

Semi-Implicit Pairwise Descent for Nonlocal Continuum Mechanics

XUKUN LUO*, Institute of Software, Chinese Academy of Sciences and UCAS, China

XIAO CHENG*, Institute of Software, Chinese Academy of Sciences and UCAS, China

YUZHONG GUO, Institute of Software, Chinese Academy of Sciences, China

YING QIAO, Institute of Software, Chinese Academy of Sciences, China

WENCHENG WANG, Institute of Software, Chinese Academy of Sciences, China

XIAOWEI HE[†], Institute of Software, Chinese Academy of Sciences, China

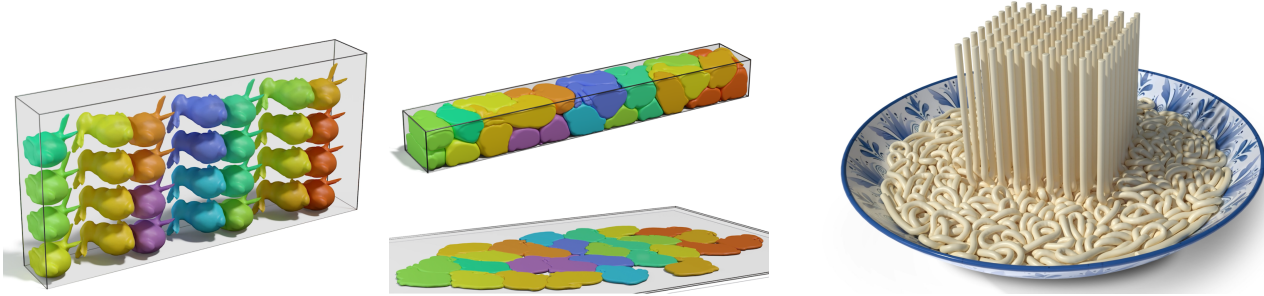


Fig. 1. **Left:** 28 bunnies (1.03M tetrahedra, 225K vertices) are compressed and deformed by the walls of a transparent box upon impact. Our solver maintains **real-time** performance with an average cost of 3ms per step ($h = 3ms$). **Right:** Noodles (1.8M tetrahedra, 697K vertices) falling into a bowl with 2.62M active collision pairs, computed at 14.3ms per step ($h = 3ms$).

We propose Semi-Implicit Pairwise Descent (SIPD), a unified nonlocal pairwise framework for simulating large-scale hyperelastic materials involving complex contact and friction. By reformulating the Finite Element Method (FEM) equations of motion into a pairwise force representation from a nonlocal perspective, our approach avoids costly Hessian computations, leading to a reduction in per-iteration computational overhead. Furthermore, we propose an analytical projection strategy for projecting our Hessian-free coefficient matrices to positive semi-definiteness. And we treat contact and friction as a unified anisotropic elastic energy, allowing for a seamless integration into the elastic solver framework. We mathematically prove that our method is unconditionally stable and numerically convergent. Experimental results demonstrate that SIPD achieves real-time performance for million-scale simulations even under intricate contact and friction conditions.

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

*Equal contribution.

[†]Corresponding author.

UCAS: University of Chinese Academy of Sciences.

Authors' Contact Information: Xukun Luo, luoxukun2022@iscas.ac.cn, Institute of Software, Chinese Academy of Sciences and UCAS, Beijing, China; Xiao Cheng, chengxiao25@ios.ac.cn, Institute of Software, Chinese Academy of Sciences and UCAS, Beijing, China; Yuzhong Guo, guoyuzhong@iscas.ac.cn, Institute of Software, Chinese Academy of Sciences, Beijing, China; Ying Qiao, qiaoying@iscas.ac.cn, Institute of Software, Chinese Academy of Sciences, Beijing, China; Wencheng Wang, whn@ios.ac.cn, Institute of Software, Chinese Academy of Sciences, Beijing, China; Xiaowei He, xiaowei@iscas.ac.cn, Institute of Software, Chinese Academy of Sciences, Beijing, China.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 1557-7368/2026/0-ART000
<https://doi.org/XXXXXXX.XXXXXXX>

Additional Key Words and Phrases: nonlocal continuum mechanics, peridynamics, finite element analysis, semi-implicit successive substitution method, hyperelasticity, contact

ACM Reference Format:

Xukun Luo, Xiao Cheng, Yuzhong Guo, Ying Qiao, Wencheng Wang, and Xiaowei He. 2026. Semi-Implicit Pairwise Descent for Nonlocal Continuum Mechanics. *ACM Trans. Graph.* 00, 0, Article 000 (2026), 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

The fast, robust, and accurate simulation of hyperelastic materials involving discontinuous contact forces has long been a fundamental goal in computer graphics applications. Since explicit time integration schemes impose strict limitations on the time step size, semi-implicit [Bridson et al. 2005; Schroeder et al. 2011] and fully implicit [Liu et al. 2013; Macklin et al. 2016; Smith et al. 2018] time integration schemes are increasingly favored for real-time applications. Moreover, with the significant increase in GPU computing power over the past decade, the development of implicit solvers capable of fully leveraging GPU advantages for deformable body dynamics becomes a new trend. However, fully addressing the challenges of simulating hyperelastic materials with both nonlinear behavior and discontinuous contact forces on GPUs requires not only a straightforward extension of traditional numerical models but also a deeper understanding and reformulation of the underlying problem.

Within classical continuum mechanics, the behavior of hyperelastic materials is typically modeled using partial differential equations. When the continuum is discretized into tetrahedra or other control

volumes using finite element analysis, the governing equations reduce to a finite nonlinear system in terms of the nodal positions [Irving et al. 2004; Sifakis and Barbic 2012; Teran et al. 2005]. To solve the nonlinear system, which may include various constraints arising from contact, sequential algorithms typically employ the Newton-Raphson solver. This solver begins by linearizing the elastic forces and then calculates the energy Hessian. However, ensuring that the stiffness matrix remains positive definite is challenging, particularly when arbitrary forms of discontinuous contact forces are included [Kang et al. 2025; Shi 1992]. Although a straightforward implementation of the Newton-Raphson solver on GPUs can significantly enhance performance due to increased computational power, the algorithm is not well suited to GPU architectures because of the high dynamic memory overhead associated with storing the Hessian matrix. To address this issue, Wang and Yang [2016] propose using a diagonalized Hessian matrix to calculate the descent direction, ensuring positive definiteness by simply checking the values of the diagonal elements. Chen et al. [2024b] instead propose Vertex Block Descent (VBD) and derive a 3×3 Hessian matrix for each vertex to update each vertex's position by solving a small local linear system. However, they still need to calculate the fourth-order tensors by taking the derivative of the first Piola-Kirchhoff stress tensor $\mathbf{P}$ with respect to the deformation gradient $\mathbf{F}$, i.e., $\partial \mathbf{P} / \partial \mathbf{F}$. Moreover, VBD does not guarantee a descent direction because each vertex block may not be positive definite.

We present the very GPU-friendly semi-implicit pairwise descent to solve the nonlinear optimization problem. Unlike the Newton-Raphson solver that requires calculating the second-order derivative of the energy function, our key innovation relies on reformulating the variational problem into a nonlocal peridynamics framework depending only on the first-order derivative of the energy density function, in which both internal and boundary forces are represented in a unified pairwise form. Thus, we efficiently construct the 3×3 coefficient matrix for each vertex using only the first Piola-Kirchhoff stress $\mathbf{P}$ and the deformation gradient $\mathbf{F}$ obtained from the constitutive model. During each iteration, a novel semi-implicit splitting based on [Lu et al. 2023] and an SVD-free analytical projection strategy are employed to ensure that the positive component of the coefficient matrix used for implicit integration remains positive definite. Consequently, the 3×3 coefficient matrix assembled at each vertex is guaranteed to be positive definite. Our method can uniformly accommodate scenarios involving discontinuous contact and friction, achieving real-time simulation for million-scale scenes.

2 Related Work

Continuum mechanics centers on describing the macroscopic behavior of matter through partial differential equations (PDEs). Since Terzopoulos et al. [1987] pioneered this theory to computer graphics, the Finite Element Method (FEM) has become the standard for deformation modeling. Sin et al. [2013] subsequently extended this framework to hyperelastic models within an implicit Euler integration scheme. Kugelstadt et al. [2018a] employed operator splitting to separately solve the shear and bulk energies for efficient computation. Schneider et al. [2018] proposed an adaptive scheme for adjusting the order of finite element basis functions, achieving high-fidelity results even on coarse meshes. While FEM

excels at modeling continuous deformations, it inherently struggles with physical discontinuities like fracture and contact, where spatial derivatives become ill-defined.

To bridge this gap, the nonlocal continuum mechanics, exemplified by Peridynamics (PD) [Silling et al. 2007], replaces differential operators with integral operators using a pairwise function $\psi(\mathbf{x}, \mathbf{x}')$ within a specific horizon. This formulation effectively captures discontinuous behavior, including crack initiation, strain localization, and fracture. Leveraging nonlocal theory, Levine et al. [2014] reformulated mass-spring systems to achieve efficient brittle fracture simulation. Similarly, Chen et al. [2018] developed a plastic model within the PD framework for ductile fracture, and Han et al. [2021] introduced a nonlocal diffusion model for temperature-dependent fracture simulation. Regarding solution strategies, He et al. [2017] proposed a corotational peridynamics formulation based on Projective Dynamics, while Lu et al. [2023] extended this to hyperelastic materials and developed a semi-implicit acceleration scheme.

Nevertheless, PD models often lack the precision of FEM in representing continuous behaviors, leading researchers [Dorduncu et al. 2024] to explore coupling strategies that utilize both models to handle continuous and discontinuous regions respectively. For instance, Yu et al. [2025] developed a coupling strategy based on force equilibrium, employing FEM for elastic continua and PD for crack regions. This coupling paradigm has also informed the integration of FEM with the Material Point Method (MPM) [Li et al. 2024]. However, these hybrids often face challenges in numerical stability and implementation complexity.

