Basic theory behind (X)PBD

Slime Piki(Taisei Kunimi)

Contents

1 Introduction

1.1 What is (X)PBD?

PBD is one of the simulation theories that basically simulates soft-body or elastic–body.

PBD (Position Based Dynamics) proposed at [4] is a popular method because of its stability and ease of implementation.In datail, PBD computes physical simulation only using positions inside the Iterations and all we have to do is compute displacement and modify them. In other words, we don't have to use complicated numerical analysis theories, it sounds pretty good.

But, in contrast to ease of implementation, it isn't easy to understand PBD's background theory. This is the problem when modifying PBD depending on your purpose.

If you start your research from the original PBD paper[4], you will wonder how the authors derive constraints' formulations or why this solver works well. Or you start from XPBD [7], you will be confused by the suddenly appeared Lagrange multiplier or energy potential that we don't know how to handle. Unfortunately, we can't know much from them and it may be common in literature search, there is no clear path to learning them. Then, I decided to write a guidebook on the underlying theory of PBD.

1.2 Difference from existing PBD coursenote

Actually, there are some course notes on PBD written by authors who published papers on PBD and XPBD, e.g. [13]. These course notes describe the basic style of PBD and its extensions. But there is the same problem we saw in [4] and [7], that is, how to implement is described but why this method works well is not. Thus, I believe that this document isn't meaningless. Well then, let's start the journey to XPBD!

1.3 Learning Path

Fig. 1 shows the shortest path to understand (X)PBD. You don't have to read this note from head to tail because this note covers wide topics around (X)PBD. If you only want to learn how (X)PBD works, you read this note along with Fig. 1's path. In addition, the dashed circles could be skipped if you are in a hurry.

As the extra, you can use this note to learn Projective Dynamics also, yay! If you want to do so, the shortest path is fig. 2.

2 The history of PBD

I think starting from history is a good way to learn something because there are no leaps in logic and it will be easy to understand where we are. However, there is certainly redundancy, so you can skip this section to save time. I'll make an effort to write that you can understand everything even if you skipped this section.

The history of PBD can be roughly divided into three parts. Let them are pre-PBD, post-PBD and post-XPBD.

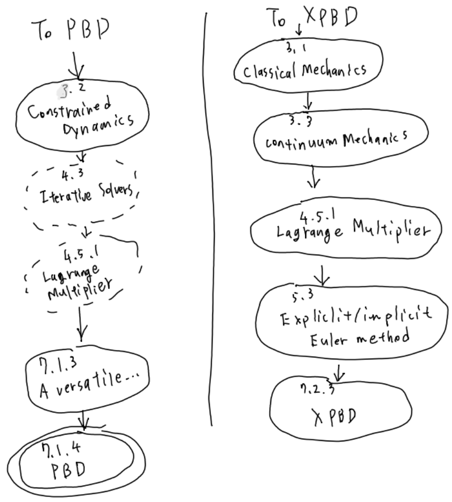

Fig. 1: Path to (X)PBD

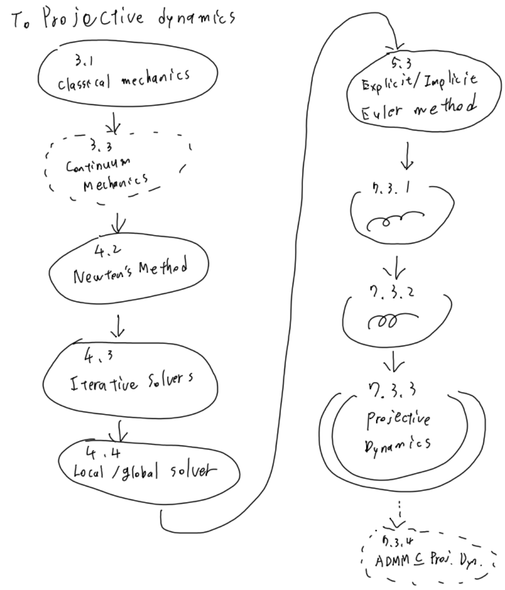

Fig. 2: Path to (X)PBD

2.1 Pre-PBD

The flow of the pre-PBD began from the appearance of constraint dynamics([14], [15] and [16]) through "*Large Steps in Cloth Simulation*"[1] used constraints as shape representation and simulates cloth with energy form constraints' gradients, "*Advanced Character Physics*"[2] introduces position-based approach derived from Verlet's integration scheme with the distance constraints and "*A Versatile and Robust Model for Geometrically Complex Deformable Solids*"[3] generalize [1]'s method. And, finally, "*Position Based Dynamics*"[4] introduced the generalized constraints method from [3] into [2]'s position-based simulation to use various constraints.

2.2 Post-PBD

From the published PBD paper, a lot of study about it has been done. The central development is resolving well-known PBD's drawback whose result depends on Iteration times, at "*XPBD: position-based simulation of compliant constrained dynamics*"[7].

To understand XPBD, we need some knowledge of continuum mechanics. The theory directly used in XPBD is derived from "*Interactive simulation of elastic deformable materials*"[6] that uses the theory for Lagrangian mechanics.

Besides that, some papers also influence XPBD, "*Efficient simulation of inextensible cloth*"[17] and "*Strain Based Dynamics*"[18] are examples of them. The former introduces the projection method described later. The latter invites physically based constraints to PBD.

