Vol.2019-CG-175 No.5
2019/9/28

アダプティブな位置ベースの物理シミュレーション

梅谷　信行[1,a)]　小山　裕己[2]　中川　展男[1,3]　蜂須賀 恵也[1]

概要：位置ベース力学 (Position-based Dynamics) は、その計算効率の良さと実装の単純さから、ゲームなどの対話的なアプリケーションで広く用いられている物理シミュレーション手法の一つである。従来の位置ベース力学ではメッシュの離散化は不変であることが前提とされていたが、本論文では、更に計算の効率を向上するために、メッシュの離散化を動的に切り替えるアダプティブな拡張を提案する。従来の位置ベース力学ではメッシュの解像度を高くすると変形が柔らかくなるなどの、解が解像度に依存するという問題があったが、本論文は変分法を取り入れてこの問題を解決した。

キーワード：Position-based Dynamics, 位置ベース力学、アダプティブ・シミュレーション

1. はじめに

アダプティブなシミュレーションとは、計算力学において一般的に広く用いられる方法で、計算メッシュの密度を解の変化が大きな箇所が細かくなるように、変化が小さな箇所では粗くなるように計算途中に逐次的に調整することで、精度を保ちつつ、計算の自由度を減らすことで効率を向上することができる。有限要素法や、境界要素法など、多くのシミュレーション手法においてアダプティブなシミュレーション法が適応されている。しかしながら、これらの手法はメッシュを変化させると、それに伴い、係数行列を再構築する計算コストが発生するので、メッシュを切り替えることで得られる計算速度の向上は限定的である。

位置ベース力学（Position-based Dynamics,PBD）はゲームなどの対話的なアプリケーションによって広く用いられる物理シミュレーション手法である。位置ベース力学は行列の計算が必要ないので、メッシュが変化した時に余計にかかる計算コストが非常に低い。筆者らはこの性質に着目して、アダプティブシミュレーションを位置ベース力学に適応する手法について模索した。

位置ベース力学は、質点の集合とそれらに課される制約条件から構築させる。計算メッシュが変化すると一般的に質点の分布や、制約条件が変化してしまうために、メッシュを変化させることは難しかった。

この課題に解決するために筆者らは変分法に基づいた位置ベース力学を提案する。変分法とは汎関数を最小化することで解を得る手法であるが、この汎関数をメッシュに依存しないように定義することで、メッシュが変化しても解が変化しないようにすることが可能である。

2. 関連研究

本論文では、我々は位置ベース法のフレームワーク上でアダプティブなシミュレーションをする手法を提案する。位置ベース法については、最近のサーベイ論文を参考にされたい [2]. コンピュータグラフィックスにおいて、アダプティブ・シミュレーションは色々な力学シミュレーション法に応用されている [10]、しかしながら位置ベース法については未だに適応された例はない。

コンピュータ・グラフィックスにおいてアダプティブ・シミュレーションの重要な応用例は布のシミュレーションである。Narain らによって衣服のアダプティブなシミュレーション [15] が提案された後、アダプティブシミュレーションは折り紙 [14] や、視点によって解像度を変える高速化手法 [5] や摩擦を含む布のシミュレーション [6] などに応用されてきた。

我々の時間積分の方法は、変分陰的オイラー法 [11] に基づいている。このような変分原理に基づく時間積分の方法を取り入れた手法は他に ADMM [16] や Projective Dynamics 法 [3] などが挙げられる。PBD と変分陰的オイラー法との関係は Liu ら [7] によって最初に指摘され、後に Bouaziz らがによって更に議論された [3]。

