State-of-the-art Improvements and Applications of Position Based Dynamics

Abstract

The emergence of position-based simulation approaches has quickly developed a group of new topics in the computer graphics community. These approaches are popular due to their advantages, including computational efficiency, controllability, stability and robustness for different scenarios, whilst they also have some weaknesses. In this survey, we will introduce the concept of the baseline position based dynamics (PBD) method and review the improvements and applications of PBD since 2018, including extensions for different materials and integrations with other techniques.

Keywords: position-based approaches, basic concepts, recent improvements, new applications

1. Introduction

Dynamic simulation of 3D models is an integral part in the field of computer animation. There exist many works developing advanced simulation techniques, which have been reviewed in the papers by Jan et al.[1] and Gibson and Mirtich[2]. However, it will always be how to make the models appear realistic animation in real-time that values the performance at last. This fundamental criterion could be achieved by shape deformation methods, which could be roughly classified into geometry-based and physics-based ones. The pure geometry approaches, including Free Form Deformations (FFDs)[3], joint-related approaches[4,5] and data-driven methods[6], manipulate the control structure of 3D models to change shapes. The physics-based approaches use forces to compute acceleration based on physical laws such as Newton's second law and update the corresponding velocities with time integration, including impulse-based methods[7], mass-spring system[8], and finite element method (FEM)[9]. Since the geometry-based methods do not involve any physics of the surface deformation but directly work on the positions, their generated animations appear unnatural. The physics-based methods generate more realistic animations but also have drawbacks like overshooting problems and expensive computational costs as they require velocity and acceleration layers and usually involve heavy numerical calculations. Thus, they cannot fulfil the criterion either.

Since position based dynamics (PBD) methods well balance animation realism and efficiency and meet the mentioned criterion better than other methods, they have recently become a focal point in computer graphics and exceedingly popular in simulating dynamic systems due to its speed, robustness and simplicity. PBD was first proposed by Müller et al.[10] It inherits the advantages of geometry-based methods to omit the velocity and acceleration layer but directly works on the positions, with considering the solution to a quasi-static problem. These allow PBD to perform well in real-time simulation with visually plausible deformation and good controllability.

The core algorithm of position-based approaches and some previous applications have been detailedly introduced in the papers by Jan et al.[11,12] The position-based approaches were initially proposed for interactive environments to simulate solid objects. It was later demonstrated that they could also be involved in the simulation of fluids, articulated rigid bodies and other application scenarios.

In this survey, we will review the improvements and applications of position-based methods for dynamic scenarios in recent years. Since the survey papers[11,12] have thoroughly reviewed the previous works related to PBD before 2018, we will only mention some of them for better reading but focus more on the advanced progress since 2018. The review will first present the basic theory of PBD in Section 2. Section 3 and Section 4 will respectively introduce the recent improvements and applications with the position-based methods. In Section 5, some future research directions will be suggested to inspire the researchers in the field.

2. Basic concept of PBD

In this section, we will take particle-based dynamic objects as an example to introduce the basic algorithm of PBD according to the papers.[10-12]

2.1 The Algorithm Overview

In a particle-based system, a dynamics object could be represented by a set of N vertices and M constraints. The i^{th} ($i \in [1,2,\dots,N]$) vertex has the following attributes: mass m_i, position $\boldsymbol{x}_i$, and velocity $\boldsymbol{v}_i$. The set of M constraints is applied to modify the position and velocity attributes of the N vertices at the next time step, computed by a group of functions C_j ($j \in [1,2,\dots,M]$).

The i^{th} point of the dynamic object, with its mass m_i and initial position $\boldsymbol{x}_i^0$ and velocity $\boldsymbol{v}_i^0$, is simulated at a timestep Δt as follows:

(1) for i in $[1, N]$:
(2) initialise $\boldsymbol{x}_i = \boldsymbol{x}_i^0, \boldsymbol{v}_i = \boldsymbol{v}_i^0, w_i = 1/m_i$
(3) do $\boldsymbol{v}_i = \boldsymbol{v}_i + \Delta t\, w_i\, \boldsymbol{f}_{external}(\boldsymbol{x}_i)$
(4) do $dampVelocities(\boldsymbol{v}_i)$
(5) do $\boldsymbol{p}_i = \boldsymbol{x}_i + \Delta t\, \boldsymbol{v}_i$
(6) do $generateCollisionConstraints(\boldsymbol{x}_i \rightarrow \boldsymbol{p}_i)$ for all vertices i
(7) for $times$ in $[1,\ solverIterations]$:
(8) do $projectConstraints(C_1, C_2, \dots, C_M, \dots, C_{M+M_{coll}},\ \boldsymbol{p}_{1,\dots,}\boldsymbol{p}_N)$
(9) do $\boldsymbol{v}_i = (\boldsymbol{p}_i - \boldsymbol{x}_i)/\Delta t$
(10) do $\boldsymbol{x}_i = \boldsymbol{p}_i$
(11) do $\boldsymbol{v}_i = velocityUpdate(\boldsymbol{v}_i)$
(12) do $\boldsymbol{x}_i^1 = \boldsymbol{x}_i, \boldsymbol{v}_i^1 = \boldsymbol{v}_i$

Line (2) uses the current attributes of each vertex to initialise the state variables. Lines (3)-(5) allow some external forces like gravity, which cannot be computed as positional constraints, to be allied to the system through a simple symplectic Euler integration step. Line (4) is an optional damping step for improving the simulation performance. Here $\boldsymbol{p}_i$ are only used as predictions. In addition to the fixed constraints C_j ($j = 1, 2, 3, \dots, M$) , line (6) generates M_{coll} non-permanent collision constraints at the beginning of each timestep. Lines (7)-(8) use a solver, which considers both the fixed constraints $C_1 \sim C_M$ and collision constraints $C_{M+1} \sim C_{M+M_{coll}}$, to correct the predicted position $\boldsymbol{p}_i$. Line (11) computes friction and restitution coefficients to modify the velocities of colliding vertices. In the last line (12), the corrected position $\boldsymbol{p}_i$ will be used to update the velocity and position attributes $\boldsymbol{v}_i$ and $\boldsymbol{x}_i$ of the vertex at the next time step.

2.2 Damping

The quality of PBD simulations can be improved by incorporating a damping term $\boldsymbol{C}\dot{\boldsymbol{X}}$ into Newton's second law where $\dot{\boldsymbol{X}}$ is a velocity vector and $\boldsymbol{C}$ is a damping matrix. In line (4) of the simulation algorithm, the velocities are damped to improve the stability of the dynamic simulation through decreasing temporal oscillations of the point positions. Among all dampings, point damping,[13] which improves the point stability, and spring damping,[14] which conserves the linear and angular momentum, are most often used due to their appropriateness.

2.3 Collision

In line (6) of the simulation algorithm, extra collision constraints are generated for the following correction steps. The collision constraints can be divided into two groups: the first group is between dynamic and static objects, and the second group is between dynamic objects.

The collision between dynamic and static objects could be handled as continuous and static collisions depending on whether the ray $\boldsymbol{x}_i \rightarrow \boldsymbol{p}_i$ for i^{th} vertex crosses an object or is in the internal space of an object.

If the ray crosses an object, the generated collision constraint is handled as a continuous collision:

$$C(\boldsymbol{p}_i) = (\boldsymbol{p}_i - \boldsymbol{q}_c) \cdot \boldsymbol{n}_c$$

where $\boldsymbol{q}_c$ stands for the intersection point, and $\boldsymbol{n}_c$ denotes the surface normal at $\boldsymbol{q}_c$.

