Subspace-Preconditioned GPU Projective Dynamics with Contact for Cloth Simulation

Xuan Li
UCLA
xuan.shayne.li@gmail.com

Yu Fang
UCLA
University of Pennsylvania
squarefk@gmail.com

Lei Lan
University of Utah
lei.lan@utah.edu

Huamin Wang
Style 3D Research
wanghmin@gmail.com

Yin Yang
University of Utah
Style 3D Research
yin.yang@utah.edu

Minchen Li
UCLA
Carnegie Mellon University
minchernl@gmail.com

Chenfanfu Jiang
UCLA
Style 3D Research
chenfanfu.jiang@gmail.com

Figure 1: Kick (High-Res). Our method can efficiently simulate a high-resolution version (more than 120K nodes) of the Kick animation in CIPC [Li et al. 2021] at 23s per frame, which is $6.5\times$ **faster than a heavily optimized and GPU accelerated CIPC solver. All collisions are robustly handled with intricate wrinkles captured on the cloth, highlighting the efficacy of our approach in handling fine-detailed garment simulations in complex animation scenarios.**

ABSTRACT

We propose an efficient cloth simulation method that combines the merits of two drastically different numerical procedures, namely the subspace integration and parallelizable iterative relaxation. We show those two methods can be organically coupled within the framework of projective dynamics (PD), where both low- and high-frequency cloth motions are effectively and efficiently computed. Our method works seamlessly with the state-of-the-art contact handling algorithm, the incremental potential contact (IPC), to offer the non-penetration guarantee of the resulting animation. Our core ingredient centers around the utilization of subspace for the expedited convergence of Jacobi-PD. This involves solving the reduced global system and smartly employing its precomputed factorization. In addition, we incorporate a time-splitting strategy to handle the frictional self-contacts.

Specifically, during the PD solve, we employ a quadratic proxy to approximate the contact barrier. The prefactorized subspace system matrix is exploited in a reduced-space LBFGS. The LBFGS method starts with the reduced system matrix of the rest shape as the initial Hessian approximation, incorporating contact information into the reduced system progressively, while the full-space Jacobi iteration captures high-frequency details. Furthermore, we address penetration issues through a penetration correction step. It minimizes an incremental potential without elasticity using Newton-PCG. Our method can be efficiently executed on modern GPUs. Experiments show significant performance improvements over existing GPU solvers for high-resolution cloth simulation.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0315-7/23/12…$15.00
https://doi.org/10.1145/3610548.3618157

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**.

KEYWORDS

subspace, projective dynamics, domain decomposition, quasi-Newton methods

ACM Reference Format:
Xuan Li, Yu Fang, Lei Lan, Huamin Wang, Yin Yang, Minchen Li, and Chenfanfu Jiang. 2023. Subspace-Preconditioned GPU Projective Dynamics with Contact for Cloth Simulation. In *SIGGRAPH Asia 2023 Conference Papers (SA*

Conference Papers '23), December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3610548.3618157

1 INTRODUCTION

In cloth simulation, a fine and high-resolution discretization is often needed for rich and vivid effects like detailed wrinkles, folds, and creases, which coarser models cannot produce. The complexity, however, increases disproportionally w.r.t to the degrees of freedom (DOFs) of the system, making high-resolution simulation prohibitive for time-critical applications, whereas animating virtual garments at an interactive rate is always desired.

It is well understood that the primary obstacle for high-resolution cloth animation lies in the computational cost associated with the system solve for numerical integration at each time step. The cloth dynamics is nonlinear, and commonly used solvers rely on incremental linearization of the equation of motion e.g., see [Baraff and Witkin 1998]. Collisions and self-collisions, which are ubiquitous in cloth simulation, impose extra difficulties. Traditional methods often leverage soft repulsion to handle collisions [Tang et al. 2018; Wu et al. 2020]. These approaches require careful parameter tuning to prevent undesirable artifacts like cloth interpenetration. The state-of-the-art solution to contact modeling is incremental potential contact (IPC) [Li et al. 2020], which utilizes log barriers and continuous collision detection (CCD) to strictly maintain objects' separation. The joint optimization with cloth elasticity is numerically challenging as the collision component is considerably stiffer and of higher frequency. The increased resolution also vastly complicates the spectrum of the system. While a wide range of numerical algorithms are available, they are proven only effective in limited or specific situations. For instance, gradient-based strategies [Wang and Yang 2016] are quite parallelizable and efficient, but the performance declines quickly for stiffer instances. On the other hand, direct solvers are robust against numerical stiffness when paired with line search [Wolfe 1969, 1971], but they are highly sequential and expensive for large-scale problems.

The pros and cons of existing algorithms endorse their strong complementarity, which forms our key rationale, illustrated in Fig. 2. Specifically, we seek algorithmic synergy between direct and iterative numerical procedures for efficient high-resolution cloth animations. We argue that parallel local relaxation schemes e.g., Jacobi or Guass-Seidel are most effective if low-frequency residuals can be pre-eliminated. The latter happens to be the task in which subspace methods excel. The coordination of those two simulation modalities collectively delivers combined efficiency and convergence that existing methods can hardly match. Model reduction suppresses the system into a low-rank subspace, wherein the low-frequency residual can be solved efficiently. At the other end of the spectrum, the remaining high-frequency components become localized, making them an ideal target for parallel GPU solvers.

In this paper, we propose a novel subspace-preconditioned projective dynamics (PD) framework [Bouaziz et al. 2014] for cloth simulation. At each global step of PD, we solve for the displacement field within the subspace to capture low-frequency motion modes, with a pre-factorized subspace global matrix. The high-frequency details are dealt with using parallel Chebyshev-Jacobi relaxation [Wang 2015; Wathen 2008] on the original full-order system. We adopt a time-splitting scheme along with a quadratic contact proxy [Xie et al. 2023] to handle complex contacts. A novel modified Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is incorporated to progressively integrate the quadratic contact proxy into the pre-factorized subspace global matrix, which circumvents the need for re-factorization. Our comparative evaluation demonstrates remarkable performance gain, enabling another 10× acceleration over the state-of-the-art simulation techniques – like IPC, the resulting animation is guaranteed to be free of any interpenetrations.

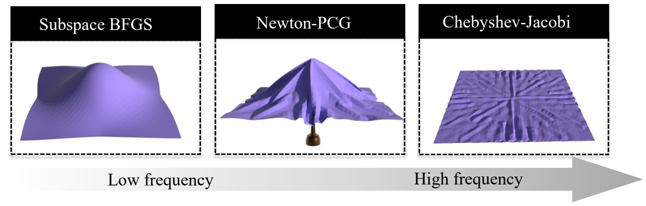


Figure 2: Subspace simulation, Newton method and Chebyshev-Jacobi excel at reducing residuals in different frequency ranges. Our method combines the advantages of Subspace BFGS and Chebyshev-Jacobi to achieve similar outcomes as Newton-PCG, but with improved performance.

2 RELATED WORK

2.1 Cloth Simulation

Cloth and thin shell simulation, ubiquitous in daily life, continue to be central in the realm of computer graphics and animation [Choi and Ko 2005a; Gingold et al. 2004; Grinspun et al. 2003]. Modern cloth animation workflows largely incorporate an implicit time integration scheme, a practice pioneered by Baraff and Witkin [1998]. Given the reduced stretchability of many fabrics, strain limiting is extensively used to prevent over-constraint [Goldenthal et al. 2007; Provot et al. 1995; Thomaszewski et al. 2009; Wang et al. 2010].

