

XPBD: 拘束動力学準拠の位置ベース物理シミュレーション

Miles Macklin

Matthias Müller

Nuttapong Chentanez

NVIDIA

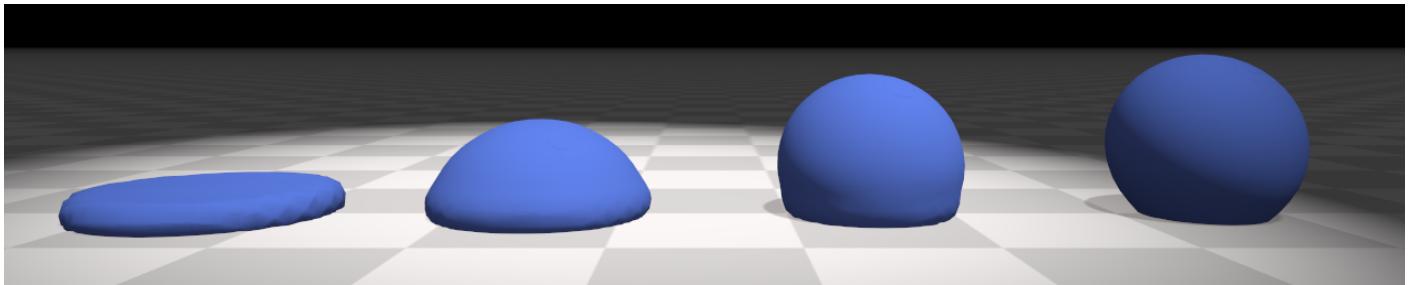


図1:この例では、弾性体の体積保存拘束、ストレッチ拘束、せん断拘束の相対剛性値を変化させたときの効果を見ることができます。通常の PBD と異なり、我々の手法では、ユーザーが弾性体のタイムステップとイテレーション回数に依存しない形で制御することができます。このことでアセット制作がとても簡単にになります。

概要

我々は、イテレーション回数とタイムステップの大きさによって位置ベース物理(PBD)の拘束の剛性値が変化してしまう問題に長い間取り組んできました。我々は、PBD にシンプルな拡張を加える手法を提案します。この手法は、正確で効果的に任意の弾性的、散逸的なエネルギーを陰的に解くことを可能とします。加えて、我々の手法は、拘束による力の見積もりを求める事ができるので、より広い領域での応用を可能とします。例えば、触覚フィードバックが求められる場面で効果的です。

我々のアルゴリズムと、より計算コストの高い非線形ソルバとを比較することで、PBD のシンプルさと堅牢さを保ったまま、視覚的に似た結果を生み出す事がわかりました。

Keywords: physics simulation, constrained dynamics, position based dynamics

Concepts: •Computing methodologies → Real-time simulation; Interactive simulation;

1 導入

Position-Based Dynamics [Müller et al. 2007] は、ゲームやインタラクティブアプリケーションの変形物体のリアルタイムシミュレーションでよく知られている手法です。この手法は、特にシンプルさとロバストさで特に魅力的で、近年は、映画や医療シミュレーションなどゲーム以外の分野でもよく知られています。

その人気が高まるにつれ、PBDの制限の問題もより大きくなってきました。よく知られた制限は、PBDの振る舞いが、タイムステップのタイムステップの大きさとシミュレーションのイテレーション回数に

依存していることです [Bender et al. 2014b]。特に、イテレーション回数が増えたり、タイムステップが減少した際に拘束が任意の堅さになってしまいます。このパラメータの組み合わせは、特に様々な種類のマテリアルがある場合に問題になります。例えば、近くにある硬い剛体とインタラクションするソフトボディなどです。このシナリオでは、あるオブジェクトの剛性値を調整するためにイテレーション回数を増やすと、他のあらゆるオブジェクトの振る舞いが意図せず変わってしまうのです。このことで、剛性値の係数をグローバルにしばしば再調整する必要が生じます。再利用可能なシミュレーションアセットを作るのは極めて難しくなります。単一のアセットでも、イテレーション回数依存は問題です。例えば、クロスのモデルの伸びと曲げの拘束の相対的な堅さを変えたい時などです。状況を更に悪くすることにイテレーション回数の降低は非線形で、パラメータの直感的に調整が難しいのです。剛性値をシンプルに調整する事も同様に難しいです。

近年のバーチャルリアリティの復興に伴い、高い忠実性やリアルタイムシミュレーションでの物理表現の改善欲求が高まってまいりました。同時に、触覚フィードバック機器の広範囲にわたる利用により、正確な力の推定が可能な手法が求められるようになってきました。PBD は、きちんと定義された拘束力のコンセプトを持っているわけではありません。そして、そのことが正確さが速度よりも重要ではない領域かシミュレーションが二次的な効果と考えられる場合での利用に制限されていました。

この論文では、拡張された Position-Based Dynamics (XPBD) アルゴリズムを提案します。我々の手法は、明確な弾性ポテンシャルエネルギーのコンセプトに対応した新しい拘束式の導入によりイテレーションとタイムステップ依存で剛性値が変わってしまう問題の解決に注力しています。PBD の総ラグランジュ乗数のコンセプトの導入により、陰的な時間離散する手法を導きました。この手法により、力に依存した効果や機器での利用が可能な拘束力の推定が可能となりました。

まとめると、我々の主要な貢献は:

- PBD の拘束を拡張して明確な弾性エネルギーや消失ポテンシャル力との直接対応を可能とした。
- PBD へ総ラグランジュ乗数の概念を導入することで、タイムステップとイテレーション回数に依存せずに解くことを可能としました。
- 陰解法タイムステップスキームとして非線形ニュートンソルバに基づいた手法と比較しアルゴリズムの検証を行いました。

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.