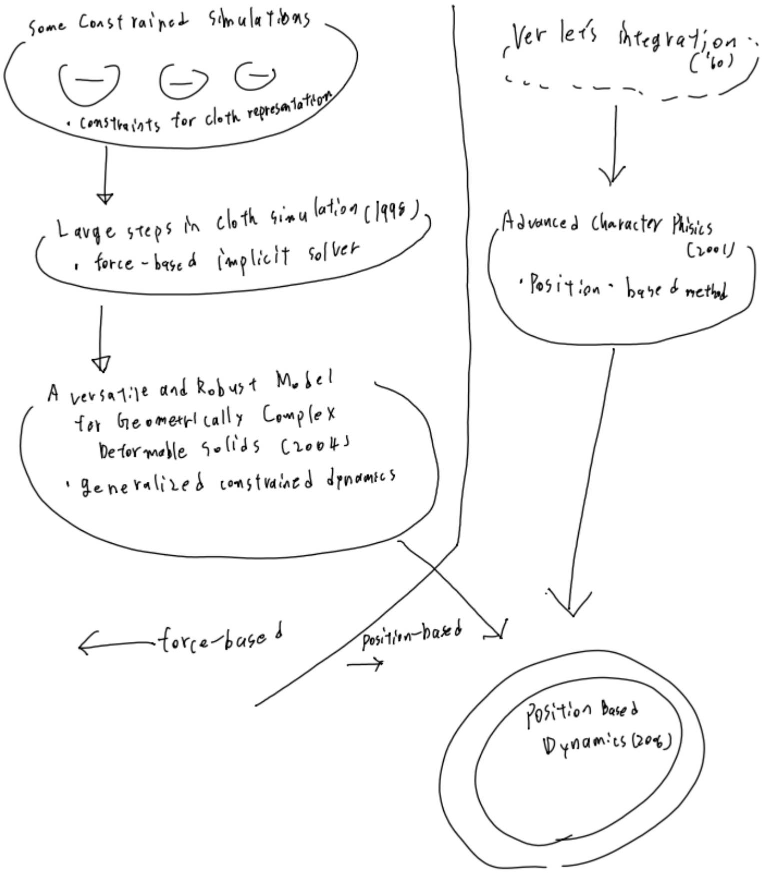

Fig. 3: history of pre-PBD

Although there is relevance to XPBD, the key idea to construct XPBD is [6] and aren't the last two papers. In other words, to understand XPBD, we truly only have to read [6] or get the equivalent knowledge.

But, unfortunately, the key idea of XPBD provided by [6] doesn't have a sufficient explanation of where the authors get the formulation. Then, I guess a description later that seemed reasonable enough.

2.3 Post-XPBD

Shortly after published XPBD paper, a simple but important improvement was provided at "*Small steps in physics simulation*"[8].

!!!!In preparation!!!!

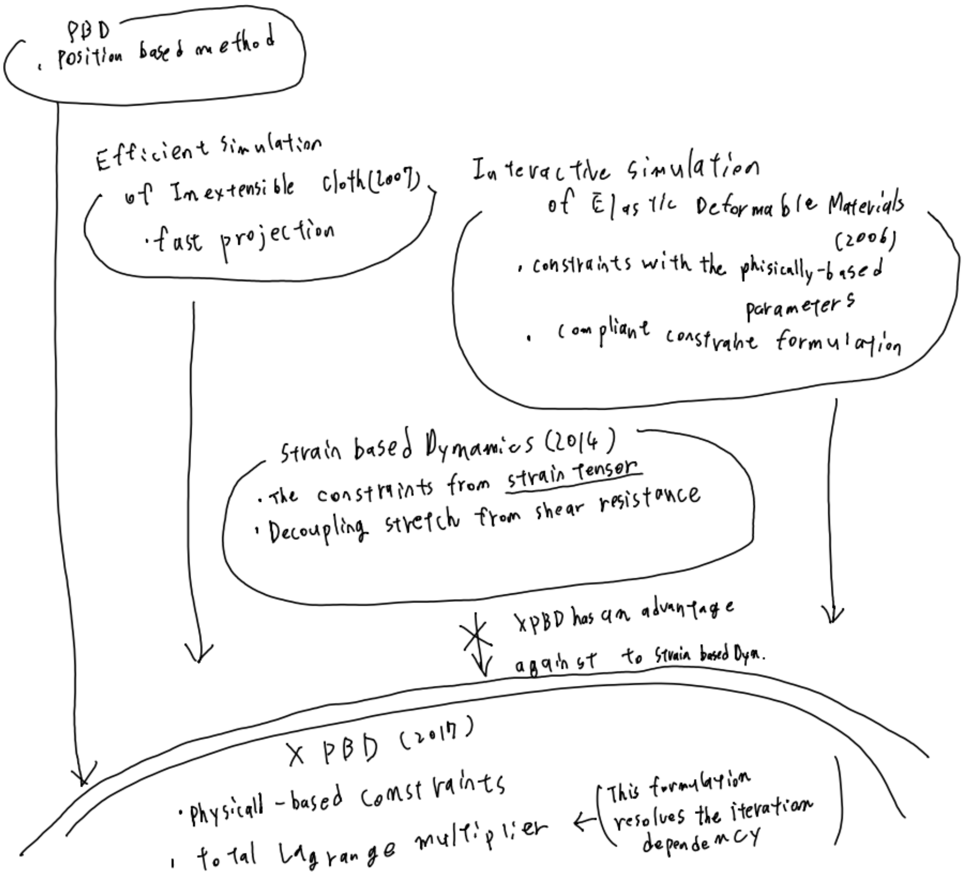

Fig. 4: history of post-PBD

3 Physics

This section is devoted to fundamental physics.

Of course, the physics simulation field takes advantage of the basics. If we only handle trivial environments, classical mechanics can achieve simulation.

