A Unified Newton Barrier Method for Multibody Dynamics

YUNUO CHEN*, University of California, Los Angeles, USA
MINCHEN LI*, University of California, Los Angeles & TimeStep Inc., USA
LEI LAN, Clemson University & University of Utah, USA
HAO SU, University of California, San Diego, USA
YIN YANG, Clemson University, University of Utah & TimeStep Inc., USA
CHENFANFU JIANG, University of California, Los Angeles & TimeStep Inc., USA

Fig. 1. **"Lying Flat".** Our primal Newton Barrier Method provides a unified simulation framework for this complex multibody system, consisting of a hammock made with rigid (stiff) rings, neo-Hookean deformable tori and circular rods, as well as an eleven-part articulated ragdoll and a piece of cloth. Accurate resolutions of the frictional contact and articulation constraints are guaranteed robustly, stably, and efficiently.

We present a simulation framework for multibody dynamics via a universal variational integration. Our method naturally supports mixed rigid-deformables and mixed codimensional geometries, while providing guaranteed numerical convergence and accurate resolution of contact, friction, and a wide range of articulation constraints. We unify (1) the treatment of simulation degrees of freedom for rigid and soft bodies by formulating them both in terms of Lagrangian nodal displacements, (2) the handling of general linear equality joint constraints through an efficient change-of-variable strategy, (3) the enforcement of nonlinear articulation constraints based on novel distance potential energies, (4) the resolution of frictional contact between mixed dimensions and bodies with a variational Incremental Potential Contact formulation, and (5) the modeling of generalized restitution through semi-implicit Rayleigh damping. We conduct extensive unit tests and benchmark studies to demonstrate the efficacy of our method.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Optimization-based Time Integration, Multibody Simulation, Frictional Contact

ACM Reference Format:
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 (July 2022), 14 pages. https://doi.org/10.1145/3528223.3530076

1 INTRODUCTION

Multibody dynamics plays an important role in both computer graphics and robotics. It is thus a long term research goal for researchers to develop more robust, controllable, accurate, and efficient multibody simulators. In a practical scene, we would have both rigid and deformable objects interacting with each other. Additionally, some of them contain articulated joints, and some others are codimensional thin manifolds. Complex scenes with all of them coupled together are essential ingredients of modern films, games,

*Equal contribution

Authors' addresses: Yunuo Chen, University of California, Los Angeles, USA, yunuoch@gmail.com; Minchen Li, University of California, Los Angeles & TimeStep Inc., USA, minchernl@gmail.com; Lei Lan, Clemson University & University of Utah, USA, lan6@clemson.edu; Hao Su, University of California, San Diego, USA, haosu@eng.ucsd.edu; Yin Yang, Clemson University, University of Utah & TimeStep Inc., USA, yin5@clemson.edu; Chenfanfu Jiang, University of California, Los Angeles & TimeStep Inc., USA, chenfanfu.jiang@gmail.com.

0730-0301/2022/7-ART66 $15.00
https://doi.org/10.1145/3528223.3530076

and robotic training environments. Accordingly we identify critical merits of a modern multibody solver: (1) guaranteed algorithmic convergence, (2) high-quality results without artifacts, (3) strong numerical stability, and (4) intuitive, user-friendly, and reliable parameter tuning. Despite a lot of research efforts devoted, it remains an open challenge to achieve all of them in a unified solver.

Many existing multibody simulation methods focus on efficiency [Wang et al. 2019] and differentiability [Geilinger et al. 2020; Macklin et al. 2019]. A major difficulty arises when new materials or constraints are incorporated, and special treatments are often applied. This is primarily due to that rigid and deformable bodies have their degrees-of-freedom (DOFs) in very different spaces, and coupling them together requires carefully designed algorithms. In addition, none of the existing multibody simulators can provide guaranteed interpenetration-free results for complex colliding objects.

Recently, Li et al. [2020a] proposed a variational integrator based on the Incremental Potential Contact (IPC) for modeling nonlinear elastic solids with guaranteed non-interpenetration. Later, IPC is proven effective in modeling contact when simulating objects with arbitrary codimensions [Li et al. 2021] and rigid bodies in SE(3) space [Ferguson et al. 2021]. More recently, Lan et al. [2022] simulated rigid bodies as stiff affine bodies, so that the trajectory of the simulated objects are also piecewise-linear. Despite an increase of DOFs, simulating rigid bodies with affine coordinates simplifies and accelerates the computation of collision-heavy scenes due to the direct applicability of highly performance optimized linear continuous collision detection (CCD) algorithms. Furthermore, with Lagrangian displacement as the primary variables, hybridizing rigid and deformable objects becomes straightforward.

We extend Lan et al. [2022] to mixed-dimensional soft-rigid coupled multibody dynamics with formulations for a wide range of linear and nonlinear, equality and inequality constraints for articulation. To the best of our knowledge our method is the first multibody simulator with guaranteed resolution of collision, contact, friction, and articulation constraints under extremely large time step sizes.

In addition to collisions and joints, another important aspect for simulating multibodies (and particularly for rigid bodies) is the modeling of restitution. Contacts between rigid bodies are inherently inelastic and highly dissipative. Additional modeling of restitution is often necessary to achieve impacts with better or more controllable energy behaviors. This is typically achieved by directly modifying the velocity-based contact constraints with a coefficient of restitution $r = v_{\text{out}}/v_{\text{in}}$ in traditional LCP-based methods [Baraff 1989; Moreau 1985; Stewart 2000]. However, challenges still remain on making these methods free of drift or interpenetration, and with guaranteed convergence all at the same time [Smith et al. 2012].

Our method models contact, friction, and joints all with potential energies. Correspondingly we design studies to investigate restitution from the perspective of energy dissipation in time integration. The resulting system allows restitution to be directly controlled by the magnitude of energy dissipation for each individual constraint.

In summary, we propose a unified optimization time integration framework for rigid-deformable coupled multibody simulation, supporting geometries in arbitrary codimensions and all common articulation constraints. Our framework guarantees algorithmic convergence and constraint satisfaction while providing robust and efficient simulations with controllable restitution for contact and joint constraints. To achieve this, we design

- an efficient change-of-variable strategy for general linear equality constraints,
- a unified formulation based on unsigned distances for nonlinear articulation constraints, and
- a general restitution model based on high-order time integration and Rayleigh damping for arbitrary inequality constraints.

We validate the efficacy of our restitution model through experimental studies. Unit tests and complex simulations are conducted to demonstrate the robustness, efficiency, and wide applicability of our multibody simulation framework.

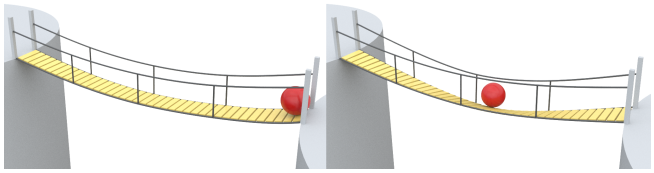


Fig. 2. **Hanging Bridge.** We drop a ball onto a hanging bridge constructed by 10 discrete elastic rods and 40 rigid boards connected via point connection constraints. The coupling between rigid bodies and codimensional deformable solids are robustly resolved.

2 RELATED WORK

2.1 Multibody Simulation

Multibody dynamics has long been a popular research topic in both computer graphics and robotics. From simulating a large number of rigid bodies, to coupling rigid and deformable solids, the key challenge is always on how to appropriately handle the contact among the simulated objects. Assuming the contacting points are known, rigid body simulation with non-interpenetration constraints can be formulated as a Linear Complementarity Problem (LCP), which dates back to Moreau [1985] and Baraff [1989]. Since then, existing works mostly focus on proposing new algorithms based on the LCP method to improve the efficiency and/or robustness.

Considering only rigid bodies, Guendelman et al. [2003] and Erleben [2007] applied shock propogation method to process contact in order following the contact graph. Kaufman et al. [2005] introduced a contact model that uses contact information at the moment of maximum compression to constrain rigid body velocities. Mazhar et al. [2015] applied Nesterov's method to accelerate solving the time integration system. Coevoet et al. [2020] adaptively merged rigid bodies with similar dynamics to reduce the number of DOFs.

Coupling rigid and deformable bodies, Shinar et al. [2008] proposed a multi-stage update that performs collisions, contacts, and stabilization in separate passes. Kaufman et al. [2008] formulated a discrete velocity-level frictional contact dynamics that reduces to a pair of coupled projections with a simple fixed-point property. Jain and Liu [2011] introduced a compact representation for an articulated character with deformable soft tissue, which can be simulated with two-way coupling in a robust and efficient manner. Bai and Liu [2014] provided a simple solution to cloth-rigid coupling using

existing cloth and rigid body simulators as-is. Li et al. [2020b] proposed a novel projective dynamics solver that takes full advantage of the pre-factorized system matrices to accelerate solving the coupling system. Some recent research also started to investigate the differentiability of the simulator, which can provide more efficient control solutions to robots [Geilinger et al. 2020; Macklin et al. 2019]. With a pure constraint view on physical modeling and a customized Gauss-Seidel solver, Position-based Dynamics methods [Deul et al. 2016; Frâncu and Moldoveanu 2017a,b; Müller et al. 2020] enable coupling rigid and deformable objects in a unified manner. However, similar to all the above works, none of them can guarantee algorithm convergence with interpenetration-free results.

By formulating noninterpenetration constraints using precise unsigned distances, and applying a customized barrier method, Li et al. [2020a] proposed the Incremental Potential Contact (IPC) model that can robustly simulate nonlinear elastodynamics with guaranteed noninterpenetration. IPC is then quickly proven effective on robustly handling contact for codimensional objects [Li et al. 2021], rigid bodies [Ferguson et al. 2021], reduced deformable bodies [Lan et al. 2021], and embedded interfaces [Zhao et al. 2022], etc. With Euclidean DOFs applied in rigid body simulation [Lan et al. 2022], rigid-deformable coupling with guarantees also becomes convenient based on IPC. We therefore construct our multibody simulator based on the optimization time integration framework with IPC.

2.2 Articulated Dynamics

In multibody systems, constraints are often applied to define various types of connections between a number of bodies, simplifying both geometric modeling and physical simulation for well-studied functional parts. Simulating such articulated systems is also of great interest to computational design and fabrication [Koyama et al. 2015; Thomaszewski et al. 2014]. With maximal or generalized coordinates, existing works on articulated dynamics also mainly focus on improving the efficiency of solving the constrained systems, which results in many different specialized treatments.