Contact is inherently characterized by discontinuous behavior, manifested as displacement discontinuities across interfaces. Thus early research typically treat collisions as impulsive forces [Bridson et al. 2002]. Modern approaches treat contact as a continuous optimization problem, via penalty methods [Chen et al. 2024b; Gast et al. 2015], dynamic penalty functions [Giles et al. 2025; Harmon et al. 2009; Narain et al. 2012; Tang et al. 2018], or Incremental Potential Contact (IPC) [Li et al. 2020, 2021]. These nonlinear optimization problems are often solved using full Newton methods [Huang et al. 2024; Li et al. 2020]. A related class of quasi-Newton elastodynamic solvers combines L-BFGS low-rank curvature updates with lagged second-order information. Liu et al. [2017] reused an FEM Hessian approximation across iterations and employed L-BFGS updates to incorporate current curvature information. Subsequent extensions employed more structured initializers, including a domain-decomposed model for mesh-based elastodynamics and a multigrid model for MPM, while retaining the L-BFGS iteration [Li et al. 2019; Wang et al. 2020]. More recently, Lu and Hu [2025] introduced Reliable Iterative Dynamics (RID), which combines a force-splitting scheme with a dual-descent formulation to reduce linearization errors and produce reliable intermediate updates. Other parallel solution strategies include position-based nonlinear Gauss-Seidel [Chen et al. 2023], preconditioned nonlinear conjugate gradient [Shen et al. 2024], Jacobi preconditioning based on the diagonal Hessian [Wang and Yang 2016], and VBD, which performs local vertex updates through graph coloring [Chen et al. 2024b]. These methods provide effective strategies for improving convergence, robustness, and parallel efficiency, but still rely on second-order Hessian information, either directly or through lagged or structured approximations. Motivated

by reducing this dependency, we seek to solve the nonlinear system using only first-order constitutive information.

3 Foundations

Since the purpose of this work is to develop a unified model capable of handling both continuous and discontinuous material behaviors, the following two sections first review fundamental concepts in both classical and nonlocal continuum mechanics. To avoid notational ambiguity, the symbol $\mathbf{x}$ is used to denote coordinates in the reference configuration, whereas $\mathbf{y}$ denotes coordinates in the deformed configuration. Vertices with lumped mass in a tetrahedral mesh are uniformly referred to as particles.

3.1 Finite Element Method

The core idea of the finite element method (FEM) is to approximate the solution of a continuous problem by breaking the domain into a finite number of elements and solving a discretized variational problem over these elements [Wriggers 2008]. For a given continuum medium, the equation of motion is expressed as

$$\rho \ddot{\mathbf{y}} = \nabla_{\mathbf{x}} \cdot \mathbf{P} + \mathbf{b}, \quad (1)$$

where $\nabla_{\mathbf{x}} \cdot \mathbf{P}$ denotes the divergence of the first Piola–Kirchhoff stress tensor $\mathbf{P}$ with respect to the reference coordinates $\mathbf{x}$, and $\mathbf{b}$ denotes the external force. When the continuum medium is discretized into a tetrahedral mesh, the deformed coordinates $\mathbf{y}$ are approximated using shape functions $N_i(\mathbf{x})$: $\mathbf{y} = \sum_i N_i(\mathbf{x}) \mathbf{y}_i$, where i denotes the vertex index and $\mathbf{y}_i$ is the deformed position of vertex i . The shape function $N_i(\mathbf{x})$ satisfies the Kronecker- δ property [Koschier et al. 2017] that $N_i(\mathbf{x}_j) = \delta_{ij}$. The deformation gradient on each tetrahedral element e then becomes

$$\mathbf{F}_e(\mathbf{x}) = \sum_{i \in \mathcal{V}_e} \mathbf{y}_i \otimes \nabla N_i(\mathbf{x}), \quad (2)$$

where $\mathcal{V}_e$ denotes the set of vertices of tetrahedral element e . Substituting this approximation into the weak form of Equation 1 and integrating over the tetrahedral mesh yields the following discretized equation of motion for particle i [Sifakis and Barbic 2012]:

$$m_i \ddot{\mathbf{y}}_i = - \sum_{e \in \mathcal{T}_i} V_e \mathbf{P}_e \nabla N_e + \mathbf{b}_i \quad (3)$$

where $\mathcal{T}_i$ denotes the set of tetrahedral elements incident to particle i , m_i and $\mathbf{b}_i$ denote the lumped mass and external force, and V_e is the volume of element e .

3.2 Peridynamics

As a typical nonlocal continuum theory, peridynamics replaces spatial derivatives with integral operators, and material points interact directly with other points within a finite distance. As a result, the governing equations remain valid even when the displacement field is discontinuous. The equation of motion in a state-based model is expressed as [Silling et al. 2007]:

$$\rho \ddot{\mathbf{y}} = \int_{\mathcal{H}_{\mathbf{x}}} \{ \mathbf{T}_{\mathbf{x}} \langle \mathbf{x}' - \mathbf{x} \rangle - \mathbf{T}_{\mathbf{x}'} \langle \mathbf{x} - \mathbf{x}' \rangle \} dV_{\mathbf{x}'} + \mathbf{b}, \quad (4)$$

where $\mathcal{H}_{\mathbf{x}}$ is a spherical neighborhood centered at $\mathbf{x}$, and $\mathbf{T}_{\mathbf{x}}$ and $\mathbf{T}_{\mathbf{x}'}$ are the force vector states at points $\mathbf{x}$ and $\mathbf{x}'$, respectively. A state of order m in peridynamics is defined to be a function that maps a bond $\xi = \mathbf{x}' - \mathbf{x}$ to a tensor of order m , typically denoted

as $\underline{\mathbf{A}}(\xi)$. Intuitively, a bond is simply the reference-space vector connecting two material points, and a force state maps each such bond to the interaction force transmitted along that connection. The neighborhood integral therefore accumulates pairwise interactions, much as FEM assembly accumulates the element forces shared by neighboring vertices. A significant difference of the momentum equation in a nonlocal continuum is that the force vector is defined as a function of two material points. The same idea is also extended to define other physical quantities. By introducing the deformation vector state $\underline{\mathbf{Y}}(\xi) = \mathbf{y}' - \mathbf{y}$, the deformation gradient in the nonlocal form is expressed as

$$\mathbf{F} = \int_{\mathcal{H}_{\mathbf{x}}} \underline{\omega}(\xi) \underline{\mathbf{Y}}(\xi) \otimes \mathbf{K}^{-1} \xi dV_{\mathbf{x}}, \quad (5)$$

where $\underline{\omega}(\xi)$ is the weighting function that maps each bond ξ to a scalar value, $\mathbf{K}$ denotes the shape tensor formulated as follows

$$\mathbf{K} = \int_{\mathcal{H}_{\mathbf{x}}} \underline{\omega}(\xi) \xi \otimes \xi dV_{\mathbf{x}}. \quad (6)$$

By comparing Equation 5 with Equation 2 through dimensional analysis, it can be observed that both equations share the same structure in computing the deformation gradient. The key difference is that Equation 2 employs local variables ($\mathbf{y}_i$ and $\nabla N_i(\mathbf{x})$), whereas Equation 5 relies on nonlocal variables ($\underline{\mathbf{Y}}(\xi)$ and $\mathbf{K}^{-1} \xi$) to evaluate the deformation gradient. Substituting the nonlocal expression of the deformation gradient into the momentum equation and integrating over $\mathcal{H}_{\mathbf{x}}$ yields the discretized momentum equation in a nonlocal form expressed as follows [Lu et al. 2023]:

$$m_i \ddot{\mathbf{y}}_i = V_i \sum_j \omega_{ij} V_j \left(\mathbf{P}_i \mathbf{K}_i^{-1} + \mathbf{P}_j \mathbf{K}_j^{-1} \right) \xi_{ij} + \mathbf{b}_i. \quad (7)$$

where the subscripts i and j are used to denote particle i and j , the bond $\xi_{ij} = \mathbf{x}_j - \mathbf{x}_i$. Other variables such as V and $\mathbf{P}$ have the same physical meanings as those in Equation 3, except they are now defined on particles.

4 A Nonlocal Pairwise Model for Continuous Deformation

In this section, we reformulate the FEM using the nonlocal concept of peridynamics introduced in Section 3.2. The resulting formulation is theoretically equivalent to the traditional FEM.

4.1 Global Optimization in a Nonlocal Formulation

Consider an object that undergoes a deformation that maps every material point from its reference space $\mathbf{x}$ to the deformed space $\mathbf{y} = \vec{\phi}(\mathbf{x})$, where $\vec{\phi}$ is an arbitrary continuous function. Without loss of generality, we consider the tetrahedron shown in Figure 2 as a Lagrangian finite control volume, where the vertex positions in the reference and deformed configurations are denoted as $\mathbf{x}_{i=1,2,3,4}$ and $\mathbf{y}_{i=1,2,3,4}$. Assuming that the deformation map is linear within tetrahedron e , the lumped force on particle i is derived from the finite element formulation in Section 3.1:

$$\mathbf{f}_{ie} = -V_e \mathbf{P}_e \nabla N_{ie}, \quad (8)$$

where $N_{ie}(\mathbf{x})$ is the linear shape function associated with vertex i and satisfies $N_{ie}(\mathbf{x}_j) = \delta_{ij}$. In classical continuum mechanics, $\mathbf{f}_{ie}$ can be interpreted as the total force acting on the boundary triangle

opposite to vertex i , as shown in Figure 2(a). To reformulate $\mathbf{f}_{ie}$ in a nonlocal form, we introduce the following theorem.

THEOREM 4.1. *Given a tetrahedron with four vertex positions denoted by $\mathbf{x}_{i=1,2,3,4}$ and linear shape functions $N_i(\mathbf{x})$ satisfying $N_i(\mathbf{x}_j) = \delta_{ij}$, we have*

$$\nabla N_i = - \sum_{j \in \mathcal{V}_e} \mathbf{K}_e^{-1} \xi_{ij}, \quad \mathbf{K}_e = \sum_{j \in \mathcal{V}_e} \xi_{ij} \otimes \xi_{ij}. \quad (9)$$

PROOF. See Supplementary Material, Section A.