If the ray is entirely in the internal space of an object, it is handled as a static collision:

$$C(\boldsymbol{p}_i) = (\boldsymbol{p}_i - \boldsymbol{q}_s) \cdot \boldsymbol{n}_s$$

where $\boldsymbol{q}_s$ stands for the closest surface point to $\boldsymbol{p}_i$, and $\boldsymbol{n}_s$ denotes the surface normal at $\boldsymbol{q}_s$.

These separately generated constraints allow the simulation to reduce significant running time as they are not generated in the following solver iterations.

The collision between two dynamic objects is more complex by simulating both objects. Taking a triangle face of one object with vertices $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$ as a reference substance, when the i^{th} point of the other object $\boldsymbol{p}_i$ moves through it, the generated collision constraint is handled as:

$$C(\boldsymbol{p}_i, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3) = \pm(\boldsymbol{p}_i - \boldsymbol{q}_1) \cdot [(\boldsymbol{q}_2 - \boldsymbol{q}_1) \times (\boldsymbol{q}_3 - \boldsymbol{q}_1)]$$

2.4 Solver

In lines (7)-(8) of the simulation algorithm, a solver is used for correcting the estimated new position $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N$ of the N vertices to satisfy the $M + M_{coll}$ constraints. The constraints could be yielded as non-linear equations and inequalities. Thus, a non-linear Gauss-Seidel iteration is introduced, which provides the ability to handle each constraint separately one after the other.

The equations and inequalities representing the constraints could be unified in the form $C(\boldsymbol{p}) \succ 0$, where the symbol $\succ$ denotes either $=$ or $\geq$. Therefore, the aim changes to finding a correction $\Delta\boldsymbol{p}$, which fulfils $C(\boldsymbol{p} + \Delta\boldsymbol{p}) \succ 0$. The nonlinear constraints can be linearised through:

$$C(\boldsymbol{p} + \Delta\boldsymbol{p}) = C(\boldsymbol{p}) + \nabla C(\boldsymbol{p}) \cdot \Delta\boldsymbol{p} + O(|\Delta\boldsymbol{p}|^2) \succ 0$$

Considering a second order accurate approximation, the above equation can be written as:

$$C(\boldsymbol{p} + \Delta\boldsymbol{p}) \approx C(\boldsymbol{p}) + \nabla C(\boldsymbol{p}) \cdot \Delta\boldsymbol{p} \succ 0 \qquad (2.1)$$

For the conservation of linear and angular momentum, $\Delta\boldsymbol{p}$ should be restricted to be in the direction of ∇C, which means only one scalar λ called a Lagrange multiplier should be found to get:

$$\Delta\boldsymbol{p} = \lambda \nabla C(\boldsymbol{p}) \qquad (2.2)$$

Substituting Eq. (2.2) into Eq. (2.1), the scalar λ is determined. Substituting it back into Eq. (2.2), the correction of an individual vertex $\boldsymbol{p}_i$ can be formulated as:

$$\Delta\boldsymbol{p}_i = -\lambda\, w_i\, \nabla_{\boldsymbol{p}_i} C(\boldsymbol{p}) \qquad (2.3)$$

where

$$\lambda = \frac{C(\boldsymbol{p})}{\sum_j w_j \left|\nabla_{\boldsymbol{p}_j} C(\boldsymbol{p})\right|^2} \qquad (2.4)$$

Finally, a stiffness parameter $k \in [0,1]$ will be introduced to define the strength of the constraints. To get a linear dependence on the stiffness parameter after n_s solver iterations, $k' = 1 - (1-k)^{1/n_s}$ is used to multiply the correction $\Delta\boldsymbol{p}$.

3. Recent Improvements of PBD

The main limitations of PBD could be identified as: 1) It could not efficiently achieve a convergent solution during iterations. 2) The dependency of the results on the stiffness parameter k, timestep Δt and number of iterations n_{iter} could not be completely eliminated, which causes insufficient simulation accuracy. 3) The handling order of different constraints could affect the simulation results.

In this section, we will review recent improvements of PBD, including advanced algorithms tackling the main limitations and the progress of extensions for other objects like fluids and cloth.

3.1 Improvements in Convergence Problem

As described in Section 2, the iterative solver is the most crucial step in PBD simulation. It helps the system get better corrections after each iteration, leading to more realistic deformation results. However, PBD uses a nonlinear Gauss-Seidel solver to iterate the projections, which makes the projection propagation slow and hard to achieve a convergent solution. Many methods have been proposed to improve the convergence efficiency of the solver, like Hierarchical Position Based Dynamics (HPBD),[15] and the second-order accurate multistep method applied with a second-order backward differentiation formula (BDF2).[16] HPBD defines a multi-grid based mesh to make the error corrections propagate faster while the tearing algorithm requires further improvement. By introducing BDF2 into PBD, only the previous timestep information is used, which makes projection convergence faster, but the method is probably uncompetitive for stretchy materials. Besides, other methods have been proposed to further

Figure 1: A twisted rope simulation results from the XPBD-based rigid body simulation method.[20]

speed up error propagation, like the method of Long Range Attachments (LRA),[17] which is best suited for inextensible character clothing, but does not provide benefits for unattached environmental cloth such as flying papers.

3.2 Improvements in Dependence Problem

For evaporating the dependence of the simulation results on the stiffness parameter k, timestep Δt and number of iterations n_{iter}, many methods have been proposed. Among them, extended position based dynamics (XPBD) and projective dynamics (PD) are highlighted since they not only make progress in this problem, but also significantly develop position-based approaches into separate research topics. This subsection will introduce these two methods, including their own developments.

3.2.1 Extended Position Based Dynamics

XPBD was first presented in the paper by Miles et al.[18] One of its initial functions is to address the dependence problem of PBD. It is achieved by associating a compliance $\alpha = \frac{1}{k}$ with each constraint. With this trivial modification to the PBD solver, a stiff solution could be reached regardless of the timestep size. Besides, XPBD could return a consistent solution to provide accurate constraint force predictions for force-related effects.

Based on XPBD, Macklin et al. proposed a method[19] to split every timestep into n isometric substeps and perform an iteration of XPBD in each substep. This method dramatically improves the achievable stiffness with a low computational cost. In addition, Müller et al. presented an XPBD-based method to precisely resolve small spatial and temporal details in the rigid body simulation.[20]

Figure 1 shows its ability to simulate a twisted rope. In the paper by Romeo et al.[21,73] a modification of XPBD was used to define the simulation of muscle dynamics through modifying the distance constraints between mesh vertices in the solver. Moreover, XPBD has been used to achieve differential parameter identification and shape control of linear objects by adding extra geometrical constraints for real-to-sim robotic manipulation.[22]

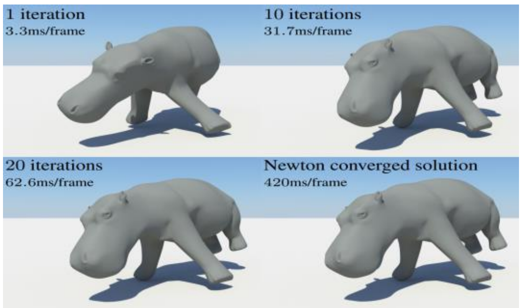

Figure 2: The comparison between the simulation results of a hippopotamus model with PD after 1, 10, and 20 iterations, and Newton's physical method. It could be seen that after 10 iterations, the simulation results are visually similar.[23]

3.2.2 Projective Dynamics

Projective Dynamics (PD) was proposed as a modification of PBD.[23] It applies extra constraints on the solver to improve its robustness for non-uniform meshing with distinct resolutions since the improved solver allows it to handle object interactions implicitly. With the additional constraints, PD also decreases the dependence between the stiffness and the iteration number. Moreover, it is even faster convergent than Newton's method, so it costs significantly less computational time, as shown in Figure 2.