XPBD [9] は陰的オイラー法に基づいて、位置ベース法を拡張したもので、元の位置ベース法に意味のある物性値

[1] 東京大学
[2] 産業技術総合研究所
[3] 株式会社ポリフォニー・デジタル
a) n.umetani@gmail.com

Vol.2019-CG-175 No.5
2019/9/28

を使えたり、解が時間ステップへに非依存性などの性質を持たせることができる。我々は XPD をさらに変分原理に基づく陰的オイラー法を適応することにより、アダプティブなシミュレーションに適応させる。変分原理では汎関数の最小化により解を求めており、解の一貫性を保ったままメッシュを変化させることができる。本手法と XPB との顕著な違いは以下の 2 つに要約される：XPB では陰的オイラー法を線形化することで全ての自由度を一度にまとめて更新するが、我々は (i) ブロック・ガウス・ザイデル法を用いてある要素に関係する自由度のみについて線形化を行い、逐次的に解を更新する。その際に (ii) 要素に関係する複数の制約条件を同時に満たすように解を更新することで、汎関数の最適化の計算を効率化する。その他、本論文ではメッシュの解像度に依存しない制約条件の構築方法についても提案する。

Projective Dynamics 法 [3] や、準ニュートン法を使ったその一般化 [8] は、位置ベース法と比較してより速い収束を実現するが、大規模行列の逆行列を前計算する必要がある。従って、メッシュの接続関係や頂点数が動的に変化するような場合は高いコストの再計算が必要になるので、適していない。

Algorithm 1 Our simulation loop

1: predict positions $\bar{\mathbf{x}} = \mathbf{x^n} + \mathbf{\Delta t v^n} + \mathbf{\Delta t^2 g}$
2: $\mathbf{x} = \bar{\mathbf{x}}$
3:
4: // minimize $\mathcal{W}$ with the Gauss-Seidel method
5: **for** each iteation **do**
6:     **for** each element $e \in \mathcal{E}$ **do**
7:         // minimize $\mathcal{W}$ element-wise with the Newton's method
8:         solve $\Delta \boldsymbol{\lambda}_e$ ▷ 9 in 3.3
9:         compute $\Delta \mathbf{x_e}$ ▷ 8 in 3.3
10:         update points $\mathbf{x} := \mathbf{x} + \mathbf{\Delta x_e}$
11:     **end for**
12: **end for**
13:
14: $\mathbf{x^{n+1}} = \mathbf{x}$
15: $\mathbf{v^{n+1}} = (\mathbf{x} - \mathbf{x^n})/\mathbf{\Delta t}$
16:
17: **if** remeshing is necessary **then**
18:     perform remeshing
19:     interpolate values for newly added points
20: **end if**

3. 変分陰的オイラー法に基づく位置ベース法

本手法では、時間積分に変分陰的オイラー法 (Variational Implicit Euler Method) [3], [4], [7], [11], [16] を時間積分に使用する。変分陰的オイラー法では現時点での解から次のステップにおける解を汎関数最小化によって求める。

変分陰的オイラー法では汎関数を最小化することで次の時間ステップにおける解を得る 。この汎関数 $\mathcal{W}$ は要素あたりの弾性ポテンシャル・エネルギー $\mathcal{U}_e$ の全要素の合計と運動エネルギー $\mathcal{K}$ の和で表すことができる。

$$\mathcal{W} = \mathcal{K} + \sum_{e \in \mathcal{E}} \mathcal{U}_e \qquad (1)$$

本論文ではまず、弾性エネルギーについて、第 3.1 章で述べた後に、第 3.2 章で運動エネルギーについて説明し、最後に変分オイラー法を使った解の更新方法について解説する。この汎関数を計算メッシュにできるだけ依存しないよう定義することで、アダプティブシミュレーションにおいて計算メッシュの切り替えから生じる影響を小さくする。

アルゴリズムの概要を図 1 に示す。各時間ステップにおいて、まず中間的な速度の予測 $\bar{\mathbf{x}}$ を計算する (line 1 in 1)。その後、中間的な速度 $\bar{\mathbf{x}}$ を汎関数を最小化することで $\mathbf{x}$ に更新する (line 4–13 in 1). 最終的に陰的オイラー法によって次の時間ステップにおける速度を計算する (line 16 in 1).

