XPBD: Position-Based Simulation of Compliant Constrained Dynamics

ALEXANDRA PILIPYUK, Institut Polytechnique de Paris, France, Feb 2024

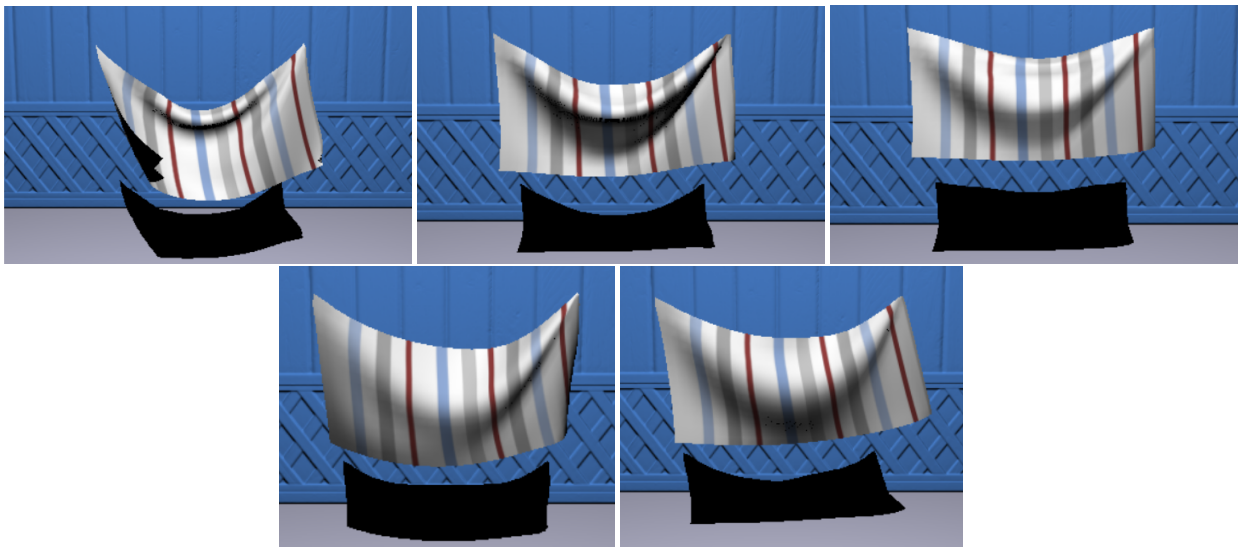

Fig. 1. Comparisom of PBD and XPBD methods on a peace of cloth. Upper row: three results of PBD with iterations 5, 20, 40. Lower row: results of XPBD with iterations 20, 40. We can observe that the difference of stiffness of the cloths is visually noticeable on the PBD results, and not noticeable on XPBD results.

Position Based Dynamics (PBD) is a powerful method for simulation without heavy and complex computations of equations from physics laws but with visually pleasant results. Extended position-based dynamics (XPBD) introduces an extension to PBD that aims for soliving the long-standing problem of iteration count and time step dependency, providing the higher physical accuracy of the simulation, stability, and lower dependence on the time step and number of iterations. This is a project about discovering PBD and XPBD, their properties and variations, implementing them, and comparing their results to each other.

1 INTRODUCTION

Original PBD [9] has a well known limitation that its behavior is dependent on the time step and iteration count of the simulation. Specifically, constraints become more stiff as the iteration count increases, or as the time step decreases. It may cause problems when having several object made of different materials in the same simulation. Moreover, the effects of iteration count are non-linear, making it difficult to intuitively adjust parameters, or to eliminate the problem by rescale stiffness values. XPBD [4] introduces additional measures that allow us to solve constraints regardless of the time step and iteration count.

In this project, I am implementing and presenting XPBD algorithm with attachment, stretching and bending constraints, showing the result on the example of a cloth and a cube.

The steps were the following:

- Study and implement PBD with the three constraints
- Study XPBD, search for materials
- Implement the stretching constraint for XBPD
- Derive and implement the bending constraint for XPBD
- Implement dampling
- Compare the results

Author's address: Alexandra Pilipyuk, Institut Polytechnique de Paris, France, Feb 2024.