While Bouaziz et al. have proved that PD could be applied to simulate many materials like cloths, shells and solids due to its robustness and simplicity,[23] it has been further used to simulate cosserat rods.[24] For accurately simulating the twisting and bending deformation of Cosserat rods, the angular momentum should also be preserved as the linear momentum. Thus, the system is modified to $\{\boldsymbol{q}_n, \boldsymbol{v}_n, \boldsymbol{\omega}_n\}$ where $\boldsymbol{\omega}_n$ denotes the angular velocities for vertices at timestep t_n. The good quality of the visual result is shown in Figure 3. Furthermore, Solar et al. compared PD and PBD in terms of the computation time to convergence and concluded that PD converges faster to a mesh-independent solution than PBD.

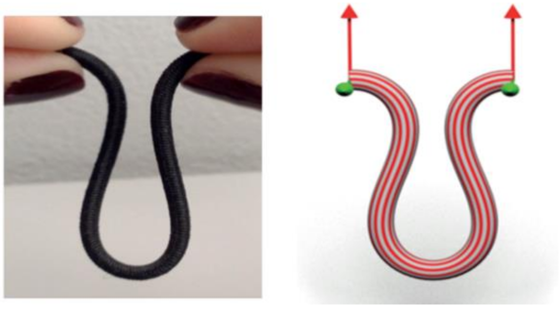

Figure 3: The comparison between a real elastic rod and the simulated rod using PD with the same parameters.[24]

Li et al.[25] proved that PD could also be implemented to simulate deformable characters with the articulated skeleton. The deformable character models consist of rigid-body parts (bones) coupled with deformable parts (flesh). Thus, extra rigid-body constraints and joint constraints are added to the global step of the PD solver. They use affine constraints to group n_f flesh vertices and n_b bone vertices and reorder them. Their proposed method could generate the simulation of articulated characters stably and efficiently with less joint error and similar quality to the state-of-the-art rigid body simulator.

3.3 Improvements in Other Limitations

Regarding the order problem in PBD, only a few improved methods have been proposed. For example, Gu et al.[26] have proposed a sorting method for dealing with the order of the basic constraints mentioned above to improve the realism and efficiency of cloth simulation. In spite of the improvement,[26] there is still a long way to go to get a systematic solution to the order problem.

There are also other limitations with the PBD solver, like the lack of momentum constraints, which makes the material stiffness related to the timestep size and the number of iterations. Besides XPBD, which addresses this limitation, Dahl and Bargteil[27] presented a simple approach to tackle the angular momentum loss problem, and Bender et al.[12] explained the impossibility of efficiently parallelising the Gauss-Seidel version of PBD due to the constraint averaging problem. This limitation makes the number of iterations depend on the number of constraints. Eliminating this dependency is also a significant issue to concern.

3.4 Improvements in Extensions

As mentioned above, PBD was first developed to simulate particle-based dynamic objects. Then it has been extended to other scenarios, including cloth, fluid and rigid body simulation. This subsection will review these extensions and significant improvements made by these extensions.

3.4.1 Cloth Simulation

Besides the essential stretching constraint, i.e. distance constraint between neighbour vertices,[28] bending constraint between adjacent triangles $(\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_2)$ and $(\boldsymbol{x}_1, \boldsymbol{x}_4, \boldsymbol{x}_2)$ is also required to be addressed in cloth simulation. Bender et al.[29] introduced another isometric bending constraint for inextensible surfaces, which is suitable for garment simulation. By applying additional positional constraints to represent constant densities in the PBD solver, Ali et al.[30] proposed a real-time cloth simulation method, which generates different wind effects of cloth, tackling the problem of dynamically preserving cloth overlays and wrinkles. Other collision constraints like self-collisions within cloth, cloth balloons and strain energy constraints are applied in different scenarios to simulate different cloth conditions.

3.4.2 Fluid Simulation

Fluids were first modelled as a system of particles,[31] with special constraints maintaining a minimum distance from each other, called Smoothed particle hydrodynamics (SPH) method. Nevertheless, it could cause problems like failure to reach hydrostatic equilibrium when coming to rest. Macklin and Müller[32] integrated the position-based concepts and SPH, and proposed position based fluids (PBF), which adds density constraint into the solver. Shao et al.[33] introduced the position constraint solved by PBD into SPH to achieve stable interactions between fluid and solid. They proved that the vorticity of the fluid particle system has been significantly smoothed due to PBD. In their work, PBD is applied to improve the visual results of SPH simulation.

Based on PBF, Köster and Kruger[34] presented a method to achieve remarkable improvements in

Figure 4: Rigid body simulations. Top: a collision scene with 2000 rigid bodies colliding with each other and the ground.[36] Bottom: a collision scene with 5000 rigid boxes falling on the ground.[37]

performance in specific scenarios, which uses fine-grained level-of-detail (LOD) information of each particle in the simulation system to alter their positions adaptively. Another modification to PBF has been proposed by Geyer.[35] Their method allows fluid particles to fit the size of the fluid shape, which reduces particle numbers and computational costs. It tackles the interaction problem among fluid particles of different sizes in PBF.

3.4.3 Rigid Body Simulation

The basic PBD method is not only applied to the particle-based system but also extended to simulate rigid bodies as shown in Figure 4 (top image) by introducing joint and contact constraints.[36] It is a truism that a particle has three translational degrees of freedom (DOF), whilst a rigid body has three extra rotational ones to represent its orientations. However, since Newton's second law is only applied to particles, it should be extended to Newton-Euler equations to contain rotational parts, with rigid bodies viewed as groups of infinite numbers of particles.

In the paper by Frâncu and Moldoveanu[37], a new formulation of PBD based on non-linear convex optimization was presented to integrate friction and contact into the projection solver for simulating rigid and elastic bodies among which the rigid body simulation result is shown in the bottom image of Figure 4. Adapting PBD as an alternative discrete algorithm for multi-agent crowd simulation, a set of positional constraints were formulated and integrated into PBD solver to offer a numerical framework for real-time crowd simulation that is robust, stable, and easy to implement.[38] Through using separation planes for adding extra constraints into the PBD solver to allow flexible collision avoidance, multi-agent simulations are achieved.[39]

Macklin et al.[40] proposed a framework to integrate an off-the-shelf linear solver into PBD to add rigid and deformable contact. This method is based on a non-smooth Newton iteration, leading to good performance in robotics simulation scenarios with dexterous manipulation. Furthermore, a differentiable framework was developed to integrate optimal robot design, model-based motion control, and system identification into PBD, leading to higher design efficiency and better estimation accuracy.[41] By utilising position variables only to reformulate articulated body dynamics simulation as an energy minimisation problem without involving calculations of high-order derivatives, position-based articulated dynamics (PBAD), which is another advanced solver, was proposed to simulate articulated body dynamics with full implicit integration, bringing in overall speedup over conventional methods under very large timestep sizes but with weaknesses of failing to avoid numerical dissipation totally and achieve as accurate results as Lie-Group integrators.[42]

In deformable object simulation approaches with position-based concepts, real-time performance is always a significant criterion fulfilled by the majority of the above methods. However, large time steps and numerical approximations of the PBD solver will cause visual artefacts like numerical damping and "explosions". For correcting the artefacts, Dinev et al.[43] presented a post-processing energy-projection method to produce visually plausible and stable motion with real-time performance.

