An Efficient Multilevel Preconditioned Nonlinear Conjugate Gradient Method for Incremental Potential Contact

YU ZHANG*, S-Lab, Nanyang Technological University, Singapore
XING SHEN*, Shanghai AI Laboratory, China
KEMENG HUANG†, University of Hong Kong, China
WEI CHEN, Zhejiang University, China
YIN YANG, University of Utah, United States of America
TAKU KOMURA, University of Hong Kong, China
TIANTIAN LIU, Independent Researcher, China
XINGANG PAN, S-Lab, Nanyang Technological University, Singapore

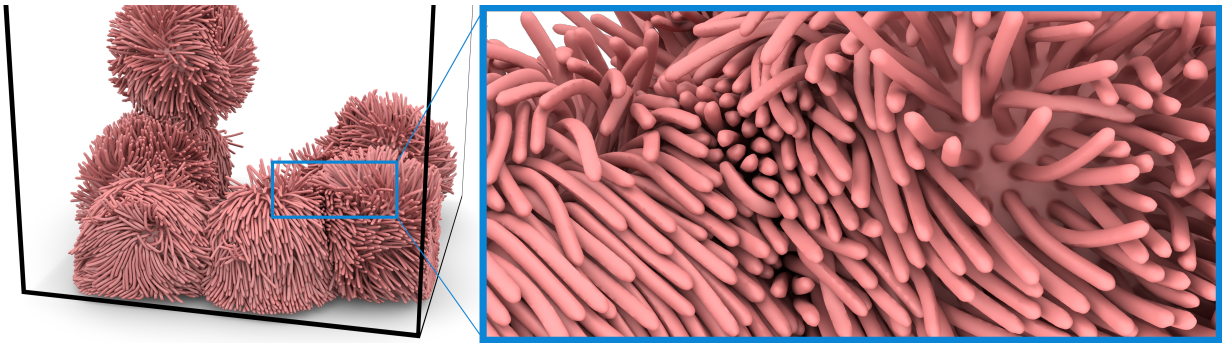


Fig. 1. Eight puffer balls are dropped from mid-air and collide repeatedly. The scene contains more than 1.14 M vertices, 2.72 M tetrahedra, and over 1 M peak active contacts. The high degree of freedom and rapidly changing contact sets challenge classical Jacobi-PNCG, leading to both prohibitive runtimes and penetration issues. In contrast, our solver advances efficiently (99 seconds / frame), and attains convergence quality comparable to a full Newton method. The zoom-in view shows that our method is able to maintain accurate penetration-free result even in highly complex large-scale collision scenarios.

Incremental Potential Contact (IPC) guarantees intersection-free simulation but suffers from high computational costs due to the expensive Hessian assembly and linear solves required by Newton's method. While Preconditioned Nonlinear Conjugate Gradient (PNCG) avoids Hessian assembly, it has historically struggled with poor convergence in stiff, contact-rich scenarios due to the lack of effective preconditioners; simple Jacobi preconditioners fail to capture the global coupling, while advanced hierarchy-based preconditioners like Multilevel Additive Schwarz (MAS) are computationally prohibitive to rebuild at every nonlinear iteration. We present MAS-PNCG, a method that unlocks the power of hierarchical preconditioning for nonlinear optimization. Our key technical innovation is a Sparse-Input Woodbury update algorithm that incrementally adapts the fine-level MAS components to rapidly evolving contact sets. This bypasses the need for full preconditioner rebuilds, reducing maintenance cost to near-zero while capturing the complex spectral properties of the contact system. Furthermore, we replace heuristic PNCG search directions with a Hessian-aware 2D subspace minimization that optimally combines the preconditioned gradient and previous direction. We also apply a fast per-subdomain conservative CCD method that ensures penetration-free trajectories while avoiding overly restrictive global step sizes. Experiments demonstrate that our MAS-PNCG outperforms state-of-the-art Newton-PCG solvers, GIPC and StiffGIPC, both preconditioned with MAS up to 5.66× and 2.07× respectively.

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

Additional Key Words and Phrases: IPC, PNCG, Contact Dynamics

1 INTRODUCTION

Contact handling lies at the heart of physically-based simulation. The faithful resolution of collision determines whether a simulation looks realistic or suffers from visible artifacts like interpenetration and jittering. The Incremental Potential Contact (IPC) method [Harmon et al. 2009; Li et al. 2020] marked a significant advance by reformulating contact dynamics as barrier-based unconstrained optimization, guaranteeing intersection-free results while maintaining high accuracy. This approach has since been adopted across diverse applications including multi-material simulations [Li et al. 2024; Xie et al. 2023], interlocked rigid components [Tang et al. 2023], high-order meshes [Ferguson et al. 2023], sticky interactions [Fang et al. 2023], and differentiable simulation [Huang et al. 2024b].

However, IPC's reliance on Newton-type solvers [Gast et al. 2015; Hecht et al. 2012] creates a computational bottleneck. At each iteration, Newton methods must assemble the full Hessian matrix

*Equal contributions.
†Corresponding author.

Authors' Contact Information: Yu Zhang, S-Lab, Nanyang Technological University, Singapore, yucrazing@gmail.com; Xing Shen, Shanghai AI Laboratory, China, shenxing@pjlab.org.cn; Kemeng Huang, University of Hong Kong, China, kmhuang@connect.hku.hk; Wei Chen, Zhejiang University, China, weichen12@zju.edu.cn; Yin Yang, University of Utah, United States of America, yangzzzy@gmail.com; Taku Komura, University of Hong Kong, China, taku@cs.hku.hk; Tiantian Liu, Independent Researcher, China, ltt1598@gmail.com; Xingang Pan, S-Lab, Nanyang Technological University, Singapore, xingang.pan@ntu.edu.sg.

and then solve a linear system involving it. For large-scale scenarios with thousands of contact pairs, this becomes prohibitively expensive. Existing acceleration strategies face a fundamental tension: approximation-based methods [Lan et al. 2023, 2022, 2021; Li et al. 2023] sacrifice IPC's accuracy for speed, while exact GPU-accelerated Newton solvers like GIPC [Huang et al. 2024a] and StiffGIPC [Huang et al. 2025] preserve accuracy but remain bound to the cost of Hessian assembly and linear solves at each Newton iteration.

Preconditioned Nonlinear Conjugate Gradient (PNCG) [Shen et al. 2024] offers an alternative that sidesteps Hessian assembly entirely, using only gradient evaluations to iteratively descend toward the energy minimum. This yields dramatically lower per-iteration costs and naturally handles degenerate configurations. However, existing PNCG implementations for IPC rely on simple preconditioners like the Jacobi matrix, which fail to capture the complex coupling between vertices induced by stiff materials and dense contacts. The result is slow convergence that negates the per-iteration savings.