2 RELATED WORKS

The PBD family of algorithms continues to be developed and improved by the research team in Nvidia. The original PBD approach [9] meets different extensions and improvements. XPBD [4] propose one of the extensions. A comprehensive review of recent PBD extensions can be found in [2].

Another, more efficient approach is Hierarchical Position Based Dynamics [5]. XPBD focuses on extending the PBD framework to handle non-linear constraints and dynamic changes in topology, making it particularly suitable for soft body simulations and deformable objects. In contrast, HPBD combines PBD with traditional methods like Finite Element Method (FEM) or Mass-Spring Systems to achieve higher fidelity in modeling complex deformations, often used in scenarios requiring high accuracy such as medical simulations or engineering analyses.

One of the interesting works based on XPBD is a Rigid Body Simulator [7] where the authors propose the modifications to XPBD allowing to simulate rigid body mechanics.

3 APPROACH

3.1 Technical details

The main algorithm consists of steps similar to the original PBD and can be found in the paper [4]. The difference of XPBD is in having an additional Lagrange multiplier λ to replace the stiffness coefficient k and the scaling factor s from PBD. λ is a unique value

for each constraint and it is updated on each iteration alongside updating the positions p_i.

That is, in the original PBD, to update the single point p_i for the constraint $C(p_1, ..p_n)$ the scaling factor is represented by

$$s = \frac{C(p_1, .., p_n)}{\sum_j w_j |\nabla_j C(p_1, .., p_n))|^2}$$

where n is a number of points, $\nabla_i C(p_1, .., p_n)$ denotes $\frac{\delta C(p_1,..,p_n))}{\delta p_i}$, and w_i is the inverse mass of the current point. The update of the position is

$$\Delta p_i = -s \cdot w_i \cdot \nabla_i C(p_1, ..p_n)$$

With the Lagrange multiplier we are updating it and the positions as following:

$$\Delta\lambda = \frac{-C(p_1, .., p_n) - \tilde{\alpha}\lambda}{\sum_j w_j |\nabla_j C(p_1, .., p_n))|^2 + \tilde{\alpha}}$$

$$\Delta p_i = \Delta\lambda \cdot w_i \cdot \nabla_i C(p_1, .., p_n)$$

Here, the stiffness k is replaced by the compliance, or the inverse stiffness, α, and having $\tilde{\alpha} = \frac{\alpha}{\Delta t^2}$.

Then it comes for the particular constraints formulations and deriving their gradients, which might be tricky. Unlike in the PBD paper, in this paper authors don't provide the examples of computing particular functions leaving this exercise to the reader.

3.2 Stretching constraint

This is the mostly used constraint and it can be found in the implementations of XPBD ([1], [10]). One of the authors of the paper has a series of videos explaining different simulation techniques, including XPBD [8], however the implementation he provides slightly differs from the algorithm showed in the paper, it looks like a simplified version of it.

Remembering the updating formulas from the previous section and a stretching constraint function with d_0 for the rest length of the edge between p_1 and p_2 given by

$$C(p_1, p_2) = |p_1 - p_2| - d_0$$

with

$$\Delta_i C(p_1, p_2) = \pm \frac{p_1 - p_2}{|p_1 - p_2|},$$

we receive the following updates after the substitution:

$$\Delta\lambda = \frac{d_0 - |p_1 - p_2| - \tilde{\alpha}\lambda}{w_1 + w_2 + \tilde{\alpha}}$$

$$\Delta p_1 = w_1 \cdot \Delta\lambda \cdot \frac{p_1 - p_2}{|p_1 - p_2|}$$

$$\Delta p_2 = -w_2 \cdot \Delta\lambda \cdot \frac{p_1 - p_2}{|p_1 - p_2|}$$

The question is how to choose the compliance parameter α. The authors of the paper come into play here with the article [6], showing the compliance values for different materials.

On the Fig. 2 I present the result of my implementation for stretching constraint with different compliance values.