Considerable research has focused on refining material modeling to accurately emulate cloth's mechanical behaviors. For example, Volino et al. [2009] introduced a nonlinear and anisotropic tensile model based on continuum mechanics. Cloth bending is closely tied to the parameterization of the dihedral angle [Volino et al. 1995], with Breen et al. [1994] leveraging linear beam theory to link bending moment and curvature. Bridson et al. [2005] developed the bending mode of a hinge-based element orthogonal to in-plane deformations. In the context of inextensible fabrics, discrete mean curvature approximates bending, yielding a quadratic energy with a constant Hessian [Bergou et al. 2006; Wardetzky et al. 2007]. The bending model introduced by Choi and Ko [2005b] effectively encapsulates compression and buckling. Furthermore, Kim [2020] unveiled the interconnections between the model proposed by Baraff and Witkin [1998] and classical Finite Element Method (FEM). Greater realism in cloth models can be achieved by incorporating captured real-world data [Feng et al. 2022; Miguel et al. 2012, 2013; Wang et al. 2011]. Simulating garments at the yarn level, although computationally intensive, is feasible [Cirio et al.

2014; Kaldor et al. 2008], by condensing dynamics of individual yarn strands. Transitioning this approach onto triangle meshes results in much more tractable computations [Sperl et al. 2020].

2.2 Collision Handling

Developing accurate contact models is crucial in mechanics, robotics, and computer graphics [Andrews et al. 2022; Johnson and Johnson 1987]. Traditional methods often handle contacts as constraint-based linear complementarity problems (LCP) [Baraff 1994; Kaufman et al. 2008], resolved using projected Gauss-Seidel (PGS) method. An alternate approach uses quadratic programming (QP) [Macklin et al. 2020; Redon et al. 2002], compatible with more flexible solving techniques like projected gradient descent [Mazhar et al. 2015], mass splitting [Tonge et al. 2012], and the augmented Lagrangian method [Takahashi and Batty 2021]. Penalty methods [Bridson et al. 2002; Tang et al. 2012; Xu et al. 2014] are also employed to handle complex self-contacts. In the context of PD, Ly et al. [2020] introduced iterative refinement of contact forces during local steps.

While traditional methods mainly model contact through approximated constraints utilizing signed distances or volumes, recent approaches like the incremental potential contact (IPC) model [Li et al. 2020] use precisely calculated unsigned distances for better robustness and accuracy. IPC approximates contact as a conservative force with a barrier potential, providing a controllable efficiency-accuracy tradeoff and ensuring penetration-free, convergent results for general contacts of codimensional solids [Li et al. 2021], rigid bodies [Ferguson et al. 2021; Lan et al. 2022a], hybrid multibody systems [Chen et al. 2022], and FEM-SPH coupled domains [Xie et al. 2023], etc. Despite its effectiveness, IPC's computational burden stems from the barrier function and the continuous collision detection in each nonlinear solver iteration. Recent endeavors have also concentrated on accelerating IPC through reduced-order models [Lan et al. 2021], projective dynamics [Lan et al. 2022b], block coordinate descent [Lan et al. 2023], and time splitting [Wang et al. 2022; Xie et al. 2023], the majority of which also employ GPU acceleration.

2.3 Subspace Simulation

Reduced-Order Modeling (ROM) offers a way to speedup the simulation of deformable bodies by using linear subspaces [Barbič and James 2005; Sifakis and Barbic 2012]. These subspaces, usually built using modal analysis [Choi and Ko 2005c; Hauser et al. 2003; Pentland and Williams 1989] and its first-order derivatives [Barbič and James 2005; Yang et al. 2015], simplify the model by removing less critical degrees of freedom (DOFs). This approach finds application in solids [An et al. 2008; Barbič and Zhao 2011; Yang et al. 2015], fluids [Kim and Delaney 2013; Treuille et al. 2006], and computational design problems [Xu et al. 2015]. Alternative approaches include geometric shape coarsening, akin to skin rigging to prescribe the dynamics of a fine model. For instance, Capell et al. [2002] deformed an elastic body using an embedded skeleton, Gilles et al. [2011] employed 6-DOF rigid frames, Brandt et al. [2018]; Faure et al. [2011] employed scattered handles, and Martin et al. [2010] used sparsely-distributed integrators for rods, shells, and solids.

Recent work has started to investigate nonlinear low-dimensional manifolds for ROM, with neural networks being used to construct these spaces [Lee and Carlberg 2020]. This approach can require smaller latent space dimensions compared to linear methods [Fulton et al. 2019; Shen et al. 2021]. There has also been significant progress in data-driven latent space dynamics [Lusch et al. 2018], with neural networks being used to learn the evolution of the entire latent space [Wiewel et al. 2019]. To construct a subspace with sparse basis for general cloth dynamics, we stick with linear subspaces and use 2D B-spline functions as the building block.

3 BACKGROUND

In this section, we provide a brief overview of the Projective Dynamics (PD) and Incremental Potential Contact (IPC) techniques, with a specific focus on cloth simulation, to ensure self-containment.

3.1 Projective Dynamics for Cloth Simulation

In the absence of collision, the PD solver employs the following optimization time integration for time stepping:

$$\min_{\boldsymbol{x}} \frac{1}{2h^2}\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|_M^2+\frac{E_{\text{mem}}}{2}\sum_t \|\boldsymbol{F}_t-\boldsymbol{R}(\boldsymbol{F}_t)\|^2+\frac{E_{\text{bend}}}{2}\sum_e \|\boldsymbol{x}\|_{\boldsymbol{Q}_e}^2. \quad (1)$$

Here, h is the time step size, $\tilde{\boldsymbol{x}} = \boldsymbol{x}^* + \dot{\boldsymbol{x}}^* h + gh^2$ is the predictive position with $\boldsymbol{x}^*$ being the current position, $\boldsymbol{F}_t$ denotes the deformation gradient of triangle t, $\boldsymbol{R}(\boldsymbol{F})$ represents the closest rotation matrix to $\boldsymbol{F}$, E_{mem} and E_{bend} correspond to the membrane stiffness and bending stiffness, respectively, and $\boldsymbol{Q}_e$ is the local quadratic bending stiffness matrix for inner edge e, as outlined by Bergou et al. [2006].

Rather than directly optimizing the energy equation (1), PD decouples it by introducing auxiliary rotations $\boldsymbol{R}_e^k$ for each triangle, enabling optimization through a global-local alternating approach:

$$\begin{aligned}&\min_{\boldsymbol{x}^k} \frac{1}{2h^2}\|\boldsymbol{x}^k-\tilde{\boldsymbol{x}}\|_M^2+\frac{E_{\text{mem}}}{2}\sum_t \|\boldsymbol{F}_t^k-\boldsymbol{R}_t^k\|^2+\frac{E_{\text{bend}}}{2}\sum_e \|\boldsymbol{x}^k\|_{\boldsymbol{Q}_e}^2,\\ &\boldsymbol{R}_t^{k+1} = \boldsymbol{R}(\boldsymbol{F}_t^k).\end{aligned} \quad (2)$$

The global step involves solving a linear system with a fixed system matrix $\boldsymbol{H} = \frac{1}{h^2}\boldsymbol{M} + \boldsymbol{K}_{\text{mem}} + \boldsymbol{K}_{\text{bend}}$, where $\boldsymbol{K}_{\text{mem}}$, $\boldsymbol{K}_{\text{bend}}$ are membrane energy Hessian and bending energy Hessian at the rest shape. The local step can be executed efficiently in parallel. This alternating procedure continues until convergence is achieved. A precomputed Cholesky factorization can be applied to solve the global step. However, as the resolution increases, the computational time grows significantly. Moreover, multiple updates are required for the rotation matrices to ensure accuracy. To mitigate this, it is common to solve the global step inexactly using a few or even just one Jacobi iteration, while updating the rotation matrices as frequently as possible. To accelerate convergence, Chebyshev acceleration techniques can be applied, as suggested by Wang [2015]. However, a substantial number of iterations are typically still required for high-resolution scenes.

