



(a) $\nu = 0.3, E = 6.9 \times 10^7$



(b) $\nu = 0.4, E = 1.0 \times 10^6$

図 5　ポアソン比 ν とヤング率 E を変化させることで，さまざまな弾性変形表現ができる

4.3　Slave パーティクルの削除

Slave パーティクルは Master パーティクルに追従させて移動する．物体が結合する場合，図 4 のように Master パーティクル同士が近づくため，間にある Slave パーティクルの数が多くなってしまう．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．

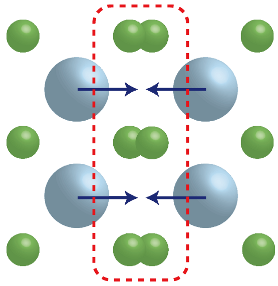
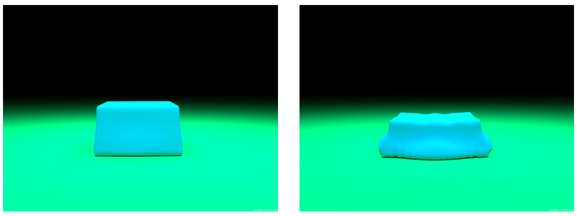


図 4　赤枠で囲まれた部分の Slave パーティクルが過密になっている

5.　実験結果と考察

実行環境は，CPU:Intel Corei7-4930K 3.40GHz である．メッシュの生成には OpenVDB[13] を使い，Mitsuba[11] でレンダリングを行っている．各シーンのパラメータと Master パーティクルおよび Slave パーティクルの数を表 1 に示す．



(a) $\theta_s = 1.5 \times 10^{-2}$ $\theta_c = 4.5 \times 10^{-3}$

(b) $\theta_s = 2.5 \times 10^{-3}$ $\theta_c = 7.5 \times 10^{-4}$

図 6　降伏点 θ_c, θ_s によって弾塑性表現を変化させることができる

図 5，6 は，物性値を変化させたときのシミュレーション結果である．図 5 は，ポアソン比 μ とヤング率 E によって弾性体表現におけるやわらかさを変化させられること示している．図 5(a) では，硬い弾性体を表現したため衝突した後大きく跳ねていることがわかる．一方，図 5(b) ではやわらかい弾性体を表現しているので衝突した後大きく変形していることがわかる．

図 6 では，降伏点 θ_c, θ_s を変化させることでさまざまな塑性変形の表現が可能であることを示している．図 6(a) よりも図 6(b) のほうが降伏点が小さいため，大きく塑性変形していることがわかる．

図 7 と図 8 はそれぞれ異なる形状の障害物に衝突させた場合のシミュレーション結果である．図 7 は障害物に衝突したあと分離する様子を示している．Slave パーティクルの追加によって分離表現が可能であることがわかる．図 8

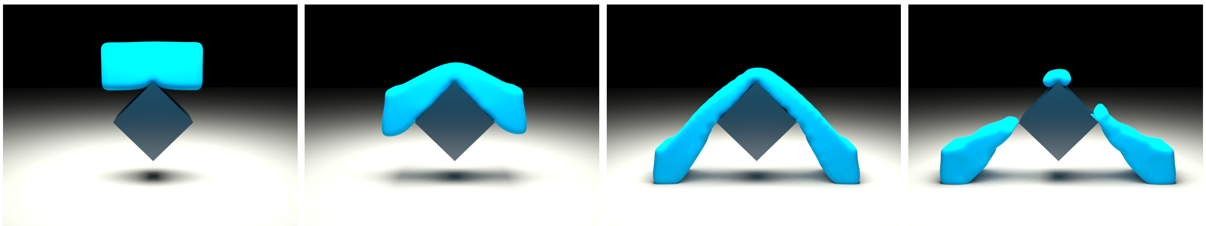


図 7　Slave パーティクルを追加することで物体が分離するような表現が可能である

は障害物の形状を変えて，分離した物体同士が結合するようにしたシーンである．障害物に衝突し分離したあと，先に分離した部分と結合する様子が表現できている．

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提案した．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せることで，計算空間に制限されない安定した計算を実現した．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 3 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，現在までは基本形状の物体でのみ安定した計算が行えることを確認した．今後はさまざまな形状の物体について，安定した計算が可能であるか検証する必要がある．

参考文献

[1] A.Stomakhin, C.Schroeder, C.Jiang, L.Chai, J.Teran and A.Selle: Augmented MPM for phase-change and varied materials, *ACM Transactions on Graphics*, Vol. 33, No. 4 (2014).