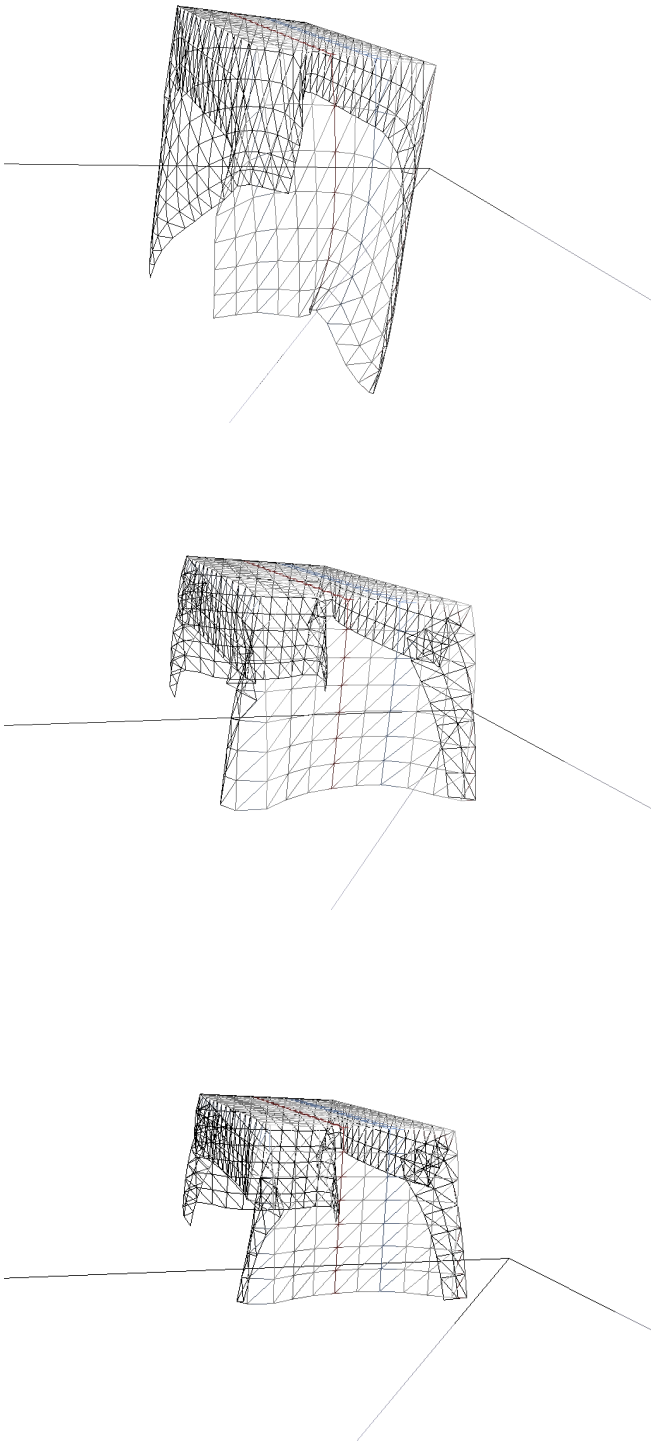

Fig. 2. Example of the stretching constraint with compliance values 1e-3, 1e-5, 1e-9. The first picture shows the dependence on the mesh structure.

3.3 Bending constraint

For the bending constraint, I couldn't find a solution for XPBD that would use it, so I needed to derive it by myself. After careful studying the XPB paper and its appendix, I came up with formulations:

$$\Delta\lambda = \frac{\varphi_0 - arccos(d) - \tilde{\alpha}\lambda}{\frac{1}{1-d^2}\sum_j w_j |q_i|^2 + \tilde{\alpha}}$$

$$\Delta p_i = \Delta\lambda w_i \frac{q_i}{\sqrt{1-d^2}}$$

using the notations from PBD appendix for q_i, d, and that

$$\Delta_i C(p_1, p_2, p_3, p_4) = \frac{-q_i}{\sqrt{1-d^2}}$$

The results of the implemented bending constraint with different compliance values can be found on the Fig.3. For bending, the compliance values given by Miles Macklin [6] cannot be applied, because they correspond to the material stiffness given by Young's modulus, but in case for bending we need the coefficient for angle and not for linear stretching, which yields different values for the compliance.