本論文では XPBD [9] の論文で使われている記法を用いている。この論文ではメッシュに属する三角形や辺などの要素の集合として、$\mathcal{E}$ という記号を用いる。また、ある量に下付き文字をつけると、e その量は要素ごとに定義された値を表すとする。例えば $\mathbf{x_e}$ は要素 e に属する節点位置を表している。

3.1 弾性ポテンシャルエネルギー

$\mathbf{C_e}$ を要素 e に関係する制約条件だとする。ここで、$\mathbf{C_e}$ はスカラー値であっても、ベクトル値であっても良いとする。Macklin らによる XPB [9] の論文と同様に、次のように弾性ポテンシャルエネルギー $\mathcal{U}$ をメッシュの中の全ての要素 $e \in \mathcal{E}$ の二乗和と定義する：

$$\mathcal{U} = \sum_{e \in \mathcal{E}} \mathcal{U}_e, \quad \text{where} \quad \mathcal{U}_e = \frac{1}{2} \mathbf{C_e^T}(\mathbf{x_e}) \boldsymbol{\alpha}_\mathbf{e}^{\mathbf{-1}} \mathbf{C_e}(\mathbf{x_e}), \qquad (2)$$

ここで、$\mathcal{U}_e$ は要素 e の内部のポテンシャルエネルギーを表している。また、$\boldsymbol{\alpha}_e$ はコンプライアンスとと呼ばれる、柔らかさを表す行列である。

汎関数がメッシュの影響を受けないようにするためには要素あたりの弾性ポテンシャルエネルギー $\mathcal{U}_e$ は、要素の基準配置における体積 V_e に比例するように定義されなければならない $\mathcal{U}_e \propto V_e$。この条件を満たすことによって、メッシュの解像度が変化しても、弾性ポテンシャルエネルギーへの影響を少なくすることができる。

要素弾性ポテンシャルエネルギーが $\mathcal{U}_e$ 体積に比例するためには、要素の制約 $\mathbf{C_e}$ の全ての要素が、要素体積の根に反比例しなければならない $\mathbf{C_e} \propto \sqrt{\mathbf{V_e}}$。この要求は一般的に用いられる位置ベース力学の制約 [2] を単純にスケールすることで得られる。この論文では三角形メッシュを用いた服の位置ベース・シミュレーションでよく用いられる制約についてアダプティブシミュレーションをするために、どのよう制約をスケールするかについて解説する。

Vol.2019-CG-175 No.5
2019/9/28

距離の制約.

距離の制約は2点間の距離が初期値と等しいという制約条件で [13]、位置ベースのシミュレーションでは頻繁に用いられる。三角形メッシュの場合には、アダプティブ・シミュレーションを実現するために拘束条件をスケールする必要ない。なぜなら辺の長さは既に要素体積の根 $\sqrt{V_e}$ に比例しているからである。要素に属する三つの辺の制約を一つのベクトル形式にまとめることによって、三角形への制約は次のように書くことができる。

$$\mathbf{C}_e = \{l_{12} - L_{12},\ l_{23} - L_{23},\ l_{31} - L_{31}\}^T \tag{3}$$

ここで、L_{ij} と l_{ij} は、それぞれ点 i と j をつなぐ辺の変形前と後の長さである (図 1)。

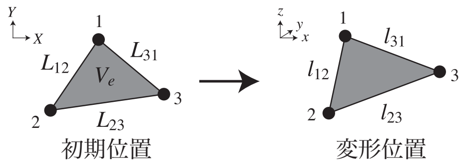

図 1 三角形に定義された拘束条件

歪みの制約.

Müller らはグリーン・ラグランジュ歪みがゼロであるという、に基づく制約条件を提案した [12]。歪みは無次元量であり、要素の体積に依存しない。そこで、Servin らの手法 [17] と同様に我々は制約条件を体積の根 $\sqrt{V_e}$ によってスケールする。三角形の変形前の座標が二次元である場合は次のように制約を書くことができる

$$\mathbf{C}_e = \sqrt{V_e}\,\{E_{XX}, E_{YY}, E_{XY}\}^T, \tag{4}$$

ここで、E_{XX}, E_{YY}, E_{XY} はグリーン・ラグランジュ歪み $\mathbf{E}$ の、参照配置における X 軸と Y 軸に関する要素である。ここで、コンプライアンス行列 $\boldsymbol{\alpha}$ は例えば、平面応力状態を仮定してた線形弾性体の応力-歪み行列を使って定義することができる [9]。

エネルギー制約.