3.2 Incremental Potential Contact

IPC [Li et al. 2020, 2021] is an approach that handles contact constraints using smooth log barriers on unsigned distances to ensure separations between objects. It provides a robust method for processing collisions, where the log barrier potential is included as an

Algorithm 1 Timestepping of subspace-preconditioned PD

if it is the first time step **then**
  Construct subspace basis sparse matrix $\mathcal{P}$. ▹ Sec. 4.2
  Factorize the reduced-order global matrix $\mathcal{P}^\top \boldsymbol{H}\mathcal{P}$. ▹ Sec. 4.3
end if
Update predictive position $\tilde{\boldsymbol{x}}$.
Run a reduced-order global step w.o. contact for an initial guess.
Construct quadratic barrier proxy at current state $\boldsymbol{x}^*$.
Initialize subspace BFGS history.
while not converged **do** ▹ Sec. 4.4.1
  Run 2 iterations of subspace BFGS and update the history.
  Run 5 fullspace Jacobi iterations.
  Run PD local projections in parallel.
end while
Run penetration correction step. ▹ Sec. 4.4.2

additional energy term in the optimization-based time integration. By combining continuous contact detection (CCD), IPC can guarantee that there are no penetrations as long as the initial placement of objects is non-overlapping.

In the context of collisions between discrete meshes, collisions are classified into point-triangle and edge-edge collisions. The contact potential is defined as follows:

$$B(x) = \sum_{P,T} b(\mathrm{dist}(P,T)) + \sum_{E_1,E_2} b(\mathrm{dist}(E_1,E_2)), \tag{3}$$

where b is a smooth log barrier function of unsigned distance:

$$b(d) = \begin{cases} -(d-\hat{d})^2 \log(d/\hat{d}), & 0 < d < \hat{d}, \\ 0, & d \geq \hat{d}. \end{cases} \tag{4}$$

Here, (P,T) represents an arbitrary point-triangle pair, and (E_1, E_2) represents an arbitrary edge-edge pair. The candidate pairs can be efficiently identified and filtered using bounding boxes. The parameter $\hat{d}$ controls the size of the contact zone. Inside this zone, the energy tends to infinity as the contact pair gets closer to each other, indicating the presence of contact forces. Outside the zone, there are no contact forces, and objects are considered separated. During the search for a energy-decreasing positional increment, CCD is employed to find an upper bound on the step size, such that when the step size is smaller, penetrations can always be avoided.

To incorporate frictional forces into the optimization, the IPC method integrates a locally smoothed semi-implicit Coulomb friction into a potential energy. For each contact point $\boldsymbol{x}_k$ with sliding basis $\boldsymbol{T}_k$ and normal contact force λ_k, the local friction is defined as

$$f_k(\boldsymbol{x}_k) = -\mu\lambda_k \boldsymbol{T}_k f_1(\|\boldsymbol{u}_k\|)\frac{\boldsymbol{u}_k}{\|\boldsymbol{u}_k\|}, \tag{5}$$

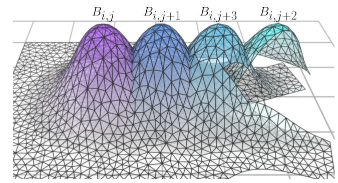
where μ is the frictional coefficient, $\boldsymbol{u}_k$ is the relative tangential velocity, f_1 is a function smoothly increase from 0 to 1 in the region $[0, \epsilon_v h]$ with ϵ_v controling the region of static friction.

4 METHOD

4.1 Algorithm Overview

Our subspace preconditioned PD pipeline is summarized in Algorithm 1. We elaborate further details in the following subsections.

4.2 Construction of Subspace



Clothing items are typically composed of several flat fabric pieces, connected by stitches. Taking this into account, it is sufficient to utilize basis functions defined in $\mathbb{R}^2$.

Suppose the cloth domain Ω is divided into multiple patches interconnected by stitches: $\Omega = \cup_{i=1}^{k}\Omega_k$. For each cloth patch Ω_k, we employ the As-Rigid-As-Possible (ARAP) parameterization technique [Liu et al. 2008] to obtain a bijective parameterization $\bar{\Omega}_k$. In certain cases, additional patch decompositions might be required to ensure bijectivity. Next, we embed each $\bar{\Omega}_k$ into a regular 2D Cartesian grid and employ Material Point Method (MPM) quadratic spline shape functions on the grid points [Jiang et al. 2016] as the basis for one spatial dimension of $\mathbb{R}^3$. Specifically, each basis is a product of two one-dimensional quadratic B-splines and is discretized on mesh nodes (see the inset figure):

$$B_{ij}(X) = N(u/\Delta x - i)N(v/\Delta x - j). \tag{6}$$

Here, (i,j) denotes a grid index, u and v represent the parameterization of X, Δx corresponds to the spline's kernel size and the spacing of the 2D grid, and $N(x)$ is defined as:

$$N(x) = \begin{cases} \frac{3}{4} - x^2, & |x| < \frac{1}{2}, \\ \frac{1}{2}(\frac{3}{2} - |x|)^2, & \frac{1}{2} \leq |x| < \frac{3}{2}, \\ 0, & \frac{3}{2} \leq |x|. \end{cases} \tag{7}$$

We decouple the three dimensions of the ambient space of Ω, meaning that the complete sparse basis matrix $\mathcal{P} = \mathcal{B} \otimes \mathcal{I}_3 \in \mathbb{R}^{3M\times 3N}$ is represented by the Kronecker product between the spline basis matrix $\mathcal{B} \in \mathbb{R}^{N\times M}$ for scalar functions and the 3D identity matrix, where M is the number of bases and N is the number of vertices.

Given a reference state of mesh position $\boldsymbol{x}_0$, we express the states in the subspace as $\{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{x}_0 + \mathcal{P}y, y \in \mathbb{R}^{3M}\}$. The displacement w.r.t. the reference state is constrained within the linear space expanded by the bases in $\mathcal{P}$. This basis satisfies the partition of unity property and is C^1-continuity w.r.t. the parameterization space.

Bending on Stitch. If decompositions are necessary to achieve non-overlapping parameterization, such as for a tube-shaped cloth, we propose an approach to incorporate bending energy on the artificially generated stitches. As depicted in Figure 3, assume an artificial stitch passing through the shared edge of two triangles, namely (v_1, v_2, v_3) and (v_4, v_5, v_6). We introduce two bending energies, each with half of the original bending stiffness, on the 4-stencils (v_1, v_2, v_3, v_4) and (v_4, v_5, v_6, v_1). By doing so, the original energy is now split into two parts. As the stitch penalty pulls the vertices v_2 and v_5, as well as v_3 and v_6, closer together, the combined energy of these two parts will recover the energy prior to the decomposition.