However, classical mechanics isn't enough to simulate complex objects or get plausible results. For example, Newtonian mechanics-based methods suffer from soft-body simulation, and so on.

The algorithm of the soft-body simulation isn't trivial, and there is no definitive solution at present. Therefore, many methods have been studied to solve this problem.

Based on the above perspective, in this section, I also introduce some physical theories that are frequently used in soft/elastic body simulation, Constrained dynamics, and Continuum mechanics. In particular, continuum mechanics is vital to understanding XPBD.

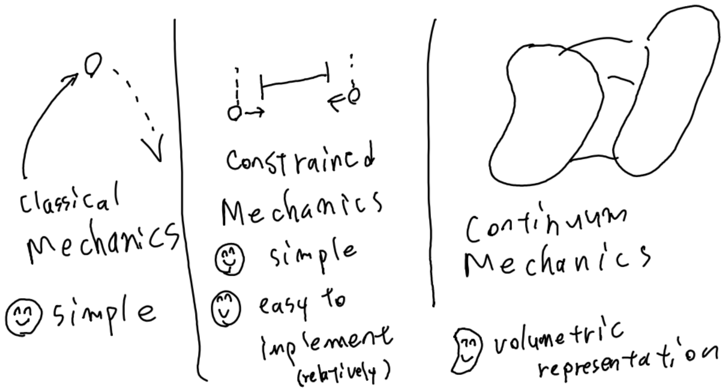

Fig. 5: Comparison of the physics

Although there are many other important theories, such as fluid mechanics or thermodynamics, I won't explain them because this document's subject is (X)PBD which focuses on soft/elastic body (actually, PBD can be used in fluid simulation...).

3.1 Classical mechanics

Let's start the physics course with classical mechanics.

3.1.1 Position, Velocity and, Acceleration

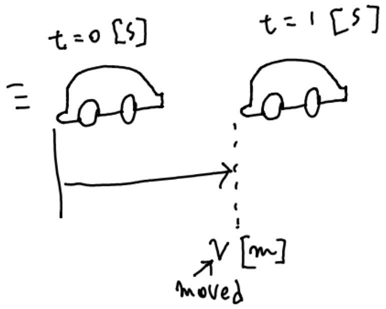

Fig. 6: constantly moving car

First, please imagine a vehicle running on a line at a constant speed. Next, we measure the vehicle's position once per second and start your stopwatch with zero. From now on, time is measured in seconds and denoted as $t = (currenttimeinseconds)$ Let the first measured position x_0, the second position x_1, and so on.

Because the vehicle has a constant speed, the distance it moves between a single step is the same. In other words, $x_n - x_{n-1}$ is an invariant for any natural integer n.

The moved distance per second is called velocity whose unit $[(distance)/(time)]$ e. g. $[meter/second]$. Following that, we can say the vehicle's velocity is $x_n - x_{n-1}$[m/s](this is the abbreviation of meter/seconds) in this example. Let the velocity denote v[m/s] then the relation between the current position at t second x_t and initial position x_0 is

$$x_t = x_0 + vt. \tag{1}$$

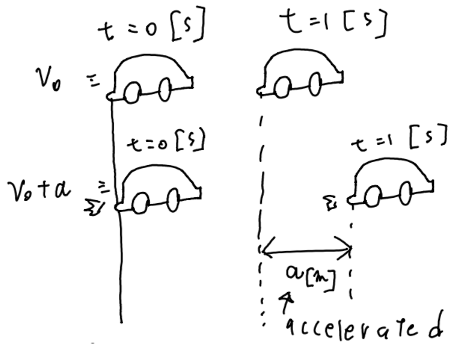

Fig. 7: constantly accelerating car

Let's start the second case. Imagine a vehicle running on a line, and its speed increases constantly. Like the former example, let the speed at time t be denoted as v_t.

Like the relation between distance and velocity in the former example, the increased speed between a single step is constant, and $v_t - v_{t-1}$ is an invariant for any natural number t. We call the increased amount acceleration. You may suppose the accerelation to have a unit $[(velocity)/(time)]$. Unfortunately, the supposed unit won't be used. Let's remind that velocity has the unit $[(distance)/(time)]$, and put this into $[(velocity)/(time)]$ then we earn commonly used unit $[(distance)/(time)^2]$ e. g. [m/s^2].

We can find a relation between the velocity and the constant acceleration a, that is,

$$v_t = v_0 + a \cdot t. \tag{2}$$

Next, we look at the relation between the traveled distance, the velocity, and the acceleration. Simply putting 2 into 1 dosen't work because the result

$$x_t = x_0 + t(v_0 + at) \tag{3}$$

says that the car runs in the constant speed $v_0 + at$ from start measuring.

What we have to do is find a geometric relation between the distance and the velocity. Let's revisit the first example and see eq.(1).

The traveled distance increases v_0 per second. Now, we consider a small timestep $\delta_t(\Delta_t << 1)$ and calculate the distance the vehicle traveled ($x << 1$ says that x is too small compared to 1). We can easily get the answer

$$v_0\Delta_t[\mathrm{m}]. \tag{4}$$

The answer can be seen as a rectangle whose width is Δ_t and height is v_0. Calculating total distance is interpreted as collecting such rectangles until the width being t. The consequence is depicted in Fig.(8).