When implementing the bending constraint, one needs to treat carefully the fact that for angles $\pi n, n \in \mathbf{N}$ the derivative is not defined. So for the points where that is the case, I don't update the position. Another option is to substitute a slightly shifted value $\pi+\epsilon$. To see what may happen when this exceptional case is not processed, the reader can watch a supplemental video named failure.mp4.

3.4 Alternative solutions for bending

In the resources I found about XPBD, people prefer not to deal with the bending constraints based on the angle, but propose alternative methods. One of them is to use the stretching constraint between two vertexes that belong to the same triangles as the edge. This way we can avoid complex computations. It works well with flat models, like for example cloth. However, when doing so, the order in which constraints are satisfied will matter. Solving the bending constraint by adjusting the distance will not be representative until the distances for stretching constraint are satisfied, and we will "loose" iterations for bending. Firstly, I have tried this method, but it seems to be less sensitive to the compliance and harder to tune.

Another approach, that is interesting to consider, is to take only stretching and volume constraints, but use a different type of mesh - tetrahedral mesh. When each face of the original mesh is not just a triangle, but a face of a tetrahedron. Thus, conservation of the volume for each tetrahedron would play the role of the bending constraint. The limitation of this approach is that tetrahedral mesh are less common than meshes with 2D cells, so it will require extra effort on creating a good mesh representation.

3.5 Damping

Damping was implemented as proposed in formula (26) of the XPBD paper and based on the hint to compute the gradient from [3]. With damping, the simulated cloth stops its movements. Without the damping, it has waves going from the bottom to the top unstoppably.