3.5 Conclusion

In this section, we mainly review the latest improvements in the PBD algorithm. They tackle the main limitations of PBD, convergence problem,[15-17,23,26] and unavoidable dependency of stiffness between timestep size and iteration numbers,[18] and handle order problem.[27]

We survey the works on improved PBD algorithms and conclude their contributions, advantages, and drawbacks in Table 1. As shown in the table, there is no method to correct all the limitations, or even to provide a perfect solution to any drawbacks. This is because that position based approaches have some inherent deficiencies.

However, these limitations are recently being addressed as some of the improved algorithms have been extended to independent research fields, like PD[23] and XPBD[18]. These extended XPBD and PD algorithms have been applied in more complex environments, which are surveyed in Table 2.

Table 1: Contributions, pros and cons of improved PBD algorithms

Reference	Methods	Contribution	Pros	Cons
Müller et al.[15]	HPBD	Change the nonlinear Gauss-Seidel Solver of PBD into a nonlinear multigrid-based algorithm.	Accelerate error correction to achieve convergence faster.	1) Poor quality hierarchical meshes cause visual artifacts for very low iteration counts; 2) More difficult for the hierarchical solver to be parallelized; 3) Bending constraints in higher levels are not considered.
English et al.[16]	BDF2	Use a BDF2-based second order accurate multistep constrained scheme for position projection.	1) Accelerate fast projection; 2) Sharply decrease numerical damping.	1) Not suit stretchy materials; 2) Insufficient smoothness; 3) Time splitting causes perturbations.
Kim et al[17]	LRA	Efficiently enforce global inextensibility implemented into PBD.	1) Speed up error propagation to suit inextensible character clothing; 2) Generate more vivid behaviour of cloth stretching.	1) Poor quality for simulating unattached environmental cloth; 2) Not realistic with high resolution.
Miles et al.[18]	XPBD	1) Improve the PBD constraints representing physics; 2) Introduce a multiplier to address time step and iteration count stiffness dependence; 3) Provide a physical solver-based validation scheme.	1) Decrease time step and iteration counting stiffness dependence; 2) Increase the accuracy of PBD but only with trivial modifications.	1) Not suitable for applications requiring high accuracy; 2) Not ignorable convergence cost.
Bouaziz et al.[23]	PD	Introduce energy potentials to PBD by a local/global step to connect it with the physical system.	1) Decrease the dependence problem of stiffness; 2) Achieve faster convergence; 3) High accuracy.	1) Implicit damping problem; 2) Not suit fluid simulation due to the alteration of the global system; 3) Hard materials are not considered.
Gu et al.[26]	Order Sorting	Propose an appropriate constraint adjustment sequence.	Improve convergence, realism, and efficiency.	Different triangulations affects simulation results.
Dahl and Bargteil[27]	Momentum Compensation	Optimise the correction step during the timestep to preserve the global momentum.	1) Require negligible computational cost; 2) Increase accuracy of PBD.	Only apply in extreme cases with angular momentum preservation.

Table 2: Contributions, pros and cons of extended XPBD and PD algorithms

Reference	Based Methods	Contribution	Pros	Cons
Macklin et al.[19]	XPBD	Split timesteps into substeps and iterate once in each substep.	1) Less stretching and higher stiffness; 2) Reduce constraint error and damping over implicit integrator; 3) More stable and robust over explicit integrator.	1) Computational cost slightly increases; 2) Ineffective in reducing velocity error; 3) Residual depends on iteration order; 4) High iteration counts require double precision floating point.
Müller et al.[20]	XPBD	Precisely resolve small spatial and temporal details for simulating rigid bodies.	1) Increase energy conservation and accuracy; 2) Easily manipulate the environment with large mass ratios and frequent rotation change.	1) Unable to damp out high-frequency vibrations; 2) High requirement of computing devices; 3) Unstable in complex scenarios.
Romeo et al.[21,73]	XPBD	Add an anisotropic component to distance condtraints and modify distance constraints in XPBD to allow muscles and fascia to contract.	1) Highly controllable; 2) Efficiently generate more realistic muscle dynamic results.	Only apply in muscle and fascia related cases.
Liu et al.[22]	XPBD	1) Add extra geometrical constraints to simulate differential linear objects; 2) Introduce a differentiable framework for constraint solving; 3) Define the problem	1) Accurate and robust simulation results; 2) Meet the real-time requirement.	Not consider the coupling between the collision handing and rigid-deformable.

		of rope-like objects in the real-to-sim context.		
Soler et al.[24]	PD	1) Introduce angular velocities; 2) Add extra constraints and potentials to PD; 3) Introduce potential weights for the rod.	1) Improve the accuracy of rod simulation; 2) Numerically robust; 3) Fast convergence with few iterations.	1) Not faster than PBD in other scenarios except for the hanging rods simulation; 2) Adding attachment constraints causes legible computational cost.
Li et al.[25]	PD	1) Propose a PD-based monolithic method to simulate articulated soft characters; 2) Integrally formulate the vertices; 3)Enforce the rigid-body and joint constraints exactly.	Improve the performance	Lack of support of joint limits, which causes the separate rotation of bones.

Table 3: Contributions, pros and cons of improvements in specific scenarios

Scenarios	Reference	Contribution	Pros	Cons
Cloth Simulation	Bender et al.[29]	Integrate continuum mechanical formulation into PBD to simulate deformable solids and cloth.	Able to handle complex physical effects like isotropy, anisotropy, elastoplasticity, and lateral contraction.	1) Visual plausibility; 2) Stiffness also depends on iteration number and time step size; 3) Do not converge to the solution as simulation mesh.
	Ali et al.[30]	1) Generate wind effects of cloth; 2) Tackle cloth overlay and wrinkle preservation problem.	1) Improve incompressibility and convergence; 2) Realistic cloth simulation in a real-time environment.	Not able to model solid objects due to lacking inclusion of physical constraints.
Fluid Simulation	Monaghan et al.[31]	Initially propose a particle-based system SPH for fluid simulation.	1) No grid requirements; 2) Set the physical equations as constraints; 3) Handle complex physics in a 3D environment.	1) Low accuracy; 2) Hard to converge; 3) Computationally expensive.
	Macklin and Müller[32]	Add density constraint into the PBD solver to simulate fluid (PBF)	1) Fast computation; 2) Apply PBD in simulating fluids with promising results.	1) Particle stacking problem; 2) Slow convergence in large number particles system; 3) Dependent parameters cannot be adjusted separately.
	Shao et al.[33]	1) Combine the position constraints from PBD to release penetration issues; 2) Add a vorticity constraint to stabilise diffusion.	1) More realistic results; 2) Make the solid boundary handling of SPH more stable; 3) Less time cost.	Insufficient phenomena experiments.
	Köster and Kruger[34]	Use fine-grained LOD to adaptively alter solver iterations.	1) Improve the performance of PBF; 2) Maintain an average density while adaptively altering the particle positions.	1) Insufficient adaption models are tested; 2) More constraints should be considered.
	Geyer[35]	Allow fluid particles to fit the fluid shape for improving PBF.	1) Reduce computational cost; 2) Improve performance.	1) Poor quality in complex particle systems; 2) The alternative criteria for the adaption should be improved.
Rigid Body Simulation	Deul et al.[36]	Introduce joint and contact constraints into PBD to simulate rigid bodies.	1) Controllable; 2) Fulfil scenarios with large-scale rigid bodies; 3) Support coupling between deformable and rigid bodies.	Missing some components in PBD-based rigid body simulation, like motors.
	Francu et al.[37]	1) Provide a contact and friction model to simultaneously simulate rigid and flexible bodies; 2) Integrate friction and contact into the projection solver	1) Stable and realistic simulation results; 2) Faster computation.	1) Some poor visual artifacts; 2) Need further smooth friction cone methods.
	Weiss et al[38]	1) Simulate crowds within the PBD framework; 2) Generate several constraints to improve realism.	1) Exploit efficiency and stability in crowd simulation; 2) Produce more flexible and emergent behaviour.	1) not able to simulate real pedestrians; 2) Lack of parameter tuning.
	Sharma et al[39]	Use separation planes to achieve multi-agent simulation;	1) Avlleviate the collision among agents; 2) High controllability; 3) High genericity to be integrated with other crowd techniques.	More application fields should be considered.
	Macklin et al.[40]	1) Introduce smooth friction; 2) Provide a new preconditioner; 3) Provide a compliance formulation to support hyper-elastic materials; 4) Approximate geometric stiffness.	1) Improve convergence; 2) More robust; 3) First time in introducing non-smooth formulations for interactive applications	1) Do not deal with elastic collisions, or energy-preserving integrators; 2) Work well on particle-based objects but badly in rigid bodies; 3) The performance needs further improvement.
	Liu et al.[41]	1) Propose a differentiable framework for rigid body simulation; 2) Formulate a	1) Lead higher efficiency and accuracy; 2) Could be applied in	Not enough comparison with other techniques.

