

# A Cubic Barrier with Elasticity-Inclusive Dynamic Stiffness

RYOICHI ANDO, zozo, Japan

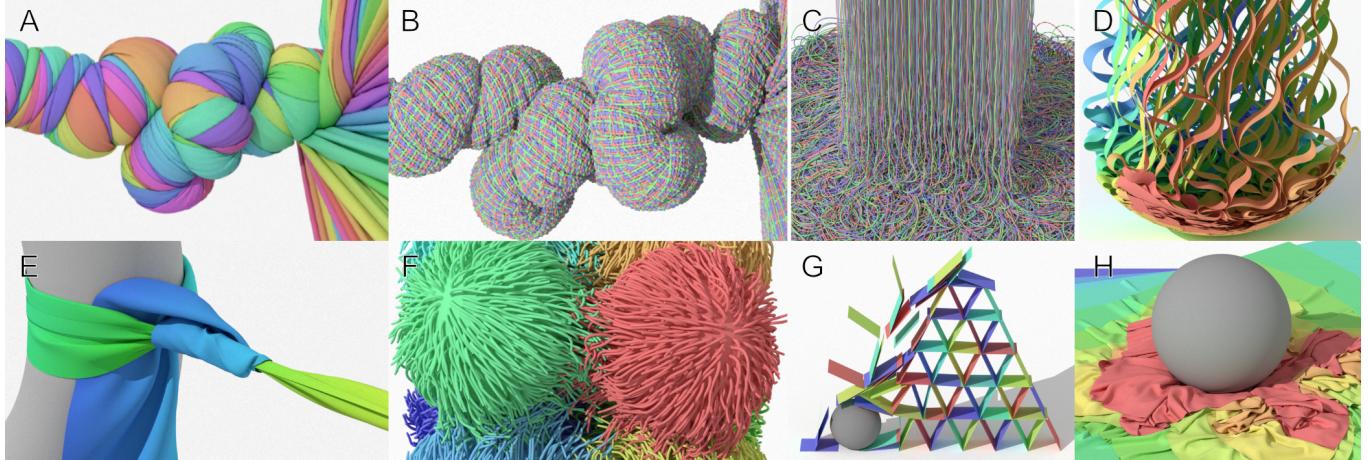


Fig. 1. Five cylindrical shells are individually twisted and bundled together (A), A single woven cylinder is twisted forming intricate buckles (B), A set of lengthy strands are poured into a bowl (C), Fluttering ribbons are poured into a bowl followed by a strong impact from a falling sphere (D), A fishing knot is tightened (E), Eight squishy hairy balls are compressed and released (F), A house of cards collapses with a gentle touch by a rolling sphere (G), Ten sheets are smashed onto the ground by a fast falling heavy sphere (H). Peak contact counts are 168.35 million (A), 17.59 million (D), 6.49 million (C), 6.04 million (B), 5.63 million (H), 2.89 million (F). Average time per video frame is 194s (A), 745s (B), 247s (C), 184s (D), 3.30s (E), 196s (F), 5.04s (G), 77.35s (H).

This paper presents a new cubic barrier with elasticity-inclusive dynamic stiffness for penetration-free contact resolution and strain limiting. We show that our method enlarges tight strain-limiting gaps where logarithmic barriers struggle and enables highly scalable contact-rich simulation.

CCS Concepts: • Computing methodologies → Physical simulation.

Additional Key Words and Phrases: collision, contact

ACM Reference Format:

Ryoichi Ando. 2024. A Cubic Barrier with Elasticity-Inclusive Dynamic Stiffness. *ACM Trans. Graph.* 43, 6, Article 224 (December 2024), 13 pages. <https://doi.org/10.1145/3687908>

## 1 INTRODUCTION

Exact contact resolution is challenging [Kaufman et al. 2005; Takahashi and Batty 2021; Tomcin et al. 2014; Won and Lee 2019]. Thin shells are particularly difficult, as even slight penetration can halt the entire simulation, leaving it stuck and struggling to untangle from self-intersections [Baraff et al. 2003; Volino and Magnenat-Thalmann 2006], except when the explicit order of shell layers is known [Buffet et al. 2019; Lee et al. 2023; Lewin 2018]. This makes

purely volumetric approaches ill-suited for codimensional colliders [Allard et al. 2010; Chen et al. 2023; Faure et al. 2008; Tang et al. 2012]. If collisions are unresolvable, one may resort to a rigid impact zone [Harmon et al. 2008; Huh et al. 2001; Provot 1997], rigidifying and delaying the collision resolution to the next step [Bridson et al. 2002; Ye et al. 2017] or deferring it to another relaxation loop until all intersections are resolved [Wu et al. 2020]. Unfortunately, there is no guarantee that all intersections will be resolved in this manner [Tang et al. 2018]. One solution is to use the inexact solution as guidance within the admissible space [Wang et al. 2023] or fall back to the interior point method [Wu et al. 2020].

Collisions are found via cubic equations [Bridson et al. 2002; Provot 1997] or a parity check [Brochu et al. 2012; Wang et al. 2022]. However, both methods need to deal with degenerate cases and limitations in floating-point accuracy [Wang et al. 2021; Wang 2014].

The incremental potential contact (IPC) [Li et al. 2020, 2021] circumvents the issues by inserting a logarithmic barrier between objects that are nearly touching to ensure penetration-free contacts.

*The Problem.* When logarithmic barriers are minimized with a Newton's solver, the search direction obtained by dividing the force by curvature yields an extremely small magnitude near gap zero.

Huang et al. [2024] were the first to identify this issue and proposed clamping the curvature. This prevents search direction locking but instead produces excessively large directions. This is more discussed in the supplementary material. Lan et al. [2023] segregated domains into multiple non-adjacent colors, solving each color exactly in parallel, and merged them until global convergence was

Author's address: Ryoichi Ando, ryoichi.ando@zozo.com, ZOZO, Chiba, Japan.

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

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2024/12-ART224 \$15.00 <https://doi.org/10.1145/3687908>

achieved. This way, locked search directions do not affect the entire update, but the inherent issue remains.

### 1.1 Our Solution

We begin with a weak cubic barrier

$$\psi_{\text{weak}}(g_i, \hat{g}, \kappa_i) = \begin{cases} -\frac{2\kappa_i}{3\hat{g}_i} (g_i - \hat{g}_i)^3, & \text{if } g_i \leq \hat{g} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where  $g_i$  denotes the gap distance with respect to  $i$ -th constraint.  $\kappa_i$  and  $\hat{g}_i$  denote the stiffness constant and the maximal gap with respect to the  $i$ -th constraint, respectively.

The above energy is  $C^2$  at  $g_i = \hat{g}_i$  and yields search directions that are not too small or too large near the critical point. The cost we have to pay with this replacement is voiding the desirable property of the energy being infinite at gap zero. We first attempt to fix this by reinterpreting  $\kappa_i$  as a function of  $x$ , a vector comprising a set of mesh vertices, and letting  $\kappa_i(x) \rightarrow \infty$  as  $g_i \rightarrow 0$ . Unfortunately, this again introduces search direction locking. To resolve this, we employ a semi-implicit approach: we treat  $\kappa_i(x)$  as constant when computing the derivatives of its energy potential. With this, the search direction locking is no longer an issue.

Throughout the paper, we employ the bar notation  $\bar{\kappa}_i$  to indicate that the variable is semi-implicitly evaluated. The ordering of the different  $\bar{\kappa}_i$  updates during our solver algorithm are covered in Section 3. In this paper, we present two key contributions

- A cubic energy that improves strain-limiting.
- A dynamically evaluated constraint stiffness using elasticity energies, which robustly enlarges the contact gap.

## 2 RELATED WORK

Contact resolution consists of four parts (a) culling [Bridson et al. 2002; Lan et al. 2022; Li et al. 2020], (b) detection [Brochu et al. 2012; Wang et al. 2022, 2021], (c) resolve [Baraff et al. 2003; Chen et al. 2023; Kim and Eberle 2022; Ye et al. 2017], and (d) energies [Kim and Eberle 2022; McAdams et al. 2011; Zhao et al. 2022a]. We refer readers to the survey by Andrews et al. [2022]. Below, we may simply write a gap distance as  $g$  or  $g_i$ .

### 2.1 Logarithmic Barrier Functions

Logarithmic barriers [Chen et al. 2022; Huang et al. 2024; Lan et al. 2023; Li et al. 2020, 2021; Zhao et al. 2022a] are among the most relevant counterparts. An example from Li et al. [2021] is

$$\psi_{\ln}(g) = \begin{cases} -\kappa (g - \hat{g})^2 \ln(g/\hat{g}), & \text{if } g \leq \hat{g} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Another difference from our method, aside from search direction locking, is the way stiffness is selected. To select  $\kappa$ , Li et al. [2020] proposed to balances two forces. Let  $f_{\text{contact}}$  and  $f_{\text{ext}}$  be a pair of forces representing contact and an external system, and  $f_{\text{contact}}$  be a barrier force without stiffness coefficient. With this,  $\kappa$  is chosen such that  $\kappa = \arg \min_{\kappa} \frac{1}{2} \|f_{\text{ext}} + \kappa f_{\text{contact}}\|^2$ . Taking a derivative with respect  $\kappa$ , substituting  $f_{\text{ext}} = -\nabla \Psi_{\text{total}}$  and setting the expression equal to zero, one obtains  $\kappa = (f_{\text{contact}} \cdot \nabla \Psi_{\text{total}}) / \|f_{\text{contact}}\|^2$ . As reported, this can yield invalid values. To avoid them, lower and upper bounds for  $\kappa$  are defined. First, a maximum curvature  $c_{\max}$

is computed  $g = 10^{-8}\hat{g}$ . With  $c_{\max}$  defined, the lower bound of  $\kappa$  is set to  $\kappa_{\min} = 10^{11}\bar{m}/c_{\max}$ , where  $\bar{m}$  is a lumped mass. Once  $\kappa_{\min}$  is computed, the upper bound is set to  $\kappa_{\max} = 100\kappa_{\min}$ . This strategy needs additional special care. When  $g$  becomes below  $10^{-9}\hat{g}$  and is also decreasing between Newton's steps,  $\kappa$  needs to be doubled. Also, choosing an adequate  $\kappa$  reportedly remains difficult [Shen et al. 2024]. Our stiffness selection differs in following aspects

- Our stiffness is parameter-free in the sense that it does not involve any numbers such as  $10^{-9}$ ,  $10^{11}$ , or  $100$ .
- Ours does not set any lower or upper bounds for  $\kappa$ .
- Ours does not require fail-safes such as doubling stiffness.

Another differences from the IPC [Li et al. 2020] is that, since we employ a distance evaluation using a fixed contact normal as detailed in the supplementary material, we do not require the full Hessians of triangle-point and edge-edge distances, which introduce analytical complexity [Shi and Kim 2023] and require mollification [Huang et al. 2024].

### 2.2 Quadratic Energies

Quadratic energies [Baraff and Witkin 1998; Bridson et al. 2002; Kim and Eberle 2022; McAdams et al. 2011] take the form  $\psi_{\text{quad}}(g) = \kappa(g - \hat{g})^2/2$ . Unlike (2), quadratic energies are not  $C^2$  continuous at  $g = \hat{g}$  when  $\psi_{\text{quad}}(g) = 0$  if  $g \geq \hat{g}$ . Tang et al. [2012] addressed the issue through continuous penalty forces. Instead of evaluating impulse forces using only the current state, they advocate examining the path between steps and integrating the penetration along the trajectory. Such a strategy may mitigate the issue to some extent but comes at the cost of added complexity, making it nontrivial to employ implicit formulations.