Redon et al. [2005] presented an adaptive algorithm that can simplify articulated body simulation to requested levels of detail. Mirtich et al. [1994] introduced an impulse-based method to handle contact between objects. Weinstein et al. [2006] applied the impulse-based approach to treat articulation and contact, where the complexity is linear both in the number of bodies and the number of contacts. Sueda et al. [2011] introduced a framework that combines Lagrangian and Eulerian approaches for robust, efficient, and accurate simulations of massively constrained systems. Xu et al. [2014] enables stable simulation of penalty-based frictional contact involving many articulated rigid objects. Tournier et al. [2015] and Andrews et al. [2017] proposed to use a geometric stiffness term to accelerate multibody simulation based on constraint projections. Enzenhöfer et al. [2019] presented a method based on the block Bard-type algorithm for efficiently factorizing the stiff systems of articulated dynamics. Peiret et al. [2019] presented a non-overlapping domain decomposition approach with Schur complement for solving stiff constrained multibody systems. Wang et al. [2019] proposed a near linear time approach to solve the linearly implicit equations of motion in multibody dynamics using a combined reduced/maximal coordinate formulation. Werling et al. [2021] built a fast and feature-complete differentiable physics engine supporting hard contact constraints for articulated rigid body simulation.

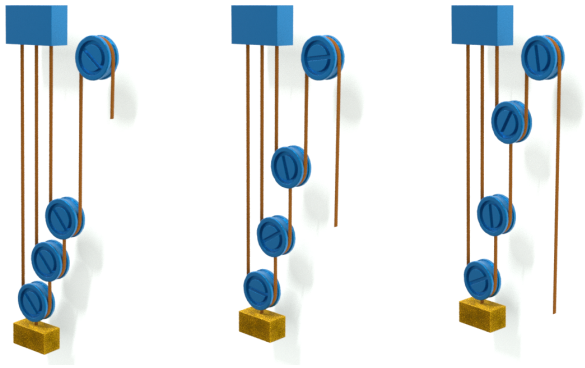


Fig. 3. **Pulley System.** We construct this pulley system with sliding constraints on the first 3 rigid pulleys, a hinge constraint on the rightmost pulley, and point connections on the discrete elastic rods and the other objects. From left to middle we show that our simulator accurately captures the mechanism of this pulley system when the right end of the rod is pulled all the way down with smaller forces to lift the heavy gold box.

Recently, Chang et al. [Chang et al. 2019] proposed a unified framework for simulating articulated mixed-dimensional objects. However, their framework specifically targets on conforming non-manifold mesh representations. Since we are performing fully implicit time integration via numerical optimization, we propose an efficient change-of-variable approach to handle all general linear equality constraints based on maximal coordinates. Then we handle all inequality and nonlinear equality articulation constraints with potential energies of unsigned distances.

2.3 Restitution Modeling

Inequality constraints in articulated systems are essentially simplifications of real-world structures, and most of these constraints are in fact due to contact locking between adjacent body parts, e.g., door hinges. Therefore, similar to contact, these inequality constraints can also exhibit restitution behaviors, which intuitively can be seen as the bounciness of the joint after a constraint is enforced. Existing works mainly focus on modeling restitution for contact constraints.

As a quantification of the restitution behavior, the coefficient of restitution has been defined in multiple ways. In Poisson's model [Routh 1905; Wang and Mason 1992], the restitution coefficient is defined as the ratio between the normal impulses in the decompression and compression phases, while in Newton's model [Whittaker 1937], it is defined as the ratio between the normal velocities in the two phases. There is also the Stronge's model [Mirtich 1996; Stronge 1991] which defines the coefficient of restitution based on the system energies before and after the impact. We refer to Ahmad et al. [2016] and Seifried et al. [2010] for detailed surveys on restitution research from a broad view.

In computer graphics, most methods model restitution following the Poisson's model [Guendelman et al. 2003; Kaufman et al. 2005; Popović et al. 2000] or the Newton's model [Choe et al. 2005; Twigg and James 2008], since they can be conveniently implemented with velocity variables. In position-based dynamics [Müller et al. 2007]

Fig. 4. **Umbrella.** We simulate an umbrella modeled by coupling a rigid skeleton with cloth. The skeleton is composed of 13 rigid thin rods, connecting to each other via point connection, hinge, and sliding constraints.

and projective dynamics [Bouaziz et al. 2014], velocities of colliding primitives are directly modified to resolve restitution behaviors. For more accurate and reliable restitution modeling, Smith et al. [2012] proposed a generalized restitution model that ensures breakaway in multi-impact problems and avoids convergence issues in inelastic collisions. In addition, Wang et al. [2017] proposed a method to enrich standard rigid-body impact models with a spatially varying coefficient of restitution map.

On the other hand, some methods also investigated restitution modeling in the energy view. Within the explicit time integration framework which preserves energy well, Harmon et al. [2009] proposed to reduce the contact force in the decompression phase to model inelastic collisons. Geilinger et al. [2020] and Bell et al. [2005] handles contact with a penalty energy on penetration depth. They applied high order time integration and damping models to achieve different coefficient of restitution. Our restitution is also modeled in the energy view, applying BDF-2 time integration to ensure low energy dissipation. Then we propose to use a semi-implicit Rayleigh damping model [Gast et al. 2015] for arbitrary inequality constraints handled by a barrier potential, achieving different coefficients of restitution with guaranteed constraint satisfaction.

3 COMPUTATIONAL FOUNDATIONS

We construct our multibody simulator within the optimization time integration framework for guaranteed robustness.

3.1 Incremental Potential Contact (IPC)

Following IPC [Li et al. 2020a], time integrating elastodynamics systems with frictional contact can be achieved via solving a nonlinear optimization problem

$$\boldsymbol{x}^{n+1} = \arg\min_{\boldsymbol{x}} \left(\frac{1}{2} \|\boldsymbol{x} - \tilde{\boldsymbol{x}}^n\|_{\boldsymbol{M}}^2 + \beta h^2 \sum_i P_i(\boldsymbol{x}) \right) \tag{1}$$

in each time step n for a new nodal position $\boldsymbol{x}^{n+1}$, followed by updating velocity $\boldsymbol{v}$ according to the time integration rule applied. Here $\boldsymbol{M}$ is the mass matrix; h is the time step size; $\tilde{\boldsymbol{x}}^n$ and β both depend on the time integration rule; and $P_i(x)$ are the potential energies for gravity, elasticity, contact, and friction, etc. The total energy being minimized is called the Incremental Potential [Li et al. 2019]. When implicit Euler is applied,

$$\boldsymbol{v}^{n+1} = \frac{1}{h}(\boldsymbol{x}^{n+1} - \boldsymbol{x}^n), \quad \tilde{\boldsymbol{x}}^n = \boldsymbol{x}^n + h\boldsymbol{v}^n, \quad \beta = 1, \tag{2}$$

while for BDF-2,

$$\begin{aligned} \boldsymbol{v}^{n+1} &= \frac{1}{2h}(3\boldsymbol{x}^{n+1} - 4\boldsymbol{x}^n + \boldsymbol{x}^{n-1}), \\ \tilde{\boldsymbol{x}}^n = \frac{1}{3}(4\boldsymbol{x}^n - \boldsymbol{x}^{n-1} + \frac{2h}{3}(4\boldsymbol{v}^n - \boldsymbol{v}^{n-1})), &\quad \beta = \frac{4}{9}. \end{aligned} \tag{3}$$

This framework naturally couples elastic solids of arbitrary codimensions with guaranteed algorithm convergence and noninterpenetration, which is simply achieved by introducing the constitutive models for shells and rods as other $P_i(\boldsymbol{x})$'s and ensuring contact among all boundary elements are processed with IPC [Li et al. 2021]. Highly stiff bodies like rigid or near-rigid objects can also be dealt with in this framework by directly hardening the Young's modulus. We could further leverage the fact that stiff bodies are hardly deformable to build a more compact representation and avoid expensive fullspace computations.

3.2 Affine Body Dynamics

Following Lan et al. [2022], we project the deformation freedoms into an affine space for a more realistic rigidity approximation of a perfectly-undeformable object. This is equivalent to using a single linear tetrahedral element to drive the dynamics of the body.

Specifically, we embed a stiff body in a virtual tetrahedron, whose four nodes are given as $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$. They are simply set as $\boldsymbol{X}_0 = (0,0,0)$, $\boldsymbol{X}_1 = (1,0,0)$, $\boldsymbol{X}_2 = (0,1,0)$, and $\boldsymbol{X}_3 = (0,0,1)$ in the material space. The position of any material point $\boldsymbol{Y_i} = (u_i, v_i, w_i)$ on the body can then be represented as:

$$Y_i = (1 - u_i - v_i - w_i)\boldsymbol{X}_0 + u_i\boldsymbol{X}_1 + v_i\boldsymbol{X}_2 + w_i\boldsymbol{X}_3, \tag{4}$$

or equivalently in matrix form, $\boldsymbol{Y} = \boldsymbol{P}\boldsymbol{X}$ with $\boldsymbol{P}_i \equiv \boldsymbol{Y_i}$. In the same way, the transformation matrix $\boldsymbol{P}$ also relates the affine coordinate $\boldsymbol{x} \in \mathbb{R}^{12}$ with the full coordinate $\boldsymbol{y}$ of all the vertices on the body. Once $\boldsymbol{P}$ is computed, we can advance the simulation with affine DOFs $\boldsymbol{x}$ and update the full DOFs at each time step using $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$.

Doing so not only reduces the total number of DOFs, but also simplifies the computation of the elasticity energy $\sum_e V_e \Psi(\boldsymbol{F}_e)$ summed over all the elements e with volume V_e and deformation gradient $\boldsymbol{F}_e$ on the body. This is because the linear tetrahedron has a uniform deformation gradient: $\boldsymbol{F}_e = \boldsymbol{F}$. Thus we have $\sum_e V_e \Psi(\boldsymbol{F}_e) = \Psi(\boldsymbol{F}) \sum_e V_e$, where $\boldsymbol{F}$ can be computed via $\boldsymbol{F} = [\boldsymbol{x}_1 - \boldsymbol{x}_0, \boldsymbol{x}_2 - \boldsymbol{x}_0, \boldsymbol{x}_3 - \boldsymbol{x}_0]\boldsymbol{D}^{-1}$. Here $\boldsymbol{D} = [\boldsymbol{X}_1 - \boldsymbol{X}_0, \boldsymbol{X}_2 - \boldsymbol{X}_0, \boldsymbol{X}_3 - \boldsymbol{X}_0] = \boldsymbol{I}$ is an identity matrix with our special choice of $\boldsymbol{X}$'s, and $\Psi(\boldsymbol{F}) = \frac{Y}{8}\|\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\|_{\mathrm{F}}^2$ with Young's modulus Y.