MiG '16, October 10 - 12, 2016, Burlingame, CA, USA

ISBN: 978-1-4503-4592-7/16/10

DOI: <http://dx.doi.org/10.1145/2994258.2994272>

2 関連研究

コンピュータグラフィックスの分野で拘束力学シミュレーションの課題に取り組んだ多くの先行研究があります。[Nealen et al. 2005]。我々の手法は、Müllerら[2007]により提案されたposition-based dynamics(PBD)に基づいています。この手法は、位置レベルでくり返しガウス・ザイデル法に基づいて拘束を解決します。Stam [2009] は、同様にガウス・ザイデル法を速度に対しての拘束を解くのに用いました。これらの手法は、拘束をローカルに扱い、その効果と実装の容易さから広く使われました。しかしながらどちらの手法もイテレーション回数が剛性値に影響する問題に悩まされました。これらの手法は、拘束をローカルに扱い、その効果と実装の容易さから広く使われました。しかしながらどちらの手法もイテレーション回数が剛性値に影響する問題に悩まされました。

多くの筆者が PBD を変形物体のシミュレーションに用いました。近年、Bender ら [2014a] は、連続体ベースの手法を提案しました。この手法は、歪みエネルギーを拘束関数に直接用いています。同時に、Müller ら [2014] は、Green-St Venant 歪みテンソルを、方向に依存せずモデルを離散化する手法を提案しました。Shape matching [Müllerら 2005] は、変形オブジェクトの幾何的な拘束を変形オブジェクトのシミュレーションを行うために PBD 拘束に導入しました。しかしながら、また、これら全ての手法は、イテレーション回数に依存して剛性値が強く依存する問題に悩まされています。その一方で、我々のコンプライアント拘束式は、従来の本質的なモデルに直接対応しています。そして明確な手法で収束します。

Servin ら [2006] は、連続モデルの従順な拘束式を提案しました。そして、半陰解法速度レベル積分器従順に依存しない形で大きな範囲の剛性値を扱えるようにしました。我々の手法は、これらの従順な拘束式を PBD に採用し、安定化のための項を追加することなく位置レベルでの拘束を可能としました。

多数の研究で、拘束を解くのにグローバルな手法を用いています。

Goldenthal ら [2007] は、PBD と似た伸びない拘束を強制するプロジェクトン手法を提案しました。しかし、拘束された系はグローバルに線形化されて拘束毎に直接法で解かれます。我々の手法は、ガウスザイデルくり返し手法と組み合わせて、彼らの高速プロジェクトンアルゴリズムのコンプライアント版とみなすことが出来ます。Tournier ら [2015] は、線形化に基づく不安定さを避けるため2次微分係数情報をもつコンプライアント拘束式を提案しました。彼らの研究と対照的に、我々の手法は、1次微分の拘束導関数のみを用います。そして、タイムステップ毎のローカル線形化を繰り返すことでグローバル線形化に関する安定性問題を避けることが出来ます。

Liu ら [2013] は、質点バネシステムのローカル/グローバルソルバを提案しました。これは、向きに関連した非線形項を扱うためにローカルソルバを用いる一方でストレッ칭ングを扱う為に事前分解されたグローバルソルバを用います。剛性値は、イテレーション回数に強く依存せず、解法は、ニュートン法のイテレーションの増加のように取り扱います。このアイデアは、Boaziz ら [2014] の Projective Dynamics の中で別の拘束へと一般化されました。Wang [2015] は、前回のイテレーションからの結果を組み合わせることでよりよい収束を得るために Chebyshev 法を用いる手法を提案しました。Narain ら [2016] は、alternating direction method of multipliers (ADMM) を時間積分の陰解法として採用しました。これらの手法は、本質的に非線形なモデルを扱うことを可能とします。そして、projective dynamics は、ADMM の特別なケースであることを示しました。

これらのグローバルなアプローチは、グローバルソルバが効果的に行えた事実に基づいており、それは、グローバル行列の事前分解を意味しています。しかしながら、拘束のトポロジーが、引き裂かれたり、破碎のように実行時に変化した場合には、グローバル行列は再分解される必要があり、それは計算コストが高くなります。これらもまた、PBD と比較して実装が複雑になります。

3 背景

position-based dynamics のアルゴリズムのコアに関して簡潔に記述します。そして、より詳細は、Bender ら [2014b] によるサーベイ論文を参照してください。PBD は、多くの拘束のプロジェクトンステップがある、Stömer-Verlet 法を用いた半陰解法ステップと考えられます。投影手法は、質量で重み付けされた修正値を持った拘束毎のローカル線形化を用いて実行されます。PBD の拘束ソルバの中の主要なステップは、下記で示す拘束毎の位置差分の計算です：

$$\Delta \mathbf{x} = k_j s_j \mathbf{M}^{-1} \nabla C_j(\mathbf{x}_i). \quad (1)$$