4.3 Subspace Projective Dynamics

In each global step, as described in Equation 2, we need to solve a linear system $\boldsymbol{H}u = \boldsymbol{b}$, where u is the displacement increment w.r.t. the previous global step $\boldsymbol{x}^k$. To optimize the quadratic energy

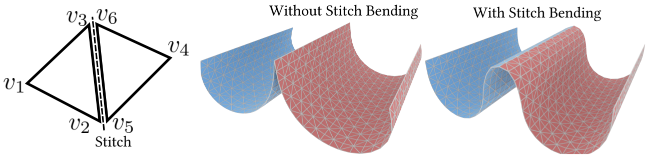


Figure 3: Stitch bending energies can recover bendings before domain decompositions.

within the subspace, we can restrict the displacement to the form $\boldsymbol{u} = \mathcal{P}y$. This leads to solving a reduced-order global system

$$\mathcal{P}^\top \boldsymbol{H}\mathcal{P}y = \mathcal{P}^\top \boldsymbol{b}. \tag{8}$$

Here, the reduced-order global system matrix $\hat{\boldsymbol{H}} = \mathcal{P}^\top \boldsymbol{H}\mathcal{P}$ has a dimension that corresponds to the number of bases in $\mathcal{P}$. This dimension can be much smaller than the total number of degrees of freedom. The advantage of this smaller matrix is that it can be prefactorized using the Cholesky decomposition, enabling efficient reuse for backsolving of Equation 8.

By exclusively solving the global step increment within the subspace, we effectively address the low-frequency modes that primarily govern the overall motion of the cloth. However, this approach tends to overlook the high-frequency details that showcase the benefits of high-resolution meshes. To reintroduce the high-frequency modes, we employ several Jacobi iterations on the original global system $\boldsymbol{Hu} = \boldsymbol{b}$, starting from the displacement obtained through the subspace solution. This two-level scheme bears a resemblance to a two-level multigrid method, wherein $\mathcal{P}$ and $\mathcal{P}^\top$ can be perceived as the restriction and prolongation matrices, respectively. Nevertheless, due to the necessity of a local projection step to update the membrane rotation reference, the right-hand side of the global linear system undergoes changes.

4.4 Subspace-Preconditioned Projective Dynamics with Contact

We follow the same strategy of Xie et al. [2023] and incorporate contact in a specifically designed time-splitting manner. At the beginning of each time step, we introduce a quadratic proxy of the contact potential into the PD solver, allowing for penetration. Subsequently, an additional correction step is incorporated to fix all penetrations while minimizing changes as much as possible.

4.4.1 PD with Quadratic Contact Proxy. The quadratic proxy is defined as the second-order expansion of the barrier potential at the initial state $\boldsymbol{x}^*$ of the current time step:

$$\hat{B}(\boldsymbol{x};\boldsymbol{x}^*) = B(\boldsymbol{x}) + \nabla B(\boldsymbol{x}^*)^\top(\boldsymbol{x}-\boldsymbol{x}^*) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^*\|^2_{\nabla^2 B(\boldsymbol{x}^*)}. \tag{9}$$

By combining this quadratic proxy, the global system of the PD solver is still linear:

$$(\boldsymbol{H} + \nabla^2 B(\boldsymbol{x}^*))\boldsymbol{u} = \boldsymbol{b} - \nabla B(\boldsymbol{x}^*) - \nabla^2 B(\boldsymbol{x}^*)(\boldsymbol{u} + \boldsymbol{x}^k - \boldsymbol{x}^*), \tag{10}$$

where $\boldsymbol{Hu} = \boldsymbol{b}$ represents the global step system without contact.

However, the global system matrix is subject to change over time. To reuse the prefactorized matrix obtained without contact, we employ subspace BFGS iterations to minimize the quadratic energy of the global step within the subspace, i.e., we solve the quadratic problem with the increment constrained to the subspace. It is important to note that the subspace global matrix is now:

$$\mathcal{P}^\top(\boldsymbol{H} + \nabla^2 B(\boldsymbol{x}^*))\mathcal{P} = \mathcal{P}^\top \boldsymbol{H}\mathcal{P}^\top + \mathcal{P}^\top \nabla^2 B(\boldsymbol{x}^*)\mathcal{P}^\top. \tag{11}$$

The prefactorized matrix $\mathcal{P}^\top \boldsymbol{H}\mathcal{P}$ can serve as the initial approximation of the Hessian for the subspace BFGS iterations at the beginning of each time step. Importantly, the global system matrix remains unchanged within a single time step, allowing for the reuse of BFGS history across different global steps within the time step. To ensure efficiency, we limit the number of BFGS iterations to 2 in each global step, effectively providing only one additional 2-rank update to the initial reduced-order Hessian matrix based on previous updates. For quadratic problems, the optimal step size for line search can be computed analytically, requiring only one matrix-vector multiplication. It is worth mentioning that at the begining of the global-local alternations, we solve one reduced-order global step, where one backsolve using the prefactorized subspace global system can give us the exact solution. This initial guess significantly decreases the number of PD iterations.

Following the subspace BFGS iterations, we perform 5 block-diagonal Jacobi iterations on the original full-order linear system (Eq. 10) to enrich high-frequency details in the solution. This choice aligns with the common practice of multigrid, which incorporates 3–5 smoothing iterations per cycle. The block size is 3 because all matrices are assembled with 3×3 blocks, during which the diagonal blocks are tracked. However, with contact proxy stiffness matrix, the eigenvalue of the iteration matrix can easily exceed 1. Here we use a modified Jacobi with automatically tuned weight. We observe that each naive Jacobi iteration is essentially a block-diagonal preconditioned gradient descent for the corresponding quadratic problem. The steepest descent step size can be analytically computed as discussed above. We use that step size as the weight for each Jacobi iterations. This can make sure that the energy for the global step in Equation 2 is always decreasing.

In summary, each global step of our solver consists of 2 L-BFGS iterations and 5 Jacobi iterations. The PD phase is terminated if the L-infinity distance between states from two consecutive global steps falls below a given tolerance ($5 \times 10^{-3}h$ in our expriments with time step size h), or if the maximum number of iterations is reached (200 in our experiments). Additionally, we apply Chebyshev acceleration to the global steps, following Liu et al. [2017]; Wang [2015], to accelerate convergence. Empirically, we found that Chebyshev weight 0.99 worked well across all our examples.

4.4.2 Penetration Correction. The above PD solver may provide a trial solution x^{tr} with penetrations. To resolve these penetrations, we solve the following energy minimization using projected Newton method combined with a non-penetration line search, following the approach used in the IPC framework:

$$\frac{1}{2h^2}\|x - x^{\text{tr}}\|^2_M + B(x). \tag{12}$$

This objective function aims to find a balance between the trial state from PD and the collision constraints. Unlike the original IPC, which solves the linear system in the Newton method with Cholesky factorization, we employ a block-diagonal preconditioned conjugate gradient (PCG) method to solve the system. On GPUs,

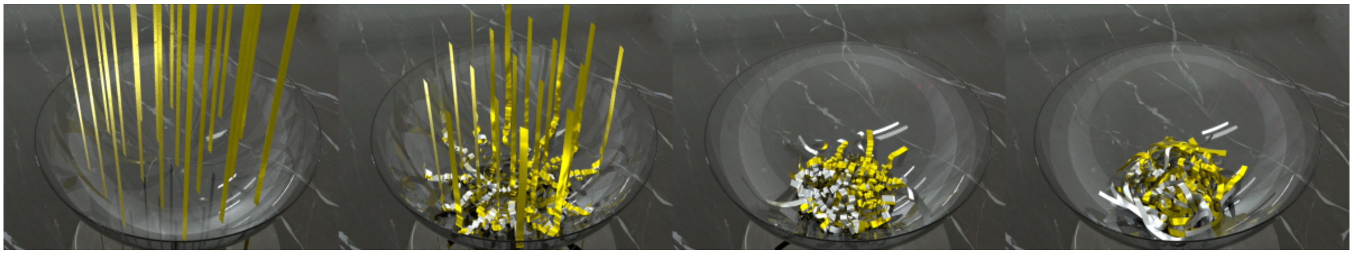


Figure 4: Ribbons. Twenty-one long ribbons are dropped into a round bowl, leading to numerous collisions and self-collisions among the ribbons.

iterative solvers are generally much faster than direct solvers. Furthermore, we use a smaller contact stiffness than the quadratic proxy. The stiffness for the proxy is slightly lower than the elasticity stiffness, while in penetration correction, the value is adjusted to 0.01×. This adjustment can reduce the condition number of the nonlinear optimization problem, facilitating faster convergence.