For other potentials $E(\boldsymbol{y})$, like the inertia, gravitational, contact, and frictional often formulated with full coordinates, we use the chain rule to evaluate the gradients and Hessians w.r.t. the affine coordinate during the optimization

$$\frac{\partial E(\boldsymbol{P}\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{P}^T \nabla E(\boldsymbol{P}\boldsymbol{x}), \quad \frac{\partial^2 E(\boldsymbol{P}\boldsymbol{x})}{\partial \boldsymbol{x}^2} = \boldsymbol{P}^T \nabla^2 E(\boldsymbol{P}\boldsymbol{x}) \boldsymbol{P}. \tag{5}$$

At first glance, this reduction still contains 6 redundant DOFs compared to rigid bodies in SE(3) [Ferguson et al. 2021]. However, keeping the trajectories piecewise linear rather than curved is a key insight for efficient and robust IPC-based rigid body simulation

[Lan et al. 2022]. In addition, the linear tetrahedron attached to each body also facilitates rigid-deformable simulations, allowing both to be formulated seamlessly in a unified framework.

4 CONSTRAINT FORMULATION

A straightforward way to handle articulation constraints is to form a nonlinear KKT system and solve it via sequential quadratic programming or similar iterative approaches as in prior works. However, it is challenging, if not impossible, to guarantee numerical convergence and stability for KKT solves. Instead, we convert the constrained optimization to an unconstrained one, in which linear equality constraints are directly handled via change of variables, while nonlinear and inequality ones are collectively modeled via (stiff) potential energies. This strategy also allows us to propose a general restitution model for arbitrary inequality constraints (see Section 5).

4.1 Linear Equality Constraints

Linear equality constraints are in the form of $\boldsymbol{c}_i^T \boldsymbol{x} = s_i$, for $i = 1, \cdots, m$. We require $m < n$ and all $\boldsymbol{c}_i$ being linearly independent. A full rank transformation matrix $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ can then be constructed, where $\boldsymbol{c}_i^T$ are the first m rows. The matrix $\boldsymbol{V}$ embodies another coordinate transformation of $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{z}$, for $\boldsymbol{U} = \boldsymbol{V}^{-1}$ since $\boldsymbol{V}$ is invertible. Similar to Eq. (5), we use chain rule to evaluate the potential gradient and Hessian w.r.t. $\boldsymbol{z}$:

$$\frac{\partial E(\boldsymbol{U}\boldsymbol{z})}{\partial \boldsymbol{z}} = \boldsymbol{U}^T \nabla E(\boldsymbol{U}\boldsymbol{z}), \quad \frac{\partial^2 E(\boldsymbol{U}\boldsymbol{z})}{\partial \boldsymbol{z}^2} = \boldsymbol{U}^T \nabla^2 E(\boldsymbol{U}\boldsymbol{z}) \boldsymbol{U}.$$

Under this new coordinate our m equality constraints now become $z_i = s_i$ as a set of Dirichlet boundary conditions. They can be enforced conveniently by setting the corresponding rows and columns in the gradient and Hessian to zero when computing search directions during the optimization.

Reduced Singular Value Decomposition and Gram-Schmidt process are two popular choices for building $\boldsymbol{V}$. However, these methods result in dense transformation basis, which induce unnecessary overheads of the Hessian computation and the linear solve during the optimization. Instead, we adopt an efficient basis extension method. Starting from the m-by-n constraint matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_m]^T$, we construct an upper-triangular matrix $\widetilde{\boldsymbol{C}}$ via Gaussian Elimination with row pivoting, so that the element with the largest absolute value in each row $\boldsymbol{c}_i$ is swapped to the diagonal. As long as constraints are linearly independent to each other, this process can be performed for all the m rows without failures. We then form the transformation matrix as $\boldsymbol{V} = \left[\widetilde{\boldsymbol{C}}^T, \widetilde{\boldsymbol{I}}^T\right]^T$, where the bottom $(n-m)$ rows are $\widetilde{\boldsymbol{I}} = \left[\boldsymbol{0}_{(n-m)\times m}, \boldsymbol{I}_{(n-m)\times(n-m)}\right]$. In other words, we insert an $(n-m)$-by-$(n-m)$ identity matrix at the bottom-right corner of $\boldsymbol{V}$ to obtain a full-rank upper-triangular matrix.

The structure of $\boldsymbol{V}$ allows an efficient back substitution for matrix multiplications involving $\boldsymbol{U} = \boldsymbol{V}^{-1}$. Next, starting from point connections, we show that a wide range of articulation joints can be formulated with linear equality constraints.

4.1.1 Point Connection. One common constraint in articulated dynamics is point-wise connections, i.e., gluing two points from different objects. Let $\{\boldsymbol{x}_1^i, \cdots, \boldsymbol{x}_{N^i}^i\}$ be the DOFs of body i. Here we use the superscript to denote the object index and the subscript for the local DOF index at each object (i.e., N^i gives the total number of DOFs at the i-th object). Any point on object i can be expressed as: $\boldsymbol{p}^i \equiv \sum_{j=1}^{N^i} \alpha_j \boldsymbol{x}_j^i$ with certain barycentric weights α. The point connection constraint between two bodies can be simply defined as:

$$\boldsymbol{p}^1 = \boldsymbol{p}^2. \tag{6}$$

Fixing a point throughout the simulation can also be conveniently enforced by fixing its world space position at its material space position:

$$\sum_{j=1}^{N} \alpha_j \boldsymbol{x}_j = \sum_{j=1}^{N} \alpha_j \boldsymbol{X}_j. \tag{7}$$

4.1.2 Hinge. A hinge constraint prescribes an axis around which objects could freely rotate. This is a nonlinear equality constraint in SE(3), and its specific form depends on the choice of the rotation parameterization. However as we do not explicitly enforce an SE(3) motion kinematically, but with a stiff penalty potential, this constraint becomes a linear one by connecting two pairs of points from the two objects (Figure 5).

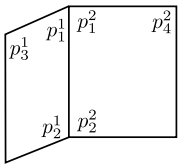


Fig. 5. **Hinge.**

$$\boldsymbol{p}_1^1 = \boldsymbol{p}_1^2, \quad \boldsymbol{p}_2^1 = \boldsymbol{p}_2^2. \tag{8}$$

This allows objects 1 and 2 to rotate around axis $\boldsymbol{p}_1^1 - \boldsymbol{p}_2^1$ freely. Oftentimes, the possible rotation range is also bounded (e.g., to prevent the penetration between the upper and lower arm). A bound constraint for the dihedral angle $\angle \boldsymbol{p}_4 \boldsymbol{p}_1 \boldsymbol{p}_3$ can also be applied as later discussed in Section 4.3.3.

4.1.3 Cone Twist. Cone twist is a gimbal-like structure where one can specify the rotation range in each axis. For each joint, let $\{\boldsymbol{n}_x, \boldsymbol{n}_y, \boldsymbol{n}_z\}$ be an orthogonal frame specified by the user in the material space, any rotation can be decomposed into 3 elementary rotations: bending around $\boldsymbol{n}_y$ axis, bending around $\boldsymbol{n}_z$ axis, and twisting around $\boldsymbol{n}_x$ axis.

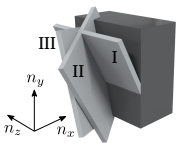


Fig. 6. **Cone Twist.**

The mapping from Euclidean DOFs to Euler angles involves costly trigonometric calculations. Alternatively, we design a special structure that enables intuitive control over each of the Euler angles. Specifically, we add three virtual links one by one for each cone twist joint – all using the hinge constraint for the corresponding rotation axis. As shown in Figure 6, the rotation axes for link I, II, and III are $\boldsymbol{n}_z$, $\boldsymbol{n}_y$ and $\boldsymbol{n}_x$ respectively. With our novel structure, the cone twist rotation is succinctly decomposed into three dihedral angles which supports independent control of the rotation ranges (again, see Section 4.3.3).

4.1.4 Sliding. When a sliding constraint limits the possible motion of a point $\boldsymbol{p}$ to be within a plane π, it requires that the displacement of the point is orthogonal to the plane's normal $\boldsymbol{n}$. If this constraint has been satisfied in previous steps, it becomes a position-level

linear constraint of

$$\boldsymbol{n}^T \sum_j^N \alpha_j (\boldsymbol{x}_j - \boldsymbol{X}_j) = 0. \tag{9}$$

If the sliding is more restrictive to be on a line, we simply use two planes π_1 and π_2 intersecting at the prescribed point trajectory, whose normals are $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$. This constraint can then be enforced by restricting the point to move within these two planes simultaneously. We next discuss the handling of nonlinear and inequality constraints, so that one can prescribe more complicated motion of $\boldsymbol{p}$ such as being on a line segment, a planar rectangular region, or along certain nonlinear trajectories.

4.2 Nonlinear Equality Constraints

Nonlinear equality constraints are essential in our maximal coordinate system when certain relations between objects are enforced in a hierarchical manner. It is also needed for constraints defined with distances. Let $c(\boldsymbol{x}) = 0$ be a general nonlinear equality constraint, we handle it with an additional potential energy

$$P_{\text{NEq}}(\boldsymbol{x}) = \frac{1}{2} k (c(\boldsymbol{x}))^2, \tag{10}$$

where k is a large stiffness that ensures the constraint is satisfied sufficiently.

4.2.1 Distance. We can maintain the distance between two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ by

$$c(\boldsymbol{x}) = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| - l = 0, \tag{11}$$

where l is the desired point-point distance. This distance constraint can be applied to any two points as long as they do not belong to the same stiff body, which will otherwise induce constraint redundancy.

4.2.2 Relative Sliding. In addition to the linear sliding constraint (Section 4.1.4), here we support point sliding relative to another simulated object. Specifically, such relative sliding constraint requires the point $\boldsymbol{p}_1^1$ from one object to move only along the direction of $\boldsymbol{p}_2^2 - \boldsymbol{p}_3^2$, two different points from another object. For simplicity, we require $\boldsymbol{p}_1^1, \boldsymbol{p}_2^2, \boldsymbol{p}_2^3$ to be collinear. This constraint can be viewed as maintaining a 0 area for triangle $\boldsymbol{p}_1^1 \boldsymbol{p}_2^2 \boldsymbol{p}_2^3$:

$$c(\boldsymbol{p}_1^1, \boldsymbol{p}_2^2, \boldsymbol{p}_3^2) = \|(\boldsymbol{p}_1^1 - \boldsymbol{p}_2^2) \times (\boldsymbol{p}_3^2 - \boldsymbol{p}_2^2)\| = 0. \tag{12}$$

4.3 Inequality Constraints

Inspired by Li et al. [2020a], we use barrier method to handle general inequality constraints $c(\boldsymbol{x}) \geq 0$ by incorporating an additional barrier term

$$P_{\text{Ineq}}(\boldsymbol{x}) = \kappa b(c(\boldsymbol{x})) \tag{13}$$

into our Incremental Potential. Here b is the barrier function [Li et al. 2020a]

$$b(c) = \begin{cases} -\left(\frac{c}{\hat{c}} - 1\right)^2 \ln\left(\frac{c}{\hat{c}}\right) & 0 < c < \hat{c} \\ 0 & c \geq \hat{c}, \end{cases} \tag{14}$$

which provides arbitrarily large repulsion to prevent $c(\boldsymbol{x})$ from being nonpositive, κ is the barrier stiffness, and $\hat{c}$ is a clamping threshold so that constraints faraway from being active are untouched.