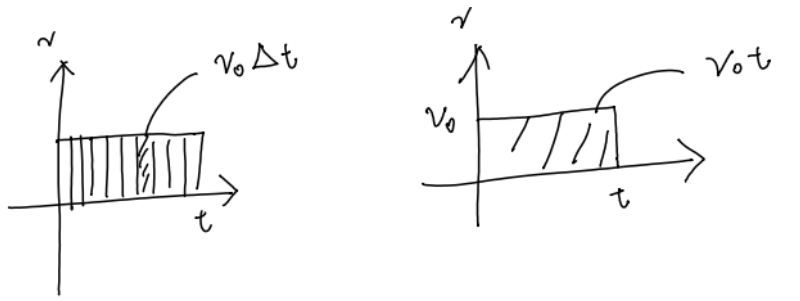

Fig. 8: Geometric aspect of the distance

Now, we expect that the total distance is the area of the graph whose x-axis is time and y-axis is velocity. In fact, the expectation is correct. I show this in a later section. For now, let the expectation be true.

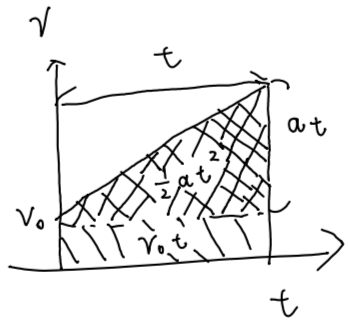

Fig. 9: The graph of the second exapmle

Return to the second example, and draw a graph of time vs velocity (Fig.(9)), then yield the correct total distance.

$$x_t = x_0 + v_0 t + \frac{1}{2}at^2. \tag{5}$$

3.1.2 The Calculus

We concluded that the relation between distance, velocity, and acceleration can be interpreted geometrically in the last section. However, we can't calculate the distance whose graph has a curve. How do we do it?

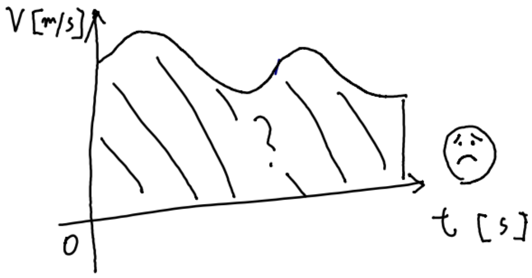

Fig. 10: The graph with the velocity curve

First, as we did in the previous section, the area is assumed to be the sum of rectangles. We fix the width rather large and pick the height by matching the rectangle's right corner with the curve. The result will be the figure below.

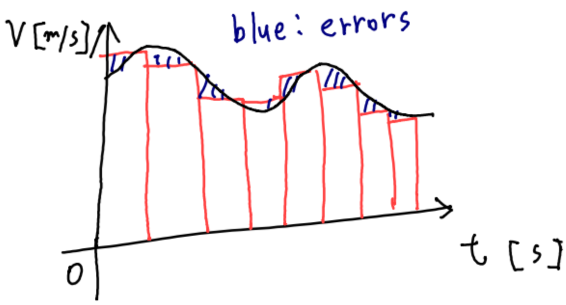

Fig. 11: The graph with the velocity curve with the large rectangles

We can see quite a large total error. This result is incorrect. Well then, let's see what happens when we set the width smaller.

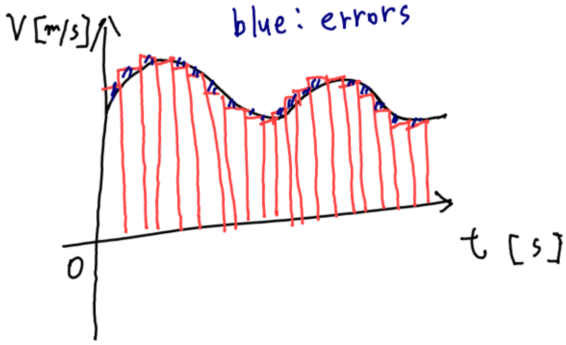

Fig. 12: The graph with the velocity curve with the small rectangles

While we didn't measure precisely, the total error looks smaller than before because they are split into small parts.

If this observation continues to the infinitesimal width, the total error also might go infinitesimal. The problem had been studied for a long time, and this manipulation is called "integration". As a consequence, it has concluded true.

However, the conclusion doesn't tell us how to calculate the total area. Thus next, we see how to calculate the sum briefly.

The formulation is straightforward, summing the rectangles which have infinitesimal width. Let the total area at a time t $A(t)$, rectangles' width dt and the velocity at time t $v(t)$, $A(t)$ is

$$A(t) = \sum_{i=1}^{t/dt} v(i \cdot dt) \cdot dt, \tag{6}$$

and bring dt to infinitesimal,

$$\lim_{dt \to 0} A(t) = \lim_{dt \to 0} \sum_{i=1}^{\infty} v(i \cdot dt) \cdot dt. \tag{7}$$

We can also write the summation in infinitely small rectangles like the notation below.

$$A(t) = \int_0^t f(t)dt, \tag{8}$$

particularly, the relation between the total displacement x and the velocity v is written as

$$x(t) = \int_0^t v(t)dt, \tag{9}$$

Please note that this equation is merely formulating the above description.

Of course, we can calculate the integration from Eq.(7) using limit theory; many functions' integrals have already been solved by someone, and we can use the results instead. That's why I won't tell you details about the integration procedures. Here, you only have to remember the relation between the displacement and the velocity.