		simulation workflow for the differentiable framework.	more complex environments and objects.	
	Pan and Manocha[42]	Present a time integrator to simulate articulated bodies.	1) Speedup conventional methods under very large timestep; 2) High stability; 3) Computation friendly for GPU.	1) Cannot alleviate numerical dissipation; 2) Less accuracy;
	Dinev et al.[43]	Propose a post-processing energy-projection method to simulate deformable objects.	Generate realistic, stable, and real-time motion;	1) Not numerically accurate; 2) Have some high-frequency oscillations

For XPBD, although initial research studies focus on decoupling the dependence problem, some researchers have late proved its genericity to be applied in various scenarios, like rigid body[20], muscle[21,73], and rod simulation[22]. Compared with PBD, more realistic and controllable results obtained with XPBD have inspired further investigations. However, the dependence problem still exists.

For PD, it introduces FEM into PBD to consider continuum mechanics and energy potentials, which accelerate convergence and improve accuracy. The extensions to PD are more focused on adding more physics-related elements to increase the realism of the simulation results, like extra constraints to represent more potentials[24,25]. However, integrating more physics will require more computations and cause more problems brought by physical limits, like joint limits[25].

The improvements in specific scenarios, including cloth, fluid, and rigid body simulation are surveyed in Table 3. Cloth simulation is the cradle of position based methods as the pioneer demonstrations are all applied on cloth. Its high-quality performance on cloth[29,30] has been well proven. However, it was later found that PBH[33], integrating with SPH[31], could also have visual plausibility in fluid simulation while inheriting the advantages of PBD[33-35]. Rigid body simulation is now being emphasized since it requires both deformations brought by particle systems and solid attributes. This is always a hard topic to balance, without losing robustness, simplicity, efficiency, and realism.

These improvements in PBD algorithm could enlighten the following-up researchers about various ways to explore.

4. Recent Applications of PBD

The recent applications related to PBD have been extended to some other fields. In this section, we will review the recent applications in the following three application scenarios: deep learning, medical field, and architecture.

4.1 Deep Learning-related Application

As a hot spot in computer graphics, deep learning (DL) has been used in many different areas, such as 3D representation, image transfer, and autonomous vehicles, and significant developments have been achieved. Due to its simplicity, efficiency and robustness, PBD has recently been integrated with DL. As a class of DL methods, graph networks (GN) have been used to estimate the constraint projection in PBD[44] and have been utilised as an accelerating method[45] for simulating rod dynamics. Compared with the original PBD method, the one applying GN could improve runtime performance. Besides improving the performance of PBD, its high-quality simulation data have been used as the input of a data-driven method for efficiently approximating physical forces.[46] Yang et al. embedded neural networks in the projection constraint step to learn and predict the physics rules for governing challenging scenarios[47]. Another neural network called anisotropic constrained-boundary convolutional neural networks (AnisoCBConvNet) was proposed by Kim et al.[48], where PBD was used as a dynamics solver to obtain surfaces data for deep learning.

In addition to PBD approaches, PBF, as a significant fluid simulation method, has also been used with DL methods. Schenck and Fox implemented PBF inside their Smooth Particle Networks (SPNets) to propose a method for computing the interaction between rigid bodies and liquids.[49] Figure 5 shows the interaction between rigid bodies and liquids.

4.2 Medical Application

In addition to the above-reviewed applications of PBD in simulating cloth, deformable objects and fluids, PBD has also been used in surgical simulation. Compared with the previous force-based and impulse-based approaches, PBD could avoid overshooting problems and easily manipulate collision constraints.

Pan et al.[50] introduced an interactive dissection method to simulate hybrid soft tissue models. This method applies energy-preserving and volume-preserving constraints on PBD to improve the visual performance in soft tissue simulation. After that, Berndt et al.[51] presented an association approach, which uses PBD to model all the dynamic objects in surgery and simulate soft tissue cuts (Figure 6), bones and body fluids by associating PBD's ability with different materials. The comparison with the previous FEM-based method proposed by Wu et al.[52] and the approach proposed by Pan et al.[50]

demonstrates that their proposed method is faster and scales better for simulating the scenarios with mixed objects.

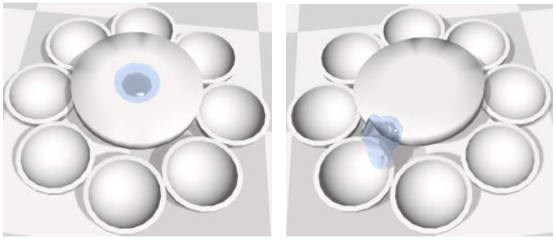

Figure 5: The interaction between rigid bodies and liquids using SPNets integrated with PBF.[49]

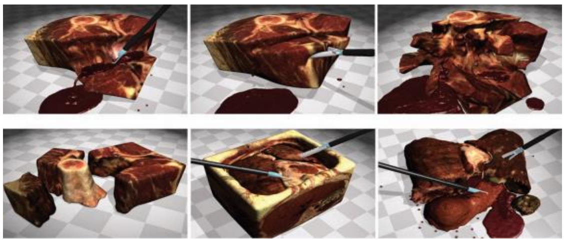

Figure 6: Simulation results from a soft tissue cut with complex materials.[51]

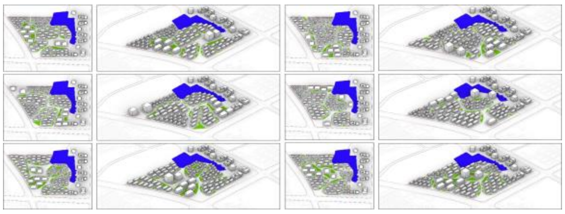

Figure 7: The simulation results with the proposed algorithm of PBD-based large-scale layout.[60]

The simulations of mitral and aortic valves were addressed with a simple material model expressed in PBD as fluids by Walczak et al.[53-55], where PBD is used for an interactive parameterization to model missing information. Han et al.[56] utilised PBD for simulating the tissue deformation in a 2D surgical framework. Xu et al.[57] addressed the dependence problem between deformation effects and iteration variables by proposing a method integrating mass spring dampers (MSD) as constraints into PBD to simulate soft tissue deformation in a laparoscopic cholecystectomy. Moreover, PBD has been used for simulating soft tissues in virtual surgeries.[58,59]

4.3 Architectural Application

PBD has further been used for architectural layout tasks, like city design. Since the architectural criteria of building arrangement could be regarded as layout constraints, Cao and Ji[60] proposed an automatic layout algorithm based on PBD to plan large-scale office parks. They modelled the layout problems as positioning and orienting n buildings $B = \{b_1, b_2, \ldots, b_n\}$. For each building $b_i \in B(i = 1,2, \ldots, n)$, attributes are attached, including a position p_i, an orientation θ_i, an associated shape mesh, a group of agent models and an inverse mass $w_i = \frac{1}{m_i}$ where m_i is the mass of the building. Then in the basic solver of the PBD algorithm, two additional steps are added for updating stiffness k and agent model AM before generating collision constraints. The overall density results obtained with the proposed method are shown in Figure 7.