Equipped with the barrier method, we formulate all the inequality constraints for articulated bodies in the form of bounded distances. This brings an intuitive and consistent formulation that processes all the constraints collectively in a unified manner.

4.3.1 Bounded Distance. To enforce an upper bound d_u on the distance between two points $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, we set

$$c(\boldsymbol{x}) = d_u - \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| \geq 0. \tag{15}$$

Similarly, lower bounds d_l can be enforced by simply setting

$$c(\boldsymbol{x}) = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| - d_l \geq 0. \tag{16}$$

Similar to the equality case (Section 4.2.1), the bounded distance can be applied to any two points that are not on the same rigid object.

4.3.2 Sliding Range. The range of sliding constraints (Section 4.1.4 and 4.2.2) can be enforced by additional bounded distance constraints. For example, to restrict point $\boldsymbol{p}$ to be within line segment $\boldsymbol{p}_1\boldsymbol{p}_2$, we can first introduce an additional point $\boldsymbol{p}_3 = 2\boldsymbol{p}_2 - \boldsymbol{p}_1$ so that $\boldsymbol{p}_2$ is the midpoint between $\boldsymbol{p}_1$ and $\boldsymbol{p}_3$, and then augment the system with two inequality constraints

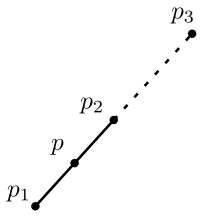


Fig. 7. **Sliding Range.**

$$c_1(\boldsymbol{x}) = \|\boldsymbol{p} - \boldsymbol{p}_3\| - \|\boldsymbol{p}_2 - \boldsymbol{p}_3\| \geq 0, \quad c_2(\boldsymbol{x}) = \|\boldsymbol{p}_1 - \boldsymbol{p}_3\| - \|\boldsymbol{p} - \boldsymbol{p}_3\| \geq 0. \tag{17}$$

4.3.3 Rotation Range. Rotation range can be applied to all rotations. Here we demonstrate how to bound the dihedral angle in a hinge constraint (Section 4.1.2). To restrict the rotation angle of point $\boldsymbol{p}$ within range $[\theta_{min}, \theta_{max}]$, we first rotate $\boldsymbol{p}_0$ (the point with zero degree) by θ_{max} and θ_{min} respectively to obtain $\boldsymbol{p}_{max}$ and $\boldsymbol{p}_{min}$. Then $\boldsymbol{p}$ must be restricted on arc $\boldsymbol{p}_{max}\boldsymbol{p}_{min}$. Let $\boldsymbol{p}_{avg}$ denote the point with angle $\theta_{avg} = \frac{1}{2}(\theta_{max} + \theta_{min})$, then $\boldsymbol{p}_{avg}$ is the midpoint of the arc. Therefore, we can avoid computing angles by expressing the rotation range constraint as a distance constraint:

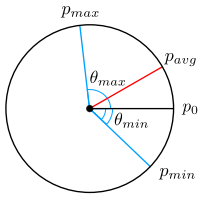


Fig. 8. **Rotation Range.**

$$c(\boldsymbol{x}) = \|\boldsymbol{p}_{max} - \boldsymbol{p}_{avg}\| - \|\boldsymbol{p} - \boldsymbol{p}_{avg}\| \geq 0. \tag{18}$$

Since cone twist (Section 4.1.3) can be decomposed into three hinge constraints, we can precisely bound each of the three angles.

5 RESTITUTION

Resolving inequality constraints with barrier potentials naturally includes the modeling of restitution. Taking contact modeled with IPC as an example, we can think of the barrier potential as the elastic potential of a virtual thin layer of material right outside the surface of every object. Then as two objects approach each other, the virtual elastic layer is compressed, accumulating and storing strain energies as kinetic energy decreases. Once the maximum compression is reached, the stored energy releases, pushing the colliding objects apart, regardless of whether the object is rigid or elastic (Figure 9).

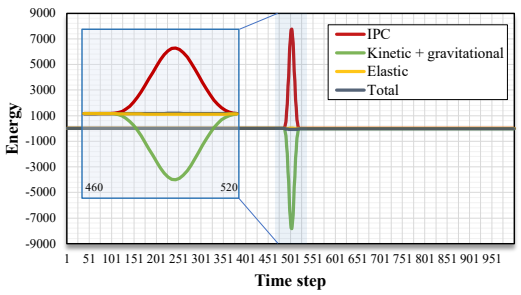


Fig. 9. **Energy Transitions.** Energy curve of a cube falling onto the ground under gravity with $h = 0.001s$. The cube hit the ground at $t = 0.5s$. We see that the yellow curve stays zero (even after zooming in), indicating accurately approximated rigid behaviors. The total energy is also well-preserved as shown by the gray curve, which is realized (as shown by the spike) by consistently transferring kinetic energies (green curve) into IPC potential (red curve) during the contact, and then the velocities are fully restored.

However, just like elasticity, this restitution behavior is significantly influenced by the numerical dissipation of the time integration scheme. For example, if implicit Euler is applied, using framerate time step sizes often tend to introduce a significant amount of numerical dissipation. Even with orders-of-magnitude smaller time step size at $h = 10^{-4}$, where time stepping becomes inefficient as much more time steps are needed, implicit Euler still damps out a non-negligible amount of energy (Figure 14). Therefore, we propose to use another unconditionally stable but 2nd-order accurate BDF-2 time integrator for drastically improved energy preservation.

Since BDF-2 is also a dissipative time integration scheme similar to implicit Euler, larger time step sizes can result in more numerical dissipation, providing less elastic bounces (Figure 14). However, purely relying on changing time step sizes for restitution control is neither intuitive nor convenient. Thus, we apply a semi-implicit Rayleigh damping model [Gast et al. 2015] to achieve robust and controllable inelastic restitution behaviors. When applied to barrier potentials for different constraints, our strategy supports separate control of each bounded articulation range.

5.1 Semi-Implicit Rayleigh Damping

For any potential energy $P(\boldsymbol{x})$, the corresponding Rayleigh damping force is

$$\boldsymbol{f}_d(\boldsymbol{x}) = -k_d \frac{\partial^2 P}{\partial \boldsymbol{x}^2}(\boldsymbol{x})\boldsymbol{v}(\boldsymbol{x}), \tag{19}$$

where $k > 0$ is the stiffness of the damping. Following Gast et al. [2015], to avoid calculating the 3rd-order derivative of $P(\boldsymbol{x})$, we discretize the stiffness matrix $\frac{\partial^2 P}{\partial \boldsymbol{x}^2}$ in the above definition to the last time step or Newton loop, just like the semi-implicit friction in IPC [Li et al. 2020a]. This let us define a damping potential

$$D(\boldsymbol{x}) = \frac{k_d}{2}\boldsymbol{v}(\boldsymbol{x})^T \left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right)^{-T} \frac{\partial^2 P}{\partial \boldsymbol{x}^2}(\boldsymbol{x}^n)\boldsymbol{v}(\boldsymbol{x}), \tag{20}$$

where its negative gradient generates a semi-implicit damping force

$$\boldsymbol{f}_d(\boldsymbol{x}) \approx -\frac{\partial D}{\partial \boldsymbol{x}}(\boldsymbol{x}) = -k_d \frac{\partial^2 P}{\partial \boldsymbol{x}^2}(\boldsymbol{x}^n)\boldsymbol{v}(\boldsymbol{x}). \tag{21}$$

Note that we project $\frac{\partial^2 P}{\partial \boldsymbol{x}^2}(\boldsymbol{x}^n)$ to positive semi-definite in the damping energy definition for valid damping behaviors [Gast et al. 2015] by zeroing out the negative Eigen values of every local stencil hessian [Liu et al. 2017]. The hessian of the damping energy is then

$$\frac{\partial^2 D}{\partial \boldsymbol{x}^2}(\boldsymbol{x}) = k_d \frac{\partial^2 P}{\partial \boldsymbol{x}^2}(\boldsymbol{x}^n)\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}. \tag{22}$$

Here we assumed linear velocity-displacement relation, so that $\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}$ is a constant multiple of the identity matrix, which is true for the linear (multi-)step time integration schemes applied in our system.

Since the semi-implicit Rayleigh damping applies to any potential energies, in our framework, it could be applied to model restitution for elasticity, contact, and all our inequality articulation constraints.

6 EVALUATION

Following Li [2020], we apply the projected Newton method to minimize the Incremental Potential (Eq.1) for solving the new configuration. We adopt the per-stencil Hessian projection [Teran et al. 2005] to ensure positive-definiteness, and so the solved search direction is always descent, which further enables backtracking line search for global convergence. Here the initial step size of line search is filtered by the additive continuous collision detection (ACCD) method [Li et al. 2021] to provide guaranteed resolution of inequality constraints, ensuring feasibility throughout.

We evaluate our method extensively by performing a unit test on each of our articulation constraints (Section 6.1), followed by a thorough analysis on our restitution model (Section 6.2). Then in Section 6.3 we demonstrate our method successfully simulate real-world complex scenes that require high accuracy and robustness to achieve the expected behavior, which are challenging for alternative simulators. The statistics of our hybrid CPU-GPU implementation is listed in Table 2, followed by a detailed performance analysis with timing breakdowns in Section 6.4. In our examples containing codimensional objects, discrete elastic shells and rods constitutive models are applied following C-IPC [Li et al. 2021].

6.1 Constraints Unit Test

Point Connections. Here we link three rods sequentially with two point connection joints (Figure 12 a). With the top-left point fixed at the initial location, this triple-rod pendulum is swinging with a chaotic behavior.

Another point connection constraint is shown in Figure 12 (a) with a piece of cloth connected to the middle point of a flexible coat rack formed by connecting five rods, demonstrating codimensional rigid-deformable coupling.