Bender らによって提案されたエネルギーに基づく制約 [1] は弾性エネルギーがゼロであるという制約をかけるものである。要素内での弾性エネルギーは要素内における弾性エネルギー密度の積分であるので、我々はそれらを係数 $1/\sqrt{V_e}$、でスケールする。サンブナン・キルヒホッフ (Saint Venant-Kirchhoff,StVK) モデルの場合は、スケール後の制約は次のように書くことができる

$$\mathbf{C}_e = \frac{1}{\sqrt{V_e}} \int_{V_e} \mathbf{S} : \mathbf{E} \mathbf{d} \mathbf{V} \tag{5}$$

ここで、$\mathbf{S}$ は第二ピオラ・キルヒホッフ応力テンソルで, $\mathbf{E}$ はグリーン・ラグランジュ歪テンソルを表している。また、$\mathbf{A} : \mathbf{B}$ はテンソル積 $\mathrm{tr}(\mathbf{A}^T\mathbf{B})$ を表すとする。

3.2 運動エネルギー

物体の運動エネルギー $\mathcal{K}$ は次のように書くことができる。

$$\mathcal{K} = \int_V \frac{1}{2}\rho \mathbf{v}^T \mathbf{v} dV = \frac{1}{2}\mathbf{v}^T \mathbf{M} \mathbf{v} \tag{6}$$

ここで ρ は基準配置における質量密度である。この $\mathbf{M_e}$ は整合質量行列 (consistent mass matrix) [19] と呼ばれるが、本論文では、非対角要素を対角に足し合わせることで対角化した、対角化質量行列 (lumped mass matrix)$\tilde{\mathbf{M}}$ を考える。ここで、陰的オイラー法において、次の時間における速度は位置の更新量を元に $(\mathbf{x} - \bar{\mathbf{x}})/\mathbf{\Delta t}$ のように計算されるということを用いると、

$$\mathcal{K} = \frac{1}{2\Delta t^2}(\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{T}} \tilde{\mathbf{M}} (\mathbf{x} - \bar{\mathbf{x}}) \tag{7}$$

のように運動エネルギーを書くことができる。

3.3 変分原理に基づく解の更新

我々の手法では、汎関数 $\mathcal{W}$ の最小化をブロック・ガウス・ザイデル法によって最小化する。つまり、全ての要素について順番に、要素単位で汎関数 $\mathcal{W}_e$ の最小化を行うことで、汎関数を最小化する。

我々の手法では、XPBD [9] と同様に、頂点位置を要素単位で以下のように更新する。

$$\Delta \mathbf{x_e} = \tilde{\mathbf{M}}_{\mathbf{e}}^{\mathbf{-1}} \frac{\partial \mathbf{C_e}}{\partial \mathbf{x_e}}^{\mathbf{T}} \mathbf{\Delta} \boldsymbol{\lambda}_{\mathbf{e}}, \tag{8}$$

ここで $\boldsymbol{\lambda}_e$ は要素ごとの制約を集めたベクトルで、$\boldsymbol{\lambda}_e = -\Delta t^2 \boldsymbol{\alpha}_e^{-1} \mathbf{C_e}$ のように定義される。このように更新することで、並進運動量や回転運動量を保存することができる。

汎関数 $\mathcal{W}$ を最小化する際に、$\boldsymbol{\lambda}_e$ を独立変数として選んで、ニュートン法を適応する。つまり、$\boldsymbol{\lambda}_e$ の更新は次のように書くことができる

$$\Delta \boldsymbol{\lambda}_e = -\left(\frac{\partial^2 \mathcal{W}_e}{\partial \boldsymbol{\lambda}_e^2}\right)^{-1} \frac{\partial \mathcal{W}_e}{\partial \boldsymbol{\lambda}_e}, \tag{9}$$