ここで、下付き文字 i は、イテレーションのインデックスを意味し、 j は、拘束のインデックスで、 $k \in [0, 1]$ は、拘束の修正毎の拘束の剛性値に適用される係数です。スケーリング項 s は、下記の式で示される拘束関数の単一のニュートンステップから求められる式です：

$$s_j = \frac{-C_j(\mathbf{x}_i)}{\nabla C_j \mathbf{M}^{-1} \nabla C_j^T}. \quad (2)$$

シンプルに位置の変化を k でスケーリングする望ましくない副作用は、タイムステップと拘束のプロジェクトンが行われる回数に剛性値が依存してしまうことです。Müller ら [2007] は、この問題に対して堅さの係数の指數関数スケーリングを用いました。しかしながら、これは、タイムステップを考慮されていませんでした。そして、複数の拘束が存在する場合は、明確な解法では収束しません。

続くセクションで、はっきりした正規化拘束を用いたスキームを開発します。これは、明確なボテンシャルエネルギーに直接対応しています。そして、これらをタイムステップとイテレーション回数に依存しない方法でどのように解くのかを示します。

Algorithm 1 XPBD simulation loop

```

1: predict position  $\tilde{\mathbf{x}} \leftarrow \mathbf{x}^n + \Delta t \mathbf{v}^n + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}_{ext}(\mathbf{x}^n)$ 
2:
3: initialize solve  $\mathbf{x}_0 \leftarrow \tilde{\mathbf{x}}$ 
4: initialize multipliers  $\lambda_0 \leftarrow \mathbf{0}$ 
5: while  $i < solverIterations$  do
6:   for all constraints do
7:     compute  $\Delta\lambda$  using Eq (18)
8:     compute  $\Delta\mathbf{x}$  using Eq (17)
9:     update  $\lambda_{i+1} \leftarrow \lambda_i + \Delta\lambda$ 
10:    update  $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \Delta\mathbf{x}$ 
11:   end for
12:    $i \leftarrow i + 1$ 
13: end while
14:
15: update positions  $\mathbf{x}^{n+1} \leftarrow \mathbf{x}_i$ 
16: update velocities  $\mathbf{v}^{n+1} \leftarrow \frac{1}{\Delta t} (\mathbf{x}^{n+1} - \mathbf{x}^n)$ 

```

4 我々の手法

我々は、extended position-based dynamics(XPBD) アルゴリズムを導くために、ボテンシャルエネルギー $U(\mathbf{x})$ から力を求めるニュートンの運動方程式から始めました。

$$\mathbf{M}\ddot{\mathbf{x}} = -\nabla U^T(\mathbf{x}). \quad (3)$$

ここで $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ は、系の状態です。PBD では、これは、単にパーティクルの位置です。しかしながら、これはあらゆる一般化された座標モデルでも表現可能です。例えば剛体の変換などです。

ここでは、勾配作用素 ∇ を偏導関数の列ベクトルとして用いることにします。

運動方程式(3) の位置レベルでの時間積分に陰解法を用います。ここで、上付き文字 n は、タイムステップのインデックスを示します。

$$\mathbf{M} \left(\frac{\mathbf{x}^{n+1} - 2\mathbf{x}^n + \mathbf{x}^{n-1}}{\Delta t^2} \right) = -\nabla U^T(\mathbf{x}^{n+1}). \quad (4)$$

我々は、extended position-based dynamics(XPBD) アルゴリズムを導くために、ポテンシャルエネルギー $U(\mathbf{x})$ は、拘束関数 $\mathbf{C} = [C_1(\mathbf{x}), C_2(\mathbf{x}), \dots, C_m(\mathbf{x})]^T$ のベクトル項にさらに特殊化されます。

$$U(\mathbf{x}) = \frac{1}{2} \mathbf{C}(\mathbf{x})^T \boldsymbol{\alpha}^{-1} \mathbf{C}(\mathbf{x}), \quad (5)$$

ここで $\boldsymbol{\alpha}$ は、剛性値の逆数に対応するブロック対角化コンプライアンス行列です。弾性ポテンシャルからの力は、 \mathbf{x} に関して U の逆勾配から求められます。

$$\mathbf{f}_{elastic} = -\nabla_{\mathbf{x}} U^T = -\nabla \mathbf{C}^T \boldsymbol{\alpha}^{-1} \mathbf{C}. \quad (6)$$

我々は、Servin ら[2006]に基づきこれをコンプライアンス拘束式へと転用します。そして、ラグランジュ乗数を導入することでその方向とスカラーラー値に力を分解します。

$$\boldsymbol{\lambda}_{elastic} = -\tilde{\boldsymbol{\alpha}}^{-1} \mathbf{C}(\mathbf{x}). \quad (7)$$

ここで $\boldsymbol{\lambda}_{elastic} = [\lambda_1, \lambda_2, \dots, \lambda_m]^T$ は、拘束の乗数です。ここから先は、明確さを損なわない範囲で下付き文字を省略していきます。式(4)の左辺に、コンプライアンス行列に含まれる形でタイムステップが含まれている、 $\tilde{\boldsymbol{\alpha}} = \frac{\boldsymbol{\alpha}}{\Delta t^2}$ として定義していることに留意してください。