[2] A.Stomakhin, C.Schroeder, L.Chai, J.Teran and A.Selle: A material point method for snow simulation, *ACM Transactions on Graphics*, Vol. 32, No. 4, p. 1 (2013).

[3] Bargteil, A. W., Wojtan, C., Hodgins, J. K. and Turk, G.: A Finite Element Method for Animating Large Viscoplastic Flow, *Computer Graphics Proceedings, Annual Conference Series, 2007* (2007).

[4] B.Jones, S.Ward, A.Jallepalli, J.Perenia and A.W.Bargteil: Deformation embedding for point-based elastoplastic simulation, *ACM Transactions on Graphics*, Vol. 33, No. 2 (2014).

[5] C.T.Dyka, P.W.Randles and R.P.Ingel: Stress Points for Tension and Instability in SPH, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 40, 2325-2341* (1997).

[6] C.T.Dyka and R.P.Ingel: An Approach for Tension Instability in Smoothed Particle Hydrodynamics, Technical report, Computer and Structures Vol.57, No.4, pp.573-580 (1995).

[7] D.Terzopoulos, J.Platt, A.Barr and K.Fleischer: Elastically Deformable Models, *SIGGRAPH '87 Proceedings of the 14th annual conference on Computer graphics and interactive techniques, 2005-214* (1987).

[8] D.Terzopoulos and K.Fleischer: Modeling Inelastic Dformation: Viscoelasticity, Plasticity, Fracture, *In Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'88), 269-278* (1988).

[9] Gerszewski, D., Bhattacharya, H. and Bargteil, A. W.: A Point-based and Method for Animating and Elastoplastic Solids, *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)* (2009).

[10] G.Irving, J.Teran and R.Fedkiw: Invertible Finite and Elements For and Robust Simulation and of Large and Deformation, *Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2004)* (2004).

[11] Jakob, W.: Mitsuba Physically Based Renderer (2015). http://www.mitsuba-renderer.org/index.html.

[12] M.Müller, R.Keiser, A.Nealen, M.Pauly, M.Gross and M.Alexa: Point Based Animation of Elastic, Plstic and Melting Object, *Eurographics/ACM SIGGRAPH Symmposium on Computer Animation* (2004).

[13] Museth, K.: VDB High-Resolution Sparse Volumes with Dynamic Topology, *ACM Transactions on Graphics*, Vol. 32, No. 3, pp. 1–22 (2013).

[14] R.Vignjevic, J.Campbell and L.Libersky: A treatment of

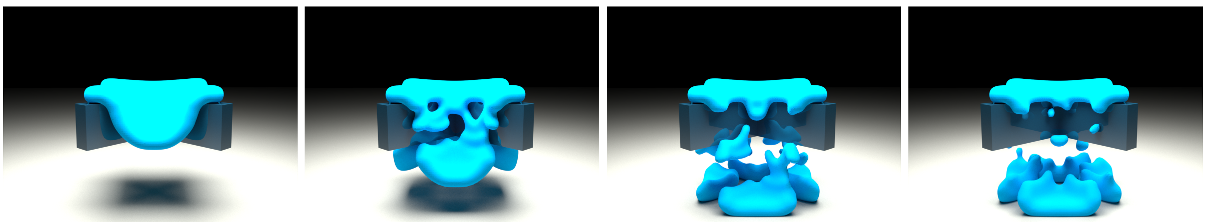


図 8　Slave パーティクルの削除によって物体同士が結合する表現が可能である

zero-energy modes in the smoothed particle and hydrodynamics method, *Comput. Methods Appl. Mech. Engrg. 184 67-85* (2000).

[15] SHAO, Y., ITO, H., SHIBATA, K. and KOSHIZUKA, S.: An Hamiltonian MPS formulation for Reissner-Mindlin shell, *Transactions of JSCES, Paper No.20120013* (2012).

[16] Wicke, M., Ritchie, D., Klingner, B. M., Burke, S., Shewchuk, J. R. and F.OBrien, J.: Dynamic Local Remeshing for Elastoplastic Simulation, *Computer Graphics Proceedings, Annual Conference Series, 2010* (2010).

[17] Zhou, Y., Lun, Z., Kalogerakis, E. and Wang, R.: Implicit Integration for Particle-based Simulation of Elastoplastic Solids, *Pacific Graphics 2013. Computer Graphics Forum, Vol.32, No.7* (2013).