A natural idea is to employ a more powerful preconditioner. The Multilevel Additive Schwarz (MAS) preconditioner [Wu et al. 2022], which combines local domain decomposition with hierarchical coarsening, has proven highly effective for accelerating linear solvers in elastodynamics and was recently adopted by StiffGIPC [Huang et al. 2025] for IPC's Newton inner loop. However, applying MAS to nonlinear PNCG faces a fundamental obstacle: MAS requires building subdomain factorizations from the system Hessian, yet PNCG deliberately avoids Hessian assembly. In Newton-PCG methods like StiffGIPC, this is not a problem because the Hessian is fixed within each Newton iteration, allowing a single MAS build to serve all inner PCG steps. But in nonlinear PNCG, the effective curvature changes *every iteration* as contacts evolve—new pairs activate when objects approach within the threshold $\hat{d}$, existing contacts stiffen as gaps shrink, and contacts release as objects separate. Rebuilding the full MAS hierarchy at each iteration would be prohibitively expensive, defeating the purpose of avoiding Newton's Hessian assembly.

Our key observation is that while IPC's effective Hessian changes rapidly, *how* it changes has exploitable structure. Each contact contributes a localized stiffness term that depends on just a few vertices (typically 4–8) and is dominated by a rank-1 component along the contact normal direction [Huang et al. 2024a]. Physically, this reflects the fact that contact forces primarily resist motion along the direction of potential penetration. This sparse, low-rank structure suggests a fundamentally different strategy: instead of rebuilding the MAS preconditioner from scratch at each iteration, we can *incrementally update* the fine-level subdomain inverses through efficient rank-1 Woodbury modifications that track contact stiffening and release, while keeping the coarse levels frozen.

Building on this insight, we develop MAS-PNCG. The coarse MAS levels, which capture global elastic coupling, are built once and updated only periodically. The fine-level subdomain inverses, which are most sensitive to local contact changes, are maintained through Sparse-Input Woodbury updates derived directly from barrier Hessian structure. Since these updates provide approximate curvature information, we can move beyond heuristic β formulas to optimal 2D subspace minimization—directly computing the best combination of preconditioned gradient and previous direction. Finally, rather than restricting the entire simulation to a single conservative step size dictated by the stiffest contact, we introduce per-subdomain CCD that allows different regions to advance at locally appropriate rates while maintaining penetration-free guarantees.

Our contributions are:

- **Incremental Spectral Maintenance via Sparse-Input Woodbury**: We introduce an algorithm to update fine-level MAS subdomain inverses using rank-1 Woodbury modifications. This allows the preconditioner to evolve with the contact constraints at negligible cost, solving the bottleneck that previously prevented the use of hierarchical preconditioners in nonlinear CG.
- **Curvature-Optimal Subspace Minimization**: We replace the heuristic β formulas of classical PNCG with a principled minimization strategy. By projecting the local Hessian approximation into the subspace spanned by the gradient and search direction, we analytically compute the optimal descent step, significantly improving convergence rates in stiff scenarios.
- **Conservative CCD with Tighter Lower Bounds**: We apply a per-subdomain collision detection scheme that maintains penetration-free guarantees while avoiding global step size restrictions, with frame-invariant lower bounds for improved tightness.

Experiments demonstrate that MAS-PNCG achieves up to 5.66× and 2.07× speedup over state-of-the-art GIPC [Huang et al. 2024a] and StiffGIPC [Huang et al. 2025] respectively, while converging to comparable accuracy in complex, large-scale scenarios. Moreover, since both GIPC and StiffGIPC already employ MAS preconditioning in their inner PCG solvers, our speedups demonstrate that the performance gains stem not from MAS alone, but from the synergy of our Sparse-Input Woodbury updates and 2D subspace minimization strategy—techniques that enable efficient preconditioner maintenance and enhanced per-iteration progress within the nonlinear CG framework.

2 RELATED WORK

2.1 IPC Solvers

The IPC framework [Li et al. 2020] reformulates contact as barrier-based optimization, enabling intersection-free simulation across diverse applications [Fang et al. 2023; Ferguson et al. 2023; Huang et al. 2024b; Li et al. 2024; Tang et al. 2023; Xie et al. 2023]. However, its expensive solve cost, especially in Hessian assembly and linear system solve per iteration, severely limits its practical applications.

To accelerate IPC, recent approximation-based methods trade accuracy for efficiency, such as medial axis reduction [Lan et al. 2021], projective dynamics [Lan et al. 2022], stencil-wise descent [Lan et al. 2023] and subspace methods [Li et al. 2023]. In contrast, our method aims to deliver the acceleration while preserving accuracy comparable to original Newton solver based IPC method.

Accuracy-preserving methods such as GIPC [Huang et al. 2024a] and its enhanced version StiffGIPC [Huang et al. 2025] accelerate IPC through GPU parallelization. For these methods, however, Hessian assembly and linear solves remain the primary computational

bottleneck per Newton iteration. Instead, inspired by MAS [Wu et al. 2022], we design a convergence-enhanced nonlinear solver with a tailored multilevel preconditioner. Our approach avoids Hessian assembly entirely by using gradient-only iterations with incrementally updated preconditioners.

2.2 Nonlinear Conjugate Gradient Methods

Nonlinear CG methods have been applied to graphics problems [Wang and Yang 2016]. Shen et al. [Shen et al. 2024] applied PNCG to IPC with Jacobi preconditioning, but observed poor convergence in stiff scenarios. We address this through MAS preconditioning with Woodbury updates and curvature-aware subspace minimization.

Quasi-Newton methods like L-BFGS [Nocedal and Wright 1999] approximate Hessian information through gradient differences [Al-Baali 2014]. In contrast, our Sparse-Input Woodbury approach exploits IPC's structure where contact contributions are sparse and low-rank, deriving updates directly from barrier derivatives.

2.3 Multilevel Additive Schwarz Preconditioning

Domain decomposition methods [Smith et al. 1996; Toselli and Widlund 2004] solve large sparse linear systems by partitioning domains into overlapping subdomains. The Multilevel Additive Schwarz variant adds coarse-level corrections for faster convergence.

Wu et al. [Wu et al. 2022] introduced GPU-parallelized MAS for elastodynamics within Newton solvers. StiffGIPC [Huang et al. 2025] adapted this for IPC, applying MAS to the linear PCG inner loop.