Hinge with Rotation Range. We demonstrate hinge constraints with three boards rotating around a shared shaft (Figure 12 b). The boards are hit by balls to start rotating, where from top to bottom the different boards has a decreasing range of bounded rotations visualized by copper sectors.

Bounded Sliding. We make a three-section monocular to demonstrate our bounded sliding constraints, where sliding ranges are set to prevent inner sections from detaching the outer ones (Figure 12 c). As observed in our experiments, using sliding constraint, this

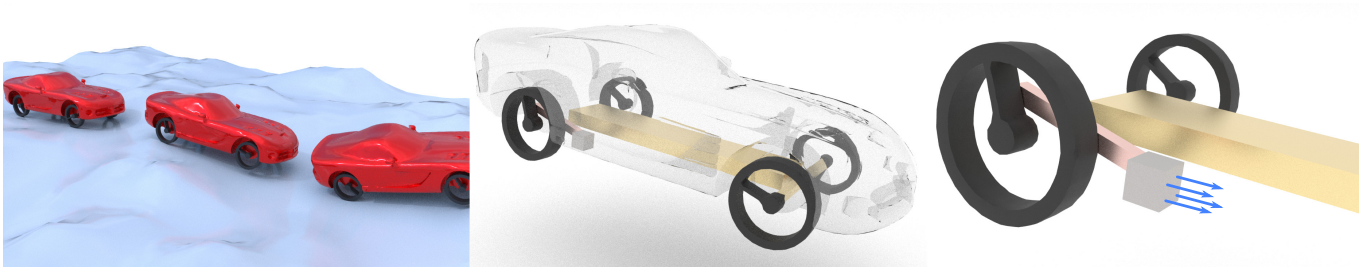


Fig. 10. **Terrain Navigation with Cylinder-Driven Vehicle.** We simulate a vehicle modeled with multiple rigid bodies driving on a terrain (left). The vehicle is triggered by a cylinder engine connected to the back-right wheel with all other wheels passive (middle and right). As our system accurately resolve the frictional contact, the vehicle can navigate the terrain smoothly as we pull the pistol with a constant force.

Fig. 11. **Ragdolls.** 60 ragdolls at random initial height are dropped on the ground without any interpenetrations (left and middle). Cone twist constraints are applied at each joint with specific rotation ranges imitating the mechanism of human bodies (right). For example, at the elbow, the rotation ranges for the two bending components are $[-135^\circ, 0^\circ]$ and $[-5^\circ, 5^\circ]$, and for twisting, $[-10^\circ, 10^\circ]$.

three-section monocular can be simulated 10× faster than directly resolving the internal contact of a real-world sliding structure.

Cone Twist. We demonstrate our cone twist constraints with a Nunchaku, connecting two thin cuboids at the end point with the ranges of the three rotations separately controlled (Figure 13). Here we move one section of the Nunchaku horizontally, vertically, and twist it around its central axis. The motion propagates to the other section with some delay as the rotation limits are reached.

Friction. We follow Li et al. [2021] to handle friction for codimensional objects, and verified our implementation through the same unit test. We place a $0.5m$-wide piece of square cloth on an inclined slope. The inclination angle is set at 26.565° so the critical friction coefficient of the slope is 0.5. For each time step we iteratively update the normal force magnitude and the tangent operators until convergence to a fully implicit solution. Our system obtains stiction with friction coefficient $\mu = 0.5$, and obtains sliding even at $\mu = 0.49$, which well-matches the real-world behaviors.

6.2 Restitution Study

To extensively study the behavior of our restitution model, we first list all related simulation parameters in Table 1, and then design experiments to analyze each of them in detail.

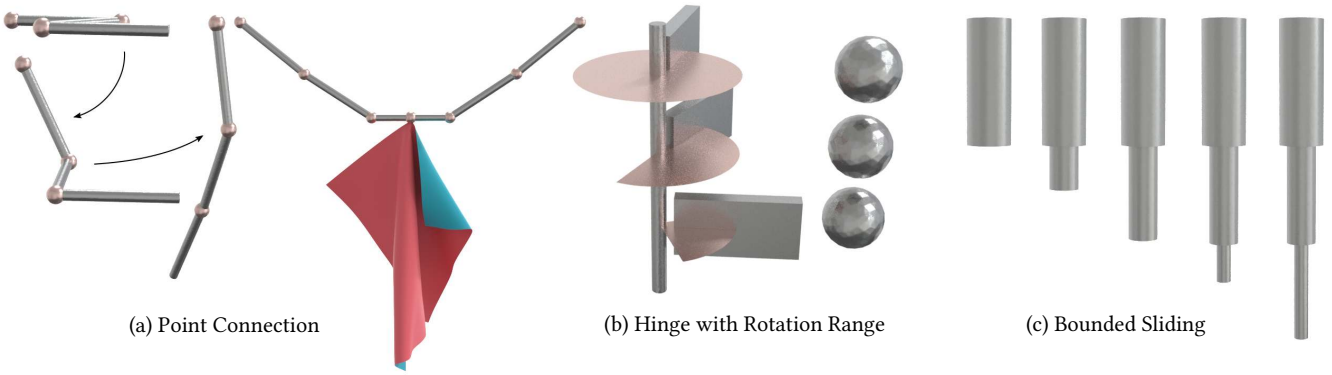
Table 1. Parameters relevant to constraint restitution.

time integration rule
time step size h
damping coefficient k_d
barrier stiffness κ and $\hat{c}$
Young's modulus Y

As elaborated in Section 5, our restitution model is realized by rapid energy transitions between the kinetic form and the barrier potential form during the activation of constraints. Therefore, the restitution behavior definitely relies on the energy behaviors of the simulation. This includes the numerical dissipation from a specific time integration rule. Here we focus on exploring implicit Euler and BDF-2. For these dissipative time integrators, different time step sizes h also result in different levels of dissipation. Then there is also our semi-implicit Rayleigh damping energy, where larger damping coefficient k_d introduces more energy loss.

Furthermore, barrier stiffness κ and the threshold distance $\hat{d}$ both influence the smoothness of our barrier potential, and so they both also affect the energy behavior since the truncation errors of the numerical time integration rules are highly dependent on the smoothness of the potential energies.

Last but not least, since our rigid bodies are essentially stiff elastic bodies reduced in the affine space, we need to make sure that the



(a) Point Connection (b) Hinge with Rotation Range (c) Bounded Sliding

Fig. 12. **(a) (left) Point Connection.** A triple-rod pendulum. **(a) (right) Point Connection with Cloth.** A piece of cloth hanging on a flexible coat rack. The left and right images show the initial and final static frame. **(b) Hinge with Rotation Range.** We use balls to hit the boards which can rotate around a shared shaft. From top to bottom, the rotation ranges of the boards are: 360°, 180° and 90°. **(c) Bounded Sliding.** Simulating this three-section monocular with our bounded sliding constraints results in a more than 10× speedup compared to directly simulating the real-world sliding structure.

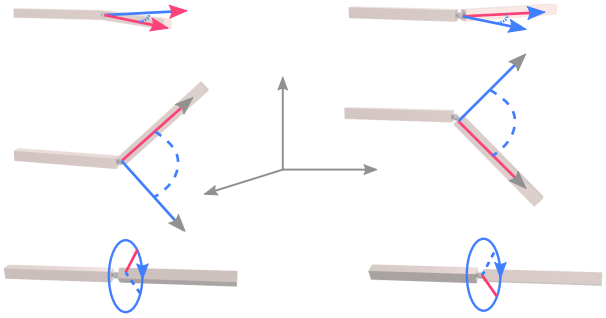


Fig. 13. **Cone Twist with Rotation Ranges.** A Nunchaku is used to demonstrate our cone twist constraint. The left section is moved horizontally (top), vertically (middle), and twisted (bottom), with the right section following its motion due to the separately controlled rotation ranges.

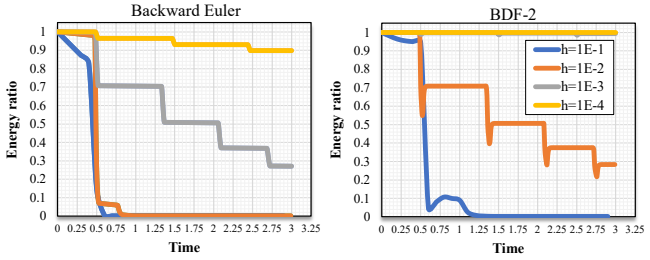


Fig. 14. **Implicit Euler v.s. BDF-2 at different h.** Energy curves of a cube falling onto the ground under gravity simulated with Implicit Euler and BDF-2 at $h = 0.1s$, $0.01s$, $0.001s$ and $10^{-4}s$. Here we plot the energy ratio calculated by treating the initial energy value as 1, and the energy value when cube stays static on the ground as 0. Here the first bounce happens at $0.5s$, where the energy decrease less when we use BDF-2 instead of implicit Euler, and when smaller time step sizes are applied.

Young's modulus Y is large enough so that during the kinetic and barrier energy transfer, no energies are absorbed in the elasticity potential as vibration.

We design two unit tests to analyze the above parameters:

Falling Box. A $1m$-wide cube free-falling onto the ground from $1m$ height with downward pointing gravity $g = -8m/s^2$.

Rotating Board. A board rotates around a shaft with initial angular velocity $1.25\ rad/s$. It is designed to reach the maximum rotation angle limit and then bounce back.

Next, we proceed our study by first showing the keys of obtaining energy conserved results, and then demonstrate controllable restitution via our semi-implicit Rayleigh damping model.

6.2.1 Energy Conservation. To start with, we compare implicit Euler and BDF-2 time integration on the falling box example with $Y = 10^9 Pa$, $\kappa = 10^5 Pa$, $\hat{d} = 0.064m$, and $k_d = 0$ at different timestep sizes. Here we plot the scaled energy value where the initial value is 1, and the value at the final resting configuration is 0 (Figure 14). We see that at frame-rate timestep size $h = 0.01s$, results given by implicit Euler damps out nearly 90% of the energy at first bounce, and $h = 0.1s$ is even more dissipative. But for BDF-2, the energy only decreases around 30% for each bounce at $h = 0.01s$, although the result is also extremely damped at $h = 0.1s$. This is because BDF-2 is also a dissipative time integrator. As we decrease the timestep size, even at $h = 10^{-4}s$, there are still obvious energy losses with implicit Euler, but for BDF-2, at $h = 0.001s$, the energy is already conserved up to 99%. These results clearly suggest using BDF-2 for significantly better energy preservation compared to implicit Euler.