We also propose an early-stop strategy to reduce the number of Newton iterations required in this step. The objective of this phase is to resolve collisions while preserving momentum as much as possible. By observing this, we can increase the tolerance for DOFs that are involved in contacts, while focusing on preserving momentum primarily for DOFs that are not in contact. By prioritizing momentum preservation for non-contact DOFs and allowing for a slightly larger tolerance for contact DOFs, we can reduce the number of Newton iterations required in the penetration resolution step while still achieving satisfactory results. In our experiments, the tolerances on non-contact DOFs and contact DOFs are $10^{-2}h$ and $10^{-1}h$ respectively.

4.4.3 Friction. We also introduce an approach designed specifically for our time-splitting contact model incorporating frictional effects. This model utilizes the Hessian of the frictional energy as a damping matrix. We note that the friction coefficient μ no longer holds a physical meaning but still controls the magnitude of the frictional forces. In this friction proxy, we expand the frictional contact potential at $x_k = x_k^* + 10h\epsilon_v\dot{x}_k^*/\|\dot{x}_k^*\|$, and remove the contributions of tangential velocities smaller than ϵ_v. That is, we only adopt dynamic friction Hessians for damping along the tangential directions. Our fuzzy friction proxy can be seamlessly integrated into the contact proxy and incorporated into the PD solve loops using the BFGS algorithm.

5 GPU IMPLEMENTATION

Our algorithm has been implemented to run efficiently on a single GPU using CUDA 12.1. To avoid write-write conflicts during the assembly of global gradients and Hessian matrices, we did not utilize coloring algorithms like the one proposed in Fratarcangeli et al. [2016]. Instead, we found that using atomic add operations on our GPU was already efficient enough and simpler to implement. For matrix-vector products in the Jacobi solver, CG method, and subspace restriction/prolongation, we employed cuSPARSE, a library for efficiently performing operations on compressed sparse row (CSR) matrices in CUDA. In addition, the dense Cholesky factorization of $\mathcal{P}^\top \boldsymbol{H}\mathcal{P}$ was implemented using cuSOLVER, which enabled efficient factorization of the reduced-order global matrix. At each Newton iteration during the penetration correction, CCD is required on each search direction to guarantee non-penetration. We use the patch-based GPU collision culling from Lan et al. [2022b] to efficiently reduce the number of candidates.

6 EXPERIMENT

We implemented our algorithm on a desktop workstation with an NVIDIA RTX 3090 GPU and an Intel Core i9-10920X 3.5-GHz CPU with 12 cores. We also follow [Macklin et al. 2019] using simulation substeps for optimized performance.

6.1 Compare with Chebyshev-Accelerated Jacobi-PD

We compare our method with [Wang 2015], a classic GPU-accelerated PD algorithm using Jacobi method to solve the global system inexactly. In this test, we simulate a piece of table cloth (250K vertices, 500K triangles) with two upper corners fixed. Here, we exclude collision and self-collision processing in both methods to only showcase the performance-wise difference. For the method described in Wang [2015], we performed three separate experiments with 1,000, 2,000, and 3,000 Jacobi iterations, respectively. We also simulate the scene using Newton's method as the reference. The results of the comparison are illustrated in Figure 6. Even with 3,000 iterations, Wang [2015] exhibits more discrepancies with Newton's results compared to our method. Furthermore, our method demonstrates faster computation times compared to Wang [2015] when utilizing only 1000 iterations, despite the presence of artifacts in their results.

6.2 Benchmarks

We further compare our method with two known IPC-based cloth simulation methods, namely CIPC [Li et al. 2021] and PD-IPC [Lan et al. 2023], in multiple high-resolution cloth simulation setups. The original CIPC implementation was on the CPU, which is quite expensive. To avoid misleading benchmarks from different platforms, we re-implemented CIPC on the GPU, and we refer to our own implementation as GPU CIPC. GPU CIPC port most costly computations to CUDA including collision detection, culling, CCD, and Newton solve, and it already exhibits notable performance gains compared to its CPU counterpart. Nevertheless, our method (proposed in this paper) offers further speedups. PD-IPC is our closest competitor as it is also based on the PD. However, they utilize simplified formulations for the membrane and bending energies

Figure 5: Funnel. Three pieces of cloth are dropped onto a shallow funnel. A sphere pushes the layers of cloth completely through the hole of the funnel.

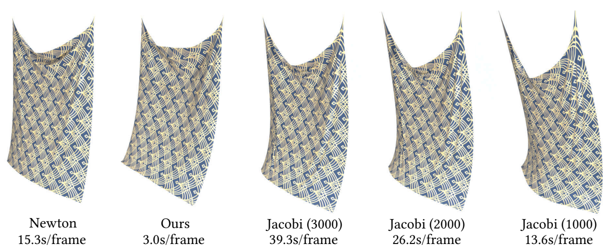


Figure 6: The comparison between our method and Wang [2015] on a cloth hanging experiment. The cloth used in the experiment consists of 250K vertices and 500K triangles. Our method demonstrates closer agreement with the results obtained using Newton's method, but is much faster.

compared to our method. We made best efforts to visually match their results with ours under the same numerical settings.

In the comparative analysis, we used a frame duration of 0.04s. We try our best to tune the time step size to ensure a fair comparison of performance among different methods. In the case of CPU IPC, the most efficient time step size is typically the frame duration. This is because the direct solver used in CPU IPC is not sensitive to the condition numbers of the Hessian matrix. However, for most iterative methods, including ours, substepping is beneficial they are more dependent on the condition numbers of the problem. The per-frame computational costs of different approaches are listed in Table 1.

Ribbons (Figure 4). We simulate the behavior of 21 long ribbons dropped into a bowl. The ribbons interact with each other, resulting in multiple collisions and self-collisions. In this example, we achieve 22× acceleration compared to GPU CIPC, 34× acceleration compared to CPU CIPC, and 2.9× acceleration compared to PD-IPC. We note that PD-IPC utilizes simplified strain and bending models. In this example, it is not able to capture fine-detailed wrinkles, resulting in fewer contact interactions compared to our method.

Cloth on sphere (Figure 7). A square cloth is dropped onto a sphere, with a ground surface located beneath the sphere. There are persistent contact between the center of cloth with the top of shpere, and persistent contact between the cloth and the ground. The contact results in numerous intricate wrinkles. We achieve 8.9× acceleration compared to GPU CIPC, 75× acceleration compared to CPU CIPC and 1.8× acceleration with PD-IPC.