Harmon et al. [2009] also recognized this issue and proposed cumulative quadratic energy layers. In doing so, they incrementally add multiple quadratic energy terms with slopes of varying steepness, ensuring that the barrier approaches infinity as the gap diminishes to zero. The main difference is that ours considers dynamic elasticity, while their method only considers the gap.

### 2.3 Constrained Optimization

Penetration-free dynamics can be solved via constrained formulations such as the active set method [Verschoor and Jalba 2019], gradient descent [Tang et al. 2018], primal/dual descent [Macklin et al. 2020], projection [Harmon et al. 2008; Kaufman et al. 2008; Müller et al. 2007; Wu et al. 2020].

Constrained optimizations that involve Lagrangian multipliers introduce extra degrees of freedom and lead to indefinite systems when solved together with other forces. Such systems could be converted into symmetric positive definite [Takahashi and Batty 2022]; however, this results in either increased degrees of freedom or denser systems. Another alternative is Alternating Direction Method of Multipliers (ADMM) [Overby et al. 2017] or its variants [Macklin et al. 2016]. Our method can be seen as a variant of projective dynamics in the sense that after a global solve (the Newton's step) we project stiffness (a semi-implicit evaluation) [Bouaziz et al. 2014], and as such, shares some similarities.

One may use a Hessian-free projection [Bouaziz et al. 2014; Macklin et al. 2016; Müller et al. 2007], which only utilizes first-order

derivatives. However, simulating stiff systems such as nearly inextensible clothes with this method requires a long convergence time. An exception is the work of Lan et al. [2023], which showed that a colored Gauss-Seidel method can be leveraged to solve stiff systems.

Wang et al. [2023] introduced a two-way approach that solves constrained optimization inexactly and uses it as guidance to conservatively update variables within an admissible space, similar to the interior point method, which our method also employs.

## 2.4 Nitsche's Method

Nitsche's method has proven to be a powerful tool for handling contacts in mechanical engineering [Chouly et al. 2017, 2022; Gustafsson et al. 2020]. In addition to a purely penalty-based method, Nitsche's method also adds a virtual work needed to separate two touching objects. A consequence is that their stiffness  $\gamma$  does not need to be set high to enforce contact resolution. This makes the entire system mildly stiff, making the linear system tractable.

Our work shares some similarity with Nitsche's method in that both methods include elastic terms that are dynamically evaluated, but Nitsche's method still requires stabilizer parameter selection, with  $\gamma$  often chosen based on discretization parameters such as the squared inverse of element size [Griebel and Schweitzer 2003; Sanders et al. 2009]. In contrast, our method determines stiffness without any prior knowledge of the scene or discretization.

## 2.5 Strain Limiting

Strain limiting was initially advocated by Provot [1997] and popularized in graphics by Bridson et al. [2002]. Both methods are based on explicit integration, which can limit scalability. To address this, Goldenthal et al. [2007] proposed a semi-implicit method that formulates the problem as a projection constraint solver. This technique was further modified to inequality constraints by Jin et al. [2017]. They demonstrated that instead of using equality constraints, correcting only elongated edges would produce more detailed wrinkles and avoid membrane locking.

In general, achieving exact satisfaction of strain limiting constraints is computationally expensive. Wang et al. [2010] proposed a multi-resolution solver to accelerate the convergence. Kim et al. [2012] introduced a long-range attachment technique that approximates inextensibility for real-time applications. The application of strain limiting is not limited to cloth; it is also applied for hair [Müller et al. 2012], ribbon [Shen et al. 2015] and rods [Zhao et al. 2022b]. Many of the above are edge-based, focusing on controlling stretch deformations along individual edges. Thomaszewski et al. [2009] proposed a continuum-based strain limiting technique that allows for accurate control of stretch and shear deformations with individual limits. Finally, Li et al. [2021] introduced both anisotropic and isotropic strain limiting using a logarithmic barrier inequality constraint energy, which enables tight coupling with elasticity and collision energies.

## 2.6 Yarn-Level Contact

Yarn-level collisions in graphics started with the work of Kaldor et al. [2008], which introduced a unique collision energy:  $\psi(g) = 1/g^2 + g^2$ , encoding that yarns should not be too close to each other

but separated by some margin. Yarn-level collisions involve a large number of contacts and can be expensive. To mitigate this cost, contact force evaluation is made efficient through linearization [Kaldor et al. 2010].

Yuksel et al. [2012] proposed an adaptive time-stepping approach to prevent pull-through, which refers to tunneling artifacts of yarns. Instead of solving multivariate nonlinear equations, they propose to place subdividable spheres. The crossing spheres indicate the possibility of pull-throughs, and as such, the step size is reduced accordingly. Cirio et al. [2014] extended the method of Sueda et al. [2011] for woven fabrics based on explicit knowledge of the woven pattern. Additionally, for distant yarn-yarn collisions, they created a thin signed field around the fabric sheet, and its value was used to detect collisions. Cirio et al. [2017] generalized the concept of utilizing explicit knowledge about the knitted pattern to simulate various types of fabrics.

Despite the aforementioned techniques, accurately simulating yarn-level contact remains computationally expensive. This limitation is addressed via the homogenization of yarn deformations elastodynamically [Sperl et al. 2020], mechanically [Sperl et al. 2021] and with machine learning [Feng et al. 2024].

## 3 METHOD OVERVIEW

Key highlights of our method consist of two techniques: a cubic function and elasticity-inclusive dynamic stiffness, each with distinct practical effects

- Our cubic energy is best suited for scenarios where gaps become extremely small, such as in strain limiting. Our experiments show that its superior effects over logarithmic barriers are not observed in contact handling, but it still robustly performs in these cases.
- Our elasticity-inclusive stiffness is most effective for contacts, as it robustly enlarges contact gaps, eliminating the need for manual parameter tuning.

### 3.1 Objective Function

Our solver inherits many components from CIPC [Li et al. 2021]; hence, we only focus on key differences. Implementation details are provided in the supplementary material. Our goal is to minimize the following objective

$$\psi_{\text{total}}(\mathbf{x}) = \psi_{\text{dyn}}(\Delta t, \mathbf{x}) + \sum_i \psi_{\text{weak}}(g_i, \hat{g}, \bar{\kappa}_i), \quad (3)$$

with respect to  $\mathbf{x}$  encoding mesh vertices, where  $\psi_{\text{dyn}}$  encodes step-size  $\Delta t$  dependent inertia and elastic energies [Martin et al. 2011] (e.g., hyper-elasticity and bending).  $\psi_{\text{weak}}$  encodes contacts, boundary conditions, and strain limiting. Once again,  $\bar{\kappa}$  is semi-implicitly updated when computing the derivatives of respective types at every Newton's step. This also means that the objective function itself changes slightly at each Newton step.

### 3.2 Time Integration

Ideally, we seek the point of (3) where the gradient of the objective vanishes to near numerical precision. However, this requires many iterations, even if we could solve it fully implicitly. Therefore, we

minimize (3) via a customized Newton's method as illustrated in Algorithm 1.  $0 < \alpha \leq 1$  is a line search fractional coefficient encoding the maximal contact-aware (or more formally, constraint-aware) feasible step size. Operators  $\nabla_x^2$  and  $\nabla_x$  are second and first derivatives with respect to  $x$ . Although this algorithm does not achieve accurate converged solutions, we find that it still produces visually satisfactory results without instabilities. We provide a summary below. More details are available in the supplementary material.

**3.2.1 Our Temporal Integrator Overview.** The key idea behind our integrator is that the optimized position is a function of the step size  $\Delta t$ . When  $\Delta t = 0$ , the optimized position clearly does not move at all. Also, when  $\Delta t$  is sufficiently small, we can assume that the positional trajectory is nearly linear within the step.

From these two observations, we can make the following assumption: when the search direction is reduced by a line search factor  $\alpha$ , this solution can be interpreted as a solution at a time  $\alpha\Delta t$ . However, the same analogy does not apply when multiple Newton steps are involved, as it becomes unclear how to interpret line search scaling factors after the first step.

To make our strategy applicable for multiple Newton steps, we consider a virtual particle in purely inertial motion, moving with a constant velocity. In this specific case, the inertial potential energy is given by  $m_p\|\mathbf{x} - \mathbf{z}\|^2/2$ , where  $\mathbf{z}$  is a target inertia position at  $t + \Delta t$  and  $m_p$  is the particle mass. When the particle advances by  $\alpha_1(\mathbf{z} - \mathbf{x})$  for the first time, the remaining path to  $\mathbf{z}$  is  $(1 - \alpha_1)(\mathbf{z} - \mathbf{x})$ . In the next Newton step, when the displacement is shrunk by  $\alpha_2$ , the particle moves by  $\alpha_2(1 - \alpha_1)(\mathbf{z} - \mathbf{x})$ . We can generalize this idea to predict the accumulated step sizes and arrive at the update rule in Line 3 of Algorithm 1. The accumulated step size is recorded as  $\beta$ .

**3.2.2 Target Maximal  $\beta_{\max}$ .** Our time integrator indicates that reaching exactly  $\beta = 1$  is not possible unless the line search reports no collision, which is very expensive in contact-rich scenarios. To address this, we set a maximum value for  $\beta$ , denoted as  $\beta_{\max}$ . For all the examples, we set  $\beta_{\max} = 1/4$ . All the steps in our examples have met this criterion to exit the loop, including contact frictional cases, which were explicitly enforced to take 32 Newton steps. A larger value, such as  $\beta_{\max} = 1/2$ , is possible; however, we found that  $\beta_{\max} = 1/4$  provides a better balance between accuracy and performance.

**3.2.3 Error Reduction Pass.** In the above, we only assumed inertial motion; however, the actual simulation also involves gravity and elasticity forces. Therefore, the trajectory is nonlinear within  $\Delta t$ . However, we consider that the above prediction still serves as a good approximation because, for small  $\Delta t$ , the inertia term  $m_p/\Delta t^2$  becomes dominant. For this reason, before exiting the optimization loop, we reconfigure the objective function with  $\beta\Delta t$  and run an additional error reduction Newton's step.

**3.2.4 Extended Search Direction.** When updating vertex positions with the search direction, we pay careful attention to ensure that the vertices do not settle too close to the constraint limits. Otherwise, the stiffness may become too large in the next Newton step, leading to simulation instability.

We achieve this by extending the search direction by 25%, and obtain the line search scaling factor  $\alpha$  that encodes the maximal

---

**Algorithm 1** Simulation Step

---

```

1: function ADVANCE_STEP( $\Delta t, \mathbf{x}$ )
2:    $\beta \leftarrow 0$ 
3:   while  $\beta < \beta_{\max}$  do
4:      $\mathbf{x}, \alpha \leftarrow \text{inner\_step}(\Delta t, \mathbf{x})$             $\triangleright$  Newton's Step
5:      $\beta \leftarrow \beta + (1 - \beta)\alpha$ 
6:   end while
7:    $\mathbf{x}_{\_} \leftarrow \text{inner\_step}(\beta\Delta t, \mathbf{x})$         $\triangleright$  Error Reduction Pass
8:   return  $\mathbf{x}, \beta\Delta t$                                  $\triangleright$  Actual Advanced Step
9: end function

10: function INNER_STEP( $\Delta t, \mathbf{x}$ )
11:    $H, f \leftarrow \nabla_x^2 \psi_{\text{total}}, -\nabla_x \psi_{\text{total}}$ ,       $\triangleright \bar{\kappa}$  updates applied during
    evaluation.
12:    $\mathbf{d} \leftarrow H^{-1}f,$ 
13:    $\alpha \leftarrow \text{Constraint Line Search from } \mathbf{x} \text{ to } \mathbf{x} + 1.25\mathbf{d}$ 
14:   return  $\mathbf{x} + \alpha\mathbf{d}, \alpha$ 
15: end function
```

---

feasible displacement from  $\mathbf{x}$  to  $\mathbf{x} + 1.25\mathbf{d}$  (Line 13). We performed constraint-only line searches and did not observe any instabilities. More details on our line search are provided in the supplementary material.