The same reasoning can be used for the relation between the velocity at t, $v(t)$, and the acceleration at t, $a(t)$; specifically,

$$v(t) = \int_0^t a(t)dt. \tag{10}$$

Here, we can also derive the relation between the $x(t)$ and the $a(t)$ by inserting eq.(10) into eq.(9),

$$x(t) = \int_0^t v(t)dt \tag{11}$$

$$= \int_0^t (\int_0^t a(t')dt')dt. \tag{12}$$

Then, we can write this shortly as

$$x(t) = \int_0^t \int_0^t a(t')dt'dt. \tag{13}$$

Next, let's see the inverse relation: how to derive the velocity $v(t)$ from the displacement $x(t)$ (or acceleration $a(t)$ from the velocity).

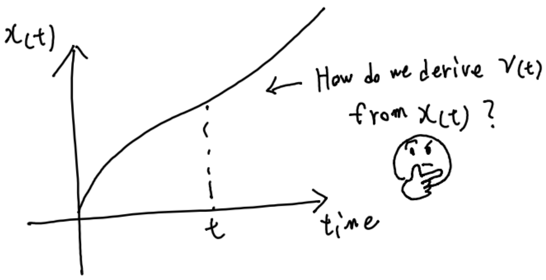

Fig. 13: How do we derive $v(t)$ from $x(t)$?

The answer to the question, "How do we derive $v(t)$ from $x(t)$?" is to start by calculating the average velocity in an interval. The average of the velocity between time t and the time elapsed Δt, $t + \Delta t$, is an elementary calculation;

$$(avg.velocitybetweentandt + \Delta t) = \frac{x(t + \Delta t)}{\Delta t} \tag{14}$$

However, what we want is the velocity at time t, then, like before, taking a small Δt, we get closer to the desired thing.

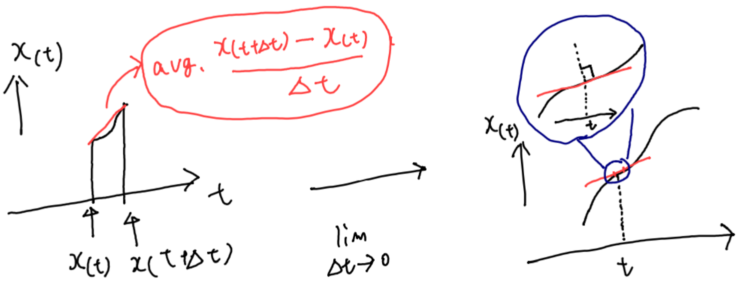

Fig. 14: How do we derive $v(t)$ from $x(t)$?

When the Δt is infinitesimal (see fig. 14), the average of the velocity is identical to the velocity at t (It's a bit paradoxical, but let this slide), that is, $v(t)$.

In mathematical terms, the above description can be written as

$$v(t) = \lim_{\Delta t \to 0} \frac{x(t+\Delta t)}{\Delta t}, \tag{15}$$

or its abbreviation notation,

$$v(t) = \frac{dx(t)}{dt}. \tag{16}$$

The manipulation is called "differentiation".

Again, the relation between the acceleration at t, $a(t)$, and the velocity $v(t)$ is denoted as

$$a(t) = \frac{dv(t)}{dt}. \tag{17}$$

Here, we can also derive the relation between the $a(t)$ and the $x(t)$ by inserting eq.(16) into eq.(17),

$$\begin{aligned} a(t) &= \frac{dv(t)}{dt} \\ &= \frac{\frac{dx(t)}{dt}}{dt}. \end{aligned} \tag{18}$$

then we can write this as

$$a(t) = \frac{d^2x(t)}{dt^2}. \tag{19}$$

All relations are gathered eq. (20) for a recap(also see fig. 15).

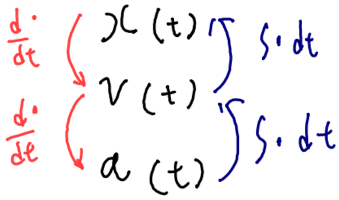

Fig. 15: The relations

$$
\begin{aligned}
x(t) &= x(t) = \int_0^t v(t)dt, && (20)\\
&= \int_0^t \int_0^t a(t')dt'dt. && (21)\\
v(t) &= \int_0^t a(t)dt && (22)\\
v(t) &= \frac{dx(t)}{dt} && (23)\\
a(t) &= \frac{dv(t)}{dt} && (24)\\
&= \frac{d^2x(t)}{dt^2}. && (25)
\end{aligned}
$$

3.1.3 Newton's Laws of Motion

In classical physics, there are principles to capture motion confirmed by experiments. They were named after the physicist who discovered them, Newton's Laws of Motion. Here, I emphasize "confirmed by experiments" because we have to admit them without proof. Once we admit them, we can construct various physical theories on them, and indeed, the theories match the natural effects. Newton's Laws of Motion consist of three laws. Then, let's see the laws.

Newton's First Law Newton's Second Law Newton's Third Law

!!!!In preparation!!!!

3.2 Constrained Dynamics

!!!!In preparation!!!!

3.3 Continuum mechanics

!!!!In preparation!!!!

3.3.1 What is tensor?

!!!!In preparation!!!!

3.3.2 Force, strain, and stress

!!!!In preparation!!!!

3.3.3 Kirchhoff stress tensor

!!!!In preparation!!!!

3.3.4 Cauchy-Green deformation tensor

!!!!In preparation!!!!

3.3.5 St. Venant strain tensor

!!!!In preparation!!!!

4 Numerical integration

!!!!In preparation!!!!

4.1 The linear solvers

!!!!In preparation!!!!

4.2 Newton's method

!!!!In preparation!!!!

4.3 Iterative solvers

!!!!In preparation!!!!

4.4 Local/global solver

!!!!In preparation!!!!

4.5 Misc. topics

!!!!In preparation!!!!

4.5.1 Lagrange Multiplier

!!!!In preparation!!!!