式の中に $\boldsymbol{\lambda}$ として、離散運動拘束式があります。

$$\mathbf{M}(\mathbf{x}^{n+1} - \tilde{\mathbf{x}}) - \nabla \mathbf{C}(\mathbf{x}^{n+1})^T \boldsymbol{\lambda}^{n+1} = \mathbf{0} \quad (8)$$

$$\mathbf{C}(\mathbf{x}^{n+1}) + \tilde{\boldsymbol{\alpha}} \boldsymbol{\lambda}^{n+1} = \mathbf{0}, \quad (9)$$

ここで、 $\tilde{\mathbf{x}} = 2\mathbf{x}^n - \mathbf{x}^{n-1} = \mathbf{x}^n + \Delta t \mathbf{v}^n$ は、予測位置、あるいは慣性位置です。この非線形システムを解くために、ニュートン法に基づいた固定点繰り返し法を考案しました。この先では、イテレーション毎にわからないうち、タイムステップの下付き文字 $(n+1)$ と、インデックスを示す下付き文字 $(i+1)$ を省略します。

式(8,9)をそれぞれ \mathbf{g}, \mathbf{h} で表すことにします。目的は、これを満たす \mathbf{x} と $\boldsymbol{\lambda}$ を見つけることです。

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \quad (10)$$

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}. \quad (11)$$

式(10,11)を線形化して、下記の線形ニュートン部分問題が得られます。

$$\begin{bmatrix} \mathbf{K} & -\nabla \mathbf{C}^T(\mathbf{x}_i) \\ \nabla \mathbf{C}(\mathbf{x}_i) & \tilde{\boldsymbol{\alpha}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{g}(\mathbf{x}_i, \boldsymbol{\lambda}_i) \\ \mathbf{h}(\mathbf{x}_i, \boldsymbol{\lambda}_i) \end{bmatrix} \quad (12)$$

ここで $\mathbf{K} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ です。このシステムは $\Delta \mathbf{x}$ と $\Delta \boldsymbol{\lambda}$ で解くことが出来て、その結果乗数の更新が可能です。

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \Delta \boldsymbol{\lambda} \quad (13)$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta \mathbf{x}. \quad (14)$$

これは、 $|\mathbf{x}_{i+1} - \mathbf{x}_i| \rightarrow 0$, $|\boldsymbol{\lambda}_{i+1} - \boldsymbol{\lambda}_i| \rightarrow 0$ のような任意の数列での運動方程式(8, 9)の陰解法での不動点解法のくり返しです。一般的にこの手法はうまく働きますが、堅牢な線形探索戦略が必要となります。そしてシステム行列の計算コストが高価になります。特に、ヘシアン行列の拘束の評価が必要となる \mathbf{K} の計算を避けたいです。

ここで、2つの近似を紹介します。これは、手法の実装をシンプルにして PBD へと戻るよう結びつけるためのものです。最初に、 $\mathbf{K} \approx \mathbf{M}$ の近似を使います。これは、幾何的な剛性値とヘシアン項の拘束を除外します。そして $O(\Delta t^2)$ の計算量でのローカルエラー修正値を導入します。この近似で、収束率が変更するかもしれません、これは、グローバルなエラーを変更しませんし不動点解法の繰り返しによる解決でもありません。特に、準ニュートン法とも捉えることが可能です。

次に、 $\mathbf{g}(\mathbf{x}_i, \boldsymbol{\lambda}_i) = \mathbf{0}$ であると仮定します。この仮定は、従来ニュートン法の繰り返し解法で正しいと思われていたことと異なりますが、 $\mathbf{x}_0 = \tilde{\mathbf{x}}$ で $\boldsymbol{\lambda}_0 = \mathbf{0}$ の時に正規化されます。加えて、拘束の勾配がゆっくり変化した場合は、小さい値に留まります。定数の場合は \mathbf{A} に向かいます。さらに、修正線形システムは、質量で重み付けされた、現在のイテレーション A_i から始まる拘束多様体の最小化問題の最適性条件と対応しています。この変分点の観点は Goldenthal ら[2007]で詳細に議論されています。そして、我々の手法は、彼らの Fast Projection アルゴリズムのコンプライアント版とみなすことも出来ます。

これらの近似の中に含まれる、線形下位問題は、下記で与えられます。

$$\begin{bmatrix} \mathbf{M} & -\nabla \mathbf{C}^T(\mathbf{x}_i) \\ \nabla \mathbf{C}(\mathbf{x}_i) & \tilde{\boldsymbol{\alpha}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{0} \\ \mathbf{h}(\mathbf{x}_i, \boldsymbol{\lambda}_i) \end{bmatrix}, \quad (15)$$

\mathbf{M} に関して、未知の $\Delta \boldsymbol{\lambda}$ に関する下記の削減されたシステムを得るためにシューアの補元を取ります。

$$[\nabla \mathbf{C}(\mathbf{x}_i) \mathbf{M}^{-1} \nabla \mathbf{C}(\mathbf{x}_i)^T + \tilde{\boldsymbol{\alpha}}] \Delta \boldsymbol{\lambda} = -\mathbf{C}(\mathbf{x}_i) - \tilde{\boldsymbol{\alpha}} \boldsymbol{\lambda}_i. \quad (16)$$

位置の更新は、下記を評価することで直接に得ることができます。

$$\Delta \mathbf{x} = \mathbf{M}^{-1} \nabla \mathbf{C}(\mathbf{x}_i)^T \Delta \boldsymbol{\lambda}. \quad (17)$$