### 3.3 Our Stiffness Design Principle

We begin by observing that the step size  $\Delta t$  and  $\hat{g}$  often satisfy e.g.,  $\hat{g}/\Delta t \approx 1\text{m/s}$ . From a numerical perspective, where both  $\Delta t$  and  $\hat{g}$  are recorded in SI units, this implies that their magnitudes are typically of the same order.

Recalling that the stiffness of the inertia Hessian (its eigenvalue) is given by  $m/\Delta t^2$ , where  $m$  is the element mass, we aim for the eigenvalue of the contact Hessian to be of the same order. This ensures that the conditioning pollution due to contact handling remains somewhat predictable. Now, observe that the second derivative (curvature) of our cubic barrier,  $\psi_{\text{weak}}$ , linearly approaches  $2\kappa$  as  $g \rightarrow 0$ . By setting  $\kappa = m/g^2$ , we can control the impact of the contact on the conditioning to be around the same order as the inertia term.

**3.3.1 Elasticity-Inclusive Stiffness.** From the above discussion, we can see that the stiffness (the eigenvalue of its Hessian) of the inertia is  $m/\Delta t^2$ . This can be further extended to account for elasticity energy: let  $H$  represent the elasticity Hessian and  $\mathbf{n}$  be a unit vector of interest, such as the contact normal. In this context, the total system stiffness in the direction of  $\mathbf{n}$  is  $m/\Delta t^2 + \mathbf{n} \cdot (H\mathbf{n})$ . The idea here is that by assigning the stiffness  $\bar{\kappa}$  of a barrier in the form of

$$\bar{\kappa} = \frac{m}{g^2} + \mathbf{n} \cdot (H\mathbf{n}), \quad (4)$$

the barrier would ideally be stiff enough to resist incoming forces. If this stiffness turns out to be still weak, the term  $m/g^2$  ensures that the barrier eventually becomes strong enough as the gap  $g$  shrinks to zero to counteract the forces. Continuous collision detection, or more generally, the line search mechanism, prevents constraint pass through.

This establishes our stiffness design principle: for each constraint type, we define the corresponding mass  $m$ , geometric gap  $g$ , direction

$\mathbf{n}$ , and the elasticity Hessian  $H$ , and substitute them to (4). All the  $\bar{\kappa}$  in the following adhere to this rule.

### 3.4 Contact Stiffness

With the knowledge above, we start by defining the stiffness for contact. The contact energy is given by  $\psi_{\text{contact}} = \psi_{\text{weak}}(\mathbf{g}_i, \hat{\mathbf{g}}, \bar{\kappa}_i^{\text{contact}})$ . We update the contact stiffness when it is requested by the computation of contact Hessians and forces during the contact proximity query. The stiffness  $\bar{\kappa}_i^{\text{contact}}$  is given by

$$\bar{\kappa}_i^{\text{contact}} = \frac{m_i}{g_i^2} + \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \cdot \left( H \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \right), \quad (5)$$

where  $m_i$  is an average vertex mass.  $\mathbf{w}_i$  is an extended contact direction computed as follows: let  $\mathbf{p}_i$  and  $\mathbf{q}_i$  denote two nearly touching positions forming a contact, and a matrix  $W_i$  is given such that  $\mathbf{p}_i - \mathbf{q}_i = W_i \mathbf{x}$ .  $\mathbf{w}_i$  is given by  $\mathbf{w}_i = W_i^T (\mathbf{p}_i - \mathbf{q}_i)$ .  $H$  is the elasticity Hessian after being enforced symmetric (semi-)positive definite.

The purpose of  $W^T$  is to redistribute the contact direction  $\mathbf{p}_i - \mathbf{q}_i$  to the vertices forming the contact, using appropriate weighting. For example, if  $W$  is an interpolation operator for two particles with equal weights (1/2, 1/2), its transpose distributes the interpolated value back to both particles equally. If the weights are (1, 0), the redistribution operator simply returns the value back to the first particle. This is necessary because  $H$  is the linear operator with respect to the vertices.

Recalling that we fix all the terms on the right hand side of (5) after evaluation, the chain-rule does not apply when taking derivatives. Therefore, 6th-order tensors of  $H$  are not necessary. We compute contact derivatives using a semi-implicit linearization [Chen et al. 2024], which we detail in the supplementary material.

### 3.5 Boundary Conditions

We wish to pin vertex  $\mathbf{x}_i$  to a fixed position  $\mathbf{p}_{\text{fixed}}$ . The energy is given by  $\psi_{\text{pin}} = \frac{1}{2} \bar{\kappa}_i^{\text{pin}} \|\mathbf{x}_i - \mathbf{p}_{\text{fixed}}\|^2$ , where  $\bar{\kappa}_i^{\text{pin}}$  is a pin stiffness. In this case, geometric gap is defined as  $\hat{g}_i^{\text{pin}} - \|\mathbf{x}_i - \mathbf{p}_{\text{fixed}}\|$ , where  $\hat{g}_i^{\text{pin}}$  is a gap tolerance for the pin condition. Hence, we use it to define the pin stiffness as

$$\bar{\kappa}_i^{\text{pin}} = \frac{m_i}{(\hat{g}_i^{\text{pin}} - \|\mathbf{x}_i - \mathbf{p}_{\text{fixed}}\|)^2} + \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \cdot (H_{ii} \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}), \quad (6)$$

where  $\mathbf{w}_i = \mathbf{x}_i - \mathbf{p}_{\text{fixed}}$ .  $m_i$  is the mass of  $i$ -th vertex.  $H_{ii}$  is a  $3 \times 3$  matrix of  $i$ -th vertex selected from  $H$ . We update the pin stiffness when they are requested during computing its Hessian and forces. Similarly, we wish to push out vertex  $\mathbf{x}_i$  from walls. We use the same contact energy for this purpose except that we use the following stiffness

$$\bar{\kappa}_i^{\text{wall}} = \frac{m_i}{(g_i^{\text{wall}})^2} + \mathbf{n}_{\text{wall}} \cdot (H_{ii} \mathbf{n}_{\text{wall}}), \quad (7)$$

where  $\mathbf{n}_{\text{wall}}$  is the wall normal and  $g_i^{\text{wall}}$  is a gap tolerance for wall boundary conditions.

### 3.6 Strain Limiting

Energy for strain limiting per triangle is given by

$$\psi_{\text{SL}}(\sigma_1, \sigma_2) = \psi_{\text{SL1}}(\sigma_1) + \psi_{\text{SL1}}(\sigma_2), \quad (8a)$$

$$\psi_{\text{SL1}}(\sigma) = \psi_{\text{weak}}(1 + \tau + \hat{\epsilon} - \sigma, \hat{\epsilon}, \bar{\kappa}_{\text{SL}}), \quad (8b)$$

where  $\sigma_j$  is a function of  $\mathbf{x}$ , representing the singular values of a deformation gradient of  $F$  on the deformed state  $\mathbf{x}$ .  $\tau$  and  $\hat{\epsilon}$  are small constants. For all the examples we set  $\tau = \hat{\epsilon}$ . If we set  $\tau + \hat{\epsilon} = 0.05$ , strain is strictly enforced within 5% of stretch.  $\tau + \hat{\epsilon}$  is also found in CIPC [Li et al. 2021]. The stiffness for strain limiting, which we present below, depends on the singular values. Therefore, we first update the singular values, and then update the stiffness just before it is requested for computing strain limiting Hessians and forces.

The geometrical gap for strain limiting is  $1 + \tau + \hat{\epsilon} - \max(\sigma_1, \sigma_2)$ . Therefore, we define  $\bar{\kappa}_{\text{SL}}$  as

$$\bar{\kappa}_{\text{SL}} = \frac{m_{\text{face}}}{(1 + \tau + \hat{\epsilon} - \max(\sigma_1, \sigma_2))^2} + \mathbf{w}_r \cdot (H_{9 \times 9} \mathbf{w}_r). \quad (9)$$

Note that the right-hand side of (9) contains the variables  $\sigma_1$  and  $\sigma_2$ , but according to the stiffness evaluation rule, they are both semi-implicitly evaluated.  $\mathbf{w}_r$  is an extended relative direction which we compute as follows: let  $\mathbf{x}_{1\dots 3}$  be three vertices of a triangle and  $\mathbf{r}_j = \mathbf{x}_j - (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3)/3$ ,  $\mathbf{w}_r$  is then  $\mathbf{w}_r = [\mathbf{r}_1^T \mathbf{r}_2^T \mathbf{r}_3^T]^T$ .  $H_{9 \times 9}$  is a block matrix where each sub  $3 \times 3$  matrix is extracted from  $H_{jk}$ , where  $jk$  is a vertex pair chosen from the three vertices, including duplicates  $j = k$ . This  $H_{9 \times 9}$  matrix encodes the stiffness of the triangle, and the action of  $\mathbf{w}_r$  is used to measure the stiffness in the tangential plane.  $m_{\text{face}}$  is the triangle mass. Note that  $\mathbf{w}_r$  is not normalized for scaling consistency with singular values.

More specifically, singular values are independent of the mesh size and encode the relative stretch from the rest pose. Therefore, the denominator on the first term would have been scaled by a squared representative mesh length, because the gap is squared there. Here, we choose  $\|\mathbf{w}_r\|$  as the representative length. The normalized constraint direction is expressed as  $\mathbf{w}_r / \|\mathbf{w}_r\|$ , so the second term would have been  $\frac{\mathbf{w}_r}{\|\mathbf{w}_r\|} \cdot (H_{9 \times 9} \frac{\mathbf{w}_r}{\|\mathbf{w}_r\|}) = \frac{1}{\|\mathbf{w}_r\|^2} (\mathbf{w}_r \cdot (H_{9 \times 9} \mathbf{w}_r))$ . Rescaling by  $\|\mathbf{w}_r\|^2$  removes the denominator, leaving simply  $\mathbf{w}_r \cdot (H_{9 \times 9} \mathbf{w}_r)$ .

It should be noted that when strain limiting exists the system stiffness becomes much stiffer than contacts because small variation in the triangular mesh results in abrupt changes in the total energy for high resolution meshes. Nevertheless, we empirically find that the  $3 \times 3$  block Jacobi preconditioner was robust enough to handle such challenges present in our examples.

When embedding (8a) into (3), we need to compute first and second derivatives of (8a) with respect to a deformation gradient. Previous literature depends on invariants [Chen and Weber 2017; Kim and Eberle 2022; Panetta 2020] but it is not clear how to express arbitrary isotropic energies using invariants in the form of (8a), such as e.g.,  $\sum_k (\sigma_k - 1)^3$  and  $-\sum_k \log(\sigma_k)$ . One could realize it as shown by Li et al. [2022] but still needs to solve implicit equations. Other methods [Stomakhin et al. 2012; Xu et al. 2015] contain a degenerate case  $\sigma_1 = \sigma_2$ . To the best of our knowledge, explicit expressions remains unclear. To this end, we provide an explicit eigen system in the supplementary material.