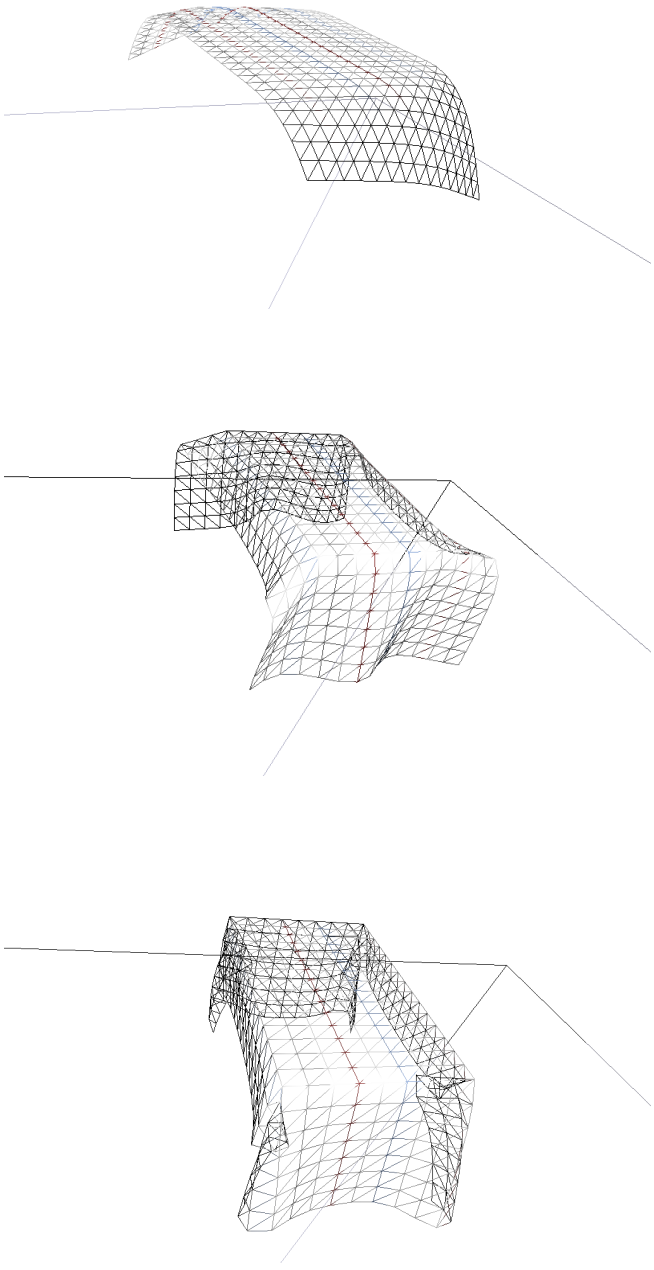

Fig. 3. Example of the bending constraint with compliance values 0, 0.5, 5. The first picture shows the dependence on the mesh structure

4 OBSERVATIONS & FUTURE WORKS

XPBD is a method that shows good simulations, however it is not without the cons. I would like to summarise some drawbacks of the XPBD method below.

- As mentioned during the presentation in the class, XPBD is quite a slow method. On my not very powerful computer, the simulation became noticeably slow at around 60 iterations for a cloth of size 30x15 points.
- The computations of derivatives and exceptional cases might be tricky
- It might be tricky to find the proper values for the compliance.
- The result heavily depends on the mesh structure, which is well seen on the Fig. 2 and Fig. 3. This could be eliminated by using the constraints that would rely not only on direct neighboring vertexes separately but on all the neighbors together or vertexes in the nearest surrounding area.

There are plenty of ways to improve the existing implementation. The possible directions of extensions are:

- Add other constraints: volume, cloth balloons,...
- Add support of tetrahedral mesh
- Improve the interaction
- Extension for rigid bodies

5 CONCLUSION

Thus, I proposed the implementation of XPBD method. It shows a nice simulation of soft bodies on the example of a cloth and a cube. Through its iterative approach and flexibility, XPBD enables to efficiently simulate complex dynamic systems while maintaining stability and accuracy. The XPBD has a potential as a powerful tool for dynamic simulation in various applications, paving the way for further advancements in the field.

Despite the room for improvement of the current implementation, working on this project provided me with valuable insight into the concepts of PBD and XPBD for simulation, as well as to their potential extensions and applications. Through this experience, I gained a deeper understanding of the challenges involved in the PBD-family approaches and experience with implementing and tuning them.

REFERENCES

[1] blackedout01. 2023. Writing a soft body cube in C/C++ using XPBD: Devlog episode 1. https://youtu.be/MgmXJnR62uA?si=ByMqxBDtwa7P7rwx

[2] Junheng Fang, Lihua You, Ehtzaz Chaudhry, and Jian Zhang. 2023. State-of-the-art improvements and applications of position based dynamics. *Computer Animation and Virtual Worlds* 34, 5 (2023), e2143. https://doi.org/10.1002/cav.2143 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cav.2143

[3] korzen303. 2020. Distance Constraints Damping in XPBD. https://www.gamedev.net/forums/topic/707328-distance-constraints-damping-in-xpbd/

[4] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: Position-Based Simulation of Compliant Constrained Dynamics. https://doi.org/10.1145/2994258.2994272

[5] Matthias Müller. 2008. Hierarchical Position Based Dynamics., Vol. 8. 1–10. https://doi.org/10.2312/PE/vriphys/vriphys08/001-010

[6] Matthias Müller. 2016. XPBD slides and stiffness. http://blog.mmacklin.com/2016/10/12/xpbd-slides-and-stiffness/

[7] Matthias Müller. 2020. Detailed Rigid Body Simulation using Extended Position Based Dynamics.

[8] Matthias Müller. 2023. Ten Minute Physics. https://github.com/matthias-research/pages/blob/master/tenMinutePhysics/14-cloth.html

[9] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position Based Dynamics. *Journal of Visual Communication and Image Representation* 18 (04 2007), 109–118. https://doi.org/10.1016/j.jvcir.2007.01.005

[10] nobuo nakagawa. 2017. XPBD implementation. https://github.com/nobuo-nakagawa/xpbd/tree/master