A crucial distinction separates these prior applications from ours. In Newton methods, MAS preconditions a *fixed* linear system $\mathbf{Hd} = -\mathbf{g}$ within each iteration—the Hessian and MAS hierarchy remain constant throughout the inner solve. In contrast, we apply MAS to nonlinear PNCG where effective curvature changes every iteration as contacts evolve. The challenge is maintaining preconditioner quality across rapidly changing landscapes without explicit Hessian matrices. We address this through Sparse-Input Woodbury updates exploiting the sparse, low-rank structure of contact barrier Hessians.

2.4 Continuous Collision Detection

CCD ensures intersection-free trajectories by computing time-of-impact before penetration. Recent robust methods include conservative advancement [Li et al. 2021], inclusion-based root-finding [Wang et al. 2021], exact root parity [Wang et al. 2022], and GPU-scalable algorithms [Belgrod et al. 2021].

Standard CCD computes a global step size, causing one stiff contact to restrict motion everywhere. Our per-subdomain approach computes localized step sizes, allowing different regions to advance appropriately while maintaining penetration-free guarantees.

3 BACKGROUND

3.1 Incremental Potential Contact

IPC formulates contact simulation as minimizing:

$$E(\mathbf{x}) = \underbrace{\frac{1}{2}(\mathbf{x}-\tilde{\mathbf{x}})^\top \mathbf{M}(\mathbf{x}-\tilde{\mathbf{x}})}_{\text{inertia}} + \underbrace{h^2\Psi(\mathbf{x})}_{\text{elasticity}} + \underbrace{B(\mathbf{x})}_{\text{contact barrier}} + \underbrace{D(\mathbf{x})}_{\text{friction}}, \quad (1)$$

where $\tilde{\mathbf{x}} = \mathbf{x}^t + h\mathbf{v}^t + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$ incorporates previous state and external forces. The contact barrier $B(\mathbf{x}) = \kappa \sum_{c\in C} b(d_c(\mathbf{x}))$ activates when distance $d < \hat{d}$, with $b(d)$ a logarithmic function ensuring non-penetration.

3.2 Preconditioned Nonlinear Conjugate Gradient

PNCG generates iterates $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_{k+1}$, where the search direction

$$\mathbf{p}_{k+1} = -\mathbf{P}_{k+1}\mathbf{g}_{k+1} + \beta_k \mathbf{p}_k \quad (2)$$

combines the preconditioned gradient $\mathbf{P}_{k+1}\mathbf{g}_{k+1}$ with the previous direction $\mathbf{p}_k$ via conjugate parameter β_k.

3.3 Multilevel Additive Schwarz Preconditioner

MAS combines domain decomposition with hierarchical coarsening. The domain is partitioned into D subdomains with selection matrices $\mathbf{S}_d$. The level-0 preconditioner aggregates local inverses:

$$\mathbf{M}_{(0)}^{-1} = \sum_{d=1}^{D} \mathbf{S}_d^T (\mathbf{S}_d \mathbf{A} \mathbf{S}_d^T)^{-1} \mathbf{S}_d. \quad (3)$$

The full MAS preconditioner extends this hierarchically:

$$\mathbf{P} = \mathbf{M}_{(0)}^{-1} + \sum_{l=1}^{L} \mathbf{C}_{(l)}^T \mathbf{M}_{(l)}^{-1} \mathbf{C}_{(l)}, \quad (4)$$

where $\mathbf{C}_{(l)}$ are coarsening matrices.

4 METHOD

4.1 Localized Woodbury Level-0 Update

The MAS preconditioner significantly accelerates NCG convergence [Wu et al. 2022], yet frequent full recomputations remain computationally prohibitive. Our method inherits the setup of MAS in StiffGIPC [Huang et al. 2025], the details can be found in the supplemental document. Our update strategy leverages the spectral properties of the system: coarse-level components primarily capture low-frequency error modes, which evolve relatively slowly. Consequently, these components can be temporarily frozen without significantly degrading convergence.

Furthermore, Hessian contributions from elastic potential energy tend to remain stable or change smoothly. In contrast, severe ill-conditioning arises primarily from contact barrier terms, which activate and deactivate abruptly. Fortunately, these interactions are spatially sparse, affecting only a small subset of the domain. Therefore, we focus our update strategy exclusively on the Level-0 (fine-level) subdomains to capture these local changes via efficient low-rank updates, while keeping the rest of the preconditioner fixed.

To ensure the preconditioner remains Symmetric Positive Definite (SPD), we note that the contact barrier Hessian has positive eigenvalues only in the normal direction, tangential eigenvalues are negative and thus eliminated by SPD projection. With scalar stiffness $k = b''(d_c)$ and contact normal $\mathbf{n} = \nabla d_c$, the projected Hessian reduces to:

$$\nabla^2 b(d_c(\mathbf{x})) \approx k\mathbf{n}\mathbf{n}^T. \quad (5)$$

This simplification allows us to model the Hessian change as a sum of rank-1 updates. We define the base state using the frozen Hessian snapshots $\mathbf{H}_{\text{base}}$ and inverse of each subdomain of level 0

$\mathbf{B}_d = (\mathbf{M}_0^d)^{-1}$ from the last global restart. The current Hessian is then approximated as:

$$\tilde{\mathbf{H}} = \mathbf{H}_{\text{base}} + \mathbf{U}\mathbf{U}^T, \tag{6}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$ is the update matrix collecting the rank-1 contributions $\mathbf{u}_i = \sqrt{k_i}\mathbf{n}_i$ from m active contacts. Since the update takes the form $\mathbf{U}\mathbf{U}^T$, the Woodbury formula simplifies with an identity coupling matrix. This formulation enables efficient low-rank updates to the Level-0 preconditioner for capturing rapidly changing contact states, as detailed below.