Continuing our exploration with BDF-2 at $h = 0.001s$, we test different Young's modulus in the falling box example. We observed that, with smaller Young's modulus like $10^5 Pa$, the energy is well-conserved but the bouncing height is not even half of the original (Figure 15). This is because a significant amount of energy has been transferred into the elastic potential as the object starts to vibrate. When Young's modulus gets $10^7 Pa$, the bouncing height is almost equal to the original, and further increasing Young's modulus does not significantly improve or reduce the bouncing height. Notice that unless an elastic object is with infinitely large Young's modulus, it can always vibrate at a certain frequency. As the object gets stiffer, its frequency of vibration increases, which can exceed the limit that our discrete time integration could capture at certain time step sizes. This

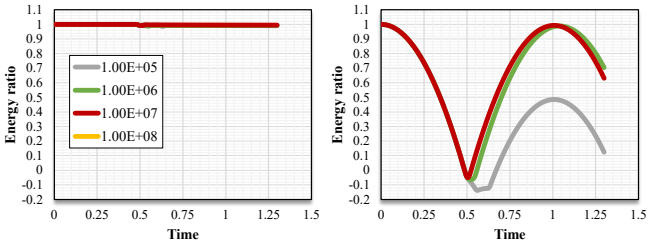


Fig. 15. **Different Young's Modulus.** The total energy plot (left) and the gravitational energy plot (right) of the falling box example with $Y = 10^5$, 10^6, 10^7, and 10^8. With smaller Young's modulus, although the total energy is well-conserved, the bouncing height is not fully restored as object starts vibrating and some energy is transferred into the elastic potential.

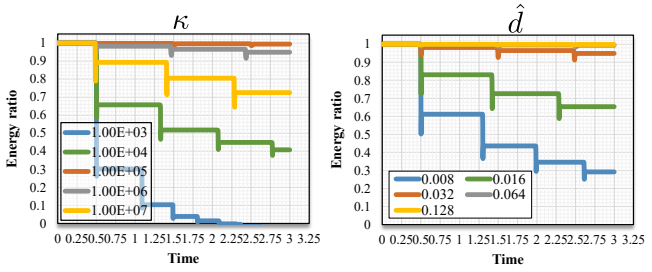


Fig. 16. **Different κ and $\hat{d}$.** Total energy plots of the falling box experiment with different κ (left) or $\hat{d}$ (right). We see that either a too large or too small κ can result in energy losses due to sharpened barrier functions. For $\hat{d}$, energy behavior is monotonically improved as $\hat{d}$ gets larger, a too large $\hat{d}$ can result in large gaps between colliding objects.

is exactly what is happening here – we are avoiding the vibrations by not capturing it. According to our extensive experiments, $Y = 10^9 Pa$ is generally enough.

The most interesting part comes with the κ and $\hat{d}$ which together control the smoothness of the barrier potential. Since our barrier is a piecewise function with local support, larger $\hat{d}$ corresponds to larger support regions, which clearly leads to a smoother function with fixed κ. But note that a too large $\hat{d}$ could result in unnecessary repulsion forces even when the constraint is well-satisfied, which potentially results in artifacts in the dynamics. For κ, a too small value can make the function sharp near 0, while a too large value can make it sharp near $\hat{d}$. Therefore, both κ and $\hat{d}$ need to be set appropriately for accurate restitution modeling. Our experiments on the falling box example with different κ and $\hat{d}$ suggest that $\kappa = 10^5 Pa$ and $\hat{d} = 0.064m$ are the optimal values for $h = 0.001s$ (Figure 16). However, different time step sizes and/or velocities can likely have different optimal values.

Without loss of generality, let us focus on contact constraints and implicit Euler time integration (so $v^{n+1} = (x^{n+1} - x^n)/h$), and look at the restitution process in the three consecutive time steps in a 1D ideal setting:

(1) **Approach.** objects approach each other with relative velocity $v_0 < 0$ and distance $d_0 > \hat{d}$;
(2) **Contact.** objects get closer to each other with distance $d_1 < \hat{d}$, and they are still approaching with relative velocity $v_1 \in (v_0, 0)$ as contact force just activates;

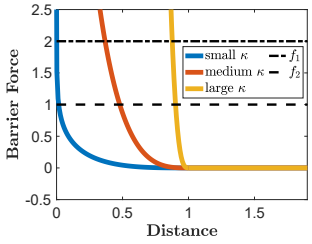


Fig. 17. **Barrier Force Plot with different κ.** With either a too large (yellow curve) or a too small (blue curve) κ, the slope of the barrier force will be extremely large between the two horizontal lines $y = f_1$ and $y = f_2$, resulting in small distance differences, and thus small bouncing velocities.

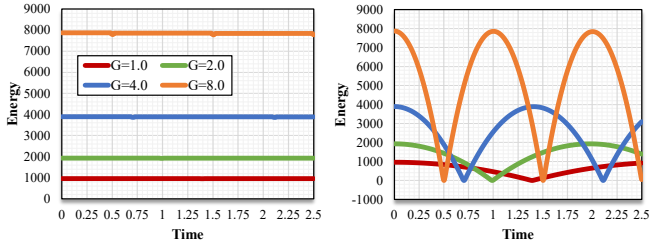


Fig. 18. **Different Gravity.** The total energy plot (left) and the gravitational potential energy plot (right) of the falling box example with Gravity $g = 1m/s^2$, $2m/s^2$, $4m/s^2$, and $8m/s^2$. With different gravity, the box hit the ground with different velocity, where elastic bounces for all the cases can be simulated with our method using the same set of parameters.

(3) **Rebound.** objects start to separate with relative velocity $v_2 > 0$, but their distance d_2 is still slightly smaller than $\hat{d}$.

Then the Newton restitution coefficient is approximately $r = v_2/(-v_0)$, and with implicit Euler we have $v_2 = (d_2 - d_1)/h < \hat{d}/h$. This relation indicates that, for perfect elastic impact ($r = 1$) to be resolved, we should at least have

$$\frac{\hat{d}}{h} > v_0, \tag{23}$$

which resembles a lot to the CFL conditions [Courant et al. 1967] in Eulerian-based simulations. Here κ decides the tightness of the above bound, where a reasonable κ should sufficiently separate d_1 and d_2 by providing a smoothly varying barrier potential. This is because the exact value of d_1 and d_2 are essentially calculated according to the contact force magnitude needed at the contact and rebound steps. Let us denote the force needed as f_1 and f_2 respectively, then we know that $f_2 - f_1 = \int_{d_1}^{d_2} \frac{\partial f}{\partial d} \boldsymbol{d}d$. Applying the Mean Value Theorem, we obtain $f_2 - f_1 = \frac{\partial f}{\partial d}(d_\xi)(d_2 - d_1)$ with $d_1 < d_\xi < d_2$. This clearly indicates that an either too large or too small κ will result in a large $\frac{\partial f}{\partial d}(d_\xi)$ and so a small $d_2 - d_1$ given fixed f_2 and f_1 (Figure 17).

In our falling box experiments with BDF-2, we observed that at $h = 0.001s$, using $\hat{d} = 0.064m$ and $\kappa = 10^5 Pa$ is able to simulate perfectly elastic contact for $v_0 \leq 4m/s$ (Figure 18), which already covers a wide range of complex real-world scenarios. We leave developing a numerical time integration scheme with unconditional energy conservation to future work.

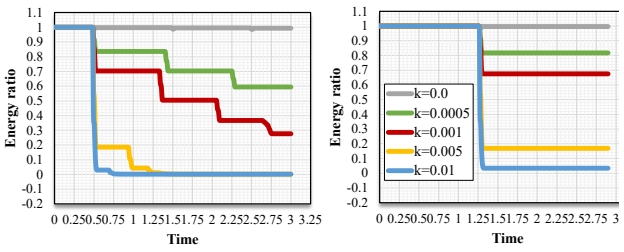


Fig. 19. **Contact and Constraint Damping.** Energy plots of the Falling (left) and the Rotating Board (right) examples with semi-implicit Rayle damping at different stiffnesses for controllable restitution behaviors.

6.2.2 Controllable Restitution. Being able to simulate energy conserving restitution behaviors, next we demonstrate controllable restitution behavior with different expected loss of energy realized by our semi-implicit Rayleigh damping model. For both the falling box and rotating boards experiments, we apply different damping stiffness k_d and obtained different energy dissipation at each bounce, resulting in different restitution behaviors (Figure 19). This demonstrates the generality of our restitution model on inequality articulation constraints besides contact.

6.2.3 Discussion. Our restitution model has several advantages over the traditional Newton's model within the LCP framework. With the Newton's model, achieving perfectly elastic behaviors without numerical instabilities requires trials and errors on setting the restitution coefficient close to 1 but not exactly 1. It is unclear how the coefficient relates to the stability of the system. With our model, perfectly elastic restitution can be achieved by directly applying the extensively studied energy-conserving time integrations, which also enables analyzing restitution in consistent with the numerical solution of PDEs. In addition, our Rayleigh damping model achieves controllable inelastic restitution for arbitrary inequality constraints, which could be conveniently applied to versatile customized constraints from researchers and practitioners in different communities.

6.3 Complex Real-World Scenes

Windmill (Comparison with Bullet). We start by constructing a windmill with a hinge constraint linking the blades to a fixed stand. A heavy green box drops, trigger the rotational motion of the wheel (Figure 20). The wheel blades then collide with two piles of boxes, scattering them all over the ground. We simulate this scene using our method and the Bullet library [Coumans 2015]. We find that Bullet often produces body-body intersections especially for fast-moving objects (e.g., the green box gets "buried" under the ground). Our method not only produces the accurate, realistic and interpenetration-free result but also runs much faster. In this example, our method delivers a 1.4× speedup over Bullet.

Hanging Bridge. We design a hanging bridge with 10 discrete elastic rods and 40 rigid boards, all linked together with point connection constraints (Figure 2). Then we place a ball at one end of the bridge. The ball begins to rotate forward under accurately resolved friction, until it deviates from the center line of the bridge and falls down as the boards lean towards one side.

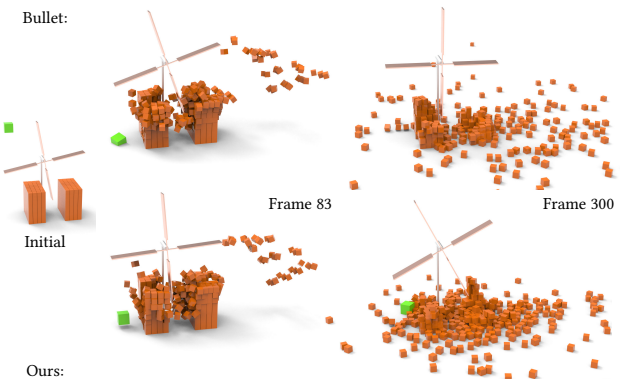


Fig. 20. **Windmill (Comparison with Bullet).** The windmill is composed of a fixed stand and four wheel blades (via a hinge constraint). We drop a heavy box (in green) onto one of its blades. The falling impact then rotates the windmill to collide its wheel with two stacks of cubes. Many intersections are present in the Bullet simulation result e.g., the green box deeply embeds into the floor. Our method is more efficient than Bullet (1.4× faster) and free of any body-body interpenetration.