4.5.2 LU decomposition

!!!!In preparation!!!!

4.5.3 Schur decomposition

!!!!In preparation!!!!

5 Numerical physics

!!!!In preparation!!!!

5.1 Verlet's integration

!!!!In preparation!!!!

5.2 Mass-spring system

!!!!In preparation!!!!

5.3 Explicit/implicit Euler method

!!!!In preparation!!!!

5.4 Shape matching

!!!!In preparation!!!!

6 Column : In the terminology mess

!!!!In preparation!!!!

6.1 Newton*

!!!!In preparation!!!!

6.2 Euler*

!!!!In preparation!!!!

6.3 Jacobi*

!!!!In preparation!!!!

6.4 Lagrange*, Hamilton*, Hesse*

!!!!In preparation!!!!

7 Interplet the papers

!!!!In preparation!!!!

7.1 To PBD

!!!!In preparation!!!!

7.1.1 *Large steps in cloth simulation*[1]

!!!!In preparation!!!!

7.1.2 *Advanced Character Physics*[2]

!!!!In preparation!!!!

7.1.3 *A Versatile and Robust Model for Geometrically Complex Deformable Solids*[3]

!!!!In preparation!!!!

7.1.4 *Position Based Dynamics*[4]

!!!!In preparation!!!!

7.2 To XPBD

!!!!In preparation!!!!

7.2.1 *Geometric, Variational Integrators for Computer Animation*[5]

!!!!In preparation!!!!

7.2.2 *Interactive simulation of elastic deformable materials*[6]

!!!!In preparation!!!!

7.2.3 *XPBD : position-based simulation of compliant constrained dynamics*[7]

!!!!In preparation!!!!

7.2.4 *Small steps in physics simulation*[8]

!!!!In preparation!!!!

7.3 Bonus section : Projective Dynamics

!!!!In preparation!!!!

7.3.1 *Example-based elastic materials*[9]

!!!!In preparation!!!!

7.3.2 *Fast simulation of mass-spring systems*[10]

!!!!In preparation!!!!

7.3.3 *Projective Dynamics: Fusing Constraint Projections for Fast Simulation*[11]

!!!!In preparation!!!!

7.3.4 *ADMM $\subseteq$ projective dynamics: fast simulation of general constitutive models*[12]

!!!!In preparation!!!!

7.4 Post-XPBD

!!!!In preparation!!!!

A Inportance of papers

A lower number means more important. The papers' names are aligned in the lexicographic order as possible as I can.

1. You must read these papers if you want to understand (X)PBD. But if you want to understand them completely, I recommend reading the others also.

   - "*Position Based Dynamics*" [4] is one of the main subjects of this document.

   - "*XPBD : position-based simulation of compliant constrained dynamics*" [7] addressed a numerical artifact that makes dependency between stiffness and Iteration count or size of the time-step.

2. These papers are vital in understanding (X)PBD or have a strong impact on the field.

   - "*Advanced Character Physics*" [2] introduced position based simulation method derived from Verlet's integration scheme and combined distance constraints. The framework of PBD can be seen here. Specifically,using distance and angular constraints, modifying verticies' position directoly and solving constraints with some iterations. This paper is easier to read than PBD, but the ideas appear here and pseudo codes are presented. I recommend reading this before read PBD.

   - "*Large steps in cloth simulation*" [1] solved implicit integration with constraints by conjugate gradient method for off-line simulation. The scheme isn't used at PBD, but this one gives us a perspective of present physical simulations.

   - "*Projective Dynamics: Fusing Constraint Projections for Fast Simulation*" [11] provided a position-based method that uses energy formulation and local/global solver. The method has a local Jacobi-like solver and a linear global equation one. The solvers enable robust and fast simulation without safeguards against singular or indefinite Hessians. If you want to understand state-of-the-art physics simulation methods, including PBD, it's better to read this.

   - "*ADMM $\subseteq$ projective dynamics: fast simulation of general constitutive models*" [12] is worth reading because ADMM is actively researched now because of its robustness, parallelizability, and simplicity.

3. These papers offer interesting discussions around PBD, deepen your understanding of PBD, or description of basic physical simulation scheme.

   - "*Constraint Methods for Neural Networks and Computer Graphics*" [16] describes the constraint methods for neural networks and computer graphics. It may not be easy to read because of 150 pages. But, if you have time, it's more worth reading this paper than some papers published before this one.

   - "*Example-based elastic materials*" [9] provides a concept of the elastic manifold and optimization method for deforming into an artist-desirable state. However, this paper lacks reference to the manifold projection methods that are apparently well-known in the geometric optimization field.This paper helps us understand Projective dynamics, but I don't know Projective Dynamics's relevance with (X)PBD for now.

- "*Fast simulation of mass-spring systems*"[10] describes the local/global type solver to the mass-spring system clearly. Thus this paper helps us understand the variant solvers.

- "*Interactive simulation of elastic deformable materials*"[6] introduces physical parameters into constrained dynamics; the spirit is inherited by XPBD and there is an interesting discussion about the integrator. But I think this paper does explain the concept poorly,e. g. the integrator provided here, equation (20) lack of explanation, etc. Therefore, this one classified here.

- "*Nucleus: Towards a unified dynamics solver for computer graphics*"[19] is a good introduction to constrained dynamics because it shows its implementation aspect. However, it doesn't describe how constraints are resolved, so it isn't full-contained.