4.4 Conclusion

So far, researchers have paid much attention to how to apply position based methods in various fields and utilise their irreplaceable capabilities. In this section, we mainly review the latest applications of PBD in different fields: deep learning, medical field, and architectural industry, which have been surveyed in Table 4. Especially in the medical field, PBD has been widely used as a simulation tool to get the deformation results of human tissues, including organs, blood, and bones.[50-59] Though PBD is not accurate enough, its fast computation could help beginners become more familiar with the human structure and train them with basic surgeries.

These applications demonstrate that PBD is robust and could be integrated with other techniques. Thus, how to further apply PBD in different industries remains a research topic for the following-up researchers.

5. Future Research Directions

In Sections 3 and 4, we have summarised the state-of-the-art improvements of PBD and their applications. In spite of these remarkable progress, there are still many open challenges that future work on PBD and their applications needs to address. In this section, we suggested main research directions in the future.

Table 4: Survey of PBD applications in different fields

Fields	Reference	Contribution
Deep Learning	Shao et al.[44]	Use GN to estimate the constraint projection in PBD
	Shao et al.[45]	Use GN to accelerate PBD-based rod dynamic simulation
	Holden et al.[46]	Use the deformation data from PBD as the input of the data-driven method
	Yang et al.[47]	Embed NN in projection constraint to predict positions
	Kim et al.[48]	Use the deformation data from PBD as input
	Schenck and Fox[49]	Implement PBF inside SPNets to compute the interaction between solid and fluid
Medical	Pan et al.[50]	Improve visual results; Add energy and volume-preserving constraints to PBD
	Berndt et al.[51]	Use PBD to model dynamic objects in surgery
	Wu et al.[52]	Compare the simulation results of PBD and FEM in surgery
	Walczak et al.[53, 54, 55]	Use PBD to express mitral and aortic valves simulation
	Han et al.[56]	Utilise PBD for simulating tissue deformation in a 2D surgical framework
	Xu et al.[57]	Solve the dependence problem between deformation and iteration variables; Integrate MSD as constraints into PBD
	Tagliabue et al.[58], Liu et al.[59]	Use PBD to simulate soft tissues in virtual surgeries
Architectural	Cao et al.[60]	Use PBD to plan large-scale layout; Specify the mass of each particle in PBD

5.1 Improvement Tackling Limitations

In Section 3, we have identified the main drawbacks of PBD algorithm and introduced new improvements. However, none of them gave a complete solution to perfectly satisfy all the limitations. Particularly the main intrinsic limitation of PBD - the material stiffness depends on the number of iterations, the time step, and the ordering of the constraints - is still a significant direction to explore. In addition to the improvements like PD[23] and XPBD[18] reviewed in Subsection 3.2, some other methods were also being explored to tackle these problems such as the method presented by Deul et al.[65] who introduced an alternation algorithm of direct solver and non-linear Gauss-Seidel solver called the KKT/GS solver into the PBD framework to avoid convergence problems when realistically simulating complex scenes with rods. It indicates that improving the algorithm of PBD to overcome the limitations is still a challenge of further improvement.

5.2 Integration With Other Computational Methods

Some research studies have integrated PBD with other computational methods such as finite element methods to improve PBD simulation quality, and how to balance accuracy and efficiency of PBD has not been well investigated. In the future, there will be more research studies in this direction.

In existing work, nodal finite element methods and PBD have been integrated to develop PD,[23] and a hybrid deformation model has been proposed to implement PBD and FEM on different meshes,[61] the computations of exponential spring potential energy functions (ESPEFs) were added to the XPBD simulation to enrich the hyperelasticity and generate more dynamic motions,[62] and a control volume method (CV) and continuous collision detection (CCD) were introduced into XPBD to establish a new method for simulating folded membrane structures.[63] From these research studies, we see a significant potential and need for further research in combining PBD with other computational methods, since such combinations could achieve more visual-plausible results.

In the work by Löschner et al.[64], the accuracy and efficiency of selected integration methods for computer graphics applications were evaluated. It raises another future topic - how to achieve the balance between quality and efficiency of PBD. Combining PBD with various integration methods could provide an effective approach to this topic.

5.3 Integration With Software Packages

Due to its advantages of efficiency, stability, and controbility, PBD has been integrated into some software packages. In spite of this, integrating PBD into more software packages is still a future direction.

In the game industry, efficiency will always be a priority. Houdini, as one of the current mainstream software tools for modeling and visual effects, has integrated the PBD algorithm to create deformations. Besides, PBD has also been applied in Unity 3D. Lee et al.[66] solved simulation problems of tetrahedral models in Unity 3D with their proposed Internal shape preserving constraint (ISPC) algorithm to integrate PBD with game techniques. Khan et al.[67] used PBD to simulate the correct physical behaviour of deformable objects in Unity 3D environment. In the future, we expect to see more work in integrating PBD into other software packages including Maya, Unreal Engine, and Blender to help designers efficiently create visually plausible deformations for computer animation and games.

5.4 Integration With Extended Reality

The PBD's advantages of efficiency, stability, and controbility also make it well applicable to extended reality (XR). Some researchers have started the research in this direction.

Extended reality is an umbrella term that covers a spectrum of newer and immersive technologies, including virtual reality (VR), augmented reality (AR), and mixed reality (MR). It is intended to create an environment to make the digital virtual world interact with the physical world. XR has been more often used in medical fields recently since it could bring more sense of immediacy for beginners. As an efficiency method to simulate tissue deformations, PBD has shown its integration with VR and AR in medical applications. Wu et al.[68] built a VR-based surgical system to train surgeons, which simulates the basic deformation behavior of soft tissue with a parallel position based dynamics framework. Huang et al.[69] proposed an AR-based medical education system, where PBD provides a real-time solver and stable time-integration scheme.

Apart from its applications in the medical field, extended reality can be applied in many other fields such as emtertainment and video games industries and marketing. How to intergrate PBD into XR with applications in more fields could attract more research interests in the future.

5.5 Cloud Computing for PBD

While not the main focus of this survey, recent research studies have developed cloud computing for numerical simulation. Zuo et al.[70] have proved that using cloud computing for numerical simulation can reduce the solution time to 1%. Thus, introducing cloud computing into PBD could also be a promising direction to further improve its efficiency.

5.6 Improvements on Volumetric Models

The first step of simulating deformations of 3D surface models with PBD is to convert them into volumetric models such as tetrahedral meshes. A few researchers have initiated the work of improving volumetric models.