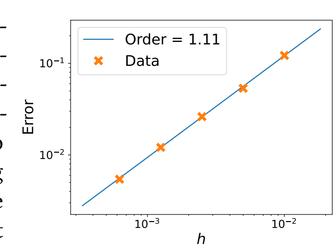
Table 1: Average computational cost per frame (s) in the comparisons.

Experiment	#V	$\hat{d}$ (m)	#Basis	#Sub-step	Ours	PD Step Percentage	GPU CIPC	CIPC	PD-IPC
Ribbons	285K	2e-4	7119	16	**46**	43.3%	1033	1579	134
Cloth on sphere	252K	1e-4	8427	8	**27**	37.2%	241	2040	49
Funnel	232K	5e-4	9432	10	**23**	63.3%	362	2740	253
Reef knot	104K	3e-4	7665	8	**10**	81.9%	90	144	18

Funnel (Figure 5). To evaluate the robustness of our contact handling, we conduct a test involving three pieces of cloth dropped onto a shallow funnel. As a scripted sphere passed through the hole of the funnel at a constant speed, multiple collisions occurred between the layers of cloth and between the cloth and the funnel. This scenario presented a significant challenge due to the substantial compression experienced by the cloth while passing through the hole. In this example, our method exhibited remarkable performance gains compared to other approaches. Specifically, we achieved a 15× acceleration compared to GPU IPC, a 119× acceleration compared to CPU IPC, and an 11× acceleration compared to PD-IPC. These results highlight the robustness and efficiency of our contact handling technique in challenging cloth simulations.

Reef knot (Figure 9). Two curved ribbons are initially intertwined and then pulled in opposite directions to form a knot. It is worth mentioning that, to apply our method, each ribbon is decomposed into three pieces combined with stitch bendings, as depicted in Figure 3. Our method manages to generate a tiny and tight knot. And we achieve 9× accelerateion comapred to GPU-IPC, 14× acceleration compared to CPU IPC, and 1.8× acceleration compared to PD-IPC.

6.3 Ablation Study



6.3.1 Convergence Under Refinement. To evaluate the accuracy of our cloth solver, we perform a convergence under refinement test w.r.t. the time step size h. We begin by capturing a contact-rich snapshot of the "cloth on sphere" experiment and using it as the initial state. We then simulate with CIPC using h=1e-4 for a duration of 0.1 seconds. The final state obtained from the CIPC simulation serves as the reference. We then use our cloth solver to simulate with

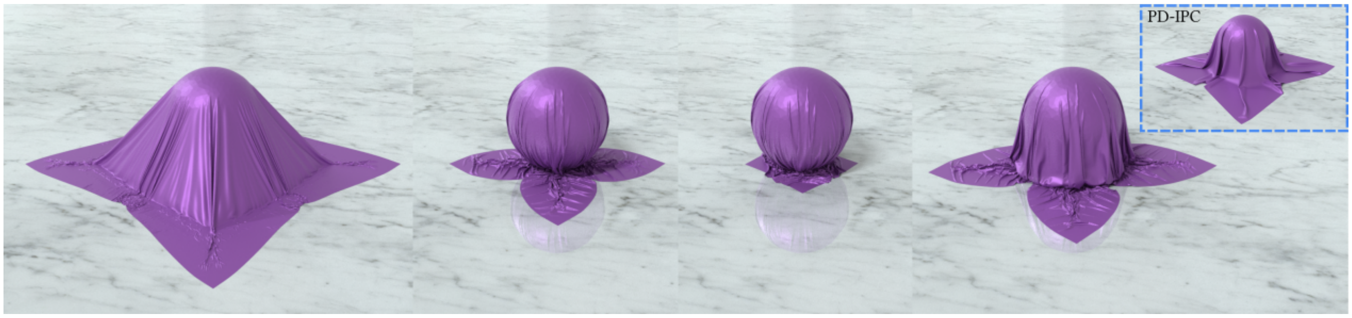


Figure 7: Cloth on sphere. A square cloth is dropped onto a sphere and form numerous intricate wrinkles. PD-IPC utilizes simplified strain and bending models, resulting in inability to capture fine-detailed wrinkles.

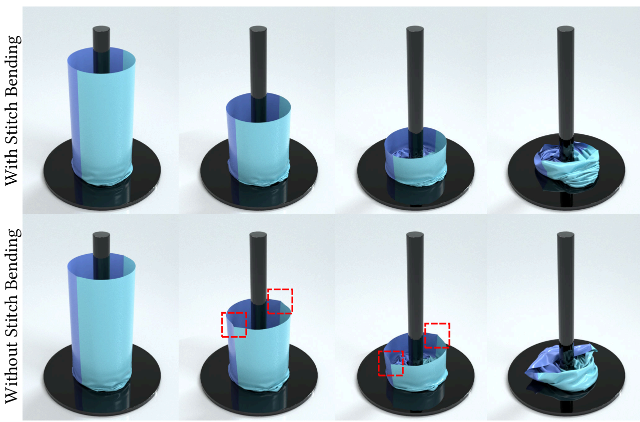
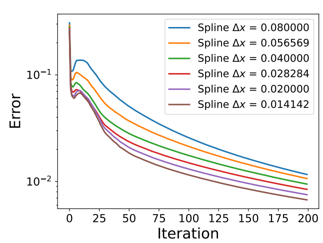


Figure 8: Stitch bending is critical for preserving the correct bending stiffness along the artificially generated seam, which does not exist on the original cloth surface.

consecutively halved time steps, starting from $h = 0.01s$, in order to estimate the convergence order of the L^2-error to the reference. Shown in the inset figure, the estimated convergence order is 1.11.

6.3.2 Number of Spline Bases. The rate of convergence in the PD phase depends on the number of spline bases. In our analysis, we take a contact-rich snapshot and focus on a single substep. The outcome from the CIPC solver serves as the reference state, against which we evaluate the convergence of our BFGS-based PD solver across varying numbers of spline bases. In our framework, the number of spline bases is controlled by the spacing between spline centers. As depicted in the inset figure, a smaller spline spacing Δx typically leads to less iterations for convergence. However, extremely small Δx values can result in too much computational time of the backsolve step on GPU. We estimated that this component's timing is approximately $O(\frac{1}{\Delta x}^{1.9})$. Despite efforts, we haven't identified an optimal empirical compromise between timing and PD convergence speed. Nevertheless, maintaining the number of bases slightly below 10K for a 100K-vertex cloth tends to yield favorable overall speedups.



6.4 Example with Stitch Bending

Some shapes needs artificial decompositions to obtain 2D ARAP parameterizations, such as a cylinder. Here we simulated a cylindrical cloth mesh, which is decomposed into two pieces. Stitch bending is critical for preserving the correct bending stiffness along the artificially generated seam, which does not exist on the original cloth surface. Without stitch bending, the cloth exhibits an unrealistically large fold along the seam, as shown in Fig. 8.

6.5 Comparisons to Hyper-Reduced Projective Dynamics

Our work differs from hyper-reduced PD (HRPD) [Brandt et al. 2018] in several key aspects. The subspace design in our method is tailored for cloth and shell structures, whereas HRPD targets volumetric solids. Additionally, we solve the full-order system, while HRPD directly simulates within the reduced subspace. Furthermore, the B-spline bases in our approach inherently satisfy partition of unity on regular grids. This is advantageous for simulating cloth, as HRPD bases lack this property. To demonstrate, we compare HRPD to our method by directly simulating the subspace dynamics without Jacobi relaxation (ours used 1200 bases, while HRPD used 1497 bases). As shown in Fig. 10, the hyper-reduced subspace exhibits severe locking under large deformations, while our method does not.