### 3.7 Matrix Assembly

Our matrix assembly is based on CAMA [Tang et al. 2016], and as such we only present differences. We chose explicit matrix assembly over matrix-free methods because, despite its name, explicit assembly is more memory efficient. Interested readers are referred to our supplementary material for this reasoning. Implementation details are also available in the supplementary material.

The most memory consuming part in CAMA is the fill-in pass of matrix entries, which involves entry duplicates before being compressed. We minimize this by re-using contact pairs from the previous step, and use them as a cached index table. When assembling the contact matrix, we look for the cache first and if it exists, we atomically merge to the value associated with the cache. The result is an order of magnitude efficient memory saving. For more efficiency, we have three sets of matrices resembling a cache hierarchy;  $3 \times 3$  block diagonal, index-table fixed based on the mesh connectivity, and the contact matrices. When incrementally adding a matrix entry, we try former first and move on to the later ones if missed.

### 3.8 Frictional Contacts

Friction is not our contribution, but it is important for practical contact simulations. For this purpose, we use the following quadratic potential

$$\psi_{\text{friction}} = \frac{1}{2} \bar{\kappa}_i^{\text{friction}} \|P_i W_i (\mathbf{x} - \mathbf{x}^{\text{prev}})\|^2, \quad (10)$$

$$\bar{\kappa}_i^{\text{friction}} = \mu f_i^{\text{contact}} / \max(\varepsilon, \|P_i W_i (\mathbf{x} - \mathbf{x}^{\text{prev}})\|), \quad (11)$$

where  $P_i$  is the projection operator  $P_i = \mathbf{I} - \mathbf{n}_i \mathbf{n}_i^T$  where  $\mathbf{n}_i$  is the contact normal.  $\mathbf{x}^{\text{prev}}$  is the mesh vertices from the previous time step.  $\mu$  and  $f_i^{\text{contact}}$  denote the friction coefficient and the contact force, respectively.  $\varepsilon$  is a small constant, which we set to  $\varepsilon = 0.01$  mm. This is needed because  $\mathbf{x} = \mathbf{x}^{\text{prev}}$  at the very first Newton step.  $f_i^{\text{contact}}$  is obtained when appending contact barriers to the system matrix; thus, it is the first derivative of the barrier potential. Note that scaling of the potential by either the mass or area should not be included because  $f_i^{\text{contact}}$  accounts for it.

Since the stiffness  $\bar{\kappa}_i^{\text{friction}}$  depends on the contact forces, we update the stiffness after the contact forces are computed. This stiffness value is then used to compute the Hessians and forces.

Readers may notice that the stiffness (eigenvalue) of the Hessian above becomes  $\mu f_i^{\text{contact}} / \varepsilon$  at the very first Newton step, which can be very high depending on the magnitude of  $\mu f_i^{\text{contact}}$ . If this issue must be addressed, one may cap  $\bar{\kappa}_i^{\text{friction}}$  by  $\partial^2 \psi_{\text{weak}} / \partial g_i^2$  (the maximal eigenvalue of the contact barrier) when computing the Hessian of the frictional potential while retaining the force itself intact. As we will show, a sufficient number of Newton's steps (which we suggest is 32) is needed for the resulting animations to closely match the specified friction coefficient  $\mu$ .

## 4 RESULTS

We use the notation <xxx.mp4> for videos in the supplementary material. We used a  $3 \times 3$  block diagonal-Jacobi PCG method with a tolerance of a relative residual of  $10^{-3}$  in the  $L_\infty$  norm. Our runtime

hardware is a single RTX4090 on a headless Ubuntu/Linux platform hosted by vast.ai<sup>1</sup>. The host side is implemented with Rust [Matsakis and Klock 2014] for safety reasons, and the device side is implemented with C/C++ CUDA [Cook 2012] and Eigen [Guennebaud et al. 2010].

The important numbers for Figure 1 are summarized in Table 1. The numbers for the videos shown in the insets are summarized in the supplementary material. We overlay dynamically changing numbers and timings in the video to let readers more objectively evaluate our work. The number of element counts, such as triangles and tetrahedra, is available in the video.

We eliminate duplicates in contact counts that arise in edge-edge and point-triangle queries. We detail how we achieve this in the supplementary material. We use the Stable Neo-Hookean model [Smith et al. 2018] for volumetric elasticity, the Wikin-Baraff model [Kim 2020] or ARAP [Stomakhin et al. 2013] for shells, and the Hooke spring [Liu et al. 2013] model for rods. Air dynamics is defined as a quadratic energy that penalizes differences between velocity and the ambient wind in the normal direction. This is detailed in the supplementary material.

All examples run in single precision, including device-side linear algebra operations in PCG. Some host-side operations (e.g.,  $\alpha$  and  $\beta$  coefficients in PCG) and pre-computations (e.g., mesh triangulation) are performed in double precision because they do not impact runtime efficiency or memory usage at all. All examples are explicitly checked intersection-free at the end of all the steps using an edge-triangle intersection test [Möller and Trumbore 2005]. Interested readers are referred to the code posted on our public GitHub repository<sup>2</sup>. Note that we do perform edge-edge continuous collision detection (CCD), but edge-edge intersection checks are not carried out. Once a pass-through occurs, intersections cannot be detected at discrete timings.

#### Bending and Rod Stiffness.

We used variants of dihedral angle-based bending energies for both shells and rods [Grinspun et al. 2003] and applied Hooke's law for rod stretching energies. Bending coefficients and stiffness values are listed in the table on the right inset.

For shells, the energy is given by  $\sum_j \frac{1}{2} (\kappa_{\text{bend}}^{\text{shell}}) h_j l_j \theta_j^2$ , where  $\kappa_{\text{bend}}^{\text{shell}}$  and  $l_j$  are the shell bending stiffness and hinge length.  $h_j$  is the thickness, which is twice the gap distance. The angle  $\theta_j$  is the angle between the two normals incident to the hinge. The bending energy for rods is similarly given by  $\sum_j \frac{1}{2} m_j (\kappa_{\text{bend}}^{\text{rod}}) \theta_j^2$ , where  $\theta_j$  is the angle between the two edge normals incident to the vertex.  $m_j$  is a vertex mass. These energy derivatives are directly plugged into the global force vector, which is recorded in SI units. Rods do not have twisting energies. Strain limiting is not enforced for rod examples. Rod's stretching energy is given by  $\sum_j \frac{1}{2} \kappa_{\text{rod}} m_j^{\text{rod}} (l_j - l_{0,j})^2$ , where  $m_j^{\text{rod}}$  is a rod mass,  $l_j$

Scene	$\kappa_{\text{bend}}^{\text{shell}}$	$\kappa_{\text{bend}}^{\text{rod}}$
Fig 1A	500	
Fig 1B		$10^{-3}$
Fig 1C		5.0
Fig 1D	1000	
Fig 1E	40	
Fig 1G		$10^6$
Fig 1H	2.5	
§ 5.11	1.0	

<sup>1</sup><https://vast.ai>

<sup>2</sup><https://github.com/st-tech/ppf-contact-solver>

Table 1. Timing breakdown (formatted as Average of all / Average of 10 highest) and important scene numbers of our examples. #N refers to the average of Newton’s steps.  $\bar{\mu}, \bar{\rho}, \bar{\lambda}$  refer to Young’s modulus, density and the Poisson’s ratio. Numbers for the insets are provided in the supplementary material. Newton’s steps for Fig 1G are fixed.

Scene	Mat Assembly	PCG	Line Search	Time Per Step	Per Frame	Contact Count	Step Size
Fig 1A	19.27s (39.97s)	21.96s (59.63s)	3.06s (6.31s)	109s (263s)	194s (714s)	100.57M (168.35M)	10.0ms
Fig 1B	540ms (644ms)	334ms (502ms)	30ms (126ms)	13.13s (28.56s)	745s (2036s)	5.50M (6.04M)	5.0ms
Fig 1C	622ms (1.02s)	69ms (195ms)	41ms (78ms)	2.38s (5.32s)	247s (395s)	4.37M (6.49M)	1.0ms
Fig 1D	1.58s (3.70s)	212ms (1.45s)	260ms (521ms)	5.14s (42.18s)	184s (1206s)	5.81M (17.59M)	1.0ms
Fig 1E	19ms (52ms)	10ms (74ms)	1ms (6ms)	149ms (972ms)	3.30s (27.90s)	6.14K (17.05K)	1.0ms
Fig 1F	1.47s (4.88s)	1.56s (2.31s)	214ms (646ms)	10.62s (122s)	196s (704s)	1.34M (2.89M)	1.0ms
Fig 1G	21ms (31ms)	52ms (130ms)	1ms (4ms)	3.04s (5.41s)	5.04s (9.86s)	58.76K (93.14K)	10.0ms
Fig 1H	455ms (1.23s)	48ms (296ms)	34ms (132ms)	2.11s (10.88s)	77.35s (217s)	2.78M (5.63M)	1.0ms

Scene	#Vert	Shell ( $\bar{\mu}, \bar{\rho}$ (kg/m <sup>3</sup> ), $\bar{\lambda}$ )	Tet ( $\bar{\mu}, \bar{\rho}$ (kg/m <sup>3</sup> ), $\bar{\lambda}$ )	Mass Rat	Dimension (m)	$\hat{g}$	$\tau + \hat{\epsilon}$	#N
Fig 1A	8.21M	10 MPa, $10^3$ , 0.25	N/A	N/A	$5.33 \times 5.13 \times 5.18$	1.75mm	N/A	1.0
Fig 1B	2.72M	N/A	N/A	N/A	$2.50 \times 1.00 \times 1.00$	0.12mm	N/A	8.7
Fig 1C	2.25M	N/A	N/A	N/A	$0.50 \times 80.00 \times 0.50$	0.50mm	N/A	2.0
Fig 1D	1.71M	100 KPa, $10^3$ , 0.25	1 MPa, $2 \times 10^3$ , 0.35	1.10	$1.90 \times 179.65 \times 1.40$	1.00mm	5%	1.4
Fig 1E	56.47K	1 MPa, $10^3$ , 0.25	N/A	N/A	$4.76 \times 1.29 \times 0.86$	3.00mm	10%	2.0
Fig 1F	6.40M	N/A	150 KPa, $10^2$ , 0.45	N/A	$2.05 \times 2.05 \times 2.05$	0.25mm	N/A	1.6
Fig 1G	43.54K	30 MPa, $10^3$ , 0.25	500 KPa, $10^3$ , 0.35	5.80	$8.04 \times 5.44 \times 1.00$	1.00mm	5%	32
Fig 1H	600.77K	100 KPa, $10^3$ , 0.25	25 MPa, $5 \times 10^4$ , 0.35	41.24	$2.83 \times 1.60 \times 2.83$	1.00mm	5%	2.8

is the current edge length and  $l_{0,j}$  is the  $j$ -th rest edge length. Rod stretch stiffness  $\kappa_{\text{rod}}$  is  $10^5$  for Figure 1B and  $10^3$  for Figure 1C.