ここで、汎関数 $\mathcal{W}$ の $\boldsymbol{\lambda}_e$ に対する勾配ベクトルとヘシアン行列は次のように計算される。

$$\frac{\partial \mathcal{W}_e}{\partial \boldsymbol{\lambda}_e} = \frac{1}{\Delta t^2}\left[(\mathbf{x}_e - \bar{\mathbf{x}}_e)^T \frac{\partial \mathbf{C_e}}{\partial \mathbf{x}_e}^T + \mathbf{C}_e\right] \tag{10}$$

$$\frac{\partial^2 \mathcal{W}_e}{\partial \boldsymbol{\lambda}_e^2}, = \frac{1}{\Delta t^2}\left[\frac{\partial \mathbf{C}_e}{\partial \mathbf{x_e}} \tilde{\mathbf{M}}_e^{-1} \frac{\partial \mathbf{C}_e}{\partial \mathbf{x_e}}^T + \frac{\boldsymbol{\alpha}_e}{\Delta t^2}\right]. \tag{11}$$

ここで、本手法で計算されるヘシアン行列 (11) は正定値対称であることに注意されたい。また、この行列の次元は N_c を要素内の制約とした場合、$N_c \times N_c$ で、三角形要素を使った場合、高々三次元であり、逆行列を計算するコストは極めて低い。

頂点位置の更新は $\Delta \mathbf{x_e}$ 独立変数 $\Delta \boldsymbol{\lambda}_e$ に式 (8) のように依存するので、$\Delta \boldsymbol{\lambda}_e$ についての最小化問題を解けば良い。ここでは、拘束条件を一つづつ最適化するのではなく、

図 2 異なる制約を用いた場合の布が垂れ下がるのシミュレーション結果の比較

Umetani らによる研究 [18] のように要素内で定義されている複数の拘束条件を一緒に最適化することで、より効率の良い収束を実現する。

4. 結果

図 2 に異なる制約条件を用いた場合に、布の垂れ下がりのシミュレーションにおいて解像度を変えた時に、どのように結果が影響したかを示す。解像度を変えた場合に、本論で提案したように制約条件をスケールした場合は、シミュレーション結果が大きく影響を受けることはないことが確認された。

5. おわりに

本論文では効率の高いアダプティブ・シミュレーションを実現するために、変分原理に基づいた位置ベース力学を提案した。今後は動的にメッシュを切り替えた際の解への影響などを、さまざまな例題のシミュレーションを通して確認したいと考えている。

参考文献

[1] Bender, J., Koschier, D., Charrier, P. and Weber, D.: Position-based simulation of continuous materials, *Comput. Graph.*, Vol. 44, pp. 1–10 (online), DOI: 10.1016/j.cag.2014.07.004 (2014).

[2] Bender, J., Müller, M. and Macklin, M.: A Survey on Position Based Dynamics, *EG 2017 - Tutorials*, (online), DOI: 10.2312/egt.20171034 (2017).

[3] Bouaziz, S., Martin, S., Liu, T., Kavan, L. and Pauly, M.: Projective Dynamics: Fusing Constraint Projections for Fast Simulation, *ACM Trans. Graph.*, Vol. 33, No. 4, pp. 154:1–154:11 (online), DOI: 10.1145/2601097.2601116 (2014).

[4] Gast, T. F., Schroeder, C., Stomakhin, A., Jiang, C. and Teran, J. M.: Optimization Integrator for Large Time Steps, *IEEE Trans. Vis. Comput. Graph.*, Vol. 21, No. 10, pp. 1103–1115 (online), DOI: 10.1109/TVCG.2015.2459687 (2015).

[5] Koh, W., Narain, R. and O'Brien, J. F.: View-Dependent Adaptive Cloth Simulation with Buckling Compensation, *IEEE Trans. Vis. Comput. Graph.*, Vol. 21, No. 10, pp. 1138–1145 (online), DOI: 10.1109/TVCG.2015.2446482 (2015).