The work by Lee et al.[66] proposed and integrated an internal shape preserving constraint (ISPC) generation algorithm into the position based dynamics, which reduces the number of nodes filling the interior of 3D models and computation time of tetrahedral models while achieving similar volume maintenance and physical properties of tetrahedral models. In the work by Angles et al.[71], a bundle of cosserat rods was used as an alternative to tetrahedral meshes for modelling muscles and PBD simulation of volume invariant rods was coupled to a surface mesh to simulate soft-body deformation with high efficiency. Without using tetrahedral meshes, triangle meshes were deformed via a Verlet integration framework of PBD simulation to reduce the computational cost, which provides the simulation of deformable surfaces on the selected area of the 3D mesh with a desired range[72].

The above work has shown the advantage of raising computational efficiency of PBD simulation through improvements of volumetric models. More research studies in this direction could be expected in the future.

5.7 Other directions

Except for the future work discussed above, some other directions could also be considered, including representing smooth surface problematic, achieving parallel and robust collision of simplices, investigating hierarchical representation (multi-scale particles), and improving convergence for parallel solvers. All these directions are worthy of investigation by the researchers in the PBD field.

6. Conclusion

In this paper, we have provided a survey about recent developments and applications in position-based approaches. We first introduced the baseline algorithm of the original position based dynamics for particle systems. Then, we reviewed the improvements of position-based approaches since 2018, including some crucial extensions such as projective dynamics and XPBD. The simulation methods of other materials like rigid bodies, fluids and cloth based on PBD as well as the modification to adapt to different scenarios were also discussed. Next, we presented some recent applications of PBD in medical-related simulation, integration with deep learning, and architectural layout arrangement. At last, we suggested research directions for future work. Hope this review could help beginners develop a quick understanding of position based dynamics and give following-up researchers some hints about future studies.

Acknowledgements

Special thanks to the supervisory team for supervision and review. All authors have read and agreed to the published version of the manuscript.

References

[1] Jan B, Erleben K, Trinkle J. Interactive simulation of rigid body dynamics in computer graphics. Computer Graphics Forum 2014; 33(1):246-270.
[2] Gibson SFF, Mirtich B. A survey of deformable modeling in computer graphics. Technical report, Mitsubishi Electric Research Laboratories, 1997.
[3] Sederberg TW, Parry SR. Free-form deformation of solid geometric models. Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques; 1986. p. 151-160.
[4] Ladislav K, Žára J. Spherical blend skinning: a real-time deformation of articulated models. Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games; 2005. p. 9-16.
[5] Yang X, Somasekharan A, Zhang JJ. Curve skeleton skinning for human and creature characters. Computer Animation and Virtual Worlds 2006; 17(3-4):281-292.
[6] Brett A, Curless B, Popović Z. Articulated body deformation from range scan data. ACM Transactions on Graphics 2002; 21(3):612-619.
[7] Vincent MB. Impulse-based dynamic simulation of rigid body systems. University of California, Berkeley, 1996.
[8] Chadwick JE, Haumann DR, Parent RE. Layered construction for deformable animated characters. ACM SIGGRAPH Computer Graphics 1989; 23(3):243-252.
[9] Capell S, Burkhart M, Curless B, Duchamp T, Popović Z. Physically based rigging for deformable characters. Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 2005. p. 301-310.
[10] Müller M, Heidelberger B, Hennix M, Ratcliff J. Position based dynamics. Journal of Visual Communication and Image Representation 2007; 18(2):109-118.
[11] Jan B, Müller M, Macklin M. A survey on position based dynamics. Proceedings of the European Association

for Computer Graphics: Tutorials; 2017. p. 1-31.
[12] Jan B, Müller M, Otaduy MA, Teschner M, Macklin M. A survey on position-based simulation methods in computer graphics. Computer Graphics Forum 2014; 33(6):228-251.
[13] Terzopoulos D, Fleischer K. Deformable models. The Visual Computer 1988; 4(6):306-331.
[14] Andrew N, Müller M, Keiser R, Boxerman E, Carlson M. Physically based deformable models in computer graphics. Computer Graphics Forum 2006; 25(4):809-836.
[15] Müller M. Hierarchical position based dynamics. Proceedings of Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS); 2008. p. 1-10.
[16] English E, Bridson R. Animating developable surfaces using nonconforming elements. ACM SIGGRAPH 2008 papers; 2008. p. 1-5.
[17] Kim T-Y, Chentanez N, Müller-Fischer M. Long range attachments - a method to simulate inextensible clothing in computer games. Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 2012. p. 305-310.
[18] Miles M, Müller M, Chentanez N. XPBD: position-based simulation of compliant constrained dynamics. Proceedings of the 9th International Conference on Motion in Games; 2016. p. 49-54.
[19] Macklin M, Storey K, Lu M, Terdiman P, Chentanez N, Jeschke S, Müller M. Small steps in physics simulation. Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '19). Association for Computing Machinery, New York, NY, USA, 2019. Article 2, p. 1-7.
[20] Müller M, Macklin M, Chentanez N, Jeschke S, Kim T. Detailed rigid body simulation with extended position based dynamics. Computer Graphics Forum 2020; 39(8):101-112.
[21] Romeo M, Monteagudo C, Sánchez-Quirós D. Muscle simulation with extended position based dynamics. Proceedings of Spanish Computer Graphics Conference, 2018. p. 1-10.
[22] Liu, F, Su, E, Lu, J, Li, M, & Yip, MC. Differentiable robotic manipulation of deformable rope-like objects using compliant position-based dynamics. 2022. arXiv preprint arXiv:2202.09714.
[23] Bouaziz S, Martin S, Liu T, Kavan L, Pauly M. Projective dynamics: Fusing constraint projections for fast simulation. ACM transactions on graphics 2014; 33(4):1-11.
[24] Soler C, Martin T, Sorkine-Hornung O. Cosserat rods with projective dynamics. Computer Graphics Forum 2018; 37(8):137-147.
[25] Li J, Liu T, Kavan L. Fast simulation of deformable characters with articulated skeletons in projective dynamics. Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2019. p. 1-10.
[26] Gu Y, Yang K, Lu C. Constraint solving order in position based dynamics. Proceedings of the 2017 2nd International Conference on Electrical, Automation and Mechanical Engineering, 2017. p. 191-194.
[27] Dahl A, Bargteil A. Global momentum preservation for position-based dynamics. Proceedings of Motion, Interaction and Games, 2019. p. 1-5.
[28] Jakobsen T. Advanced character physics. Proceedings of Game Developers Conference, 2001. p. 1-16.
[29] Bender J, Koschier D, Charrier P, Weber D. Position-based simulation of continuous materials. Computers & Graphics 2014; 44:1-10.
[30] Ali IR, Mohammed MA, Al-Sharify TA, Kolivand H. Real-time cloth simulation on virtual human character using enhanced position based dynamic framework technique. Baghdad Science Journal 2020; 17(4):1294-1303.
[31] Monaghan JJ. Smoothed particle hydrodynamics. Annual Review of Astronomy and Astrophysics 1992; 30(1):543-574.
[32] Macklin M, Müller M. Position based fluids. ACM Transactions on Graphics 2013; 32(4):1-12.
[33] Shao, X, Liao, E, Zhang, F. Improving SPH fluid simulation using position based