#### 4.1 Noodle Bowl

Figure 1C shows an example with 2.25 million vertices in which 624 lengthy rods are poured into an invisible hemispherical bowl, totaling 6.49 million contacts. After all the rods are poured into the container, they settle into a static equilibrium state without freezing [Daviet et al. 2011; Erleben 2004; Schmidl and Milenkovic 2004]. Each video frame took 247s and the 2.0 Newton’s steps are consumed to advance a step. For rods, we applied an absolute offset thickness of 1.50mm to ensure they do not touch at their radii, as relying solely on our dynamic stiffness does not guarantee such absolute separation.

Also, to stabilize the horizontal positions of the noodles at the far top before being poured, we applied a weak artificial quadratic energy that penalizes horizontal displacements for vertices located outside the camera’s visibility.

#### 4.2 Woven Cylinder

Figure 1B shows a woven cylinder made of 1626 rods and 2.72 million vertices being twisted and buckled. This is particularly challenging because treatments that rely on explicit knowledge of yarn structures [Cirio et al. 2014, 2017] are not applicable due to distant yarn collisions. Since prior works depend on small steps to prevent pull-through [Kaldor et al. 2008, 2010; Yuksel et al. 2012], these methods are subject to self-intersections at arbitrarily large steps. Each video frame took 745s and the 8.7 Newton’s steps are consumed to advance a step. We have also applied an absolute offset of 1.50mm for

the same reason as described in §4.1. The rotational angular velocity is  $\pi$  radians per second.

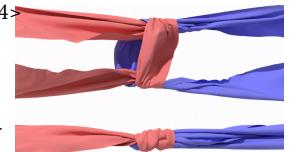
#### 4.3 Guided Fitting

The right inset <fitting.mp4> is an example staging a set of cloth patterns from Li et al. [2021] for guided fitting. To set up this configuration, we used a simple quadratic energy  $|g - \hat{g}|^2$  for stitch energy, omitting both the inertial term and gravity during staging. Each video frame took 891ms and the 2.1 Newton’s steps are consumed to advance a step. We used two step sizes:  $\Delta t = 1\text{ms}$  during staging and  $\Delta t = 10\text{ms}$  for the rest. Strain limiting is set to 10%. We tried a more severe strain limiting, but some amount of stretch was necessary to allow seams to be stitched together during staging.



#### 4.4 Reef and Fishing Knots

Figure 1E and the right inset <knot.mp4> illustrate two different types of knots being tightened, with the reef knot from Harmon et al. [2009]. In these examples, we used co-rotational distortion energy [Kim and Eberle 2022; Lan et al. 2022; Stomakhin et al. 2013] because the Baraff-Witkin model requires explicit knowledge of orthogonal bases. Each video frame took 3.30s for the fishing knot example and 2.29s for the reef knot. Average Newton’s steps are 2.0 for the fishing knot and 2.1 for the reef knot example.

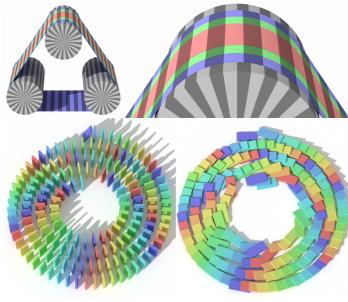


## 4.5 Squishy Balls

Figure 1F demonstrates a challenging collision scenario in which eight squishy balls from Zheng and James [2012] are tightly packed inside a spherical container and then released into the air. This example consists of 24.68 million tetrahedral elements and 6.40 million vertices. Each video frame took 196s and the 1.6 Newton’s steps are consumed to advance a step. Despite our method being semi-implicit, it is worth highlighting that during the compression of the squishy balls, they remain stable. Note that when the eight balls are released, we switched to ten-times slower slow-motion mode; otherwise, the balls would fly away too quickly.

## 4.6 Frictional Contacts

Figure 1G shows a collapsing house of cards ( $\mu = 0.4$ ). We highlight that towards the end of the animation, the entire scene comes to a stop, with static friction maintaining the delicate balance of the standing cards. The inset top <belt.mp4> features rotating cylinders, where three belts are tightly tensioned ( $\mu = 0.5$ ).



As the cylinders begin to rotate, each layer of the belt responds to the rotation beneath it, making all three belts to rotate together. When loosely tightened, as also seen in the same video, these straight patterns no longer hold.

The inset bottom <domino.mp4> is a domino cascade ( $\mu = 0.1$ ). After the entire domino collapse, a short period of equilibrium is reached, followed by an abrupt but slight sliding motion triggered by the transition from static friction to sliding friction.

As we will discuss, a sufficient number of Newton’s steps are needed for accurate frictional contacts. We used 32 steps and  $\Delta t = 10\text{ms}$  for the above three examples. This is illustrated in Algorithm in the inset. Average time for a video frame is 5.04s for the card example, 22.21s for the belt example and 10.78s for the domino example.

---

```

1: for  $n_{\text{iter}} = 1$  to 32 do
2:    $x, \alpha \leftarrow \text{inner\_step}(\Delta t, x)$ 
3:    $\beta \leftarrow \beta + (1 - \beta)\alpha$ 
4: end for
```

---

**4.6.1 Unresolved Frictional Contacts.** For some other examples in Figure 1, positive frictional coefficients  $0 \leq \mu \leq 1$  are provided, as listed in the table on the right inset. Unlisted examples simply have  $\mu = 0$ . In these examples, Figure 1G is the only one where frictional contacts are adequately resolved, as most of the others use only a few Newton’s steps with smaller step sizes. For these unresolved cases, the frictional coefficients are not physical and need to be selected heuristically.

For example, in Figure 1B, we initially set  $\mu = 0$ , but it resulted in untangled woven rods. Therefore, we used the smallest coefficient to prevent this issue. We have also capped  $\bar{\kappa}_i^{\text{friction}}$  by  $\partial^2 \psi_{\text{weak}} / \partial g_i^2$

for safety reasons because we know contact forces can be enormous towards the end. We did not cap  $\bar{\kappa}_i^{\text{friction}}$  for other examples. Limitations on the frictional contacts are discussed in more detail in §6.

## 4.7 Ribbons and Stacked Sheets

Figure 1D demonstrates the presence of both weak and tight contacts with a strong impact, with a mass ratio of different co-dimensional objects being nearly one. In this example, lengthy ribbons are gently poured into a transparent hemispherical bowl. Once the ribbons have nearly settled, a sudden impact from a fast-falling sphere compresses the ribbons all the way down, making the sphere bounce on the pile of ribbons several times. Each video frame took 184s and the 1.4 Newton’s steps are consumed to advance a step.

A similar test is also performed in Figure 1H, where ten sheets slowly fall onto the ground until a fast-falling sphere strongly smashes them. In this example, the mass ratio difference between all the sheets and the sphere is 41.24. Each video frame took 77.35s and the 2.8 Newton’s steps are consumed to advance a step.

Since our method does not implicitly adjust contact stiffness, a scenario such as this, where multiple sheets are stacked and experience a sudden compressing impact, would pose a numerical challenge. Our method handles such scenarios without difficulty such as oscillatory instabilities or PCG divergence.

## 4.8 Highly Localized Strong Contact

The right inset <main.mp4> conducts a stress test inspired by Wang et al. [2023], involving five sheets gently draped on the tip of a needle and subjected to the impact of a falling sphere. In this setup, the sphere is 43.06 times heavier than the total sheets and the tip of the needle is represented with a single vertex. We also conducted another set of three tests in which five sheets fall at a high speed and are subject to a highly localized impact at the tip. This test set is composed of three initial speeds: 5m/s, 10m/s and 20m/s with a step size of 1ms.



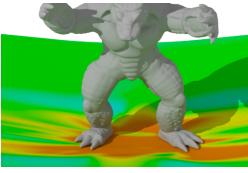
For all the tests in this section, we activated two-way coupled strain limiting with a strict upper bound of 5%, posing greater challenges than Wang et al. [2023]. As we will discuss in §5.1, our cubic barrier is essential for this stress test to pass; a logarithmic counterpart fails.

## 4.9 Five Cylinders Twisted

Figure 1A shows an example of five cylindrical sheets from Li et al. [2021], individually twisted and bundled, peaking at 168.35 million contacts. For such a significant number of contacts, our cached contact matrix assembly is indispensable; otherwise, the peak memory allocation does not fit within the 24 GB of device memory for the RTX4090. We highlight that despite such a high contact count, the running time per frame remains around 714s even at intense moments. It is also worth noting that the average time per frame was 194s. The vertex and face counts were 8.21 million and 16.41 million, respectively. Since the movements were slow, a single Newton step was sufficient for many time steps.

#### 4.10 Strain Limited Trampoline

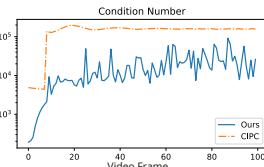
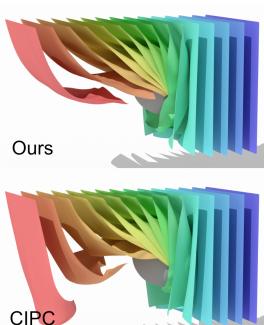
The right inset <main.mp4> demonstrates high tolerance for severe strain limiting with an upper bound of 1%, even when subjected to a high impact collision, such as an armadillo with a mass difference ratio of 32.91 hitting the sheet from the normal direction.



This setup is particularly challenging because a weakly coupled system may struggle to maintain a delicate balance sustained by highly sensitive forces of strain limiting and deformable dynamics. The initial falling speeds are 5m/s and 25m/s, respectively with  $\Delta t = 1\text{ms}$ .

#### 4.11 Comparisons

In the right inset <curtain.mp4> top, we hung 15 sheets subjected to undulation by a moving sphere with  $\Delta t = 10\text{ms}$  and a minimal of 8 Newton's steps. Additionally, in the bottom we simulated the same scene using a publicly available implementation<sup>3</sup> of CIPC. This shows that our inexact Newton's method produces animations similar to those of CIPC when a sufficient number of Newton's steps are selected.



We also plot the condition number of both methods in log-scale at the very first Newton's step as shown in the inset and observed that the CIPC yields larger ones than ours.

Note that, despite the issue of conditioning has been reported [Li et al. 2021] it is shown to work with a PCG [Huang et al. 2024]. Although our work is a variant of IPC, our results are more consistent with those of Huang et al. [2024] in that we did not encounter convergence issues arising from the use of logarithmic barriers, except for search direction locking. This indicates that the conditioning issues for contacts are present only in specific cases and due to algorithmic choices, and are generally not a problem in our method.

A runtime comparison for smaller scale examples is summarized in the inset <cipc-comparison.mp4>. The second column is either the numbers of Newton's steps for our method or CIPC. The third column represents the average time per frame. CIPC was run on an AMD Ryzen 7 5700X.

We emphasize that our method does not achieve exact gradient-vanishing convergence; therefore, this table should not be interpreted as a fair performance comparison to CIPC.

#Vert	#N	Time(s)
87.2K	8	1.29
	16	1.73
	32	3.01
	CIPC	212.47
23.5K	8	0.66
	16	0.85
	32	1.52
	CIPC	35.42
8.5K	8	0.49
	16	0.66
	32	1.14
	CIPC	8.70

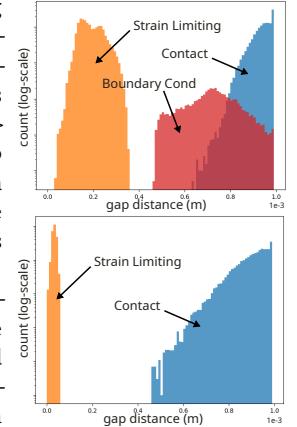
<sup>3</sup><https://github.com/ipc-sim/Codim-IPC>

For example, Wang et al. [2023] did not perform similar comparisons for fairness reasons (see §8 in their paper).