Note $\mathbf{K}_e$ is a special case of the shape tensor defined in Equation 6, obtained by applying a constant weight function $\underline{\omega}(\xi) = 1$. After substituting Equation 9 into Equation 8, the particle force $\mathbf{f}_{ie}$ is decomposed into three pairwise forces, as illustrated in Figure 2(b). The physical interpretation of $\mathbf{f}_{ie}$ can thus be understood as the actions exerted by the other particles. After summing over all tetrahedra incident to particle i , the governing equation is reformulated as

$$m_i \ddot{\mathbf{y}}_i = \sum_j \left(\sum_{e \in \mathcal{T}_i \cap \mathcal{T}_j} V_e \mathbf{P}_e \mathbf{K}_e^{-1} \right) \xi_{ij} + \mathbf{b}_i, \quad (10)$$

where V_e , $\mathbf{P}_e$ and $\mathbf{K}_e$ denote physical quantities associated with the tetrahedron that shares the edge connecting particle i and j . Accordingly, the total force acting on particle i by its neighboring particle j is formulated in the following nonlocal form:

$$\mathbf{f}_{ij} = \left(\sum_{e \in \mathcal{T}_i \cap \mathcal{T}_j} V_e \mathbf{P}_e \mathbf{K}_e^{-1} \right) (\mathbf{x}_j - \mathbf{x}_i). \quad (11)$$

Since $\mathbf{f}_{ij}$ may contain nonlinear terms, Equation 10 is preferably solved using the optimization-based implicit Euler time integration method [Gast et al. 2015]. Thus, the system can be converted into the following optimization problem:

$$\mathcal{L}(\mathbf{y}) = \frac{1}{2h^2} \sum_i m_i \|\mathbf{y}_i - \mathbf{y}_i^*\|^2 + \sum_e V_e \Psi_e(\mathbf{y}), \quad (12)$$

where $\mathbf{y}_i^* = \mathbf{y}_i + h\mathbf{v}_i$, and Ψ_e represents the strain energy density function.

4.2 A Pairwise Force Model for Hyperelastic Continua

In hyperelasticity, the strain energy density function Ψ is typically formulated as a scalar function of deformation gradient $\mathbf{F}$, i.e., $\Psi = \Psi(\mathbf{F})$. For isotropic materials, Ψ can be further simplified as a function of the three principal invariants of Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F}$, i.e., $\Psi = \Psi(I_1, I_2, J)$. Following [Smith et al. 2019], the first Piola-Kirchhoff stress $\mathbf{P}$ is then derived as:

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}} = 2 \frac{\partial \Psi}{\partial I_1} \mathbf{F} + 2 \frac{\partial \Psi}{\partial I_2} (I_1 \mathbf{F} - \mathbf{F} \mathbf{C}) + J \frac{\partial \Psi}{\partial J} \mathbf{F}^{-T}. \quad (13)$$

The above expression for $\mathbf{P}$ not only simplifies the calculation, but also makes it easier to guarantee the convergence for iterative methods in solving the nonlinear optimization problem in Equation 12. More details will be discussed in Section 5.1.

Within element e , linear interpolation gives $\mathbf{y}_j - \mathbf{y}_i = \mathbf{F}_e \xi_{ij}$ and hence $\mathbf{F}_e^{-1}(\mathbf{y}_j - \mathbf{y}_i) = \xi_{ij}$. Substituting this identity into Equation 11 yields the following pairwise force between particles i and j :

$$\mathbf{f}_{ij} = \sum_{e \in \mathcal{T}_i \cap \mathcal{T}_j} V_e \mathbf{P}_e \mathbf{K}_e^{-1} \mathbf{F}_e^{-1} (\mathbf{y}_j - \mathbf{y}_i). \quad (14)$$

Remove the shape tensor $\mathbf{K}$: To further simplify the formulation, we propose a novel solution to eliminate the inverse of the shape tensor $\mathbf{K}_e^{-1}$ by introducing the following identical relation:

$$\sum_{j \in \mathcal{V}_e} \mathbf{K}_e^{-1} \xi_{ij} = \sum_{j \in \mathcal{V}_e} w_{ij,e} \xi_{ij} \quad (15)$$

where $w_{ij,e}$ represents the geometric weight associated with the edge ij of element e (different from the weight function $\underline{\omega}(\xi)$). When the edge vectors $\{\xi_{i1}, \xi_{i2}, \xi_{i3}\}$ are linearly independent (non-degenerate tetrahedron), they form a complete basis for $\mathbb{R}^3$. Consequently, the vector $\mathbf{K}_e^{-1} \sum_j \xi_{ij}$ can be uniquely expressed as the linear combination $\sum_j w_{ij,e} \xi_{ij}$. A practical solution in calculating the weights for each tetrahedron is provided in section 2 of the supplemental document. Specifically, the weight vector $\mathbf{w}_{ie}$ for each tetrahedron vertex can be calculated as $\mathbf{w}_{ie} = \mathbf{G}_{ie}^{-1} \mathbf{1}$, where $\mathbf{G}_{ie}$ is the Gram matrix of vertex i and $\mathbf{1} = [1, 1, 1]^T$. Inserting Equation 15 into Equation 14 yields a more concise expression for the pairwise force between particles i and j :

$$\mathbf{f}_{ij} = \underbrace{\sum_{e \in \mathcal{T}_i \cap \mathcal{T}_j} w_{ij,e} (V_e \mathbf{P}_e \mathbf{F}_e^{-1})}_{\mathbf{L}_{ij} \in \mathbb{R}^{3 \times 3}} (\mathbf{y}_j - \mathbf{y}_i). \quad (16)$$

Hereafter, $\mathbf{L}_{ij}$ is referred to as the *pairwise tension tensor*, as it has the dimensions of energy per unit area. As shown in Supplementary Material, Section B, $w_{ij,e} = w_{ji,e}$; therefore, the pairwise force model in Equation 16 satisfies Newton's third law, i.e., $\mathbf{f}_{ij} = -\mathbf{f}_{ji}$.

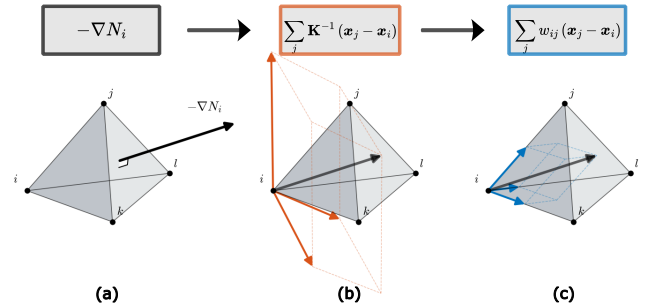


Fig. 2. Perspectives on force modeling across different simulation frameworks. (a) the FEM perspective; (b) the peridynamic perspective; and (c) our proposed perspective.

The nonlocal pairwise force model described above offers several key advantages over classical continuum mechanics-based models, particularly when applied to hyperelastic materials undergoing discontinuous deformation or contact. First, the structure of $\mathbf{L}_{ij}$ significantly simplifies the numerical modeling of arbitrary hyperelastic materials, as it involves only the first-order derivative of the energy density function with respect to the principal stretches of the deformation gradient. Second, discontinuous deformations involving contact and friction can be represented consistently by virtual tetrahedra, as detailed in Section 6. Finally, the implementation of an implicit solver is highly compatible with modern GPU architectures. This is because $\mathbf{L}_{ij}$ can be efficiently computed using only $\mathbf{P}_e, \mathbf{F}_e \in \mathbb{R}^{3 \times 3}$ and $w_{ij,e} \in \mathbb{R}$, without requiring the costly evaluation of $\partial \mathbf{P} / \partial \mathbf{F} \in \mathbb{R}^{9 \times 9}$. Moreover, symmetry and positive definiteness

can be easily guaranteed via a semi-implicit decomposition and an analytical projection strategy, as demonstrated in Section 5.

5 Semi-Implicit Pairwise Descent

Given both the internal and boundary forces in the nonlocal pairwise form, the position-based governing equation derived from Equation 12 and Equation 16 can be uniquely expressed as follows:

$$\frac{m_i}{h^2}(\mathbf{y}_i - \mathbf{y}_i^*) - \underbrace{\sum_j \sum_{e \in \mathcal{T}_i \cap \mathcal{T}_j} \mathbf{w}_{ij,e} (V_e \mathbf{P}_e \mathbf{F}_e^{-1})}_{\mathbf{L}_{ij}} (\mathbf{y}_j - \mathbf{y}_i) = 0. \quad (17)$$

To solve the nonlinear system above, a Newton-type method linearizes the nonlinear term by introducing the 9×9 Hessian $\partial \mathbf{P} / \partial \mathbf{F}$, which is expensive to compute and difficult to project to a positive-definite matrix. We instead solve the nonlinear system using the *successive substitution method* [Langtangen and Mardal 2019], a Jacobi-style iterative method that linearizes the nonlinear term by keeping it fixed at each iteration ($\mathbf{L} = \mathbf{L}^k$), without computing the Hessian.