6.6 Controllable Friction

In Figure 11, we present an experiment where we vary the friction coefficients between the cloth and the slope. The cloth used in the experiment is a long rectangle with 120K vertices, and it falls onto the slope under gravity. By increasing the friction coefficient, we observe that the cloth slides down the slope at a slower speed until it gets stuck and then undergoes turning motions.

6.7 Garment Animation

Fine-detailed garment simulations play a crucial role in the animation industry. In the animation pipeline, artist-designed character animation sequences serve as moving boundary conditions for the cloth simulations. However, simulating garments on animated characters presents significant challenges due to the dramatic motions involved, such as running, jumping, or dancing. Here we use a challenging test case from Li et al. [2021], where a character turns

Figure 9: Reef knot. Two curved ribbons are initially intertwined and then pulled in opposite directions to form a tiny and tight knot.

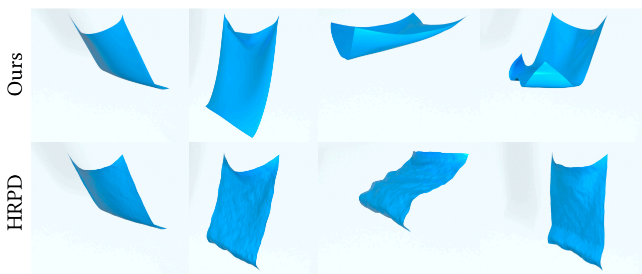


Figure 10: Our B-spline bases on regular grids naturally satisfy the property of partition of unity, while hyper-reduced bases [Brandt et al. 2018] does not, leading to severe locking under large deformations.

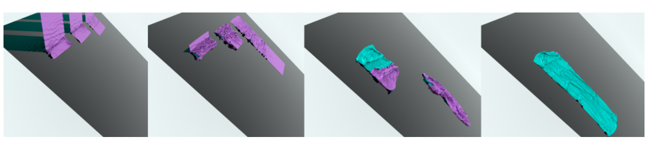


Figure 11: Controllable friction. Three pieces of 120K-node long rectangle cloth fall onto the slope under gravity. With larger friction coefficients (closer to the camera), the cloth slides down the slope at a slower rate until it gets stuck and then undergoes turning motions.

and kicks wearing a multi-layer dress (Figure 1). We subdivided the original testing garments to 120K vertices. The leg causes the dress moving in a high speed and there are intricate interactions between layers of the dress. Our method can robustly handle these collisions and resolve complex wrinkles on the cloth, with an average running time of 23 seconds per frame. This is a 6.5× acceleration compared to the GPU IPC method, highlighting the effectiveness and efficiency of our approach in handling fine-detailed garment simulations in complex animation scenarios.

7 CONCLUSION

In this paper, we propose an efficient cloth simulation method based on the projective dynamics (PD) framework. Our method combines subspace integration and parallelizable iterative relaxation techniques to effectively reduce both high-frequency and low-frequency residuals, leading to significantly improved convergence. We seamlessly integrate our method with the state-of-the-art contact handling framework, IPC, to ensure interpenetration-free results in a time-splitting manner. We have shown that our method has significant performance improvements over existing GPU solvers for high-resolution cloth simulation.

Indeed, when dealing with objects exhibiting high speeds, the time splitting error can become significant, leading to amplified damping effects. To address this issue, it would be valuable to explore adaptive substepping techniques that can enhance the accuracy of the time splitting process and alleviate the undesired damping artifacts.

Furthermore, the use of Newton's method in the penetration correction step may give rise to overshooting problems, resulting in excessive optimization iterations. Notably, the penetration correction step typically consumes a substantial portion of the computation time. To further improve the overall performance of our algorithm, it is crucial to investigate dedicated solvers specifically tailored for the penetration correction step.

ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers for their valuable feedback. We thank Pingying Chen for narrating the video. This work is partially supported by NSF 2153851, 2153863, 2023780, 2301040, 2008915, 2244651, and 2008564.

REFERENCES

Steven S An, Theodore Kim, and Doug L James. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)* 27, 5 (2008), 1–10.

Sheldon Andrews, Kenny Erleben, and Zachary Ferguson. 2022. Contact and friction simulation for computer graphics. In *ACM SIGGRAPH 2022 Courses.* 1–172.

David Baraff. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques.* 23–34.

David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques.* 43–54.

Jernej Barbič and Yili Zhao. 2011. Real-time large-deformation substructuring. *ACM transactions on graphics (TOG)* 30, 4 (2011), 1–8.

Jernej Barbič and Doug L James. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. In *ACM Trans. Graph. (TOG)*, Vol. 24. ACM, 982–990.

Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A quadratic bending model for inextensible surfaces. In *Symposium on Geometry Processing*. 227–230.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)* 33, 4 (2014), 1–11.

Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. 2018. Hyper-reduced projective dynamics. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

David E Breen, Donald H House, and Michael J Wozny. 1994. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 365–372.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 594–603.

Robert Bridson, Sebastian Marino, and Ronald Fedkiw. 2005. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*. 3–es.

Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović. 2002. Interactive skeleton-driven dynamic deformations. In *ACM Trans. Graph. (TOG)*, Vol. 21. ACM, 586–593.

Yunuo Chen, Minchen Li, Lei Lan, Hao Su, Yin Yang, and Chenfanfu Jiang. 2022. A unified newton barrier method for multibody dynamics. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.

Kwang-Jin Choi and Hyeong-Seok Ko. 2005a. Session Details: Advanced Topics on Clothing Simulation and Animation. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) *(SIGGRAPH '05)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3245700

Kwang-Jin Choi and Hyeong-Seok Ko. 2005b. Stable but responsive cloth. In *ACM SIGGRAPH 2005 Courses*. 1–es.

Min Gyu Choi and Hyeong-Seok Ko. 2005c. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Trans. on Visualization and Computer Graphics* 11, 1 (2005), 91–101.

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. 2014. Yarn-level simulation of woven cloth. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 1–11.

François Faure, Benjamin Gilles, Guillaume Bousquet, and Dinesh K Pai. 2011. Sparse meshless models of complex deformable solids. In *ACM Trans. Graph. (TOG)*, Vol. 30. ACM, 73.

Xudong Feng, Wenchao Huang, Weiwei Xu, and Huamin Wang. 2022. Learning-Based Bending Stiffness Parameter Estimation by a Drape Tester. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–16.

Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M Kaufman, and Daniele Panozzo. 2021. Intersection-free rigid body dynamics. *ACM Transactions on Graphics* 40, 4 (2021).

Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A practical gauss-seidel method for stable soft body dynamics. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–9.

Lawson Fulton, Vismay Modi, David Duvenaud, David IW Levin, and Alec Jacobson. 2019. Latent-space Dynamics for Reduced Deformable Simulation. In *Computer graphics forum*, Vol. 38. Wiley Online Library, 379–391.

Benjamin Gilles, Guillaume Bousquet, Francois Faure, and Dinesh K Pai. 2011. Frame-based elastic models. *ACM Trans. Graph. (TOG)* 30, 2 (2011), 15.

Yotam Gingold, Adrian Secord, Jefferson Y Han, Eitan Grinspun, and Denis Zorin. 2004. A discrete model for inelastic deformation of thin shells. In *ACM SIGGRAPH/Eurographics symposium on computer animation*. Citeseer.

Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient simulation of inextensible cloth. In *ACM SIGGRAPH 2007 papers*. 49–es.

Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Citeseer, 62–67.