4.1.1 Construction of Local Update Matrices. We maintain a global set of active contacts C_{curr} and compare each contact $c \in C_{\text{curr}}$ against its state in the global base set C_{base} from the last restart. Using a rotation threshold ϵ_{rot} (e.g., 25°), we determine the potential rank-1 update vectors $\mathbf{u}$ and their significance metrics ΔS by classifying each contact into two categories:

(1) **Case 1: New Contact** ($c \notin C_{\text{base}}$). For topologically new contacts, we inject the full stiffness as a pure increment:

$$\mathbf{u} = \sqrt{k_{\text{curr}}} \cdot \mathbf{n}_{\text{curr}}, \quad \Delta S = k_{\text{curr}}. \tag{7}$$

(2) **Case 2: Existing Contact** ($c \in C_{\text{base}}$). For contacts persisting from the base state, we categorize them based on the stability of their normal directions.
- **Rotated Normal ($\mathbf{n}_{\text{curr}} \cdot \mathbf{n}_{\text{base}} < \epsilon_{\text{rot}}$):** If the geometric constraint has rotated significantly, updating the direction is numerically unstable. We treat this as a **New Contact**, effectively retaining the old stiffness as a "ghost wall" while adding the new constraint to ensure robustness:

$$\mathbf{u} = \sqrt{k_{\text{curr}}} \cdot \mathbf{n}_{\text{curr}}, \quad \Delta S = k_{\text{curr}}. \tag{8}$$

- **Stable Normal ($\mathbf{n}_{\text{curr}} \cdot \mathbf{n}_{\text{base}} \geq \epsilon_{\text{rot}}$):** For directionally stable contacts, we further compare the stiffness:
  - *Stiffening ($k_{curr} > k_{base}$):* We add the stiffness differential:

$$\mathbf{u} = \sqrt{k_{\text{curr}} - k_{\text{base}}} \cdot \mathbf{n}_{\text{curr}}, \quad \Delta S = k_{\text{curr}} - k_{\text{base}}. \tag{9}$$

  - *Softening ($k_{curr} \leq k_{base}$):* We ignore stiffness decreases. Retaining the stiffer base approximation yields a safer step without risking indefiniteness. We set $\Delta S = 0$.

To balance approximation accuracy and computational overhead, we employ a *localized Top-K truncation* strategy. For each subdomain d, we rank all potential local update vectors by their significance metric ΔS and select only the top K vectors (e.g., $K = 8$) to form the subdomain update matrix $\mathbf{U}_d \in \mathbb{R}^{N_d \times K}$. This ensures the rank of the update remains low within each subdomain, keeping the subsequent per-subdomain capacitance solve efficient.

4.1.2 Localized Woodbury Solver. To maintain the efficiency of the domain-decomposed preconditioner without full rebuilds, we apply the Woodbury update locally within each subdomain d. Let $\mathbf{M}_{\text{base}}^d$ be the Level-0 subdomain Hessian frozen at the last global restart. We factorize $\mathbf{M}_{\text{base}}^d$ and cache its inverse $\mathbf{B}_d = (\mathbf{M}_{\text{base}}^d)^{-1}$ as the *base state*. The updated inverse $\tilde{\mathbf{B}}_d$ is then efficiently computed via the Woodbury formula:

$$\tilde{\mathbf{B}}_d = \mathbf{B}_d - \mathbf{B}_d\mathbf{U}_d \left(\mathbf{I}_K + \mathbf{U}_d^T\mathbf{B}_d\mathbf{U}_d\right)^{-1} \mathbf{U}_d^T\mathbf{B}_d. \tag{10}$$

For a local gradient $\mathbf{g}_d = \mathbf{S}_d\mathbf{g}$, the preconditioned direction $\mathbf{z}_d = \tilde{\mathbf{B}}_d\mathbf{g}_d$ is computed through a streamlined *capacitance system*:

(1) **Base Solve**: Compute the cached projection $\mathbf{z}_{\text{base}} = \mathbf{B}_d\mathbf{g}_d$.
(2) **Update Projection**: Form the auxiliary matrix $\mathbf{W}_d = \mathbf{B}_d\mathbf{U}_d$ and the correction vector $\mathbf{r} = \mathbf{U}_d^T\mathbf{z}_{\text{base}}$.
(3) **Capacitance Solve**: Solve the $K \times K$ system $(\mathbf{I}_K + \mathbf{U}_d^T\mathbf{W}_d)\boldsymbol{\lambda} = \mathbf{r}$.
(4) **Correction**: Update the final local direction $\mathbf{z}_d = \mathbf{z}_{\text{base}} - \mathbf{W}_d\boldsymbol{\lambda}$.

Finally, the global preconditioned gradient $\mathbf{z}_{k+1}$ is assembled by merging the updated Level-0 contributions with the frozen coarse-level corrections:

$$\mathbf{z}_{k+1} = \sum_{d=1}^{D} \mathbf{S}_d^T\mathbf{z}_d + \sum_{l=1}^{L} \mathbf{C}_{(l)}^T\mathbf{M}_{(l)}^{-1}\mathbf{C}_{(l)}\mathbf{g}_{k+1}, \tag{11}$$

where $\mathbf{M}_{(l)}^{-1}$ are cached from the last restart.

Efficiency Optimization. The localized Woodbury solver provides a critical advantage: it completely bypasses the need for global Hessian assembly and expensive matrix factorization in each iteration. By reusing the frozen base inverse $\mathbf{B}_d$ and only updating it with a low-rank matrix $\mathbf{U}_d$, we maintain an up-to-date preconditioner at minimal cost. This effectively enables the preconditioner to "evolve" with the contact state without a full rebuild, ensuring both numerical stability and high performance.

Algorithm 1: MAS-PNCG Pipeline

Input: Position $\mathbf{x}^t$, velocity $\mathbf{v}^t$, mass $\mathbf{M}$, barrier $\hat{d}$, tol ϵ, restart δ
Output: Updated $\mathbf{x}^{t+1}$
$\mathbf{x}_0 \leftarrow \mathbf{x}^t$; $\tilde{\mathbf{x}} \leftarrow \mathbf{x}^t + h\mathbf{v}^t + h^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}$; Restart $\leftarrow$ True;
for $k = 0$ **to** *IterMax* **do**
  $C \leftarrow$ ComputeConstraintSet$(\mathbf{x}_k, \hat{d})$;
  if *Restart* **then**
    $(\mathbf{P}_{\text{base}}, \tilde{\mathbf{H}}) \leftarrow$ RebuildMAS$(\mathbf{x}_k, C)$; $\mathbf{P}_{k+1} \leftarrow \mathbf{P}_{\text{base}}$;
  else
    $\mathbf{P}_{k+1} \leftarrow$ WoodburyUpdate$(\mathbf{P}_{\text{base}}, C)$;
  end
  $\mathbf{g}_{k+1} \leftarrow \nabla E(\mathbf{x}_k, \tilde{\mathbf{x}}, C)$; $\mathbf{z}_{k+1} \leftarrow \mathbf{P}_{k+1}\mathbf{g}_{k+1}$; $\mathbf{v} \leftarrow \tilde{\mathbf{H}}\mathbf{z}_{k+1}$;
  if *Restart* **then**
    $\mu \leftarrow (\mathbf{z}_{k+1}^\top\mathbf{g}_{k+1})/(\mathbf{z}_{k+1}^\top\mathbf{v})$; $\nu \leftarrow 0$;
  else
    $(\mu, \nu) \leftarrow$ Solve2DSubspace$(\mathbf{z}_{k+1}, \mathbf{p}_k, \mathbf{g}_{k+1}, \tilde{\mathbf{H}})$
  end
  $\mathbf{p}_{k+1} \leftarrow -\mu\mathbf{z}_{k+1} + \nu\mathbf{p}_k$;
  $\{\alpha_d\} \leftarrow$ ConservativeCCD$(\mathbf{x}_k, \mathbf{p}_{k+1})$;
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \sum_d \mathbf{S}_d^T\alpha_d\mathbf{S}_d\mathbf{p}_{k+1}$;
  if $\|\mathbf{z}_{k+1}\| \leq \epsilon$ ***and*** *Restart* **then**
    break;
  end
  $r_k \leftarrow |\mathbf{g}_{k+1}^\top\mathbf{z}_k|/(\mathbf{g}_{k+1}^\top\mathbf{z}_{k+1})$;
  Restart $\leftarrow (r_k > \delta)$;
  $\mathbf{z}_k \leftarrow \mathbf{z}_{k+1}$;
end
return $\mathbf{x}_{k+1}$;