## 5 VALIDATIONS

### 5.1 Search Direction Locking

First, the search direction locking mostly emerges when strain limiting exists. A real-scale gap distribution for an example where five sheets are draped on a sphere <drape.mp4> is visualized in the right inset top and §5.11 in the right bottom, which shows that the strain limiting is the most stringent constraint and thus is sensitive to floating-point accuracy.



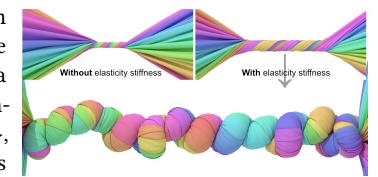
To clarify that the logarithmic barrier is a source of the issue, we have replaced  $\psi_{\text{weak}}$  with  $\psi_{\text{ln}}$  introduced in (2) while retaining other components. When we run the simulation with this replacement, examples of both §4.8 and §5.11 fail. Before the moment of failure, the minimal strain limiting gap fluctuates around an order of  $10^{-6}\text{m}$  (§4.8) or  $10^{-7}\text{m}$  (§5.11), and the condition number eventually turns to NaN (not measurable within single-precision).

This is because logarithmic barriers struggle to enlarge extremely small gaps within a limited budget of Newton's steps. We have also tried a variant of the logarithmic barrier [Huang et al. 2024], with curvature capped near the critical point as suggested, and confirmed the removal of the search direction locking. Instead, due to the sharp increase in the search direction, the line search more severely restricts the feasible step size. The above two behaviors are more clearly illustrated in the supplementary material.

An example of timing comparison for §4.8 is plotted in the inset. In this example, the running time of the quadratic stiffened logarithmic barrier can increase by up to two orders of magnitude longer than ours during challenging moments. Note that the original IPC integrator manages this by running the Newton's loop until exact convergence and solving the linear system via a direct solver in double precision, neither of which we employ. Hence, the issue should be less critical for CIPC.

### 5.2 Elasticity-Inclusive Stiffness

To validate the inclusion of an elasticity-involved term in the contact stiffness, we twisted a single cylinder with and without this term <twist.mp4>, thereby making the stiffness depend only on the squared inverse distance.



Without it, the contact gap quickly approaches its limit, increasing the system stiffness and the chance of collisions. Visually, this

manifests as an underestimated sheet thickness. One solution is to introduce offsetting to the gap [Li et al. 2021] as we did in the same video, but the runtime remains 7.3 times more expensive.

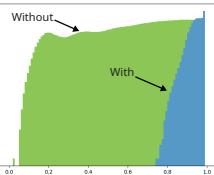
The contact gap distribution (with the y-axis log-scaled) is visualized in the inset.

### 5.3 Number of Newton's Steps

To validate that a small number of Newton's steps can still deliver visually satisfactory animations, we simulated Figure 1H with a minimum of eight Newton's steps, which tends to use more steps than one unspecified `<stack-8-newton.mp4>`.

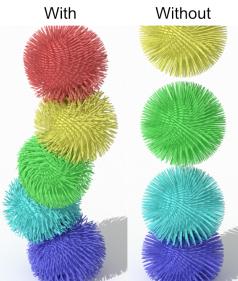
### 5.4 Quadratic Energy Function

To validate the use of cubic energy, we have replaced  $\psi_{\text{weak}}$  with a quadratic counterpart  $\psi_{\text{quad}}(g, \hat{g}, \kappa) = \frac{\kappa}{2\hat{g}}(g - \hat{g})^2$  if  $g < \hat{g}$  and 0 otherwise, which yields forces of the same order of magnitude as ours. This results in severe jitter artifacts, as seen in the inset `<drape.mp4>` because although the force is continuous at  $g = \hat{g}$ , the search direction is not. The search direction is what matters for positional updates, and our Newton's steps do not terminate at exact convergence.



### 5.5 Substep Scaled Newton's Method

To validate our step size amendment  $\beta$  in Algorithm 1, we compare results with and without it `<squishy.mp4>` shown in the right inset. Without  $\beta$  (that is, setting  $\beta = 1$ ), the resulting animation exhibits dragging artifacts.

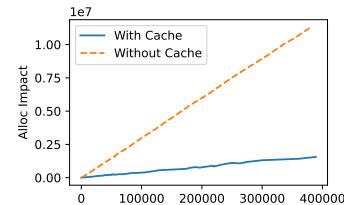


### 5.6 Strain Limiting Eigenanalysis

We provide a concise code in the supplementary material (`eigsys-1.py`) that runs on Google Colab to help readers numerically validate the correctness of our eigenanalysis of strain limiting.

### 5.7 Cached Matrix Assembly

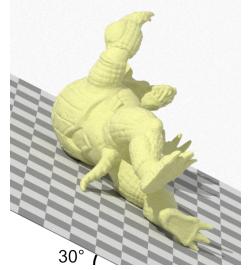
When a vanilla CAMA [Tang et al. 2016] is employed, the matrix entry counts before being merged can total to an order of magnitude greater than one being deduplicated. An example is shown in the right inset. For dozens of millions of contacts, this is important; otherwise, the GPU runs out of memory.



### 5.8 Resolved Frictional Contacts

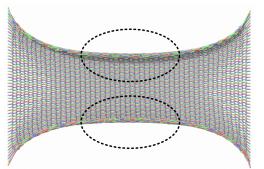
We placed a box on a  $30^\circ$  tilted slope `<friction-cube.mp4>`. When  $\mu < 0.5$ , the box slides but when  $\mu > 0.5$  it stays still. We also dropped a non-trivial deformable `<friction-armadillo.mp4>`.

When  $\mu = 0.51$  the armadillo slides down for several seconds but eventually stops as shown in the inset. For static friction, accuracy can be confirmed by closely examining the checkerboard pattern in the video. We set the Newton's steps to 32 and  $\Delta t = 10\text{ms}$  for the above two examples.



### 5.9 Yarn Curl

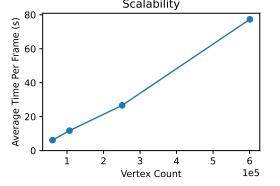
When contacts are resolved together with elasticity, it is known that a yarn-based sheet curls up when stretched. We confirmed this `<yarn.mp4>`.



### 5.10 Scalability

#### Curve By Vertex Count

We have performed an example from Figure 1H with increasing resolutions `<scalability.mp4>`. An inset on the right shows the scalability curve, which is closely linear to the vertex count.



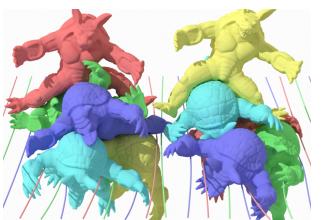
### 5.11 Locking-Free Wrinkles

Sheets are known to lock when a high stiffness is given to prevent stretch, leading to stiff-wrinkle artifacts, and strain limiting is often used to fix this. We have confirmed this in the right inset `<hang.mp4>`. This highlights the importance of strain limiting for realistic cloth simulation.



### 5.12 Wind Dynamics

An example of non-zero ambient wind is shown in `<wind.mp4>`. We confirm that five sheets flutter with periodic wrinkles that make frequent osculating contacts with other sheets.



### 5.13 Mixed-Codimensional Contacts

The right inset `<codim.mp4>` shows an example of different codimensional elements consisting of 32 rods and 1.67 million tetrahedral elements. Our method naturally handles such cases without special care.

## 6 DISCUSSION AND LIMITATIONS

**Accuracy.** Due to the proximal projection, our method exhibits minor fluctuations and does not reach the gradient-vanishing point as precisely as fully implicit schemes.

If this is important, one could achieve exact convergence by gradually diminishing the search directions as Newton's steps continue, similar to simulated annealing [Henderson et al. 2003], as we did in

<curtain.mp4> with a convergence criterion  $\|\mathbf{d}\|_\infty/\Delta t < 10^{-2}$  m/s, where  $\mathbf{d}$  is the search direction before being diminished. In other words, positions are updated by  $\gamma\mathbf{d}$ , where  $\gamma$  is a diminishing coefficient, and termination is decided by  $\mathbf{d}$ . We used the formula  $\gamma = \exp(-0.005 \times \#step)$ , but this is only experimental and not optimal. Due to the use of single-precision floating-point numbers, our achievable accuracy is lower than that of CIPC.

*Friction.* Our method does not resolve accurate friction with only a few Newton's steps. If this is important, running a sufficient number of Newton's steps is necessary, as we did in §4.6 and §5.8. Employing a colored Gauss-Seidel method [Chen et al. 2024; Fratarcangeli et al. 2016; Lan et al. 2023] and updating the friction potential every time a vertex is visited may help.

We note that when static friction is present, a contact should behave like a pin constraint. For this reason, larger steps are preferable; otherwise, any inaccuracy (mostly due to refreshing the inertial potential, which forgets where the original pin was) leads to unnoticeably small erroneous displacements that accumulate at each time step, resulting in noticeable frictional contact drift. For this matter, we chose  $\Delta t = 10$  ms for our frictional contact examples.

*Large Mass Differences.* Since we depend on a single-precision CG to invert the system, we empirically find that our method diverges when more than two orders of magnitude differences in density or stiffness are involved. A dedicated preconditioner, such as MAS [Wu et al. 2022], may help.

*Error Reduction Pass.* We included an error reduction pass at the end to ensure that we aim to minimize the correct total energy. However, the visual improvements are subtle in many cases. Therefore, if faster runtime is more important, this step could be omitted. An example without error reduction is shown in <stack-wo-err-red.mp4>.

*Logarithmic Barriers for Contact.* In our examples, logarithmic barriers with our elasticity-inclusive dynamic stiffness resolve contacts without failure because most gaps remain  $g/\hat{g} > 1/2$ . However, this does not mean that search direction locking is not an issue at all; for example, Huang et al. [2024] report that search direction locking leads to solver divergence for purely distance-based stiffness.

Also, we ran three examples without strain limiting: draping sheets on a sphere (§5.4), cylinder twisting (§5.2), and Fig.1H, all of which were compared using our cubic and logarithmic barriers <no-strain-limiting.mp4> shown in the inset. Here, numbers refer to the time per video frame. For the three cases, we observed nearly identical results, both in performance and visually.

Therefore, we conclude that our cubic barrier makes a difference only when strain limiting is present.

*Step-Sizes.* Our step-size reduced inexact Newton's method in §3.2 provides the best initial guess for purely ballistic motion and a poor guess when the inertia potential is not dominant. This implies that our initial guess may become less accurate when a large step is chosen, regardless of whether the system is invertible. For example, when we take a relatively large step size such as  $\Delta t = 100$  ms, the

animation tends to become stiffened. An example can be seen in <curtain-01-dt.mp4>.

We empirically find that a step size of from around 1ms to 10ms provides a good balance between accuracy and performance for many of our examples, except for frictional contact examples. This issue is specific to our method and does not exist in CIPC.

## 7 CONCLUSIONS

We focused on inequality constraints using barriers. Our method is based on IPC but diverges by devising a new cubic energy and a dynamic elasticity-inclusive stiffness, and opting for projective alternation for time integration. We applied our method to both contacts and strain limiting, which enables locking-free cloth simulation. We validated our method through stress tests and comparisons to logarithmic and quadratic barrier counterparts.