- "*Robust treatment of collisions, contact and friction for cloth animation*"[20]'s scheme separates physical simulation into internal parts and external parts. Therefore, we can choose the internal modeling(e.g. mass-spring) and the external modeling(e.g. collision repulsion, friction, or gravity) independently. However, the scheme is not directly related to PBD.

- "*Strain Based Dynamics*"[18]. The formulation of the strain tensor in XPBD seems to be based on "Interactive Simulation of Elastic Deformable Materials"(2006) rather than this paper. However, the formulations described here are easy to understand if you are familiar with PBD and strain tensor.

- "*Geometric, Variational Integrators for Computer Animation*"[5] presents the variational integrator from the Lagrangian/Hamiltonian physics that preserves linear/angular momenta. Its conservative quantities will be more important in fields such as robotics or accurate physical computation. But, this knowledge may be useless if you only want to understand (X)PBD.

- "*Efficient simulation of inextensible cloth*"[17] treated the constrained system as globally linearized form and solved with a direct approach at each iteration.

4. These ones have historical value, but deeper discussions are done in other papers.

   - "*Elastically deformable models*"[21] brought the formulation of elastic bodies to computer graphics.

- "*Energy Constraints On Parameterized Models*" [**?**] uses shape representations that quite different from the current ones and the constraints presented in this paper are slightly inconvenient to the current ones. Thus, we no longer have to read this.

- "*A modeling system based on dynamic constraints*" [22] uses constraints as models' motion rather than to hold a model's detail and uses linear simultaneous equations when deriving forces.The points that are difficult to understand are that the paper doesn't describe the background of the equation derivation and that the symbols are scattered too much. Fortunately, we don't have to read this paper completely to understand present constrained dynamics because the style varies from the recent ones.

5. Not be classified yet.

   - "*A Versatile and Robust Model for Geometrically Complex Deformable Solids*" [3]
   - "*Meshless deformations based on shape matching*" [23]
   - "*Fast Simulation of Inextensible Hair and Fur*" [24]
   - "*Long Range Attachments - A Method to Simulate Inextensible Clothing in Computer Games*" [25]
   - "*Position Based Fluids*" [26]
   - "*Position-based simulation of continuous materials*" [27]
   - "*Unified particle physics for real-time applications*" [28]
   - "*Air Meshes for Robust Collision Handling*" [29]
   - "*A survey on position based dynamics, 2017*" [13]
   - "*Stable Constrained Dynamics*" [30]
   - "*Small steps in physics simulation*" [8]
   - "*Non-Smooth Newton Methods for Deformable Multi-Body Dynamics*" [31]
   - "*Detailed Rigid Body Simulation with Extended Position Based Dynamics*" [32]
   - "*A Constraint-based Formulation of Stable Neo-Hookean Materials*" [33]
   - "*Physically Based Shape Matching*" [34]
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []
   - "" []

- “” []
- “” []
- “” []
- “” []
- “” []
- “” []
- “” []
- “” []

B Glossaly

B.1 symbols

B.2 terms

conjugate gradient method
This method, abbreviated as CG method is used linear equations that form is $A\boldsymbol{x} = \boldsymbol{b}$. This method often has high computational cost .
constraint
At the (X)PBD, constraint is relationship among arbitrary vertices what we want to keep during simulation. It can express not only between two vertices relation, e.g. distance a vertex to another one, but among more than three vertices, e.g. volume of the tetrahedron .

elastic-body
This is an object that resists deformation and returns to its original shape when that influence or force is removed. Sometimes, the term refers to objects with high stiffness(e.g. steel wire). .

Iteration
In computer science, iteration is the process of repeating a series of instructions multiple times. Especially at (X)PBD, the part of physical solver which treats constraints is called iteration .

Lagrangian mechanics
This is one of the physics theories that conserves momentum in addition to inertia. Variational integration is a crucial concept in this theory. Hamiltonian mechanics is also known as a more generalized form than this theory. .

soft-body
This is an object that deforms by external force. The term is often compared to ”Rigid body” which refers to an undeformable object. .

References

[1] Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '98, pp. 43–54. Association for Computing Machinery, New York, NY, USA (1998). https://doi.org/10.1145/280814.280821 . https://doi.org/10.1145/280814.280821

[2] Jakobsen, T.P.: Advanced character physics. (2003). https://api.semanticscholar.org/CorpusID:5102710

[3] Teschner, M., Heidelberger, B., Muller, M., Gross, M.: A versatile and robust model for geometrically complex deformable solids. In: Proceedings Computer Graphics International, 2004., pp. 312–319 (2004). https://doi.org/10.1109/CGI.2004.1309227

[4] Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. J. Vis. Comun. Image Represent. **18**(2), 109–118 (2007) https://doi.org/10.1016/j.jvcir.2007.01.005

[5] Kharevych, L., Yang, W., Tong, Y., Kanso, E., Marsden, J.E., Schröder, P., Desbrun, M.: Geometric, variational integrators for computer animation. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '06, pp. 43–51. Eurographics Association, Goslar, DEU (2006)

[6] Servin, M., Lacoursière, C., Melin, N.: Interactive simulation of elastic deformable materials. (2006). https://api.semanticscholar.org/CorpusID:16851165

[7] Macklin, M., Müller, M., Chentanez, N.: Xpbd: position-based simulation of compliant constrained dynamics. In: Proceedings of the 9th International Conference on Motion in Games. MIG '16, pp. 49–54. Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2994258.2994272 . https://doi.org/10.1145/2994258.2994272

[8] Macklin, M., Storey, K., Lu, M., Terdiman, P., Chentanez, N., Jeschke, S., Müller, M.: Small steps in physics simulation. In: Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '19. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3309486.3340247 . https://doi.org/10.1145/3309486.3340247