4.2 Optimal 2D Subspace Minimization

Traditional PNCG methods typically lack direct access to Hessian information, relying instead on heuristic conjugacy parameters (e.g., Polak-Ribière) and decoupled line searches. Leveraging our approximated Hessian (Eq. 6), we can directly determine the optimal search direction $\mathbf{p}_{k+1}$ within the 2D subspace spanned by the preconditioned gradient $\mathbf{z}_{k+1} = \mathbf{P}_{k+1}\mathbf{g}_{k+1}$ and the previous direction $\mathbf{p}_k$:

$$\mathbf{p}_{k+1}(\mu, \nu) = -\mu\mathbf{z}_{k+1} + \nu\mathbf{p}_k. \tag{12}$$

The optimal coefficients (μ^*, ν^*) are obtained by minimizing the local quadratic model $Q(\mathbf{p}_{k+1}) = \mathbf{g}_{k+1}^\top\mathbf{p}_{k+1} + \frac{1}{2}\mathbf{p}_{k+1}^\top\tilde{\mathbf{H}}\mathbf{p}_{k+1}$, which yields the following 2×2 system:

$$\begin{bmatrix} \mathbf{z}_{k+1}^\top\tilde{\mathbf{H}}\mathbf{z}_{k+1} & -\mathbf{z}_{k+1}^\top\tilde{\mathbf{H}}\mathbf{p}_k \\ -\mathbf{p}_k^\top\tilde{\mathbf{H}}\mathbf{z}_{k+1} & \mathbf{p}_k^\top\tilde{\mathbf{H}}\mathbf{p}_k \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{k+1}^\top\mathbf{g}_{k+1} \\ -\mathbf{p}_k^\top\mathbf{g}_{k+1} \end{bmatrix}. \tag{13}$$

This formulation eliminates the need for heuristic β selection and expensive line searches. Notably, the coefficient μ acts as a "natural step size" that accounts for second-order curvature, allowing us to adopt a unit initial trial step ($\alpha = 1.0$) which is only clamped to enforce non-penetration via our Conservative CCD (Section 4.4). Detailed derivations are provided in the Supplementary Document.

4.3 Global Restart Criterion

Although Sparse-Input Woodbury updates enhance the efficiency of managing level 0 MAS components, approximation errors can accumulate across iterations, leading to a decline in preconditioner quality and deterioration of the conjugate property. To address this, we employ a global restart strategy that monitors the orthogonality loss between the current gradient $\mathbf{g}_{k+1}$ and the previous preconditioned gradient $\mathbf{z}_k$:

$$r_k = \frac{|\mathbf{g}_{k+1}^\top\mathbf{z}_k|}{\mathbf{g}_{k+1}^\top\mathbf{z}_{k+1}}. \tag{14}$$

When r_k exceeds a threshold (e.g., $\delta = 0.3$), indicating significant loss of orthogonality in the preconditioned space, we trigger a full MAS rebuild. Evaluating in the preconditioned space better reflects the true loss of conjugacy, as it filters out the ill-conditioning of the original system.

4.4 Localized Conservative CCD

While our subspace minimization strategy provides an optimal trial step, it cannot guarantee collision-free motion in non-quadratic, contact-rich landscapes. To address the "numerical locking" inherent in global CCD, where a single difficult contact restricts the entire system, we implement Conservative CCD (CCCD) by computing safe steps $\{\alpha_d\}$ independently for each subdomain, unconstrained regions maintain optimal descent speeds while only critical subdomains are damped, ensuring that local restrictions do not compromise global convergence.

Our CCCD strategy prioritizes rapid, safe clamping over exact Time-of-Impact (TOI) calculation. The key insight is that penetration detection often signals a search direction contaminated by missing collision constraints. Rather than pursuing expensive exact root-finding to "touch" the obstacle, we delegate accuracy to the subsequent iteration—injecting the newly detected contact into the Woodbury update naturally steers the search direction away from the collision.

The CCCD algorithm examines the cubic distance function $f_c(\alpha)$ over the interval $[0, 1.0]$. Following [Yuksel 2022], we first identify the monotonic region by computing α_{mon}, the first critical point of $f_c'(\alpha)$. Within the restricted interval $[0, \min(1.0, \alpha_{\text{mon}})]$, we employ a sign-based root detection strategy: if $\text{sign}(f_c(0)) \neq \text{sign}(f_c(\alpha))$, a collision exists within the interval. Rather than precisely locating the collision point, we use a bisection strategy that iteratively halves α until either the signs match (indicating no collision) or we reach the minimal threshold α_l. This approach avoids the numerical instabilities of polynomial root-finding, requires only function evaluations, and achieves rapid convergence to safe step sizes while avoiding ACCD's potentially excessive iterations. Furthermore, We introduce an improved lower bound for CCD step sizes by directly bounding relative displacements between primitives, providing a tighter estimate compared to standard ACCD [Li et al. 2021]. Complete algorithmic details and pseudocode are provided in the Supplementary Document.

5 EXPERIMENTS AND ANALYSIS

We conducted all experiments on a desktop workstation with an `Intel Core i9-13900X` CPU and an `NVIDIA RTX 4090` GPU. Our method is built on top of the publicly available StiffGIPC code base [Huang et al. 2025], and all simulations performed using double-precision floating-point arithmetic.