In this Jacobi-style iteration, all $\mathbf{y}_j$ are fixed at $\mathbf{y}_j^k$ while computing the update $\Delta \mathbf{y}_i^{k+1} = \mathbf{y}_i^{k+1} - \mathbf{y}_i^k$ from the following linear system:

$$\left(\frac{m_i}{h^2} \mathbf{I} + \sum_j \mathbf{L}_{ij}^k \right) \Delta \mathbf{y}_i^{k+1} = - \overbrace{\left(\frac{m_i}{h^2} (\mathbf{y}_i^k - \mathbf{y}_i^*) - \sum_j \mathbf{L}_{ij}^k (\mathbf{y}_j^k - \mathbf{y}_i^k) \right)}^{\nabla \mathcal{L}_i(\mathbf{y}^k)}. \quad (18)$$

Thus, the update $\Delta \mathbf{y}_i^{k+1}$ can be solved by $-(\frac{m_i}{h^2} \mathbf{I} + \sum_j \mathbf{L}_{ij}^k)^{-1} \nabla \mathcal{L}_i(\mathbf{y}^k)$. Consequently, the position of each particle can be updated independently using only information from neighboring particles. By bypassing the Hessian evaluation inherent to second-order techniques (such as VBD method), our first-order method incurs significantly lower computational and register overheads, making it ideal for parallel implementation. Furthermore, an analytical projection strategy is proposed to ensure convergence.

5.1 Positive Definiteness and Convergence

Note when the 3×3 matrix $\mathbf{L}_{ij}$ is positive semi-definite, the coefficient matrix $(\frac{m_i}{h^2} \mathbf{I} + \sum_j \mathbf{L}_{ij}^k)^{-1}$ is accordingly positive definite, we have

$$\nabla \mathcal{L}_i^k \cdot (\mathbf{y}^{k+1} - \mathbf{y}^k) = -\nabla \mathcal{L}_i^k \cdot \left(\frac{m_i}{h^2} \mathbf{I} + \sum_j \mathbf{L}_{ij}^k \right)^{-1} \nabla \mathcal{L}_i^k \leq 0. \quad (19)$$

The direction defined by $\Delta \mathbf{y}_i^{k+1} = \mathbf{y}_i^{k+1} - \mathbf{y}_i^k$ is clearly a descent direction whenever $\|\nabla \mathcal{L}_i^k\| \neq 0$.

However, the positive definiteness of $\mathbf{L}_{ij}$ is not guaranteed under all conditions, particularly during compression. To address this issue, we present an extension of the semi-implicit successive substitution method (SISSM), originally introduced by Lu et al. [2023] for original peridynamics, to our reformulated FEM model. Its core idea is to decompose the coefficient matrix into positive and negative components (i.e., $\mathbf{L}_{ij} = \mathbf{L}_{ij}^+ + \mathbf{L}_{ij}^-$), treating them implicitly and explicitly, respectively. Specifically, the explicit treatment of the negative components means they are excluded from the coefficient matrix corresponding to the implicit variables on the left-hand side of Equation 18.

Considering the definiteness of the pairwise tension tensor for each finite element is jointly defined by the values of $\mathbf{w}$ and $\mathbf{P}\mathbf{F}^{-1}$, our solution to decompose the pairwise tension tensor $\mathbf{L}$ is as follows:

$$\begin{aligned} \mathbf{L}_{ij,e}^+ &= V_e \left(\mathbf{w}_{ij,e}^+ (\mathbf{P}\mathbf{F}^{-1})_e^+ + \mathbf{w}_{ij,e}^- (\mathbf{P}\mathbf{F}^{-1})_e^- \right) \\ \mathbf{L}_{ij,e}^- &= V_e \left(\mathbf{w}_{ij,e}^- (\mathbf{P}\mathbf{F}^{-1})_e^+ + \mathbf{w}_{ij,e}^+ (\mathbf{P}\mathbf{F}^{-1})_e^- \right) \end{aligned} \quad (20)$$

where the decomposition for $\mathbf{w}$ is done as follows

$$\mathbf{w}^+ = \begin{cases} \mathbf{w}, & \mathbf{w} > 0 \\ 0, & \mathbf{w} \leq 0 \end{cases}, \quad \mathbf{w}^- = \begin{cases} 0, & \mathbf{w} > 0 \\ \mathbf{w}, & \mathbf{w} \leq 0 \end{cases}, \quad (21)$$

and the $(\mathbf{P}\mathbf{F}^{-1})$ should be decomposed into a positive definite part and a negative definite part. This decomposition can be achieved by traditional eigenvalue projection methods based on singular value decomposition (SVD), such as the eigenvalue clamping projection strategy [Teran et al. 2005] and the absolute eigenvalue projection strategy [Chen et al. 2024a]. Although applying these projection methods to our 3×3 first Piola-Kirchhoff stress matrix $\mathbf{P}$ is much faster than applying them to a 9×9 or 12×12 Hessian matrix, we still want to avoid the costly SVD.

Thus, we propose an analytical projection strategy without SVD for isotropic hyperelastic materials, which is obtained by multiplying Equation 13 with $\mathbf{F}^{-1}$:

$$\mathbf{P}\mathbf{F}^{-1} = 2 \frac{\partial \Psi}{\partial \mathbf{I}_1} \mathbf{I} + 2 \frac{\partial \Psi}{\partial \mathbf{I}_2} (\mathbf{I}_1 \mathbf{I} - \mathbf{F}^T \mathbf{F}) + J \frac{\partial \Psi}{\partial J} (\mathbf{F}\mathbf{F}^T)^{-1}. \quad (22)$$

The above equation can be decoupled into two components: the positive definite matrices including $\mathbf{I}$, $\mathbf{F}^T \mathbf{F}$ and $(\mathbf{F}\mathbf{F}^T)^{-1}$ as well as the scalar values including $\frac{\partial \Psi}{\partial \mathbf{I}_1}$, $\frac{\partial \Psi}{\partial \mathbf{I}_2} \mathbf{I}_1$ and $J \frac{\partial \Psi}{\partial J}$, all of which can be efficiently computed without SVD. The above observation allows us to easily partition $(\mathbf{P}\mathbf{F}^{-1})$ into a positive definite part $(\mathbf{P}\mathbf{F}^{-1})^+$ and a negative definite part $(\mathbf{P}\mathbf{F}^{-1})^-$ based on their values.

For example, we present an decomposition for the stable neo-hookean energy (Equation (13) in [Smith et al. 2018]) in case of $J \geq 0$ as follows:

$$\begin{aligned} \mathbf{P}\mathbf{F}^{-1} &= \mu \mathbf{I} - \mu J (\mathbf{F}\mathbf{F}^T)^{-1} + \lambda (J - 1) J (\mathbf{F}\mathbf{F}^T)^{-1} \\ (\mathbf{P}\mathbf{F}^{-1})^+ &= \mu \mathbf{I} + \lambda J^2 (\mathbf{F}\mathbf{F}^T)^{-1} \\ (\mathbf{P}\mathbf{F}^{-1})^- &= -(\mu + \lambda) J (\mathbf{F}\mathbf{F}^T)^{-1}. \end{aligned} \quad (23)$$

For the deteriorated case where $J < 0$, we can set $(\mathbf{P}\mathbf{F}^{-1})^+ = \mathbf{P}\mathbf{F}^{-1}$ and $(\mathbf{P}\mathbf{F}^{-1})^- = 0$ to preserve positive definiteness. Therefore, the stable Neo-Hookean model [Smith et al. 2018] remains well-defined for inverted and nearly singular configurations. In our practical implementation, when $\mathbf{F}$ becomes singular, we evaluate the gradient term $\mathbf{w}_{ij} \mathbf{P}\mathbf{F}^{-1} (\mathbf{y}_j - \mathbf{y}_i)$ using its equivalent form $\mathbf{w}_{ij} \mathbf{P} (\mathbf{x}_j - \mathbf{x}_i)$, thereby avoiding an explicit evaluation of $\mathbf{F}^{-1}$. When calculating the term $\mathbf{L}$, we use the following formula

$$(\mathbf{F}\mathbf{F}^T)_\epsilon^{-1} := \frac{\text{adj}(\mathbf{F}\mathbf{F}^T)}{\max(J^2, \epsilon^2)}, \quad (24)$$

to prevent the pairwise coefficient matrix from becoming unbounded as $J \rightarrow 0$.

After inserting Equation 20 into Equation 18, the update can be reformulated into $\Delta \mathbf{y}_i^{k+1} = -(\frac{m_i}{h^2} \mathbf{I} + \sum_j \mathbf{L}_{ij}^{k+})^{-1} \nabla \mathcal{L}_i(\mathbf{y}^k)$, with $\mathbf{L}_{ij}^k$ treated explicitly on the right-hand side. As shown in Supplementary Material, Section D, the analytical projection strategy makes

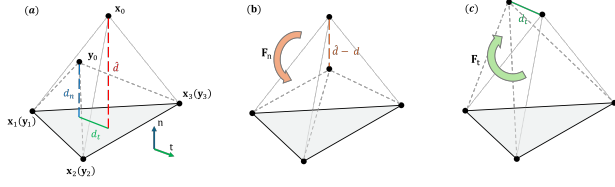


Fig. 3. Demonstration of virtual boundary elements for contact and friction. (a) the VBE in the reference configuration; (b) the VBE with normal deformation; (c) the VBE with shearing deformation.

every local update matrix positive definite and hence produces a descent direction. Combined with line search, it ensures convergence. Figures 11 and 8 further demonstrate its stability and line-search behavior in practice. The detailed SIPD procedure is provided in Algorithm 1.

6 Anisotropic Modeling for Contact and Friction

To uniformly model contact and friction within our nonlocal pairwise model, we introduce virtual boundary elements (VBE) for proximity pairs whose distances are below a given activation threshold $\hat{d}$, motivated by Wang et al. [2023] and Müller et al. [2015].