[9] Martin, S., Thomaszewski, B., Grinspun, E., Gross, M.: Example-based elastic materials. In: ACM SIGGRAPH 2011 Papers. SIGGRAPH '11. Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/1964921.1964967 . https://doi.org/10.1145/1964921.1964967

[10] Liu, T., Bargteil, A.W., O'Brien, J.F., Kavan, L.: Fast simulation of mass-spring systems. ACM Trans. Graph. **32**(6) (2013) https://doi.org/10.1145/2508363.2508406

[11] Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective Dynamics: Fusing Constraint Projections for Fast Simulation, 1st edn. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3596711.3596794

[12] Narain, R., Overby, M., Brown, G.E.: Admm $\subseteq$ projective dynamics: fast simulation of general constitutive models. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '16, pp. 21–28. Eurographics Association, Goslar, DEU (2016)

[13] Bender, J., Müller, M., Macklin, M.: A survey on position based dynamics, 2017. In: Proceedings of the European Association for Computer Graphics: Tutorials. EG '17. Eurographics Association, Goslar, DEU (2017). https://doi.org/10.2312/egt.20171034 . https://doi.org/10.2312/egt.20171034

[14] Witkin, A., Fleischer, K., Barr, A.: Energy constraints on parameterized models. SIGGRAPH Comput. Graph. **21**(4), 225–232 (1987) https://doi.org/10.1145/37402.37429

[15] Barzel, R., Barr, A.H.: A modeling system based on dynamic constraints. SIGGRAPH Comput. Graph. **22**(4), 179–188 (1988) https://doi.org/10.1145/378456.378509

[16] Platt, J.C.: Constraint methods for neural networks and computer graphics. PhD thesis, USA (1990). UMI Order No: GAX90-00594

[17] Goldenthal, R., Harmon, D., Fattal, R., Bercovier, M., Grinspun, E.: Efficient simulation of inextensible cloth. ACM Trans. Graph. **26**(3), 49 (2007) https://doi.org/10.1145/1276377.1276438

[18] Müller, M., Chentanez, N., Kim, T.-Y., Macklin, M.: Strain based dynamics. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '14, pp. 149–157. Eurographics Association, Goslar, DEU (2015)

[19] Stam, J.: Nucleus: Towards a unified dynamics solver for computer graphics. In: 2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, pp. 1–11 (2009). https://doi.org/10.1109/CADCG.2009.5246818

[20] Bridson, R., Fedkiw, R., Anderson, J.: Robust treatment of collisions, contact and friction for cloth animation. ACM Trans. Graph. **21**(3), 594–603 (2002) https://doi.org/10.1145/566654.566623

[21] Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically

deformable models. SIGGRAPH Comput. Graph. **21**(4), 205–214 (1987) https://doi.org/10.1145/37402.37427

[22] Barzel, R., Barr, A.H.: A modeling system based on dynamic constraints. SIGGRAPH Comput. Graph. **22**(4), 179–188 (1988) https://doi.org/10.1145/378456.378509

[23] Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. ACM Trans. Graph. **24**(3), 471–478 (2005) https://doi.org/10.1145/1073204.1073216

[24] Müller, M., Kim, T., Chentanez, N.: Fast simulation of inextensible hair and fur. (2012). https://doi.org/10.2312/PE/vriphys/vriphys12/039-044

[25] Kim, T., Chentanez, N., Müller, M.: Long range attachments - a method to simulate inextensible clothing in computer games, pp. 305–310 (2012). https://doi.org/10.2312/SCA/SCA12/305-310

[26] Macklin, M., Müller, M.: Position based fluids. ACM Trans. Graph. **32**(4) (2013) https://doi.org/10.1145/2461912.2461984

[27] Bender, J., Koschier, D., Charrier, P., Weber, D.: Position-based simulation of continuous materials. Computers & Graphics **44**, 1–10 (2014) https://doi.org/10.1016/j.cag.2014.07.004

[28] Macklin, M., Müller, M., Chentanez, N., Kim, T.-Y.: Unified particle physics for real-time applications. ACM Trans. Graph. **33**(4) (2014) https://doi.org/10.1145/2601097.2601152

[29] Müller, M., Chentanez, N., Kim, T.-Y., Macklin, M.: Air meshes for robust collision handling. ACM Trans. Graph. **34**(4) (2015) https://doi.org/10.1145/2766907

[30] Tournier, M., Nesme, M., Gilles, B., Faure, F.: Stable constrained dynamics. ACM Trans. Graph. **34**(4) (2015) https://doi.org/10.1145/2766969

[31] Macklin, M., Erleben, K., Müller, M., Chentanez, N., Jeschke, S., Makoviychuk, V.: Non-smooth newton methods for deformable multi-body dynamics. ACM Trans. Graph. **38**(5) (2019) https://doi.org/10.1145/3338695

[32] Müller, M., Macklin, M., Chentanez, N., Jeschke, S., Kim, T.-Y.: Detailed rigid body simulation with extended position based dynamics. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '20. Eurographics Association, Goslar, DEU (2020). https://doi.org/10.1111/cgf.14105 . https://doi.org/10.1111/cgf.14105

[33] Macklin, M., Muller, M.: A constraint-based formulation of stable neo-hookean

materials. In: Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games. MIG '21. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3487983.3488289 . https://doi.org/10.1145/3487983.3488289

[34] Müller, M., Macklin, M., Chentanez, N., Jeschke, S.: Physically based shape matching. Computer Graphics Forum **41**, 1–7 (2023) https://doi.org/10.1111/cgf.14618