Table 1 presents the experimental settings and overall performance results of MAS-PNCG across diverse simulation scenarios, ranging from the 22K-tetrahedra "Armadillo" (Figure 15) to the highly demanding "Puffer Balls" simulation (Figure 14) with 2.72M tetrahedral elements. Our method demonstrates consistent performance across various energy models, including As-Rigid-As-Possible (ARAP), Stable Neo-Hookean (SNH) [Smith et al. 2018], and Baraff-Witkin cloth [Kim 2020]. The average iteration counts scale reasonably with problem complexity, demonstrating our method's versatility while maintaining robust intersection-free guarantees.

5.1 Ablation and Comparative Studies

Preconditioner Efficacy. We benchmark the proposed MAS preconditioner against the classic Jacobi preconditioner in the "Teapots" scene (Figure 10), where 32 elastic teapots are dropped into a box. In this experimenal setting, we use a 3×3 blocked Jacobi preconditioner. All teapots are simulated with the ARAP energy model and a Young's modulus of 3×10^5. The convergence tolerance of $\epsilon = 1 \times 10^{-5}$, as looser thresholds frequently lead to noticeable artifacts.

In preliminary experiments, Jacobi-PNCG requires substantially more iterations to reach the same tolerance, leading to prohibitively long runtimes (several minutes per frame) without an iteration cap, while not altering the qualitative conclusions. Therefore, to ensure practical and fair comparisons, we impose a maximum of 10,000 PNCG iterations per time step for all solver.

Given the same time budget, MAS achieves significantly higher accuracy than Jacobi. Figure 10 contrasts the two solvers at an intermediate frame: MAS delivers smooth, physically plausible dynamics, whereas Jacobi exhibits excessive damping and severe distortion—clear symptoms of poor convergence. To achieve comparable

Table 1. Experimental Settings and Overall Performance Results of MAS-PNCG across Various Benchmarks. The "Mesh" column shows vertex/face/tetrahedron counts (faces shown only for cloth). Time step $\mathbf{h}$ is 0.01s for Armadillo (Figure 15) through Cloth Twisting (Figure 13), and 0.005s for remaining scenes. Gap distance $\hat{\mathbf{d}} = 10^{-3}$ except Teapots (Figure 10) and Dropping Letters (Figure 7) (5×10^{-4}) and Puffer Balls (Figure 14) (10^{-4}). "Avg. Iters" denotes average NCG iterations per frame. "Time" shows average wall-clock time (**s**) per frame. The abbreviations ARAP, SNH and BW represent As-Rigid-As-Possible, Stable Neo-Hookean and Baraff-Witkin cloth energy models, respectively.

Example	Mesh (#V/#F/#T)	Material	Avg. Iters	Time
Figure 15	8K / - / 22K	ARAP	29	0.11
Figure 11	10K / - / 39K	ARAP	36	0.25
Figure 9	20K / - / 80K	SNH	26	0.13
Figure 12	40K / - / 160K	SNH	33	0.21
Figure 13	20K / 40K / -	BW	62	3.29
Figure 8	110K / 179K /80K	SNH+BW	107	7.44
Figure 10	39K / - / 121K	ARAP	184	7.91
Figure 16	78K / - / 274K	SNH	76	3.39
Figure 7	105K / - / 298K	SNH	81	4.18
Figure 14	1.14M / - / 2.72M	SNH	891	99.62

visual quality, the Jacobi method requires more than ten times longer computation time and may still occasionally fail to converge.

Material Stiffness Robustness. The "Dropping Letters" scene (Figure 7) is designed to challenge the solver with extreme variations in material stiffness. We utilize the SNH model for a stack of 3D letters whose Young's modulus increases geometrically from 2.15×10^4 Pa (bottom) to 1×10^7 Pa (top), resulting in an approximately 500-fold stiffness span. Despite this extreme three–order-of-magnitude stiffness disparity, MAS-PNCG converges robustly, whereas Jacobi-PNCG frequently stalls and fails to converge. This performance difference is further illustrated by a convergence comparison across varying stiffness levels in Figure 2. This comparison confirms that MAS-PNCG maintains robust convergence even with highly stiff materials, whereas the convergence rate of Jacobi-PNCG degrades severely.

We also evaluate robustness under frictional contact. In the "Bunny Cloth" scene (Figure 8), we simulate a cloth sheet falling onto a bunny with frictional contact. The friction model we used is same with [Li et al. 2020]. MAS-PCG remains stable and accurately captures friction-induced wrinkles and sliding behavior.

Preconditioner Update Strategy. We compare three preconditioner update strategies in the "Octopus Stack" scene (Figure 11): (i) *Full Rebuild*: recomputes all hierarchy levels at every PNCG iteration; (ii) *Freeze*: keeps the preconditioner fixed between NCG restarts; and (iii) our proposed *Sparse-Input Woodbury update*. A critical question is whether our incremental Woodbury updates degrade the quality of the preconditioner compared to a ground-truth Full Rebuild. Figure 6 demonstrates that Sparse-Input Woodbury updates preserve the convergence rate of a Full Rebuild almost exactly, yet they reduce the computation time by two orders of magnitude. This confirms that our method successfully captures the spectral changes of the contact system without the computational penalty of explicit assembly. The *Freeze* strategy, which ignores these updates, fails to converge efficiently, proving that tracking contact evolution is mandatory for PNCG. Since NCG restarts occur every 3–10 PNCG iterations across all test scenes, the computational savings from lightweight Woodbury updates accumulate substantially.

We further compare MAS-PNCG with Jacobi-PNCG and the SOTA GPU Newton–PCG solver with MAS preconditioner (GIPC) [Huang et al. 2024a]. As shown in Figure 3, MAS-PNCG reaches significantly lower error levels than Jacobi-PNCG. Although Newton-PCG converges faster per iteration, MAS-PNCG achieves the same error in less total time due to its substantially lower per-iteration cost.

Conservative CCD. We compare our proposed Conservative Continuous Collision Detection (CCCD) with the existing Additive CCD (ACCD) method using the "Octopus Stack" scene. At the frame (left) shown in Figure 11 (approximately frame 60, when the octopuses first come into mutual contact) we record the average *maximum* number of CCD iterations required per NCG step over several successive frames. Figure 4 shows that CCCD dramatically reduces iteration count from over 30,000 (ACCD) to just a few dozen, while still ensuring penetration-free configurations. This reduced CCD computational burden improves stability within each NCG iteration and achieves an end-to-end speedup of $\sim 10\%$ for this test case.