dynamics. IEEE Access 2017; 5:13901-13908.
[34] Köster M, Krüger A. Adaptive position-based fluids: improving performance of fluid simulations for real-time applications. International Journal of Computer Graphics & Animation 2016; 6(3):1-16.
[35] Geyer, L. Adaptive sampling in position based fluids. Doctoral dissertation, Wien. 2022.
[36] Deul C, Charrier P, Bender J. Position-based rigid-body dynamics. Computer Animation and Virtual Worlds 2016; 27(2):103-112.
[37] Frâncu M, Moldoveanu F. Unified simulation of rigid and flexible bodies using position based dynamics. Proceedings of Workshop on Virtual Reality Interaction and Physical Simulation, 2017. p. 49-58.
[38] Weiss, T, Litteneker, A, Jiang, C, Terzopoulos, D. Position-based multi-agent dynamics for real-time crowd simulation. Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2017. p. 1-2.
[39] Sharma R, Weiss T, Kallmann M. Plane-based local behaviors for multi-agent 3D simulations with position-based dynamics. Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality, 2020. p. 214-217.
[40] Macklin M, Erleben K, Müller M, Chentanez N, Jeschke S, Makoviychuk V. Non-smooth newton methods for deformable multi-body dynamics. ACM Transactions on Graphics 2019; 38(5):1-20.
[41] Liu F, Li M, Lu J, Su E, Yip MC. Parameter identification and motion control for articulated rigid body robots using differentiable position-based dynamics. arXiv:2201.05753, 2022. p. 1-15.
[42] Pan Z, Manocha D. Time integrating articulated body dynamics using position-based collocation methods. Proceedings of International Workshop on the Algorithmic Foundations of Robotics, 2018. p. 673-688.
[43] Dinev D, Liu T, Li J, Thomaszewski B, Kavan L. FEPR: Fast energy projection for real-time simulation of deformable objects. ACM Transactions on Graphics 2018; 37(4):1-12.
[44] Shao H, Kugelstadt T, Hädrich T, Palubicki W, Bender J, Pirk S, Michels DL. Accurately solving rod dynamics with graph learning. Advances in Neural Information Processing Systems 2021; 34:4829-4842.
[45] Shao, H. Fluids, threads and fibers: towards high performance physics-based modeling and simulation. Doctoral dissertation. 2022.
[46] Holden D, Duong BC, Datta S, Nowrouzezahrai D. Subspace neural physics: fast data-driven interactive simulation. Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2019. p. 1-12.
[47] Yang S, He X, Zhu B. Learning physical constraints with neural projections. Advances in Neural Information Processing Systems 2020; 33:5178-5189.
[48] Kim, JH, Kim, SJ, Lee, J. Geometry image super-resolution with AnisoCBConvNet architecture for efficient cloth modeling. PloS one 2022; 17(8):e0272433.
[49] Schenck C, Fox D. Spnets: differentiable fluid dynamics for deep neural networks. Proceedings of Conference on Robot Learning, 2018. p. 317-335.
[50] Pan J, Bai J, Zhao X, Hao A, Qin H. Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics. Computer Animation and Virtual Worlds 2015; 26(3-4):321-335.
[51] Berndt I, Torchelsen R, Maciel A. Efficient surgical cutting with position-based dynamics. IEEE computer graphics and applications 2017; 37(3):24-31.
[52] Wu J, Westermann R, Dick C. A survey of physically based simulation of cuts in deformable bodies. Computer Graphics Forum 2015; 34(6):161-187.
[53] Walczak L, Georgii J, Tautz L, Neugebauer M, Wamala I, Sündermann S, Falk V, Hennemuth A. Using position-based dynamics for simulating the mitral valve in a decision support system.

Proceedings of Eurographics Workshop on Visual Computing for Biology and Medicine, 2019. p. 165-175.
[54] Walczak L, Goubergrits L, Hüllebrand M, Georgii J, Falk V, Hennemuth A. Using position-based dynamics to simulate deformation in aortic valve replacement procedure. Current Directions in Biomedical Engineering 2020; 6(1):20200042.
[55] Walczak L, Georgii J, Tautz L, Neugebauer M, Wamala I, Sündermann S, Falk V, Hennemuth A. Using position-based dynamics for simulating mitral valve closure and repair procedures. Computer Graphics Forum 2021; 41(1):270-287.
[56] Han Y, Liu F, Yip MC. A 2D surgical simulation framework for tool-tissue interaction. arXiv:2010.13936, 2020. p. 1-2.
[57] Xu L, Lu Y, Liu Q. Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time. Royal Society Open Science 2018; 5(2):171587.
[58] Tagliabue E, Pore A, Dall'Alba D, Magnabosco E, Piccinelli M, Fiorini P. Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery. Proceedings of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020. p. 3261-3266.
[59] Liu Y, Guan C, Li J, Yu X, Zhang S. The PBD model based simulation for soft tissue deformation in virtual surgery. Journal of Physics: Conference Series 2020; 1621(1):012043.
[60] Cao S, Ji G. Automatically generating layouts of large-scale office park using position-based dynamics. Proceedings of the 26th CAADRIA Conference, 2021. volume 1, p. 21-30.
[61] Tan, S, Zhou, H, Zheng, J. A hybrid model method for accurate surface deformation and incision based on fem and PBD. Scientific Programming 2021; Article ID 8343312.
[62] Cetinaslan, O. ESPEFs: exponential spring potential energy functions for simulating deformable objects. Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games; 2021. p. 1-11.
[63] Peng, H, Li, N, Li, F, Zhang, L, Dong, K. Development simulation of an inflatable membrane antenna based on extended position-based dynamics. Acta Mechanica Sinica 2022; 38(7):1-13.
[64] Löschner, F, Longva, A, Jeske, S, Kugelstadt, T, Bender, J. Higher-order time integration for deformable solids. Computer Graphics Forum 2020; 39(8):157-169.
[65] Deul, C, Kugelstadt, T, Weiler, M, Bender, J. Direct position-based solver for stiff rods. Computer Graphics Forum 2018; 37(6):313-324.
[66] Lee D, Kim T, Choi Y, Hong M. Volumetric object modeling using internal shape preserving constraint in Unity 3D. Intelligent Automation & Soft Computing 2022; 32(3):1541-1556.
[67] Khan, L, Choi, Y.-J, Hong, M. Cutting simulation in unity 3D using position based dynamics with various refinement levels. Electronics 2022; 11(14):2139.
[68] Wu, H, Ye, F, Gao, Y, Cong, Y, Hao, A. Real-time laparoscopic cholecystectomy simulation using a particle-based physical system. Complex System Modeling and Simulation 2022; 2(2):186-196.
[69] Huang, T, Zhang, N, Zhang, X, Liao, H. Anatomy education method using autostereoscopic 3D image overlay and mid - air augmented reality interaction. SID Symposium Digest of Technical Papers 2022; 53(S1):91-94.
[70] Zuo, Y, Dai, C, Tan, C, Zhan, T, Chen, W. Virtual cloud computing-based and 3D multi-physics simulation for local oxygen starvation in PEM fuel cell. International Journal of Hydrogen Energy 2022; 47(87): 37040-37051.
[71] Angles, B, Rebain, D, Macklin, M, Wyvill, B, Barthe, L, Lewis, JP, von der Pahlen, J, Izadi, S, Valentin, J, Bouaziz, S, Tagliasacchi, A. Viper: volume invariant position-based elastic rods. Proceedings of the ACM on Computer Graphics and Interactive Techniques; 2019. vol. 2, no. 2, p. 1-26.
[72] Cetinaslan, O. Localized constraint based deformation framework for triangle

meshes. Entertainment Computing 2018; 26:78-87.
[73] Romeo, M, Monteagudo, C, Sánchez-Quirós, D. Muscle and Fascia Simulation with Extended PositionBased Dynamics. Computer Graphics Forum 2020; 39(1):134-146.