## ACKNOWLEDGMENTS

The author thanks the anonymous reviewers for their detailed feedback and helpful comments, which significantly improved the paper. The author also thanks Nobuyuki Umetani for sharing his expertise in the Finite Element Method and Ken Hayami for insightful discussions on the convergence of the conjugate gradient method.

Finally, the author would like to thank ZOZO, Inc. for allowing him to work on this topic as part of his main workload. The author also extends thanks to the teams in the IP department for permitting the publication of our technical work and the release of our code, as well as to many others for assisting with the internal paperwork required for publication.

## REFERENCES

- Jérémie Allard, François Faure, Hadrien Courtecuisse, Florent Falipou, Christian Duriez, and Paul G. Kry. 2010. Volume Contact Constraints at Arbitrary Resolution. *ACM Trans. Graph.* 29, 4, Article 82 (July 2010), 10 pages. <https://doi.org/10.1145/1778765.1778819>
- Sheldon Andrews, Kenny Erleben, and Zachary Ferguson. 2022. Contact and Friction Simulation for Computer Graphics. In *ACM SIGGRAPH 2022 Courses* (Vancouver, British Columbia, Canada) (*SIGGRAPH '22*). Association for Computing Machinery, New York, NY, USA, Article 3, 172 pages. <https://doi.org/10.1145/3532720.3535640>
- David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54. <https://doi.org/10.1145/280814.280821>
- David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling Cloth. *ACM Trans. Graph.* 22, 3 (July 2003), 862–870. <https://doi.org/10.1145/882262.882357>
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.* 33, 4, Article 154 (July 2014), 11 pages. <https://doi.org/10.1145/2601097.2601116>
- Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph.* 21, 3 (July 2002), 594–603. <https://doi.org/10.1145/566654.566623>
- Tyson Brochu, Essex Edwards, and Robert Bridson. 2012. Efficient Geometrically Exact Continuous Collision Detection. *ACM Trans. Graph.* 31, 4, Article 96 (July 2012), 7 pages. <https://doi.org/10.1145/2185520.2185592>
- Thomas Buffet, Damien Rohmer, Loïc Barthe, Laurence Boissieux, and Marie-Paule Cani. 2019. Implicit Untangling: A Robust Solution for Modeling Layered Clothing. *ACM Trans. Graph.* 38, 4, Article 120 (July 2019), 12 pages. <https://doi.org/10.1145/3306346.3323010>
- Anka He Chen, Ziheng Liu, Yang Yin, and Cem Yuksel. 2024. Vertex Block Descent. *ACM Transactions on Graphics (TOG)* 116 (2024).
- He Chen, Elie Diaz, and Cem Yuksel. 2023. Shortest Path to Boundary for Self-Intersecting Meshes. *ACM Trans. Graph.* 42, 4, Article 146 (Jul 2023), 15 pages. <https://doi.org/10.1145/3592136>

- Renjie Chen and Ofir Weber. 2017. GPU-Accelerated Locally Injective Shape Deformation. *ACM Trans. Graph.* 36, 6, Article 214 (November 2017), 13 pages. <https://doi.org/10.1145/3130800.3130843>
- Yunuo Chen, Minchen Li, Lei Lan, Hao Su, Yin Yang, and Chenfanfu Jiang. 2022. A Unified Newton Barrier Method for Multibody Dynamics. *ACM Trans. Graph.* 41, 4, Article 66 (jul 2022), 14 pages. <https://doi.org/10.1145/3528223.3530076>
- Franz Chouly, Mathieu Fabre, Patrick Hild, Rabii Mlika, Jérôme Pousin, and Yves Renard. 2017. An Overview of Recent Results on Nitsche's Method for Contact Problems. In *Geometrically Unfitted Finite Element Methods and Applications*, Stéphane P. A. Bordas, Erik Burman, Mats G. Larson, and Maxim A. Olshanskii (Eds.). Springer International Publishing, Cham, 93–141.
- Franz Chouly, Patrick Hild, Vanessa Lleras, and Yves Renard. 2022. Nitsche method for contact with Coulomb friction: Existence results for the static and dynamic finite element formulations. *J. Comput. Appl. Math.* 416 (2022), 114557. <https://doi.org/10.1016/j.cam.2022.114557>
- Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014. Yarn-Level Simulation of Woven Cloth. *ACM Trans. Graph.* 33, 6, Article 207 (November 2014), 11 pages. <https://doi.org/10.1145/2661229.2661279>
- Gabriel Cirio, Jorge Lopez-Moreno, and Miguel A. Otaduy. 2017. Yarn-Level Cloth Simulation with Sliding Persistent Contacts. *IEEE Trans. Vis. Comput. Graph.* 23, 2 (2017), 1152–1162. <https://doi.org/10.1109/TVCG.2016.2592908>
- Shane Cook. 2012. *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. *ACM Trans. Graph.* 30, 6 (December 2011), 1–12. <https://doi.org/10.1145/2070781.2024173>
- Kenny Erleben. 2004. *Stable, Robust, and Versatile Multibody Dynamics Animation*. Ph.D. Dissertation, University of Copenhagen.
- François Faure, Sébastien Barbier, Jérémie Allard, and Florent Falipou. 2008. Image-Based Collision Detection and Response between Arbitrary Volume Objects. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Dublin, Ireland) (SCA '08). Eurographics Association, Goslar, DEU, 155–162.
- Xudong Feng, Huamin Wang, Yin Yang, and Weiwei Xu. 2024. Neural-Assisted Homogenization of Yarn-Level Cloth. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 80, 10 pages. <https://doi.org/10.1145/3641519.3657411>
- Marco Fratcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-Seidel Method for Stable Soft Body Dynamics. *ACM Trans. Graph.* 35, 6, Article 214 (November 2016), 9 pages. <https://doi.org/10.1145/2980179.2982437>
- Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient Simulation of Inextensible Cloth. *ACM Trans. Graph.* 26, 3 (July 2007), 49–es. <https://doi.org/10.1145/1276377.1276438>
- M. Griebel and M. A. Schweitzer. 2003. *A Particle-Partition of Unity Method Part V: Boundary Conditions*. Springer Berlin Heidelberg, Berlin, Heidelberg, 519–542. [https://doi.org/10.1007/978-3-642-55627-2\\_27](https://doi.org/10.1007/978-3-642-55627-2_27)
- 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* (San Diego, California) (SCA '03). Eurographics Association, Goslar, DEU, 62–67.
- Gaël Guennebaud, Benoit Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- Tom Gustafsson, Rolf Stenberg, and Juha Videman. 2020. On Nitsche's method for elastic contact problems. arXiv:1902.09312 [math.NA]
- David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous Contact Mechanics. *ACM Trans. Graph.* 28, 3, Article 87 (July 2009), 12 pages. <https://doi.org/10.1145/1531326.1531393>
- David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust Treatment of Simultaneous Collisions. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–4. <https://doi.org/10.1145/1360612.1360622>
- Darrall Henderson, Sheldon H. Jacobson, and Alan W. Johnson. 2003. *The theory and practice of simulated annealing*. Internat. Ser. Oper. Res. Management Sci., Vol. 57. Kluwer Acad. Publ., Boston, MA, 287–319. [https://doi.org/10.1007/0-306-48056-5\\_10](https://doi.org/10.1007/0-306-48056-5_10)
- Kemeng Huang, Floyd Chitalu, Huancheng Lin, and Taku Komura. 2024. GIPC: Fast and stable Gauss-Newton optimization of IPC barrier energy. arXiv:2308.09400 [cs.GR]
- S. Huh, D.N. Metaxas, and N.I. Badler. 2001. Collision resolutions in cloth simulation. In *Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No.01TH8596)*, 122–127. <https://doi.org/10.1109/CA.2001.982385>
- Ning Jin, Wenlong Lu, Zhenglin Geng, and Ronald P. Fedkiw. 2017. Inequality Cloth. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) (SCA '17). Association for Computing Machinery, New York, NY, USA, Article 16, 10 pages. <https://doi.org/10.1145/3099564.3099568>
- Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2008. Simulating Knitted Cloth at the Yarn Level. *ACM Trans. Graph.* 27, 3 (August 2008), 1–9. <https://doi.org/10.1145/1360612.1360664>
- Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2010. Efficient Yarn-Based Cloth with Adaptive Contact Linearization. *ACM Trans. Graph.* 29, 4, Article 105 (July 2010), 10 pages. <https://doi.org/10.1145/1778765.1778842>
- Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. 2005. Fast Frictional Dynamics for Rigid Bodies. *ACM Trans. Graph.* 24, 3 (July 2005), 946–956. <https://doi.org/10.1145/1073204.1073295>
- Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Trans. Graph.* 27, 5, Article 164 (December 2008), 11 pages. <https://doi.org/10.1145/1409060.1409117>
- Theodore Kim. 2020. *A Finite Element Formulation of Baraff-Witkin Cloth*. Eurographics Association, Goslar, DEU. <https://doi.org/10.1111/cgf.14111>
- Theodore Kim and David Eberle. 2022. Dynamic Deformables: Implementation and Production Practicabilities (Now with Code!). In *ACM SIGGRAPH 2022 Courses* (Vancouver, British Columbia, Canada) (SIGGRAPH '22). Association for Computing Machinery, New York, NY, USA, Article 7, 259 pages. <https://doi.org/10.1145/3532720.3535628>
- Tae-Yong Kim, Nuttапong Chentanez, and Matthias Müller-Fischer. 2012. Long Range Attachments - a Method to Simulate Inextensible Clothing in Computer Games. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Lausanne, Switzerland) (SCA '12). Eurographics Association, Goslar, DEU, 305–310.
- Lei Lan, Minchen Li, Chenfanfu Jiang, Huamin Wang, and Yin Yang. 2023. Second-Order Stencil Descent for Interior-Point Hyperelasticity. *ACM Trans. Graph.* 42, 4, Article 108 (jul 2023), 16 pages. <https://doi.org/10.1145/3592104>
- Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022. Penetration-Free Projective Dynamics on the GPU. *ACM Trans. Graph.* 41, 4, Article 69 (jul 2022), 16 pages. <https://doi.org/10.1145/3528223.3530069>
- Dohae Lee, Hyun Kang, and In-Kwon Lee. 2023. ClothCombo: Modeling Inter-Cloth Interaction for Draping Multi-Layered Clothes. *ACM Trans. Graph.* 42, 6, Article 247 (dec 2023), 13 pages. <https://doi.org/10.1145/3618376>
- Chris Lewin. 2018. Cloth Self Collision with Predictive Contacts. <https://api.semanticscholar.org/CorpusID:202704112>
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panizzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Trans. Graph. (SIGGRAPH)* 39, 4, Article 49 (2020).
- Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 170 (2021).
- Huancheng Lin, Floyd M. Chitalu, and Taku Komura. 2022. Isotropic ARAP Energy Using Cauchy-Green Invariants. *ACM Trans. Graph.* 41, 6, Article 275 (nov 2022), 14 pages. <https://doi.org/10.1145/3550454.3555507>
- Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-Spring Systems. *ACM Trans. Graph.* 32, 6, Article 214 (November 2013), 7 pages. <https://doi.org/10.1145/298363.2508406>
- M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and T. Y. Kim. 2020. *Primal/Dual Descent Methods for Dynamics*. Eurographics Association, Goslar, DEU. <https://doi.org/10.1111/cgf.14104>
- Miles Macklin, Matthias Müller, and Nuttапong Chentanez. 2016. XPBD: Position-Based Simulation of Compliant Constrained Dynamics. In *Proceedings of the 9th International Conference on Motion in Games* (Burlingame, California) (MIG '16). Association for Computing Machinery, New York, NY, USA, 49–54. <https://doi.org/10.1145/2994258.2994272>
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-Based Elastic Materials. *ACM Trans. Graph.* 30, 4, Article 72 (July 2011), 8 pages. <https://doi.org/10.1145/2010324.1964967>
- Nicholas D. Matsakis and Felix S. Klock. 2014. The Rust Language. *Ada Lett.* 34, 3 (oct 2014), 103–104. <https://doi.org/10.1145/2692956.2663188>
- Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (July 2011), 12 pages. <https://doi.org/10.1145/2010324.1964932>
- Tomas Möller and Ben Trumbore. 2005. Fast, minimum storage ray/triangle intersection. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) (SIGGRAPH '05). Association for Computing Machinery, New York, NY, USA, 7–es. <https://doi.org/10.1145/1198555.1198746>
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118. <https://doi.org/10.1016/j.jvcir.2007.01.005>
- Matthias Müller, Tae-Yong Kim, and Nuttапong Chentanez. 2012. Fast Simulation of Inextensible Hair and Fur. In *Workshop on Virtual Reality Interaction and Physical Simulation*, Jan Bender, Arjan Kuijper, Dieter W. Fellner, and Eric Guerin (Eds.). The Eurographics Association. <https://doi.org/10.2312/PE/vrphys/vrphys12/039-044>
- Matthew Overby, George E. Brown, Jie Li, and Rahul Narain. 2017. ADMM  $\supseteq$  Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Trans. Vis. Comput. Graph.* 23, 10 (2017), 2222–2234. <https://doi.org/10.1109/TVCG.2017.2730875>
- Julian Panetta. 2020. Analytic Eigensystems for Isotropic Membrane Energies. arXiv:2008.10698 [math.NA]
- Xavier Provot. 1997. Collision and self-collision handling in cloth model dedicated to design garments. In *Eurographics*. Springer Vienna, 177–189. [https://doi.org/10.1007/978-3-7091-6874-5\\_13](https://doi.org/10.1007/978-3-7091-6874-5_13)