Without loss of generality, we consider a vertex-triangle proximity pair in Figure 3(a), where (y_0, y_1, y_2, y_3) denote the vertex coordinates in the deformed configuration. The corresponding vertex coordinates in the reference configuration, where the contact and friction energies are zero, are defined as

$$\mathbf{x}_0 = \mathbf{y}_0 - (d_n - \hat{d})\mathbf{n} - d_t\mathbf{t}, \quad \mathbf{x}_{1,2,3} = \mathbf{y}_{1,2,3}, \quad (25)$$

where d_n denotes the signed distance measuring the normal deformation along the normal vector $\mathbf{n}$, while d_t denotes the unsigned distance measuring the magnitude of the shearing deformation along the tangential vector $\mathbf{t}$. To control the normal and tangential deformations independently, we construct two virtual tetrahedra, as shown in Figure 3(b) and Figure 3(c). The deformation gradients and their inverses for the normal and tangential components are then formulated separately as

$$\begin{aligned} \mathbf{F}_n &= \mathbf{I} + \left(\frac{d_n}{\hat{d}} - 1\right)\mathbf{nn}^T, & \mathbf{F}_n^{-1} &= \mathbf{I} + \left(\frac{\hat{d}}{d_n} - 1\right)\mathbf{nn}^T \\ \mathbf{F}_t &= \mathbf{I} + \frac{d_t}{\hat{d}}\mathbf{tn}^T, & \mathbf{F}_t^{-1} &= \mathbf{I} - \frac{d_t}{\hat{d}}\mathbf{tn}^T \end{aligned} \quad (26)$$

To control the deformation along the normal and tangential directions separately, we use two separate energy density functions. For the normal deformation, we apply the fourth invariant of the Cauchy-Green deformation tensor [Lin et al. 2024] $I_4(\mathbf{n}) = \|\mathbf{F}\mathbf{n}\|^2$, which is commonly used within fiber-reinforced materials [Spencer 1972; Wolper et al. 2020], to define the energy density function:

$$\Psi_n = \frac{1}{2}\kappa C_n^2, \quad C_n = \text{sgn}(J)\sqrt{\|\mathbf{F}\mathbf{n}\|^2} - 1 \geq 0, \quad (27)$$

where κ denotes the stiffness, $\text{sgn}(J)$ returns the sign of J . For shearing deformation, we define the potential as a combination of the smoothed friction-velocity relation used in IPC [Li et al. 2020] and the fourth invariant of the corotated Cauchy-Green deformation tensor [Kugelstadt et al. 2018b], defined as $I_4(\mathbf{n}) = \|(\mathbf{F} - \mathbf{R})\mathbf{n}\|^2$.

Algorithm 1: Semi-Implicit Pairwise Descent (SIPD)

Input: Mesh $(\mathbf{X}, \mathcal{T})$, surface $\mathcal{S}$, state $(\mathbf{x}^n, \mathbf{v}^n)$, time step h , iteration count K
Output: Updated state $(\mathbf{x}^{n+1}, \mathbf{v}^{n+1})$
 $\mathbf{y}^* \leftarrow \text{InertialPrediction}(\mathbf{x}^n, \mathbf{v}^n, h)$; $\mathbf{y}^0 \leftarrow \mathbf{y}^*$;
 $\mathcal{C} \leftarrow \text{DCD-BroadPhase}(\mathcal{S}, \mathbf{x}^n, \mathbf{y}^*)$;
for $k \leftarrow 0$ **to** $K - 1$ **do**
 if $k \bmod 4 = 0$ **then**
 $\mathcal{A} \leftarrow \text{DCD-NarrowPhase}(\mathcal{C}, \mathbf{y}^k)$;
 $\mathcal{V} \leftarrow \text{BuildVBES}(\mathcal{A})$;
 end
 $\mathbf{g}_i \leftarrow \frac{m_i}{h^2}(\mathbf{y}_i^k - \mathbf{y}_i^*)$; $\mathbf{A}_i \leftarrow \frac{m_i}{h^2}\mathbf{I}$;
 foreach $e \in \mathcal{T} \cup \mathcal{V}$ **in parallel do**
 Compute $\mathbf{F}_e$, $\mathbf{P}_e \mathbf{F}_e^{-1}$, and its analytical split;
 foreach vertex pair (i, j) of e **do**
 Assemble $\mathbf{L}_{ij,e}$ into $\mathbf{g}_i$ and $\mathbf{L}_{ij,e}^+$ into $\mathbf{A}_i$;
 end
 end
 $\Delta \mathbf{y}_i \leftarrow -\mathbf{A}_i^{-1} \mathbf{g}_i$; // independent 3×3 solves
 $\alpha \leftarrow \text{LineSearch}(\mathcal{L}, \mathbf{y}^k, \Delta \mathbf{y})$; $\mathbf{y}^{k+1} \leftarrow \mathbf{y}^k + \alpha \Delta \mathbf{y}$;
end
 $\mathbf{x}^{n+1} \leftarrow \mathbf{y}^K$; $\mathbf{v}^{n+1} \leftarrow (\mathbf{x}^{n+1} - \mathbf{x}^n)/h$;

For pure shearing deformation, the potential can be further simplified by neglecting rotational effects and setting $\mathbf{R} = \mathbf{I}$, yielding the following expression:

$$\Psi_t = \mu \lambda f(C_t), \quad C_t = \sqrt{\|(\mathbf{F}_t - \mathbf{I})\mathbf{n}\|^2} = 0 \quad (28)$$

where λ is the contact force magnitude, μ is the local friction coefficient, and f is a polynomial function defining a smooth displacement-based friction response. Taking derivative of both Ψ_n and Ψ_t with respect to $\mathbf{F}$ and multiplying both equations with $\mathbf{F}^{-1}$ yields the following first Piola-Kirchhoff stress tensor

$$(\mathbf{P}\mathbf{F}^{-1})_n = \kappa(1 - \frac{\hat{d}}{d_n})\mathbf{nn}^T, \quad (\mathbf{P}\mathbf{F}^{-1})_t = \mu \lambda f'(C_t)\mathbf{tn}^T. \quad (29)$$

Finally, we apply analytical projection to both $(\mathbf{P}\mathbf{F}^{-1})_n$ and $(\mathbf{P}\mathbf{F}^{-1})_t$, whose formulations are

$$\begin{aligned} (\mathbf{P}\mathbf{F}^{-1})_n^+ &= \kappa \mathbf{nn}^T, & (\mathbf{P}\mathbf{F}^{-1})_t^+ &= \mu \lambda f'(C_t) \frac{d_t}{\hat{d}} \mathbf{I}, \\ (\mathbf{P}\mathbf{F}^{-1})_n^- &= -\kappa \frac{\hat{d}}{d_n} \mathbf{nn}^T, & (\mathbf{P}\mathbf{F}^{-1})_t^- &= \mu \lambda f'(C_t) \left(\mathbf{tn}^T - \frac{d_t}{\hat{d}} \mathbf{I} \right). \end{aligned} \quad (30)$$

For edge-edge (EE) collisions, the pairwise force is derived by replacing $\mathbf{n}$ with the normalized cross product of the two edge directions. A special case arises when the two edges are parallel; in this situation, the EE pair is discarded, as the resulting tetrahedron has zero volume and therefore does not contribute to boundary handling. Compared with traditional collision-handling methods based on volume constraints [Müller et al. 2015], the anisotropic material model formulated using the fourth invariant ensures that collision responses and friction are strictly confined to the normal and tangential direction strictly, and do not introduce spurious tangential expansion in the material behavior. The scope and limitations of this smooth contact and friction formulation are discussed in Supplementary Material, Section C.

7 Results and Discussion

Table 1. Simulation parameters and performance metrics for various test scenarios.

Example Name	Number of		Material			Contact & Friction		Solver Parameters		Time per step	
	Vert.	Tet./Tri.	Type	ρ	Stiffness	κ_c	μ_c, ϵ_a	h	Iterations	avg/max	
Twisting Beam	97,290	416,852	Stable NeoHookean	1000	$E = 1e8, \nu = 0.48$	$5 \cdot 10^3$	$0.1; 10^{-2}$	3ms	5	0.76ms/0.90ms	
Dropping Bunny	225,316	1,034,600	Stable NeoHookean	1000	$E = 6e7, \nu = 0.46$	$5 \cdot 10^3$	$0.1; 10^{-2}$	3ms	4	2.8ms/3.1ms	
Cloth	9,100	17,590	StVK	1000	$E = 1e8, \nu = 0.45$	N/A	N/A	1ms	20	1.83ms/2.1ms	
Beam Fall	463	1,479	Stable NeoHookean	1000	$E \in \{4e5, 8e5, 1.6e6, 8e6\}$ $\nu = 0.45$	N/A	N/A	1ms	10/25/50	avg: 0.3ms/0.7ms/1.3ms max: 0.3ms/0.8ms/2.1ms	
Sliding Cube	24	30	Stable NeoHookean	1000	$E = 1e7, \nu = 0.45$	$5 \cdot 10^3$	$\{0.0, 0.2, 0.4, 0.6, 0.8\}; 10^{-2}$	3ms	3	0.02ms/0.03ms	
Noodle	697,005	1,849,230	Stable NeoHookean	100	$E = 1e8, \nu = 0.46$	$5 \cdot 10^3$	$0.1; 10^{-2}$	3ms	4	6.5ms/7.9ms	
Pushing bunny	32,188	147,800	Stable NeoHookean	1000	$E = 1e8, \nu = 0.46$	10^4	$0.1; 10^{-2}$	3ms	7	0.78ms/0.91ms	
Dropping Squishy Ball	167,445	478,500	Stable NeoHookean	1000	$E = 1e8, \nu = 0.46$	10^4	$0.1; 10^{-2}$	3ms	4	0.95ms/1.13ms	
Stretching Cube	64,000	355,914	Stable NeoHookean	1000	$E = 1e8, \nu = \{0.1, 0.2, 0.46, 0.48, 0.49, 0.499\}$	N/A	N/A	3ms	40	3.2ms/3.4ms	
Bunny recover	3,219	14,308	Corotated linear	1000	$E = 1e8, \nu = 0.45$	N/A	N/A	30ms	20	0.28ms/0.32ms	
Octopus recover	3,430	13,578	Stable NeoHookean	1000	$E = 1e8, \nu = 0.46$	N/A	N/A	3ms	6	0.17ms/0.19ms	
Armadillo recover	15,228	62,770	Stable NeoHookean	1000	$E = 1e8, \nu = 0.45$	N/A	N/A	3ms	20	1.52ms/1.68ms	

Table 2. Comparison of descent directions and required derivative information.