我々の手法では、もはや運動方程式を陰解法で正確に解いているとは言えないけれども、エラーは少ないので実践的です。この手法の正確さを先のセクション6で調査します。

4.1 ガウス・ザイデル更新

式(16)の線形システムのためのガウスザイデル法を考えることで PBD の話題に戻ります。インデックス j での拘束式は、ラグランジュ乗数の変化を評価することで直接計算が可能です。

$$\Delta \lambda_j = \frac{-C_j(\mathbf{x}_i) - \tilde{\alpha}_j \lambda_{ij}}{\nabla C_j \mathbf{M}^{-1} \nabla C_j^T + \tilde{\alpha}_j}. \quad (18)$$

この式が、我々の手法の核になります。拘束を解決している間、最初に $\Delta \lambda_j$ を単一の拘束として計算します。その後システムの位置と乗数を式(13,14)を用いて更新します。アルゴリズム1で手法をまとめています。行4,7,9を加えた以外はオリジナルのPBDのアルゴリズムと同一です。

$\Delta \lambda_j$ のために、式をより詳細に分析する場合、 $\alpha_j=0$ の場合を見ることができます。これは、オリジナルのPBDのアルゴリズム(2)のスケール項 s_j に正確に対応します。新しい定義から、 s_j を、無限の剛性値を持つ拘束のための追加のラグランジュ乗数の変化であると考えます。コンプライアンスの場合、 $\alpha_j \geq 0$ となり、追加の項が、分子と分母の両方に現れます。これらの項は、拘束の力が限られている時に拘束の正規化を行う役目があります。これは式(5)で与えられる弾性ポテンシャルに該当します。

式(18)の分母は、 λ_{ij} の参照を含みます。これは、現在のイテレーション i での拘束 j のトータルのラグランジュ乗数となります。システムの位置に加えて、この値を保存して更新する必要があります。拘束毎の1スカラー一分の追加のストレージが、ささやかなオーバーヘッドになります。そして、トータルの拘束力の有益な情報を提供し、力に依存した効果を導くのに使うことが可能です。(例:破壊可能なジョイント)あるいは、触覚デバイスです。

5 ダンピング

我々の手法は、陰解法タイムステップ法から導かれます。これは、いくらかの機械エネルギーの消失を自然に含んでいます。それでもなお、追加の拘束のダンピングのモデルを考えるのは有益です。そうすることで、レイリー消失ポテンシャルを定義することが出来ます。

$$D(\mathbf{x}, \mathbf{v}) = \frac{1}{2} \dot{\mathbf{C}}(\mathbf{x})^T \boldsymbol{\beta} \dot{\mathbf{C}}(\mathbf{x}) \quad (19)$$

$$= \frac{1}{2} \mathbf{v}^T \nabla \mathbf{C}^T \boldsymbol{\beta} \nabla \mathbf{C} \mathbf{v}, \quad (20)$$

ここで $\boldsymbol{\beta}$ は、拘束ダンピング係数に該当するブロック対角行列です。 $\boldsymbol{\beta}$ は、コンプライアンスのように逆数のパラメータではありません。そして、ダンピング剛性値のように設定されるべきです。ラグランジュ力学によれば、消失ポテンシャルからの力は、速度に関する D の負の勾配から求まります。

$$\mathbf{f}_{damp} = -\nabla_{\mathbf{v}} D^T = -\nabla \mathbf{C}^T \boldsymbol{\beta} \nabla \mathbf{C} \mathbf{v}. \quad (21)$$

弾性体の場合、この力ベクトルのスカラー項と拘束乗数を分離します。

$$\boldsymbol{\lambda}_{damp} = -\tilde{\boldsymbol{\beta}} \dot{\mathbf{C}}(\mathbf{x}) = -\tilde{\boldsymbol{\beta}} \nabla \mathbf{C} \mathbf{v}, \quad (22)$$

ここで、 $\tilde{\boldsymbol{\beta}} = \Delta t^2 \boldsymbol{\beta}$ を定義することでダンピング項をタイムステップに含みます。

$\boldsymbol{\lambda}_{damp}$ は、それ自身で解くことが可能で、拘束力のダンピングや弾性項それぞれの乗数を、得ることが出来ます。しかしながら、多くの場合で総拘束力を知ることのみ配慮しています。なぜならダンピング力は、弾性力と同じ方向にそって働くなければならないからです。我々は乗数を单一の式へと結合することが出来ます。

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_{elastic} + \boldsymbol{\lambda}_{damp} = -\tilde{\boldsymbol{\alpha}}^{-1} \mathbf{C}(\mathbf{x}) - \tilde{\boldsymbol{\beta}} \nabla \mathbf{C} \mathbf{v}. \quad (23)$$

(23)を再構築して、拘束を下記のように得ました。

$$\mathbf{h}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{C}(\mathbf{x}) + \tilde{\boldsymbol{\alpha}} \boldsymbol{\lambda} + \tilde{\boldsymbol{\alpha}} \tilde{\boldsymbol{\beta}} \nabla \mathbf{C} \mathbf{v} = \mathbf{0}. \quad (24)$$