Kris K Hauser, Chen Shen, and James F O'Brien. 2003. Interactive Deformation Using Modal Analysis with Constraints.. In *Graphics Interface*, Vol. 3. 16–17.

Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. 2016. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*. 1–52.

Kenneth Langstreth Johnson and Kenneth Langstreth Johnson. 1987. *Contact mechanics*. Cambridge university press.

Jonathan M Kaldor, Doug L James, and Steve Marschner. 2008. Simulating knitted cloth at the yarn level. In *ACM SIGGRAPH 2008 papers*. 1–9.

Danny M Kaufman, Shinjiro Sueda, Doug L James, and Dinesh K Pai. 2008. Staggered projections for frictional contact in multibody systems. In *ACM SIGGRAPH Asia 2008 papers*. 1–11.

Theodore Kim. 2020. A Finite Element Formulation of Baraff-Witkin Cloth. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 171–179.

Theodore Kim and John Delaney. 2013. Subspace fluid re-simulation. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–9.

Lei Lan, Danny M Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022a. Affine body dynamics: Fast, stable & intersection-free simulation of stiff materials. *arXiv preprint arXiv:2201.10022* (2022).

Lei Lan, Minchen Li, Chenfanfu Jiang, Huamin Wang, and Yin Yang. 2023. Second-order Stencil Descent for Interior-point Hyperelasticity. *ACM Transactions on Graphics (TOG)* (2023).

Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022b. Penetration-free projective dynamics on the GPU. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

Lei Lan, Yin Yang, Danny Kaufman, Junfeng Yao, Minchen Li, and Chenfanfu Jiang. 2021. Medial IPC: accelerated incremental potential contact with medial elastics. *ACM Transactions on Graphics* 40, 4 (2021).

Kookjin Lee and Kevin T Carlberg. 2020. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* 404 (2020), 108973.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49.

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 170 (2021).

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. 2008. A local/global approach to mesh parameterization. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1495–1504.

Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-newton methods for real-time simulation of hyperelastic materials. *Acm Transactions on Graphics (TOG)* 36, 3 (2017), 1–16.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications* 9, 1 (2018), 4950.

Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective dynamics with dry frictional contact. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 57–1.

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. 2020. Primal/dual descent methods for dynamics. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 89–100.

Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapong Chentanez, Stefan Jeschke, and Matthias Müller. 2019. Small steps in physics simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–7.

Sebastian Martin, Peter Kaufmann, Mario Botsch, Eitan Grinspun, and Markus Gross. 2010. Unified simulation of elastic rods, shells, and solids. In *ACM Trans. Graph. (TOG)*, Vol. 29. ACM, 39.

Hammad Mazhar, Toby Heyn, Dan Negrut, and Alessandro Tasora. 2015. Using Nesterov's method to accelerate multibody dynamics with friction and contact. *ACM Transactions on Graphics (TOG)* 34, 3 (2015), 1–14.

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A Otaduy, and Steve Marschner. 2012. Data-driven estimation of cloth simulation models. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 519–528.

Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C Schvartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A Otaduy. 2013. Modeling and estimation of internal friction in cloth. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–10.

Alex Pentland and John Williams. 1989. Good vibrations: Modal dynamics for graphics and animation. In *SIGGRAPH Comput. Graph.*, Vol. 23. ACM.

Xavier Provot et al. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*. Canadian Information Processing Society, 147–147.

Stéphane Redon, Abderrahmane Kheddar, and Sabine Coquillart. 2002. Fast continuous collision detection between rigid bodies. In *Computer graphics forum*, Vol. 21. Wiley Online Library, 279–287.

Siyuan Shen, Yin Yang, Tianjia Shao, He Wang, Chenfanfu Jiang, Lei Lan, and Kun Zhou. 2021. High-Order Differentiable Autoencoder for Nonlinear Model Reduction. *ACM Trans. Graph.* 40, 4, Article 68 (jul 2021), 15 pages. https://doi.org/10.1145/3450626.3459754

Eftychios Sifakis and Jernej Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) *(SIGGRAPH '12)*. Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages. https://doi.org/10.1145/2343483.2343501

Georg Sperl, Rahul Narain, and Chris Wojtan. 2020. Homogenized yarn-level cloth. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 48–1.

Tetsuya Takahashi and Christopher Batty. 2021. FrictionalMonolith: a monolithic optimization-based approach for granular flow with contact-aware rigid-body coupling. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–20.

Min Tang, Dinesh Manocha, Miguel A Otaduy, and Ruofeng Tong. 2012. Continuous penalty forces. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–9.

Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental collision handling for GPU-based interactive cloth simulation. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Bernhard Thomaszewski, Simon Pabst, and Wolfgang Strasser. 2009. Continuum-based strain limiting. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 569–576.

Richard Tonge, Feodor Benevolenski, and Andrey Voroshilov. 2012. Mass splitting for jitter-free parallel rigid body simulation. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–8.

Adrien Treuille, Andrew Lewis, and Zoran Popović. 2006. Model reduction for real-time fluids. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 826–834.

Pascal Volino, Martin Courchesne, and Nadia Magnenat Thalmann. 1995. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 137–144.

Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics* 28, 4 (2009), Article–No.

Huamin Wang. 2015. A chebyshev semi-iterative approach for accelerating projective and position-based dynamics. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–9.

Huamin Wang, James O'Brien, and Ravi Ramamoorthi. 2010. Multi-resolution isotropic strain limiting. *ACM Transactions on Graphics (TOG)* 29, 6 (2010), 1–10.

Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. 2011. Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG)* 30, 4 (2011), 1–12.

Huamin Wang and Yin Yang. 2016. Descent methods for elastic body simulation on the GPU. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–10.

Tianyu Wang, Jiong Chen, Dongping Li, Xiaowei Liu, Huamin Wang, and Kun Zhou. 2022. Fast GPU-Based Two-Way Continuous Collision Handling. arXiv:2211.04045 [cs.GR]

Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. 2007. Discrete quadratic curvature energies. *Computer Aided Geometric Design* 24, 8-9 (2007), 499–518.

Andy Wathen. 2008. *Chebyshev Semi- iteration in Preconditioning*. Technical Report.

Steffen Wiewel, Moritz Becher, and Nils Thuerey. 2019. Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer graphics forum*, Vol. 38. Wiley Online Library, 71–82.

Philip Wolfe. 1969. Convergence conditions for ascent methods. *SIAM review* 11, 2 (1969), 226–235.

Philip Wolfe. 1971. Convergence conditions for ascent methods. II: Some corrections. *SIAM review* 13, 2 (1971), 185–188.

Longhua Wu, Botao Wu, Yin Yang, and Huamin Wang. 2020. A safe and fast repulsion method for GPU-based cloth self collisions. *ACM Transactions on Graphics (TOG)* 40, 1 (2020), 1–18.

Tianyi Xie, Minchen Li, Yin Yang, and Chenfanfu Jiang. 2023. A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling. *ACM Transactions on Graphics (TOG)* (2023).

Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbič. 2015. Interactive material design using model reduction. *ACM Transactions on Graphics (TOG)* 34, 2 (2015), 1–14.

Hongyi Xu, Yili Zhao, and Jernej Barbič. 2014. Implicit multibody penalty-baseddistributed contact. *IEEE transactions on visualization and computer graphics* 20, 9 (2014), 1266–1279.

Yin Yang, Dingzeyu Li, Weiwei Xu, Yuan Tian, and Changxi Zheng. 2015. Expediting precomputation for reduced deformable simulation. *ACM Trans. Graph. (TOG)* 34, 6 (2015).