- Jessica D. Sanders, John E. Dolbow, and Tod A. Laursen. 2009. On methods for stabilizing constraints over enriched interfaces in elasticity. *Internat. J. Numer. Methods Engrg.* 78, 9 (2009), 1009–1036. <https://doi.org/10.1002/nme.2514> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2514>
- H. Schmidl and V.J. Milenkovic. 2004. A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics* 10, 2 (2004), 189–197. <https://doi.org/10.1109/TVCG.2004.1260770>
- Xing Shen, Runyuan Cai, Mengxiao Bi, and Tangjie Lv. 2024. Preconditioned Nonlinear Conjugate Gradient Method for Real-time Interior-point Hyperelasticity. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 96, 11 pages. <https://doi.org/10.1145/3641519.3657490>
- Zhongwei Shen, Jin Huang, Wei Chen, and Hujun Bao. 2015. Geometrically Exact Simulation of Inextensible Ribbon. *Computer Graphics Forum* 34, 7 (2015), 145–154. <https://doi.org/10.1111/cgf.12753> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12753>
- Alvin Shi and Theodore Kim. 2023. A Unified Analysis of Penalty-Based Collision Energies. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 41 (aug 2023), 19 pages. <https://doi.org/10.1145/3606934>
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (March 2018), 15 pages. <https://doi.org/10.1145/3180491>
- Georg Sperl, Rahul Narain, and Chris Wojtan. 2020. Homogenized Yarn-Level Cloth. *ACM Trans. Graph.* 39, 4, Article 48 (July 2020), 16 pages. <https://doi.org/10.1145/3386569.3392412>
- Georg Sperl, Rahul Narain, and Chris Wojtan. 2021. Mechanics-Aware Deformation of Yarn Pattern Geometry. *ACM Trans. Graph.* 40, 4, Article 168 (July 2021), 11 pages. <https://doi.org/10.1145/3450626.3459816>
- Alexey Stomakhin, Russell Howes, Craig Schroeder, and Joseph M. Teran. 2012. Energetically Consistent Invertible Elasticity. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Lausanne, Switzerland) (SCA '12). Eurographics Association, Goslar, DEU, 25–32.
- Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. Graph.* 32, 4, Article 102 (July 2013), 10 pages. <https://doi.org/10.1145/2461912.2461948>
- Shinjiro Sueda, Garrett L. Jones, David I. W. Levin, and Dinesh K. Pai. 2011. Large-Scale Dynamic Simulation of Highly Constrained Strands. *ACM Trans. Graph.* 30, 4, Article 39 (July 2011), 10 pages. <https://doi.org/10.1145/2010324.1964934>
- Tetsuya Takahashi and Christopher Batty. 2021. FrictionalMonolith: A Monolithic Optimization-Based Approach for Granular Flow with Contact-Aware Rigid-Body Coupling. *ACM Trans. Graph.* 40, 6, Article 206 (dec 2021), 20 pages. <https://doi.org/10.1145/3478513.3480539>
- Tetsuya Takahashi and Christopher Batty. 2022. ElastoMonolith: A Monolithic Optimization-Based Liquid Solver for Contact-Aware Elastic-Solid Coupling. *ACM Trans. Graph.* 41, 6, Article 255 (nov 2022), 19 pages. <https://doi.org/10.1145/3550454.3555474>
- Min Tang, Dinesh Manocha, Miguel A. Otaduy, and Ruofeng Tong. 2012. Continuous Penalty Forces. *ACM Trans. Graph.* 31, 4, Article 107 (jul 2012), 9 pages. <https://doi.org/10.1145/2185520.2185603>
- Min Tang, Huamin Wang, Le Tang, Ruofeng Tong, and Dinesh Manocha. 2016. CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation. *Computer Graphics Forum* 35, 2 (2016), 511–521. <https://doi.org/10.1111/cgf.12851> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12851>
- Min Tang, tongtong wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation. *37, 6, Article 204* (December 2018), 10 pages. <https://doi.org/10.1145/3272127.3275005>
- Bernhard Thomaszewski, Simon Pabst, and Wolfgang Straßer. 2009. Continuum-based Strain Limiting. *Computer Graphics Forum* 28, 2 (2009), 569–576. <https://doi.org/10.1111/j.1467-8659.2009.01397.x> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01397.x>
- Robin Tomicin, Dominik Sibbing, and Leif Kobbelt. 2014. Efficient enforcement of hard articulation constraints in the presence of closed loops and contacts. *Computer Graphics Forum* 33, 2 (2014), 235–244. <https://doi.org/10.1111/cgf.12322> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12322>
- Mickeal Verschoor and Andrei C. Jalba. 2019. Efficient and Accurate Collision Response for Elastically Deformable Models. *ACM Trans. Graph.* 38, 2, Article 17 (March 2019), 20 pages. <https://doi.org/10.1145/3209887>
- Pascal Volino and Nadia Magnenat-Thalmann. 2006. Resolving Surface Collisions through Intersection Contour Minimization. *ACM Trans. Graph.* 25, 3 (July 2006), 1154–1159. <https://doi.org/10.1145/1141911.1142007>
- Bolun Wang, Zachary Ferguson, Xin Jiang, Marco Attene, Daniele Panozzo, and Teseo Schneider. 2022. Fast and Exact Root Parity for Continuous Collision Detection. *Computer Graphics Forum* 41, 2 (2022), 355–363. <https://doi.org/10.1111/cgf.14479> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14479>
- Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, and Daniele Panozzo. 2021. A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection. *ACM Transactions on Graphics* 40, 5, Article 188 (Oct. 2021), 16 pages.
- Huamin Wang. 2014. Defending Continuous Collision Detection against Errors. *ACM Trans. Graph.* 33, 4, Article 122 (July 2014), 10 pages. <https://doi.org/10.1145/2601097.2601114>
- Huamin Wang, James O'Brien, and Ravi Ramamoorthi. 2010. Multi-Resolution Isotropic Strain Limiting. *ACM Trans. Graph.* 29, 6, Article 156 (December 2010), 10 pages. <https://doi.org/10.1145/1882261.1866182>
- Tianyu Wang, Jiong Chen, Dongping Li, Xiaowei Liu, Huamin Wang, and Kun Zhou. 2023. Fast GPU-Based Two-Way Continuous Collision Handling. *ACM Trans. Graph.* 42, 5, Article 167 (jul 2023), 15 pages. <https://doi.org/10.1145/3604551>
- Jungdam Won and Jehee Lee. 2019. Learning Body Shape Variation in Physics-Based Characters. *ACM Trans. Graph.* 38, 6, Article 207 (November 2019), 12 pages. <https://doi.org/10.1145/3355089.3356499>
- Botao Wu, Zhendong Wang, and Huamin Wang. 2022. A GPU-Based Multilevel Additive Schwarz Preconditioner for Cloth and Deformable Body Simulation. *ACM Trans. Graph.* 41, 4, Article 63 (jul 2022), 14 pages. <https://doi.org/10.1145/3528223.3530085>
- Longhua Wu, Botao Wu, Yin Yang, and Huamin Wang. 2020. A Safe and Fast Repulsion Method for GPU-based Cloth Self Collisions. *ACM Trans. Graph.* 40, 1 (2020), 5:1–5:18. <https://doi.org/10.1145/3430025>
- Hongyi Xu, Fushning Sin, Yufeng Zhu, and Jernej Barbič. 2015. Nonlinear Material Design Using Principal Stretches. *ACM Trans. Graph.* 34, 4, Article 75 (July 2015), 11 pages. <https://doi.org/10.1145/2766917>
- Juntao Ye, Guanghui Ma, Liguo Jiang, Lan Chen, Jituo Li, Gang Xiong, Xiaopeng Zhang, and Min Tang. 2017. A Unified Cloth Untangling Framework Through Discrete Collision Detection. *Computer Graphics Forum* 36, 7 (2017), 217–228. <https://doi.org/10.1111/cgf.13287> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13287>
- Cem Yuskel, Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2012. Stitch Meshes for Modeling Knitted Clothing with Yarn-Level Detail. *ACM Trans. Graph.* 31, 4, Article 37 (July 2012), 12 pages. <https://doi.org/10.1145/2185520.2185533>
- Chongyao Zhao, Jinkeng Lin, Tianyu Wang, Hujun Bao, and Jin Huang. 2022b. Efficient and Stable Simulation of Inextensible Cosserat Rods by a Compact Representation. *Computer Graphics Forum* 41, 7 (2022). <https://doi.org/10.1111/cgf.14701>
- Yidong Zhao, Jinyun Choo, Yupeng Jiang, Minchen Li, Chenfanfu Jiang, and Kenichi Soga. 2022a. A barrier method for frictional contact on embedded interfaces. *Computer Methods in Applied Mechanics and Engineering* 393 (apr 2022), 114820. <https://doi.org/10.1016/j.cma.2022.114820>
- Changxi Zheng and Doug L. James. 2012. Energy-based Self-Collision Culling for Arbitrary Mesh Deformations. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)* 31, 4 (Aug. 2012). <http://www.cs.cornell.edu/projects/escc>

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009