Pulley System. We design a pulley system with 4 rigid pulleys and 4 elastic rods (Figure 3). The rightmost pulley is fixed by a hinge constraint, and all other pulleys are restricted to freely move or rotate in the vertical plane with sliding constraints. All the rods are tightly fitting in the notch of each pulley, with the first two rods connecting the ceiling and the rotation axis of the middle two pulleys. Our system accurately simulates the pulleys lifting heavy objects with longer pulling distance but smaller forces. This accuracy is only achieved as we accurately resolve the frictional contact.

We also conduct a real-world experiment on the pulley system and compare it side-by-side with our simulation results. Constructed by four identical pulleys (18.14g each), the system is tested by lifting a sequence of heavy objects (from 50g to 250g). In each experiment, we record the magnitude of the dragging force needed to keep the pulley system stationary. Our simulation well-matches both the analytical solution and the real-world experiment, despite some measuring errors in the real-world data (Figure 21).

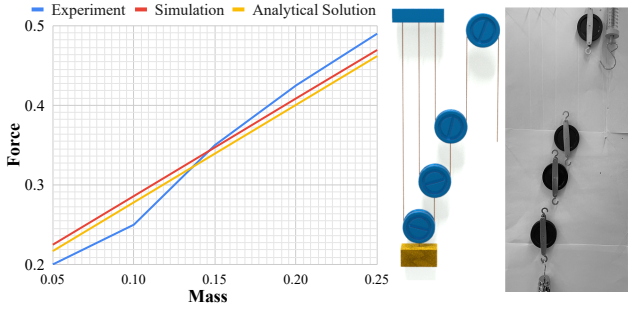


Fig. 21. **Pulley System Experiment.** We construct a pulley system in real-world and use it to lift objects with different mass. Plots of the dragging forces at stationarity shows that our simulation well-matches both the analytical solution and the real-world data, despite some measuring errors.

Umbrella. We simulate an umbrella by coupling a rigid skeleton with cloth. We design the internal supporting structure with 13 rigid thin rods connected with point connections, hinge, and sliding constraints. Then we model a pyramid-shaped cloth and connect it to the 6 outer rods on the skeleton. As we move the center pistol down, we can see intricate wrinkles formed on the cloth as it shrinks following the skeleton.

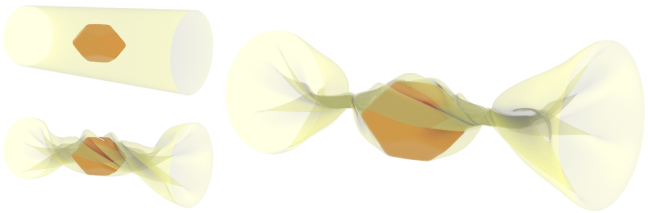


Fig. 22. **Candy.** We pack a red candy (affine body) by twisting a surrounding cylinder shell. The rigid-deformable coupling and cloth self-contact are automatically and robustly handled.

Candy. We place a rigid candy inside a cylinder shell, and then twist the shell on its both ends to wrap the candy. The shell quickly attaches the candy, and tight knots are formed on both sides. This example demonstrates the robustness of our system on simulating the friction and self-contact coupling mixed-dimensional solids even under large stresses.

Terrain Navigation with Cylinder-Driven Vehicle. We construct a self-powered cylinder engine and install it into a vehicle, connecting to the back-right wheel and make all three other wheels passive. The vehicle body, wheels, and all inner structures are all modeled with rigid bodies, connecting to each other through point connection and hinge constraints. As we pull the pistol of the cylinder engine with a constant external force, the car starts navigating the terrain smoothly due to accurately resolved frictional contact between the wheels and the ground.

Ragdolls. We drop 60 ragdolls on the ground. Each Ragdoll is composed of 11 rigid bodies, linked together by 10 cone twist joints. All joints are applied with specific rotation ranges independently set around different axes to mimic the mechanism of human body joints. For example, at the elbow, the rotation ranges for the two bending components are $[-135°, 0°]$ and $[-5°, 5°]$, and $[-10°, 10°]$ for twisting. All rotation ranges are strictly satisfied with our barrier formulation.

Precession. We simulate a unicycle with automotive pedals. We demonstrate the precession phenomenon by setting different initial velocities for the unicycle. With larger initial velocity, the unicycle is able to go farther in its direction.

"Lying Flat". Finally, we demonstrate the versatility of our unified solver by combining rigid/deformable objects of multiple codimensions. We model a hanging bed on the tree with multiple elastic tori, rigid rings, and circular discrete rods chaining together. Then we drop an articulated ragdoll and a piece of cloth onto the bed. The bed is firm and the ragdoll can safely "lying flat" as our simulator guarantees non-interpenetration even with thin geometries and complex contact configurations.

Table 2. **Simulation Setups and Statistics.** The collision detections are performed on NVIDIA 3090 GPU, while other operations are performed on Intel i9-10920X CPU. $C_{\max}$: max contact pair # per step. C_A : average contact pair # per step.

Example	Δt_{step}	N_{body}	N_{node}	μ	$C_{\max}$	C_A	iter/step	ms/iter
(Fig. 2) Bridge	1/24	51	2.3k	0.2	18	1.9	10.3	18.6
(Fig. 3) Pulley	1/24	9	16k	0.2	306	245.1	9.5	36.2
(Fig. 10) Terrain	1/24	10	34k	0.8	13	4.8	26.3	39.4
(Fig. 4) Umbrella	1/24	19	30k	0.0	491	176.7	10.2	1269.1
(Fig. 11) Ragdolls	1/100	1861	18k	0.2	17820	16590.2	43.1	1072.9
(Fig. 23) Precession	1/240	5	4k	0.8	4	0.3	3.5	16.0
(Fig. 1) "Lying Flat"	1/24	77	18k	0.2	1575	89.6	26.2	22.350
(Fig. 20) Windmill	1/240	468	1.9k	0.2	28641	9248.0	14.3	89.6
(Fig. 22) Candy	1/24	2	17k	0.4	1309	339.3	8.1	722.9

6.4 Performance analysis

For all the complex real-world scenes, we show the detailed timing breakdowns composed of Hessian/Gradient computations, collision detection (contact stencils and CCD), and linear solve (Figure 24). We observe that a scene with only rigid bodies tend to consume a far less proportion of time on linear solves than a scene that also contains deformable bodies. This is as expected because deformable bodies often introduce a much larger number of simulation DOFs. In addition, Hessian computation cost scales with both the system size and the number of linear equality constraints.

To investigate the scalability of our system, we design an experiment with $N \times N$ nunchakus (each consists of two box segments linked by a hinge constraint) distributed uniformly in a square domain and then dropped onto the ground simultaneously. We run the simulation with $N = 20, 40, 60, 80, 100$ and plot their timing per step in Figure 25. The plot shows that the timing cost of our system grows linearly as the number of objects increases. Note that our current implementation only uses GPU to perform collision detections, which as expected only takes a small proportion of the total timing (Figure 24 yellow and red) especially for large scenes.

7 CONCLUSION

In this paper we present a new multibody solver based on a unified primal Newton Barrier Method. In contrast to most existing work, our framework converts the formulation of contact, friction, and articulation constraints into unconstrained nonlinear optimization assisted with filtered line search techniques. We provide guarantees in global convergence and constraint satisfaction while maintaining practical near-interactive performance for a wide range of challenging and versatile scenes.

7.1 Limitations and Future Work

Our system can be slower than traditional rigid body solvers when there are only a few objects in simple scenarios (e.g. 2 bodies with 1 hinge constraint) due to our slightly more DOFs per body (12 vs 6). In addition, for our handling of the nonlinear equality constraints via quadratic penalty, in more extreme situations where larger penalty

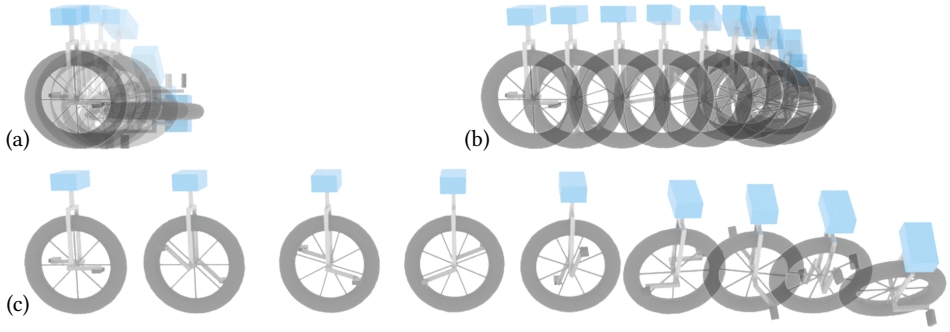


Fig. 23. **Precession.** From (a) to (c), we drop a unicycle on the ground with increasing initial angular velocities. As our method accurately simulates the dynamics and frictional contacts, different precession effects are nicely captured as faster angular velocity leads to a longer time of rolling. We show one image per 50 frames in each sub-figure until the unicycle loses balance.

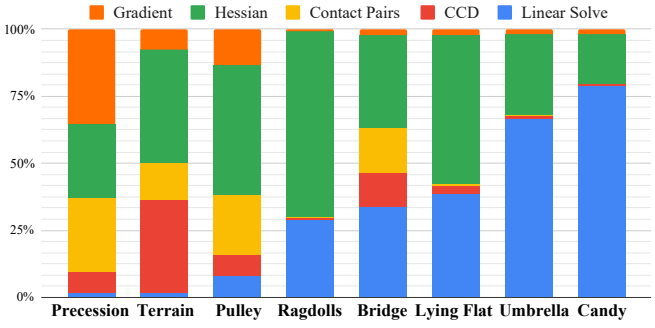


Fig. 24. **Timing Breakdown.** We plot the proportions of the main computation routines (Gradient, Hessian, contact stencils, CCD, and linear solves) for all real-world complex scenes. Examples are sorted in ascending order on the percentage of time spent on linear solve.