速度、 $\mathbf{v} = 1/\Delta t (\mathbf{x}_{n+1} - \mathbf{x}_n)$ へ代入し、 $\boldsymbol{\lambda}$ に関して線形化し下記を得ます。

$$\left[(\mathbf{I} + \frac{\tilde{\boldsymbol{\alpha}} \tilde{\boldsymbol{\beta}}}{\Delta t}) \nabla \mathbf{C}(\mathbf{x}_i) \mathbf{M}^{-1} \nabla \mathbf{C}(\mathbf{x}_i)^T + \tilde{\boldsymbol{\alpha}} \right] \Delta \boldsymbol{\lambda} = -\mathbf{h}(\mathbf{x}_i, \boldsymbol{\lambda}_i). \quad (25)$$

单一の拘束式に関して、ガウス・ザイデル更新は下記で与えられます。

$$\Delta \boldsymbol{\lambda}_j = \frac{-C_j(\mathbf{x}_i) - \tilde{\boldsymbol{\alpha}}_j \lambda_{ij} - \gamma_j \nabla C_j(\mathbf{x}_i - \mathbf{x}^n)}{(1 + \gamma_j) \nabla C_j \mathbf{M}^{-1} \nabla C_j^T + \tilde{\boldsymbol{\alpha}}_j}. \quad (26)$$

共通の事例として、 $\tilde{\boldsymbol{\alpha}}$ と $\tilde{\boldsymbol{\beta}}$ が、シンプルな対角行列の場合（ブロック対角行列とは対照的に）、 $\gamma_j = \frac{\tilde{\alpha}_j \tilde{\beta}_j}{\Delta t}$ を得ることが出来ます。

言い換えると、タイムステップは、様々なコンプライアンスと拘束のダンピングパラメータでスケールされます。ここで用いられる全ての追加項は、簡単に効果的に計算可能であることを強調しておきます。

6 結果

様々な種類の変形モデルと拘束タイプでのテストを行いました。2Dの実験では、ガウス・ザイデル繰り返し法でのCPU実装を用いました。離散運動方程式(8,9)を直接適用する従来の非線形ニュートン解法との比較を行っています。ニュートン解法は、堅牢なEigen数学ライブラリ[2010]のコレスキーフィルタを線形ソルバとして用いています。

3Dの実験では、NVIDIA GTX1070 上でのGPUベース手法でヤコビ繰り返し法を用いました。衝突は、オリジナルのPBD手法の通りに扱っています。接触にあたり、コンプライアンスは0として扱います。これは、ラグランジュ乗数を衝突の拘束として保存する必要が無いことを意味しています。

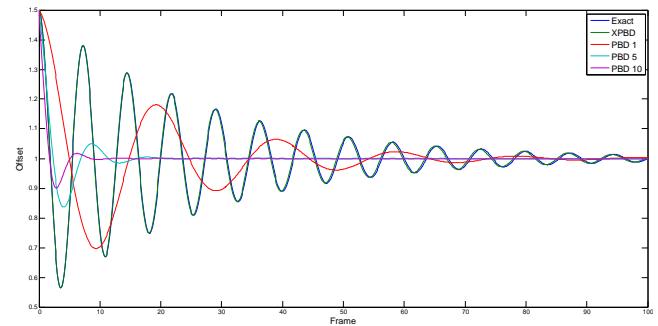


図2: コンプライアンス値 $\alpha = 0.001$ の距離拘束でモデルされたシンプルな調和振動子XPBDは、解析解にほぼ一致していますが、PBDの解は、イテレーション回数に強く依存しています。

6.1 パネ

距離拘束でモデル化されたシンプルな調和振動子の事例で手法のテストを行いました。定常位置 $x=1$ とし、初期位置 $x_0 = 1.5$ $\alpha = 0.001$ で質量 $m = 1$ とします。図2で見られる通り、PBD は、正確な振動を示しません。どのようにPBDの振動するか、ダンピングがイテレーション回数に依存して急激に動いているかを示すために1, 5, 10回でのPBDシミュレーション結果をプロットしました。

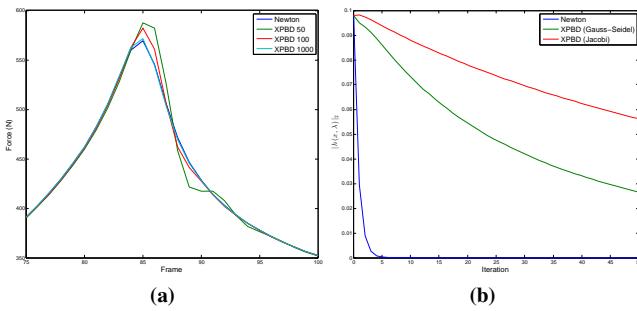


図3: (a) チェーンの例での一連のフレームの固定支点での拘束力の大きさ。(b) 梁の例でサンプルしたフレーム $A(x, \lambda)$ での残存エラーの値。ガウス・ザイデル法と、ヤコビ法の両方で繰り返し解放で線形に収束することが期待されます。

このシンプルなケースで、XPBDは、タイムステップとイテレーション回数に関わらず分析解と近い解を得られます。この例では、リファレンスとなる例でのダンピングの値は、運動方程式の陰解法での離散化の副作用です。