Method	Descent Direction Δy_i	Derivative Information
GD	$-\nabla \mathcal{L}_i$	First order
Newton	$-(\mathbf{M}/h^2 + \mathbf{H})^{-1} \nabla \mathcal{L}_i$	Full Hessian $\mathbf{H}$
VBD	$-(\mathbf{M}_{ii}/h^2 + \mathbf{H}_{ii})^{-1} \nabla \mathcal{L}_i$	Vertex Hessian block $\mathbf{H}_{ii}$
Jacobi-Pre.	$-(\mathbf{M}_{ii}/h^2 + \text{diag}\{\mathbf{H}_{ii}\})^{-1} \nabla \mathcal{L}_i$	Diagonal Hessian
Ours	$-(\mathbf{M}_{ii}/h^2 + \mathbf{L}_{ii}^+)^{-1} \nabla \mathcal{L}_i$	First-order $\mathbf{P}, \mathbf{F}$

Table 2 summarizes the derivative information required by each descent direction. Newton, VBD, and Jacobi preconditioning construct full, block, or diagonal Hessian information from the constitutive derivative $\partial \mathbf{P} / \partial \mathbf{F}$, respectively, whereas our method constructs $\mathbf{L}_{ii}^+$ directly from the first-order quantities $\mathbf{P}$ and $\mathbf{F}$. Their measured convergence with respect to iteration count and wall-clock time is reported in Figure 5.

We implement our solver and other comparison algorithms in CUDA on an NVIDIA RTX 5090 GPU using single-precision arithmetic. We adopt the stable NeoHookean model (Equation 23) for both volumetric solids and cloth. Due to our method's low register usage (under 56 registers per thread), we maintain high GPU occupancy and use a thread block size of 256 for all experiments to maximize throughput. We set $\hat{d} = 0.03$ for all collision scenes.

7.1 Collision Handling

We employ Discrete Collision Detection for collisions and friction. We restrict detection to the surface triangle meshes of our tetrahedral models. Each time step begins with a broad-phase filter to identify collision candidates. During solver iterations, narrow-phase detection extracts active collision pairs to compute contact normals and penetration depths every 4 iterations. We handle both Vertex-Face (V-F) and Edge-Edge (E-E) collisions, filtering the degenerate coplanar E-E cases.

Our formulation treats collision pairs as "virtual tetrahedra," allowing them to be stored in the same element list as elastic tetrahedra. This enables a single CUDA kernel to process elasticity, collision, and friction by only varying the $\mathbf{P}\mathbf{F}^{-1}$ matrix. Figures 1, Figures 6 and Figures 10 demonstrate the robustness and efficiency of this unified treatment.

7.2 Positive definiteness

To validate our analytical projection strategy for SIPD, we test scenarios with high Poisson's ratios, where non-positive definiteness issues are more pronounced [Chen et al. 2024a]. As illustrated in Figure 11, under conditions of a high Poisson's ratio ($\nu > 0.48$) and large deformations, the non-positive definiteness problem becomes

severe. Directly using a non-positive definite matrix for iterations leads to divergence, as seen in the VBD method without positive definiteness enforcement, which fails in $\nu > 0.48$. In contrast, our method maintains stability throughout the simulation.

Moreover, we evaluate our analytical projection strategy under stretching, compression, and bending with $\nu = 0.499$, using the eigenvalue clamping and absolute eigenvalue projection strategies as baselines (Figure 8). Our analytical projection accepts the full step $\alpha = 1$ in most iterations across all three deformation modes and therefore requires little line-search backtracking.

7.3 Stress Test

We test our method under three extreme deformation scenarios to show the stability of SIPD in Figure 15. The first one is a bunny model with randomly initialized positions around the surface of a sphere. The second one is a highly flattened armadillo. The third is an octopus model with its arms strongly stretched.

Figure 10 is a challenging frictional contact case where two beams are twisted together. Figure 6 is a large-scale collision scenario in which 81 long noodles are dropped into a bowl, producing over one million contact pairs, substantially more than the 697K vertex degrees of freedom. Our method passes these stress tests without any stability issues, demonstrating the robustness of our SIPD method under extreme conditions.

7.4 Comparisons with other descent methods

By viewing the descent direction as $\Delta y_i = -\mathbf{A} \nabla \mathcal{L}$, our solver effectively constructs a Hessian-free preconditioning matrix $\mathbf{A}$, and the traditional methods approximate the Hessian $\mathbf{H}$ directly. While the Hessian matrix improves convergence in Newton-type methods, it introduces the heavy computational burden of fourth-order tensors $\partial \mathbf{P} / \partial \mathbf{F} \in \mathbb{R}^{9 \times 9}$ and positive definite projection steps. This limitation affects even efficient vertex-wise local descent approaches like VBD. Our method, instead, derives the preconditioning matrix from second-order tensor $\mathbf{P}\mathbf{F}^{-1} \in \mathbb{R}^{3 \times 3}$ without Hessian, and provides an SVD-free positive definite projection strategy. Although the omission of the original Hessian information degrades the per-iteration convergence rate, our method involves only second-order tensor computations and is completely SVD-free. This significantly reduces the per-iteration overhead, achieving a highly effective trade-off that minimizes the total convergence time (**1.6x** faster than VBD).

To prove this, we compare our method with several mainstream methods in table 2: Gradient Descent (GD) uses the identity matrix

as the preconditioning matrix. Newton-type Methods (e.g., full Newton [Sin et al. 2013], VBD, BlockJacobi [Chen et al. 2024b], Jacobi Preconditioning [Wang and Yang 2016]) involve constructing the global Hessian $\mathbf{H}$ or its block-diagonal counterparts $\mathbf{H}_{ii}$. To exclude the influence of graph coloring on convergence rates and execution time, we implement both our method and VBD in the Coloring Gauss-Seidel (GS) and Jacobi frameworks: Our-GS, Our-Jacobi, VBD and BlockJacobi. Then, we compare the convergence behavior of these methods in Figure 5 where a large compressed model is released. As illustrated in Figure 5, although our methods (Our-GS and Our-Jacobi) exhibit a slower per-iteration convergence rate (left figure) compared to the VBD and BlockJacobi methods, they achieve a substantially shorter overall convergence time (right figure) due to a 33%-40% reduction in computational overhead brought by small register usage in Section 7.5.

Furthermore, as noted by [Chen et al. 2024b], BlockJacobi (Jacobi framework) incurs significant line search overhead, resulting in slower overall performance compared to VBD (GS framework). Conversely, our method is highly compatible with the Jacobi framework. Our projection strategy requires markedly fewer line search iterations (see Figure 8), enabling Our-Jacobi to outperform Our-GS.

7.5 Performance

In this section, we evaluate our method's advantage by analyzing register usage, which critically determines GPU parallelism. High register usage severely limits the number of active thread blocks on an Streaming Multiprocessor (SM), reducing GPU occupancy and its ability to hide memory latency. For instance, within the bottleneck kernel for computing the gradient and descent direction, even the efficient VBD method requires 102 registers per thread (with the dPdF computation alone taking at least 81) and an occupancy of 33.3%. Conversely, our approach demands only 56, giving an occupancy of 66.7%. This substantial reduction in register usage directly translates to higher instruction throughput and better hardware utilization. As a result, Figure 4 demonstrates that our methods (Our-GS and Our-Jacobi) achieve remarkable performance gains, costing only 60% and 67% of the time required by VBD and BlockJacobi, respectively.

7.6 Comparisons with Peridynamics

The mechanical response in traditional peridynamics relies heavily on the spatial distribution of the horizon, resulting in computational inaccuracies at physical boundaries caused by horizon truncation.

Our approach reformulates FEM using non-local concepts of peridynamics, achieving theoretical equivalence to standard FEM and fundamentally eliminating these boundary issues.

To verify this, we simulate a tensile test on a square plate with a central circular hole (dimensions: 10cm $\times$ 10cm, discretization spacing $dx = 0.185$ cm, hole radius: 1cm) in Figure 7. The top and bottom two layers of particles were designated as traction boundaries, subject to an outward tensile velocity of 10cm/s. We compare the velocity fields of three methods: traditional peridynamics, our SIPD, and standard FEM. For this setup, the peridynamics horizon size is set to $2\Delta x$ and the resulting system is solved using the method of [Lu et al. 2023]; the FEM system is solved using a Newton method. Starting from the same initial state, traditional peridynamics and SIPD are each run for 100 iterations, while the Newton-based FEM solver is run for 20 iterations. The result shows that peridynamics non-physical discontinuities near the hole due to aforementioned horizon truncation, whereas our method remains smooth. Disregarding minor floating-point error, our method is computationally equivalent to the standard FEM.

8 Conclusion

In summary, the proposed semi-implicit pairwise descent offers an efficient and robust solution to nonlinear optimization problems in physics-based simulation. By avoiding second-order derivatives and relying solely on first-order energy gradients, the method reduces computational complexity while remaining well suited for GPU parallelization. The reformulation within a nonlocal peridynamics framework enables a unified treatment of internal and boundary forces. The semi-implicit splitting and analytical projection strategy guarantees positive definiteness of the local stiffness matrices. As a result, the method supports stable, parallel particle updates with minimal overhead, making it particularly attractive for large-scale, real-time elastic simulations.

Despite the promising results, our approach has several limitations. Our contact and friction model remains dependent on mesh resolution, which may lead to spurious contact forces. Although we revisit FEM from a nonlocal perspective, the current method does not yet support fracture simulation. Our current penalty-based contact formulation does not strictly prevent penetration. To improve robustness in more challenging simulations involving extreme deformation and complex contact, a promising direction is to combine SIPD with IPC [Li et al. 2020], using its barrier formulation and continuous collision detection to enforce nonpenetration while retaining SIPD's Hessian-free parallel updates. Developing a compatible formulation for IPC contact and friction without sacrificing GPU efficiency is left for future work.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the New Generation Artificial Intelligence–National Science and Technology Major Project of China (No. 2025ZD0123902), the National Natural Science Foundation of China (Nos. 92570206 and 62302490), and the Basic Research Project of ISCAS (No. ISCAS-JCMS-202403).