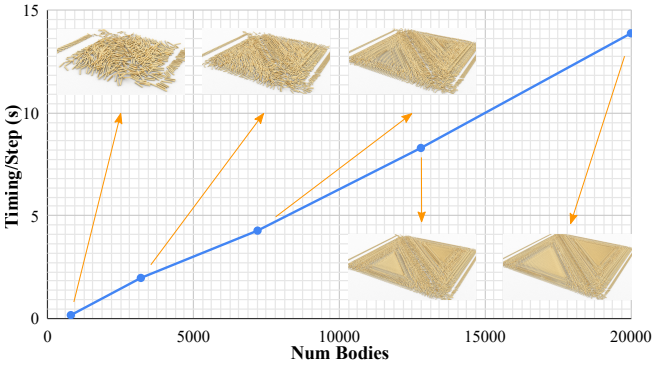


Fig. 25. **Scalability Test.** We perform a scalability test on our system by simulating a number of 2-segment nunchakus dropped onto the ground. Here we visualize the configurations at 2.4 *s*. The timing cost per step grows linearly as the number of nunchakus increases.

stiffness is required for higher accuracy, the Newton convergence will be slower due to the worse system condition. Although in our experiment we do not observe any significant performance deficiencies, it is certainly meaningful to further accelerate our method by e.g. migrating more computations to the GPU.

With our choice of the maximal coordinate system, we defined our relative sliding joint as a nonlinear equality constraint, which with generalized coordinates can be expressed as a linear function of the DOFs. However, a generalized coordinate system will result in a nonlinear polynomial relation between the reduced and full DOFs, which on the other end inject more nonlinearity to the potential energies in implicit time integration. Our system can be trivially extended to support generalized coordinate systems, and it would be interesting to compare between these two design choices.

Our restitution behavior is mainly analyzed on the BDF-2 time integration scheme. We selected BDF-2 for its unconditional stability and simplicity as it is a linear multi-step method. But note that our controllable restitution can be realized based on any time integrators with nice energy preserving behaviors. It is certainly interesting and meaningful to test our system with other stable and accurate time integration schemes such as the implicit Runge-Kutta, etc. Furthermore, designing a time integration invariant energy conserving restitution model for barrier-based inequality constraints would be extremely meaningful.

ACKNOWLEDGMENTS

We thank all reviewers for their valuable feedback, and all UCLA MultiPLES lab members for the help on setting up the pulley system. Y. C., M. L., and C. J. are supported in part by NSF (No. 2153851, 2153863, 2023780) and DOE ORNL (No. 4000171342). L. L. and Y. Y. are supported in part by NSF (No. 2016414, 2011471). H. S. thanks Qualcomm AI for the gift funding.

REFERENCES

Masniezam Ahmad, Khairul A Ismail, and Fauziah Mat. 2016. Impact models and coefficient of restitution: a review. *ARPN Journal of Engineering and Applied Sciences* 11 (2016).

Sheldon Andrews, Marek Teichmann, and Paul G Kry. 2017. Geometric Stiffness for Real-time Constrained Multibody Dynamics. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 235–246.

Yunfei Bai and Karen Liu. 2014. Coupling cloth and rigid bodies for dexterous manipulation. In *Proceedings of the Seventh International Conference on Motion in Games*.

David Baraff. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*. 223–232.

Nathan Bell, Yizhou Yu, and Peter J Mucha. 2005. Particle-based simulation of granular materials. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 77–86.

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.

Jumyung Chang, Fang Da, Eitan Grinspun, and Christopher Batty. 2019. A Unified Simplicial Model for Mixed-Dimensional and Non-Manifold Deformable Elastic Objects. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–18.

Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko. 2005. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 153–160.

Eulalie Coevoet, Otman Benchekroun, and Paul G Kry. 2020. Adaptive merging for rigid body simulation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 35–1.

Erwin Coumans. 2015. Bullet Physics Simulation. In *ACM SIGGRAPH 2015 Courses*. Article 7.

Richard Courant, Kurt Friedrichs, and Hans Lewy. 1967. On the partial difference equations of mathematical physics. *IBM journal of Research and Development* 11, 2 (1967), 215–234.

Crispin Deul, Patrick Charrier, and Jan Bender. 2016. Position-based rigid-body dynamics. *Computer Animation and Virtual Worlds* 27, 2 (2016), 103–112.

Andreas Enzenhöfer, Nicolas Lefebvre, and Sheldon Andrews. 2019. Efficient block pivoting for multibody simulations with contact. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 1–9.

Kenny Erleben. 2007. Velocity-based shock propagation for multibody dynamics animation. *ACM Transactions on Graphics (TOG)* 26, 2 (2007), 12–es.

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), 183.

Mihai Frâncu and Florica Moldoveanu. 2017a. Position based simulation of solids with accurate contact handling. *Computers & Graphics* 69 (2017), 12–23.

Mihai Frâncu and Florica Moldoveanu. 2017b. Unified Simulation of Rigid and Flexible Bodies Using Position Based Dynamics.. In *VRIPHYS*. 49–58.

Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics* 21, 10 (2015), 1103–1115.

Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. 2020. ADD: analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Transactions on Graphics (TOG)* 39, 6 (2020).

Eran Guendelman, Robert Bridson, and Ronald Fedkiw. 2003. Nonconvex rigid bodies with stacking. *ACM transactions on graphics (TOG)* 22, 3 (2003), 871–878.

David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous contact mechanics. In *ACM SIGGRAPH 2009 papers*. 1–12.

Sumit Jain and Karen Liu. 2011. Controlling physics-based characters using soft contacts. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. 1–10.

Danny M Kaufman, Timothy Edmunds, and Dinesh K Pai. 2005. Fast frictional dynamics for rigid bodies. In *ACM SIGGRAPH 2005 Papers*. 946–956.

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.

Yuki Koyama, Shinjiro Sueda, Emma Steinhardt, Takeo Igarashi, Ariel Shamir, and Wojciech Matusik. 2015. AutoConnect: computational design of 3D-printable connectors. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–11.

Lei Lan, Danny M Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022. Affine Body Dynamics: Fast, Stable & Intersection-free Simulation of Stiff Materials. *ACM Transactions on Graphics (TOG)* 41, 4 (2022).

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 (TOG)* 40, 4 (2021), 1–16.

Jing Li, Tiantian Liu, and Ladislav Kavan. 2020b. Soft Articulated Characters in Projective Dynamics. *IEEE Transactions on Visualization and Computer Graphics* (2020).

Minchen Li. 2020. *Robust and Accurate Simulation of Elastodynamics and Contact*. Ph.D. Dissertation. University of Pennsylvania.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020a. Incremental potential contact: Intersection- and inversion-free, large-deformation dynamics. *ACM transactions on graphics* 39, 4 (2020).

Minchen Li, Ming Gao, Timothy Langlois, Chenfanfu Jiang, and Danny M Kaufman. 2019. Decomposed Optimization Time Integrator for Large-Step Elastodynamics. *ACM Transactions on Graphics* 38, 4 (2019).

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

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.

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-smooth newton methods for deformable multi-body dynamics. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 1–20.

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.

Brian Mirtich and John Canny. 1994. *Impulse-based dynamic simulation*. Citeseer.

Brian Vincent Mirtich. 1996. *Impulse-based dynamic simulation of rigid body systems*. University of California, Berkeley.

Jean J Moreau. 1985. Standard inelastic shocks and the dynamics of unilateral constraints. In *Unilateral problems in structural analysis*. Springer, 173–221.

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.

Matthias Müller, Miles Macklin, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. 2020. Detailed rigid body simulation with extended position based dynamics. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 101–112.

Albert Peiret, Sheldon Andrews, József Kövecses, Paul G Kry, and Marek Teichmann. 2019. Schur complement-based substructuring of stiff multibody systems with contact. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 1–17.

Jovan Popović, Steven M Seitz, Michael Erdmann, Zoran Popović, and Andrew Witkin. 2000. Interactive manipulation of rigid body simulations. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 209–217.

Stephane Redon, Nico Galoppo, and Ming C Lin. 2005. Adaptive dynamics of articulated bodies. In *ACM SIGGRAPH 2005 Papers*. 936–945.

Edward John Routh. 1905. *The advanced part of a treatise on the dynamics of a system of rigid bodies*. MacMillan & Co.

Robert Seifried, Werner Schiehlen, and Peter Eberhard. 2010. The role of the coefficient of restitution on impact problems in multi-body dynamics. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 224, 3 (2010), 279–306.

Tamar Shinar, Craig Schroeder, and Ronald Fedkiw. 2008. Two-way coupling of rigid and deformable bodies. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Citeseer, 95–103.

Breannan Smith, Danny M Kaufman, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2012. Reflections on simultaneous impact. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–12.

David E Stewart. 2000. Rigid-body dynamics with friction and impact. *SIAM review* 42, 1 (2000), 3–39.

William J Stronge. 1991. Unraveling paradoxical theories for rigid body collisions. *Journal of Applied Mechanics* 58 (1991).

Shinjiro Sueda, Garrett L Jones, David IW Levin, and Dinesh K Pai. 2011. Large-scale dynamic simulation of highly constrained strands. In *ACM SIGGRAPH 2011 papers*.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 181–190.

Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational design of linkage-based characters. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9.

Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. 2015. Stable constrained dynamics. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

Christopher D Twigg and Doug L James. 2008. Backward steps in rigid body simulation. In *ACM SIGGRAPH 2008 papers*. 1–10.

Jui-Hsien Wang, Rajsekhar Setaluri, Doug L James, and Dinesh K Pai. 2017. Bounce maps: an improved restitution model for real-time rigid-body impact. *ACM Trans. Graph.* 36, 4 (2017), 150–1.

Yu Wang and Matthew T Mason. 1992. Two-dimensional rigid-body collisions with friction. *Journal of Applied Mechanics* 59 (1992).

Ying Wang, Nicholas J Weidner, Margaret A Baxter, Yura Hwang, Danny M Kaufman, and Shinjiro Sueda. 2019. REDMAX: Efficient & flexible approach for articulated dynamics. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–10.

Rachel Weinstein, Joseph Teran, and Ronald Fedkiw. 2006. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization and Computer Graphics* 12, 3 (2006), 365–374.

Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and Karen Liu. 2021. Fast and Feature-Complete Differentiable Physics Engine for Articulated Rigid Bodies with Contact Constraints. In *Robotics: Science and Systems*.

Edmund Taylor Whittaker. 1937. *A treatise on the analytical dynamics of particles and rigid bodies*. CUP Archive.

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.

Yidong Zhao, Jinhyun Choo, Yupeng Jiang, Minchen Li, Chenfanfu Jiang, and Kenichi Soga. 2022. A barrier method for frictional contact on embedded interfaces. *Computer Methods in Applied Mechanics and Engineering* 393 (2022), 114820.