6.2 チェーン

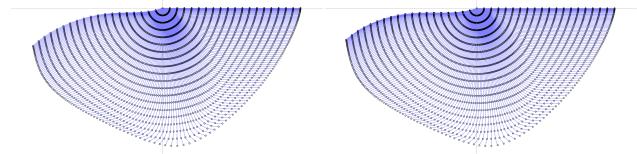


図4: 重力に従って落下する、ぶら下がっているチェーン状の2Dパーティクルのタイムラプスピュー。左:リファレンスとなるニュートンソルバ。右:50イテレーションでのXPBDの結果。

我々の手法のキーとなる利点の1つは、拘束力の大きさを返すことです。そこで重要な質問は、拘束力をどうやって正確に返すかです。PBD の更新によるエラーを定量的に計測するために、図4で示すように、少しだけ伸びるパーティクルのチェーンのシミュレーションを行いました。チェーンは、20パーティクルで構成されていて、それぞれ $m = 1.0$ で $\alpha = 10^{-8}$ の距離拘束で接続されています。チェーンのトップの固定点パーティクルで100フレーム以上に渡り、拘束の合力の大きさを測定しました。図3は、リファレンスとなるニュートンソルバと XPBD の結果を比較したものです。一連のシミュレーションに渡って、我々の手法の最大相対エラーは、イテレーション回数50回、100回、1000回の場合 6%, 2%, 0.5% でした。この結果は、多くのグラフィックスの応用において十分に許容可能だと考えています。

6.3 片持ち梁

従来の有限要素法(FEM)は、コンプライアント拘束フレームワーク [Servin ら 2006] で再定式化することが可能です。線形等方性構成モデルの場合、歪みテンソル ϵ のそれぞれの要素は、拘束関数のベクトルとして扱われます、これは三角形の要素のフォーカト記法です。

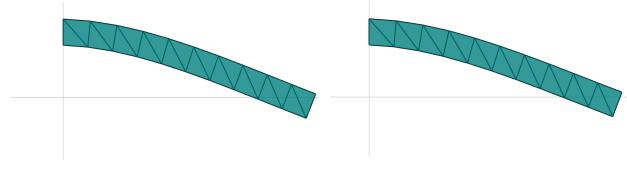


図5: 三角形のFEM要素でモデリングされた、片持ち梁。線形、等方性構成モデルで50フレームのシミュレーション。左:リファレンスとなるニュートンソルバ。右:20イテレーションでのXPBDの結果。我々の手法は、リファレンスと比べても視覚的に区別できない。

コンプライアンス行列は、剛性値行列の逆数で与えられます、言い換えるとラメ定数は、

$$\boldsymbol{\alpha}_{tri} = \mathbf{K}^{-1} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & 2\mu \end{bmatrix}^{-1}. \quad (28)$$

Strain-Based Dynamics [Müller ら 2014] に対するこの定式化のアドバンテージは、従来使われていたマテリアルのパラメータに該当するだけではなく、正確に、個々の歪みをポアソン降下のモデルへと連結させていることです。このことで、XPBDは、実験データから得られたマテリアル特性を持ったモデルのシミュレーションが可能です

アルゴリズムの正確さの評価のために、St Venant-Kirchhoff 三角形有限要素法を用いた片持ち梁のシミュレーションと比較してみました(図5)。 $E = 10^3$ のヤング率で、ポアソン比 $\mu = 0.3$ で $\Delta t = 0.008$ の線形等方性構成モデルを用いました。シミュレーションを50フレーム走らせて、XPBDのイテレーション回数が20回で、リファレンスとなるニュートンソルバと比較して視覚的に見分けることができないほど十分であることが分かりました。図3bで、イテレーション毎の残りエラーをプロットしました。

6.4 クロス

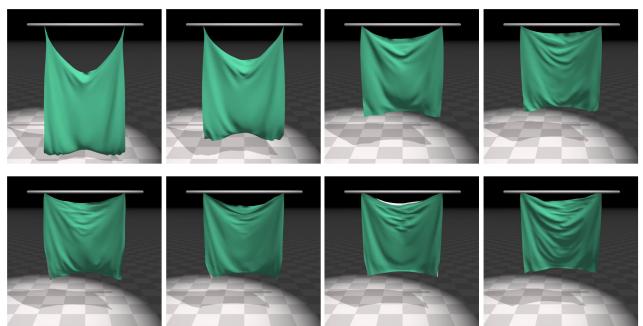


図6: それぞれ(左から右へ) 20, 40, 60, 80, 160回のイテレーション回数での吊るされたクロス。上段:PBDで、下段:XPBD。PBDの剛性値は、イテレーション回数に対して非線形に依存しています。その一方、XPBD の振る舞いは、定性的に変化しません。

アルゴリズムの性質によりイテレーション回数と独立しているかどうかを、重力に対してぶら下がっているクロスのシミュレーションでテストしました。クロスのモデルは、64x64のパーティクルのグリッドで構成されており、24,000の距離拘束のグラフで接続されています。図6では、拘束の剛性値を固定してイテレーション回数を変化させた状態です。

$$\mathbf{C}_{tri}(\mathbf{x}) = \boldsymbol{\epsilon}_{tri} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{bmatrix}. \quad (27)$$

イテレーション回数を変えた際に、PBD の振る舞いがどのように変わるのがを明確に示すために、低い拘束の剛性値 $k = 0.01$ の人工的な値を用いました。XPBD は、PBD よりも早く収束しないことに注意してください。単に、イテレーション回数を変化させたときでも一貫した振る舞いになるだけです。実際に、ゼロコンプライアンスの XPBD では、PBD の拘束剛性値 $k = 1$ と等価です。