We implement CCCD on a per-subdomain basis to avoid "numerical locking". We compare per-subdomain and global step-size selection on the "Single Bunny" scene. Figure 9 displays the same simulation frame for both strategies. Per-subdomain step sizes allow the bunny's ears to sag naturally under gravity, while the global strategy produces visibly overdamped motion even with identical convergence criteria. The global approach requires more than five times the time costing to achieve comparable visual quality.

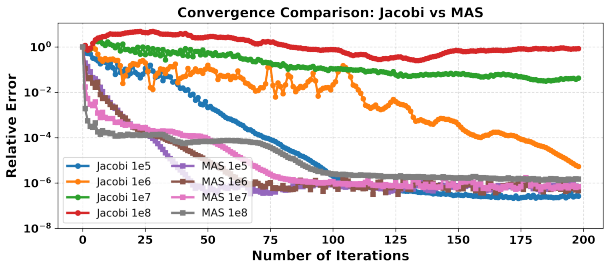


Fig. 2. Convergence behavior of MAS-PNCG and Jacobi-PNCG under varying material stiffness. MAS-PNCG maintains robust convergence even for highly stiff materials, while Jacobi-PNCG's convergence degrades severely, failing to reach low error values for high stiffness materials.

Performance. Table 2 presents a detailed performance comparison between MAS-PNCG and two SOTA Newton-based solvers: GIPC and StiffGIPC. Crutially, both baseslines already employ MAS preconditioning within their inner loops. This highlights that our performance advantage does not come from MAS itself, but from how MAS is applied. In both baselines, the high cost the Newton outer loop (e.g. Hessian rebuilding) bounds performance. By shifting to a NCG framework equipped with our our Sparse-Input Woodbury maintainance scheme, we eliminate the assembly bottleneck while retaining the spectral benefits of MAS. Also, our Hessian-aware 2D subspace minimization ensures optimal descent directions, preventing the convergence degradation typically associated with standard PNCG.

Table 2. **Performance Comparison with Timing Breakdown.** We compare the average time per frame (s) for key solver stages against GIPC [Huang et al. 2024a] and StiffGIPC [Huang et al. 2025]. Both baselines are preconditioned with MAS in their inner PCG solvers. All timing columns show per-frame time in seconds. The breakdown includes gradient and Hessian computation, search direction computation, CCD, and line search. Speedup is calculated as (Baseline / Ours) and shows: vs GIPC, vs StiffGIPC.

Example	Mesh DOFs (v, t)	Method	Grad & Hess	Moving Dir	CCD	Line Search	Total Time	Speedup
Octopus Stack	10K, 39K	GIPC	0.118	0.626	0.042	0.128	0.92	**3.71×, 1.97×**
		StiffGIPC	0.049	0.281	0.032	0.092	0.49	
		Ours	0.075	**0.023**	0.066	0.079	**0.25**	
Single Bunny	20K, 80K	GIPC	0.111	0.298	0.016	0.050	0.49	**3.64×, 1.54×**
		StiffGIPC	0.052	0.099	0.011	0.032	0.21	
		Ours	0.062	**0.019**	0.027	0.023	**0.13**	
Two Bunnies	40K, 160K	GIPC	0.257	0.752	0.031	0.127	1.18	**5.66×, 2.07×**
		StiffGIPC	0.113	0.207	0.024	0.078	0.43	
		Ours	0.090	**0.026**	0.035	0.054	**0.21**	
Armadillo	8K, 22K	GIPC	0.033	0.133	0.013	0.038	0.22	**2.12×, 1.03×**
		StiffGIPC	0.014	0.051	0.011	0.028	0.11	
		Ours	0.018	**0.011**	0.023	0.052	**0.11**	
Dragon High	78K, 274K	GIPC	1.565	7.895	0.412	0.981	11.05	**3.25×, 1.16×**
		StiffGIPC	0.845	2.281	0.227	0.465	3.95	
		Ours	1.091	**0.394**	0.692	1.141	**3.39**	

We break down the per-frame computation into several key stages: gradient and Hessian calculation, search direction computation, CCD and line search.

An observation is the dramatic reduction in search direction computation time. While GIPC and StiffGIPC spend significant time solving linear systems (0.13–7.9s depending on scene complexity), our method computes preconditioned search directions in just 0.01–0.39s by avoiding explicit Hessian assembly and leveraging efficient MAS preconditioning with Woodbury updates. This stage contributes the majority of our speedup.

Another observation is that while our method substantially reduces the bottleneck time in computing search directions, the quasi-Newton nature of PNCG requires more iterations to converge compared to full Newton methods. Consequently, the cumulative time spent on CCD and line search increases proportionally with the iteration count. Nevertheless, the dramatic savings in per-iteration direction computation far outweigh this overhead, resulting in net performance gains across all tested scenarios.

Across five benchmark scenes with varying scales (8K–78K vertices), MAS-PNCG consistently outperforms both baselines. We achieve 2.12–5.66× speedup over GIPC and 1.03–2.07× speedup over StiffGIPC. The largest gains occur in the scenario "Two Bunnies" (5.66× over GIPC, 2.07× over StiffGIPC. See Figure 12) where our incremental preconditioner updates efficiently track evolving contact configurations. Even in the challenging "Dragon High" scene with 274K tetrahedra, we maintain a 3.25× speedup over GIPC while achieving comparable accuracy.

These results confirm that our method's efficiency advantage is not simply due to using a different solver framework, but rather stems from our key algorithmic innovations: (1) Sparse-Input Woodbury updates that exploit the low-rank structure of contact Hessian changes for efficient preconditioner maintenance, and (2) 2D subspace minimization that leverages curvature information for optimal search directions. Together, these techniques enable MAS-PNCG to outperform Newton-PCG methods that already use the same MAS preconditioning.

5.2 Stress Tests

To further validate robustness and scalability under extreme conditions we ran several highly demanding benchmarks.

Stretching Armadillo. In Figure 15, we setup a large-deformation simulation scene: the armadillo is gradually stretched by a large external force, then the force is suddenly removed. The result shows that our method allows the armadillo to rebound within a few frames. This demonstrates that our method can maintain convergence comparable to that of Newton's method in large-deformation scenarios.

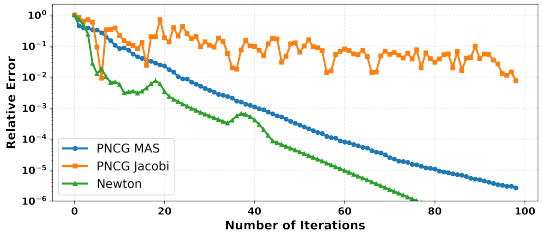
Dragon High Resolution. Figure 16 shows a free-fall scene of a high-resolution dragon model (270k tetrahedral elements). In this high-DoF setting, GIPC requires 11.05 s to simulate per frame, StiffGIPC requires 3.95 s, while our method takes only 3.39 s.