[6] Li, J., Daviet, G., Narain, R., Bertails-Descoubes, F., Overby, M., Brown, G. E. and Boissieux, L.: An Implicit Frictional Contact Solver for Adaptive Cloth Simulation, *ACM Trans. Graph.*, Vol. 37, No. 4, pp. 52:1–52:15 (online), DOI: 10.1145/3197517.3201308 (2018).

[7] Liu, T., Bargteil, A. W., O'Brien, J. F. and Kavan, L.: Fast Simulation of Mass-spring Systems, *ACM Trans. Graph.*, Vol. 32, No. 6, pp. 214:1–214:7 (online), DOI: 10.1145/2508363.2508406 (2013).

[8] Liu, T., Bouaziz, S. and Kavan, L.: Quasi-Newton Methods for Real-Time Simulation of Hyperelastic Materials, *ACM Trans. Graph.*, Vol. 36, No. 3 (online), DOI: 10.1145/2990496 (2017).

[9] Macklin, M., Müller, M. and Chentanez, N.: XPBD: Position-based Simulation of Compliant Constrained Dynamics, *Proc. MIG '16*, pp. 49–54 (online), DOI: 10.1145/2994258.2994272 (2016).

[10] Manteaux, P.-L., Wojtan, C., Narain, R., Redon, S., Faure, F. and Cani, M.-P.: Adaptive Physically Based Models in Computer Graphics, *Comput. Graph. Forum*, Vol. 36, No. 6, pp. 312–337 (online), DOI: 10.1111/cgf.12941 (2017).

[11] Martin, S., Thomaszewski, B., Grinspun, E. and Gross, M.: Example-based Elastic Materials, *ACM Trans. Graph.*, Vol. 30, No. 4, pp. 72:1–72:8 (online), DOI: 10.1145/2010324.1964967 (2011).

[12] Müller, M., Chentanez, N., Kim, T.-Y. and Macklin, M.: Strain Based Dynamics, *Proc. SCA '14*, pp. 149–157 (online), DOI: 10.2312/sca.20141133 (2014).

[13] Müller, M., Heidelberger, B., Hennix, M. and Ratcliff, J.: Position based dynamics, *J. Vis. Commun. Image Represent.*, Vol. 18, No. 2, pp. 109–118 (online), DOI: 10.1016/j.jvcir.2007.01.005 (2007).

[14] Narain, R., Pfaff, T. and O'Brien, J. F.: Fold-

ing and Crumpling Adaptive Sheets, *ACM Trans. Graph.*, Vol. 32, No. 4, pp. 51:1–51:8 (online), DOI: 10.1145/2461912.2462010 (2013).

[15] Narain, R., Samii, A. and O'Brien, J. F.: Adaptive Anisotropic Remeshing for Cloth Simulation, *ACM Trans. Graph.*, Vol. 31, No. 6, pp. 152:1–152:10 (online), DOI: 10.1145/2366145.2366171 (2012).

[16] Overby, M., Brown, G. E., Li, J. and Narain, R.: ADMM $\supseteq$ Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints, *IEEE Trans. Vis. Comput. Graph.*, Vol. 23, No. 10, pp. 2222–2234 (online), DOI: 10.1109/TVCG.2017.2730875 (2017).

[17] Servin, M., Lacoursière, C. and Melin, N.: Interactive Simulation of Elastic Deformable Materials, *Proc. SIGRAD '06*, pp. 22–32 (online), available from ⟨http://www.ep.liu.se/ecp/article.asp?issue=019article=005⟩ (2006).

[18] Umetani, N., Schmidt, R. and Stam, J.: Position-based Elastic Rods, *Proc. SCA '14*, pp. 21–30 (online), DOI: 10.2312/sca.20141119 (2014).

[19] Zienkiewicz, O. C. and Taylor, R. L.: *The Finite Element Method for Solid and Structural Mechanics*, Butterworth-Heinemann, 6th edition (2005).