クロスの例で XPBD のパフォーマンスのインパクトを測定し、イテレーション毎に少し計算コストが加わるだけ、典型的にはシミュレーション時間全体の 2% 以下になることがわかりました。詳細に関しては、表1を参照ください。

表 1: イテレーション回数を変化させた際のクロスがぶら下がっている例のステップごとのシミュレーション時間(ms)

Iterations	20	40	80	160
PBD	0.95	1.75	3.25	5.61
XPBD	0.97	1.78	3.34	5.65

6.5 ふくらませることができるバルーン

我々の手法を、ふくらませができるバルーンに適用させました。このバルーンの内部の空気圧は、グローバルの体積拘束を使ってモデル化されています。また外側の表面はストレッチ、せん断、曲げ拘束のクロスとして扱われています(図1)。追加ストレージの観点から見ると、内部の体積拘束は、追加の乗数を一つ要求します。さらに、サーフェイスの拘束ごとにもう一つ乗数が必要になります。我々の手法は、相対的な拘束の剛性値を調整しやすくなっています。例えば、サーフェイスの剛性値を変えずに、より強い体積保存性を得るためにイテレーションの回数を増やすことができます。

7 制限事項と今後の課題

$\alpha = 0$ の制限で、我々の手法は、 $k = 1$ の PBD と同一の結果になります。そして同じイテレーション回数で同様な剛性値になります。PBD と同じように、収束するよりも前の段階での少ないイテレーション回数の場合は、不十分な結果になります。ガウスサイデル法やヤコビ繰り返し法を、イテレーション毎により洗練された線形ソルバに置き換えることで収束速度を改善することができますかかもしれません。近年の研究では、Wang ら[2015]によって提案されたチェビシェフ法のような高速化繰り返し法にフォーカスされています。

我々の将来性のある手法は、時間的なコヒーレンスを利用してしています。なぜなら、総拘束力の記録を持っており、前のフレームのラグランジュ乗数に基いて拘束の解をウォームスタートすることが可能です。

Projective Dynamics [Bouaziz ら 2014]と異なり、我々の手法は、オイラー積分の陰解法の近似に過ぎません。リアルタイムのグラフィックスアプリケーションでは、エラーは許容範囲で、多くの場合で見分けがつかないと考えています。しかしながら、従来手法が、より正確さの保証が求められる場合により有効なこともあります。

8 結論

position-based dynamics のシンプルな拡張を提案しました。これは、広く知られているタイムステップと、イテレーション回数に剛性値が依存してしまう問題の解決に取り組んでいます。我々の手法は、ただひとつの追加のスカラー事前拘束の計算と保存のみを要求しますが PBD で任意の弾性体と消失エネルギーポテンシャルのシミュレーションが可能となります。加えて、力に依存した効果のため正確な拘束力の見積もりを得ることが可能となります。このことが、PBD の適応領域をより正確で、従来のマテリアルモデルに一致する応用へと広げると考えています。

なぜなら、我々の手法は、既存の PBD ソルバに小さな変更を加えるだけなので、多くの産業や実践者に簡単に採用されると考えています。

References

- BENDER, J., KOSCHIER, D., CHARRIER, P., AND WEBER, D. 2014. Position-based simulation of continuous materials. *Computers & Graphics* 44, 1–10.
- BENDER, J., MÜLLER, M., OTADUY, M. A., TESCHNER, M., AND MACKLIN, M. 2014. A survey on position-based simulation methods in computer graphics. *Computer Graphics Forum*, 1–25.
- BOUAZIZ, S., MARTIN, S., LIU, T., KAVAN, L., AND PAULY, M. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (TOG)* 33, 4, 154.
- GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. In *ACM Transactions on Graphics (TOG)*, vol. 26, ACM, 49.
- GUENNEBAUD, G., JACOB, B., ET AL., 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- LIU, T., BARGTEIL, A. W., O'BRIEN, J. F., AND KAVAN, L. 2013. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG)* 32, 6, 214.
- MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 471–478.
- MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (Apr.), 109–118.
- MÜLLER, M., CHENTANEZ, N., KIM, T.-Y., AND MACKLIN, M. 2014. Strain based dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '14, 149–157.
- NARAIN, R., OVERBY, M., AND BROWN, G. E. 2016. ADMM \supseteq projective dynamics: Fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. To appear.
- NEALEN, A., MÜLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2005. Physically Based Deformable Models in Computer Graphics. In *Eurographics 2005 - State of the Art Reports*, The Eurographics Association, Y. Chrysanthou and M. Magnor, Eds.
- SERVIN, M., LACOURSIERE, C., AND MELIN, N. 2006. Interactive simulation of elastic deformable materials. In *Proceedings of SIGRAD Conference*, 22–32.
- STAM, J. 2009. Nucleus: Towards a unified dynamics solver for computer graphics. In *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics' 09. 11th IEEE International Conference on*, IEEE, 1–11.
- TOURNIER, M., NESME, M., GILLES, B., AND FAURE, F. 2015. Stable constrained dynamics. *ACM Transactions on Graphics (TOG)* 34, 4, 132.
- WANG, H. 2015. A chebyshev semi-iterative approach for accelerating projective and position-based dynamics. *ACM Transactions on Graphics (TOG)* 34, 6, 246.