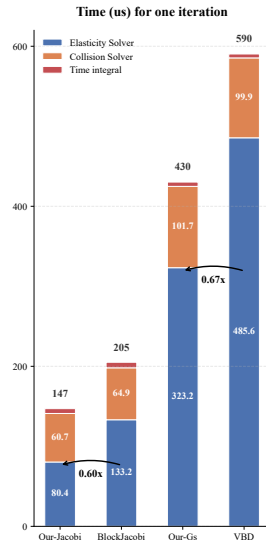


Fig. 4. Time overhead comparison in the Twisting Beam scene.

References

- Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 594–603.
- R. Bridson, S. Marino, and R. Fedkiw. 2005. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) (SIGGRAPH '05). Association for Computing Machinery, New York, NY, USA, 3–es.
- Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. 2024b. Vertex block descent. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–16.
- Honglin Chen, Hsueh-Ti Derek Liu, David IW Levin, Changxi Zheng, and Alec Jacobson. 2024a. Stabler neo-hookean simulation: Absolute eigenvalue filtering for projected newton. In *ACM SIGGRAPH 2024 Conference Papers*. 1–10.
- Wei Chen, Fei Zhu, Jing Zhao, Sheng Li, and Guoping Wang. 2018. Peridynamics-based fracture animation for elastoplastic solids. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 112–124.
- Yizhou Chen, Yushan Han, Jingyu Chen, and Joseph Teran. 2023. Position-based nonlinear gauss-seidel for quasistatic hyperelasticity. *arXiv preprint arXiv:2306.09021* (2023).
- Mehmet Dorduncu, Huilong Ren, Xiaoying Zhuang, Stewart Silling, Erdogan Madenci, and Timon Rabczuk. 2024. A review of peridynamic theory and nonlocal operators along with their computer implementations. *Computers & Structures* 299 (2024), 107395.
- 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.
- Chris Giles, Elie Diaz, and Cem Yuksel. 2025. Augmented Vertex Block Descent. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–12.
- Chengguizhi Han, Tao Xue, and Mridul Aanjaneya. 2021. A Lagrangian Particle-based Formulation for Coupled Simulation of Fracture and Diffusion in Thin Membranes. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 97–108.
- David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous contact mechanics. In *ACM SIGGRAPH 2009 papers*. 1–12.
- Xiaohei He, Huamin Wang, and Enhua Wu. 2017. Projective peridynamics for modeling versatile elastoplastic materials. *IEEE transactions on visualization and computer graphics* 24, 9 (2017), 2589–2599.
- Kemeng Huang, Floyd M Chitalu, Huancheng Lin, and Taku Komura. 2024. GIPC: Fast and stable gauss-newton optimization of IPC barrier energy. *ACM Transactions on Graphics* 43, 2 (2024), 1–18.
- G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Grenoble, France) (SCA '04). Eurographics Association, Goslar, DEU, 131–140. doi:10.1145/1028523.1028541
- Jingyu Kang, Xiaodong Fu, Qian Sheng, Xing Wang, Haifeng Ding, Xuehan Zhao, Tian Xi, and Shangwei Jiang. 2025. Highly efficient contact detection strategy of 3D discontinuous deformation analysis in continuous-discontinuous simulation. *Journal of Rock Mechanics and Geotechnical Engineering* 17, 11 (2025), 6977–6992.
- Dan Koschier, Jan Bender, and Nils Thuerey. 2017. Robust extended finite elements for complex cutting of deformables. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.
- Tassilo Kugelschadt, Dan Koschier, and Jan Bender. 2018a. Fast corotated FEM using operator splitting. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 149–160.
- T. Kugelschadt, D. Koschier, and J. Bender. 2018b. Fast Corotated FEM using Operator Splitting. *Computer Graphics Forum* 37, 8 (2018), 149–160.
- Hans Petter Langtangen and Kent-Andre Mardal. 2019. *Introduction to numerical methods for variational problems*. Springer Nature.
- Joshua A Levine, Adam W Bargteil, Christopher Corsi, Jerry Tessendorf, and Robert Geist. 2014. A Peridynamic Perspective on Spring-Mass Fracture.. In *Symposium on Computer Animation*, Vol. 10. 47–55.
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020), 49.
- Minchen Li, 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, Article 70 (jul 2019), 10 pages. doi:10.1145/3306346.3322951
- Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional incremental potential contact. *ACM Trans. Graph.* 40, 4, Article 170 (July 2021), 24 pages. doi:10.1145/3450626.3459767
- Xuan Li, Minchen Li, Xuchen Han, Huamin Wang, Yin Yang, and Chenfanfu Jiang. 2024. A dynamic duo of finite elements and material points. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.
- Huancheng Lin, Floyd M. Chitalu, and Taku Komura. 2024. Analytic rotation-invariant modelling of anisotropic finite elements. *ACM Trans. Graph.* 43, 5, Article 157 (Aug. 2024), 20 pages. doi:10.1145/3666086
- 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 (Nov. 2013), 7 pages.
- 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.
- Jia-Ming Lu and Shi-Min Hu. 2025. Reliable Iterative Dynamics: A Versatile Method for Fast and Robust Simulation. *ACM Transactions on Graphics* 44, 3, Article 29 (may 2025), 18 pages. doi:10.1145/3734518
- Zixuan Lu, Xiaowei He, Yuzhong Guo, Xuehui Liu, and Huamin Wang. 2023. Projective peridynamic modeling of hyperelastic membranes with contact. *IEEE Transactions on Visualization and Computer Graphics* 30, 8 (2023), 4601–4614.
- Miles Macklin, Matthias Müller, and Nattapong Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics (MIG '16). Association for Computing Machinery, New York, NY, USA, 49–54.
- Matthias Müller, Nattapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air meshes for robust collision handling. *ACM Trans. Graph.* 34, 4, Article 133 (July 2015), 9 pages. doi:10.1145/2766907
- Rahul Narain, Armin Samii, and James F O'brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)* 31, 6 (2012), 1–10.
- Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling simulation accuracy from mesh quality. *ACM transactions on graphics* (2018).
- Craig Schroeder, Nipun Kwatra, Wen Zheng, and Ron Fedkiw. 2011. Asynchronous Evolution for Fully-Implicit and Semi-Implicit Time Integration. *Computer Graphics Forum* 30, 7 (2011), 1983–1992.
- 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*. 1–11.
- Gen-hua Shi. 1992. Discontinuous deformation analysis: a new numerical model for the statics and dynamics of deformable block structures. *Engineering computations* 9, 2 (1992), 157–168.
- Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) (SIGGRAPH '12). Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages. doi:10.1145/2343483.2343501
- Stewart A Silling, Michael Epton, Olaf Weckner, Jifeng Xu, and E23481501120 Askari. 2007. Peridynamic states and constitutive modeling. *Journal of elasticity* 88, 2 (2007), 151–184.
- Fun Shing Sin, Daniel Schroeder, and Jernej Barbic. 2013. Vega: non-linear FEM deformable object simulator. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 36–48.
- 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.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic eigensystems for isotropic distortion energies. *ACM Transactions on Graphics (TOG)* 38, 1 (2019), 1–15.
- A.J.M. Spencer. 1972. *Deformations of Fibre-reinforced Materials*. Clarendon Press. <https://books.google.com/books?id=uqdRAAAAMAAJ>
- Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental collision handling for GPU-based interactive cloth simulation. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.
- 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.
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 205–214.
- Huamin Wang and Yin Yang. 2016. Descent methods for elastic body simulation on the GPU. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–10.
- Tianyu Wang, Jiong Chen, Dongping Li, Xiaowei Liu, Huamin Wang, and Kun Zhou. 2023. Fast GPU-based Two-way Continuous Collision Handling. *ACM Trans. Graph.* 42, 5, Article 167 (July 2023), 15 pages. doi:10.1145/3604551
- Xinlei Wang, Minchen Li, Yu Fang, Xinxin Zhang, Ming Gao, Min Tang, Danny M. Kaufman, and Chenfanfu Jiang. 2020. Hierarchical Optimization Time Integration for CFL-Rate MPM Stepping. *ACM Transactions on Graphics* 39, 3, Article 21 (apr 2020), 16 pages. doi:10.1145/3386760
- Joshuah Wolper, Yunuo Chen, Minchen Li, Yu Fang, Ziyin Qu, Jiecong Lu, Meggie Cheng, and Chenfanfu Jiang. 2020. AnisoMPM: animating anisotropic damage mechanics. *ACM Trans. Graph.* 39, 4, Article 37 (Aug. 2020), 16 pages. doi:10.1145/3386569.3392428
- Peter Wriggers. 2008. *Nonlinear finite element methods*. Springer Science & Business Media.
- Wei Yu, Jun Liu, Haibo Wang, Lei Qin, Lei Gan, Quansheng Zang, and Wenbin Ye. 2025. An implicit coupled method of scaled boundary finite element and peridynamics for fracture analysis. *Finite Elements in Analysis and Design* 252 (2025), 104453.

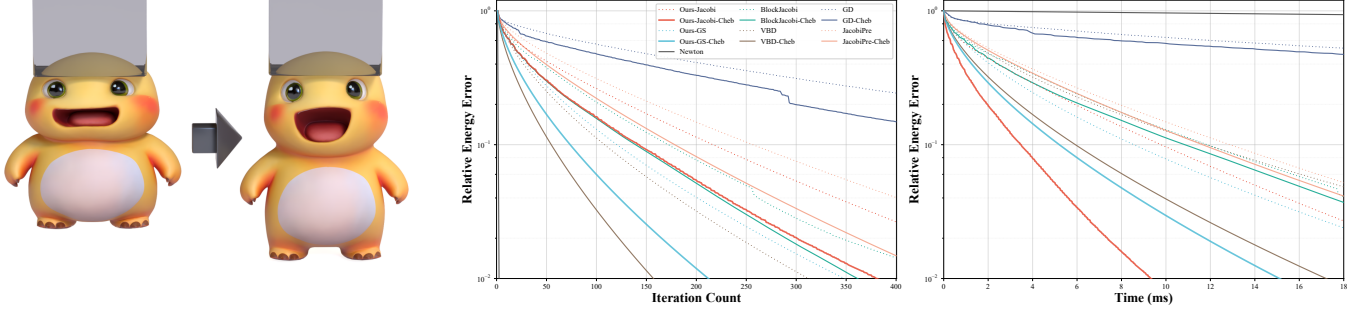


Fig. 5. **Comparison of various descent methods.** We evaluate algorithm performance using the Baby Dragon model (9k vertices, 38k tetrahedra). The model's top vertices are fixed within a glass block, and an initial compression is applied by pushing the foot vertices upward (left). The positional constraints on the feet are then instantaneously removed to simulate deformation over a single time step of $h = 33$ ms (right). The plots illustrate the relative energy error as a function of iteration count and computation time. The relative energy error is calculated by $\frac{\mathcal{L}^k - \mathcal{L}^*}{\mathcal{L}^0 - \mathcal{L}^*}$, where $\mathcal{L}^*$ is calculated by the full Newton method. All methods are implemented in a unified GPU framework, with the Chebyshev acceleration parameter set to $\rho = 0.95$.



Fig. 6. **Noodle Simulation.** 81 long noodles (1.8M tetrahedra) dropping into a porcelain bowl. Our animation is plausible and free of any collision artifact, with an average cost of 14.3ms per step ($h = 3$ ms).

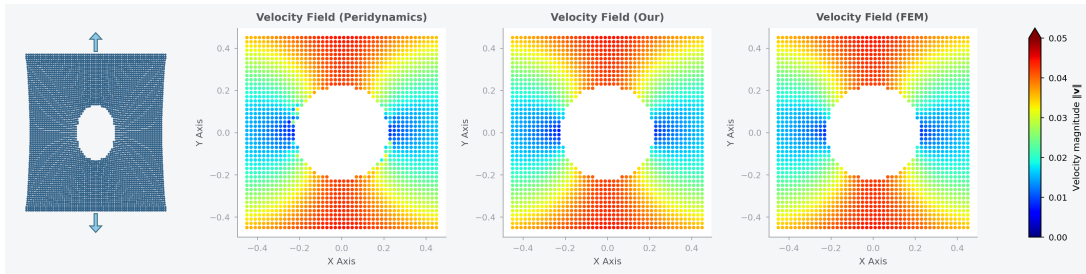


Fig. 7. **Open-hole Tension Test.** We compare the velocity fields of three methods: peridynamics, our method, and standard FEM. Peridynamics presents anomalous discontinuities around the hole boundary, an issue entirely absent in both our method and FEM.

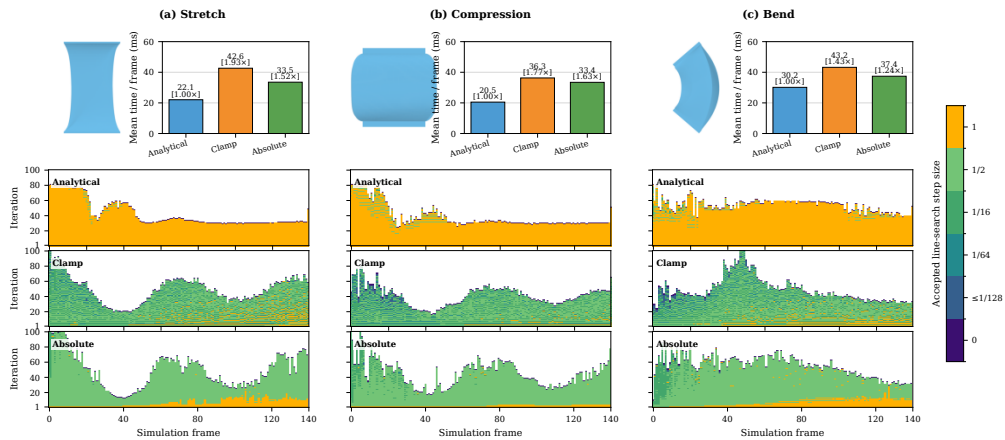


Fig. 8. **Comparison of three positive-definitization strategies over complete simulations** with $\nu = 0.499$: (a) stretching, (b) compression, and (c) bending. The top row shows the deformation and mean valid time per frame, while the heatmaps below show the accepted step size over frame and iteration. All runs terminate at a relative energy error below 10^{-4} or after 100 iterations.

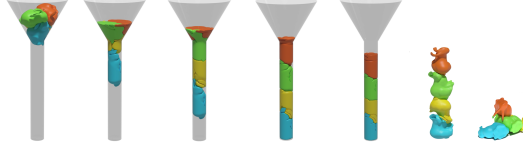


Fig. 9. **Pushing Bunny**. 4 bunnies are compressed into a narrow cylindrical cavity, followed by a sudden removal of the confinement.

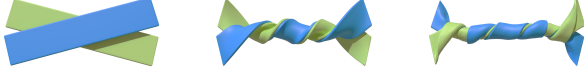


Fig. 10. **Twisting Beam**. Two beams (97K vertex, 416K tetrahedra) intertwined under complex frictional contact and buckling. Our solver delivers artifact-free results at a near real-time rate, averaging 3.5 ms per step ($h = 3ms$)

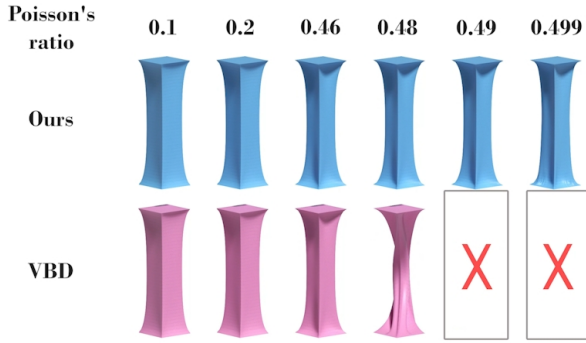


Fig. 11. **Stretching Cube**. A cube (64K vertex, 355K tetrahedra) is stretched under a varying Poisson's ratio. Our solver delivers artifact-free results from $\nu = 0.1$ to $\nu = 0.499$, while the VBD solver fails when $\nu > 0.48$.

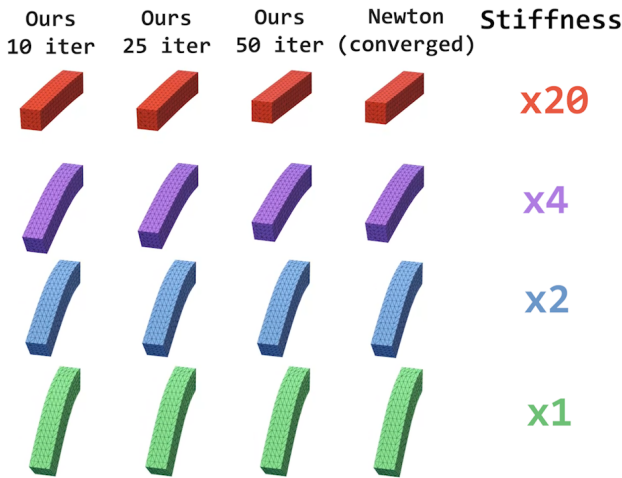


Fig. 12. **Beam fall**. Visual convergence behavior of a beam (463 vertices, 1.5K tetrahedra), illustrated through simulations with varying material stiffness and iterations per frame.

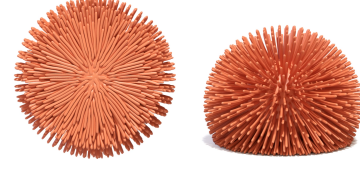


Fig. 13. **Squishy Ball**. A squishy ball with tentacles (167.4K vertices, 478.5K tetrahedra) is simulated as it drops and impacts the ground, with a per-frame compute time of 4.3 ms and a timestep of 3 ms.

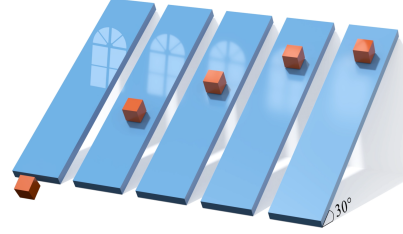


Fig. 14. **Sliding Cube**. We test different friction coefficients μ_c for a cube sliding down an inclined plane. From left to right: $\mu_c = 0.0, 0.2, 0.4, 0.6, 0.8$. Our method can robustly handle various frictional contacts.

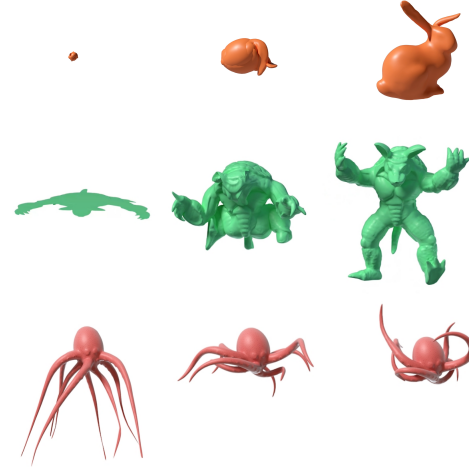


Fig. 15. **Stress Test**. Robustness to extreme initialization: **(Top)** a randomly initialized bunny model (3K vertices, 14K tetrahedra) returns to its reference configuration. **(Middle)** a highly flattened armadillo (15K vertices, 62K tetrahedra) recovers its original shape without artifacts. **(Bottom)** an octopus model (3K vertices, 13K tetrahedra) with eight tentacles recovers from a severely stretched state.



Fig. 16. **Cloth**. Draping of a cloth with two fixed points. Our current implementation models cloth as a hyperelastic membrane (without bending).