Cloth Twisting. As shown in Figure 13, we twist a narrow cloth strip through four full turns, creating dense self-collisions. Jacobi-PNCG frequently diverges or permits interpenetration in this setting. Conversely, MAS-PNCG converges reliably with low error, maintains penetration-free configurations.

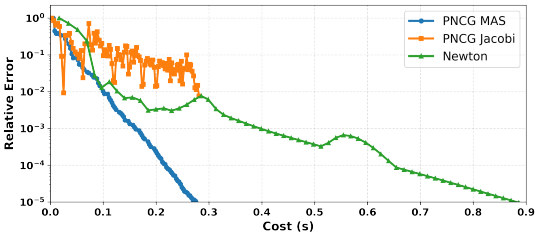
Large-Scale Simulation. The "Puffer Balls" scene (Figure 14) simulates 8 puffer balls with 1.14 M vertices, 2.72 M tetrahedra and over one million simultaneous contact pairs. Traditional Jacobi-PNCG fails due to penetrations and excessive runtime, while MAS-PNCG maintains stable, penetration-free motion and achieves convergence quality comparable to full Newton methods while remaining computationally efficient.

6 CONCLUSION

We have presented MAS-PNCG, an efficient and robust multilevel preconditioned nonlinear conjugate gradient method tailored for Incremental Potential Contact simulations. Our approach significantly accelerates convergence and enhances stability, especially for ill-conditioned systems arising from stiff materials or intricate contact configurations. This is achieved by integrating a Multilevel



(a)



(b)

Fig. 3. Convergence and performance comparison between MAS-PNCG, Jacobi-PNCG, and GPU Newton-PCG with MAS preconditioner (GIPC). a) shows the relative error curve with the number of iterations. MAS preconditioning enables significantly higher accuracy convergence than Jacobi-PNCG. b) shows the relative error curve over time. MAS-PNCG reaches target accuracy faster due to reduced per-iteration overhead.

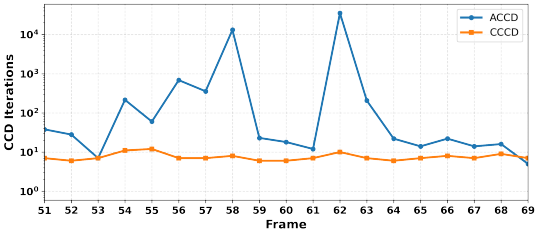


Fig. 4. Comparison of CCD iteration counts per NCG step between CCCD and ACCD methods across frames following Figure 11 (left). CCCD dramatically reduces iterations (tens vs. 30,000+) while maintaining penetration-free guarantees by prioritizing safe steps over exact TOI computation, resulting in approximately 10% end-to-end performance improvement.

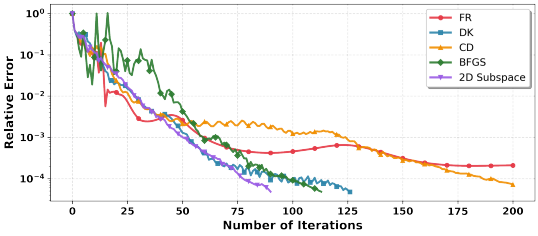


Fig. 5. Convergence comparison of different nonlinear CG formulas: Fletcher-Reeves (FR), Dai-Kou (DK), Conjugate Descent (CD), BFGS, and our 2D subspace minimization. The 2D subspace method achieves consistently faster and more stable convergence.

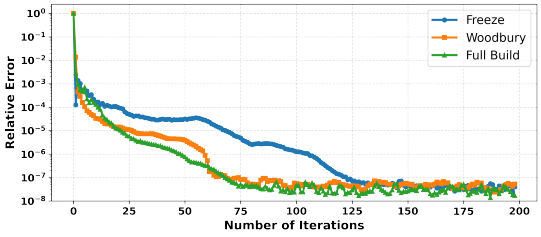


Fig. 6. Comparison of preconditioner updating strategies (the "Octopus Stack" scene). Our Sparse-Input Woodbury updating achieves convergence rates comparable to *Full Rebuild*, while converging substantially faster than the *Freeze*.

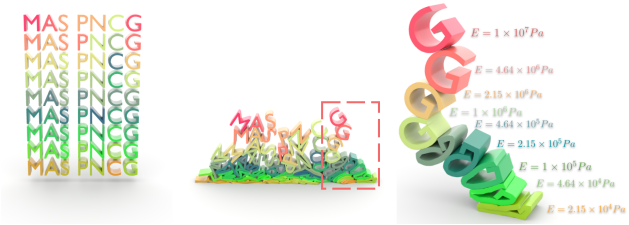


Fig. 7. **Dropping Letters**. A stack of 3D letters whose Young's moduli grow geometrically from 2.15×10^4 Pa (bottom) to 1×10^7 Pa (top). Left: the initial frame. Center: a frame during the simulation. Right: a zoom-in view of the dashed box in the center figure. MAS-PNCG remains robust across this 500× stiffness span, whereas Jacobi frequently fails to converge. Figure 2 shows the convergence comparison of different stiffness.

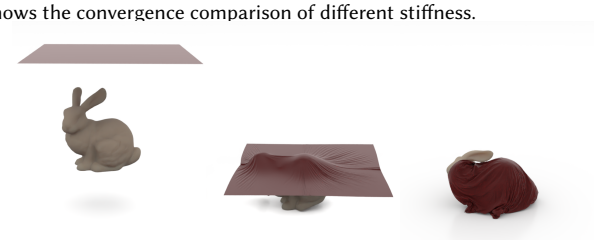


Fig. 8. **Bunny Cloth.** A cloth sheet falls onto a dynamic bunny with friction. MAS-PCG remains stable and faithfully captures friction-induced wrinkles and sliding.

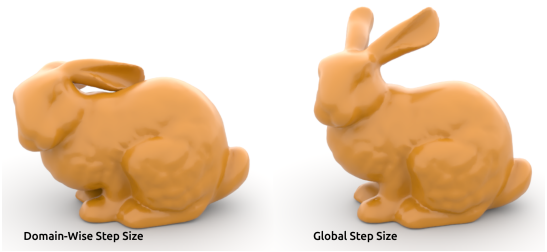


Fig. 9. **Single Bunny.** A bunny falls to the ground. With per-subdomain step sizes the bunny's ears sag naturally under gravity, whereas the global strategy yields noticeably overdamped motion despite identical termination criteria.

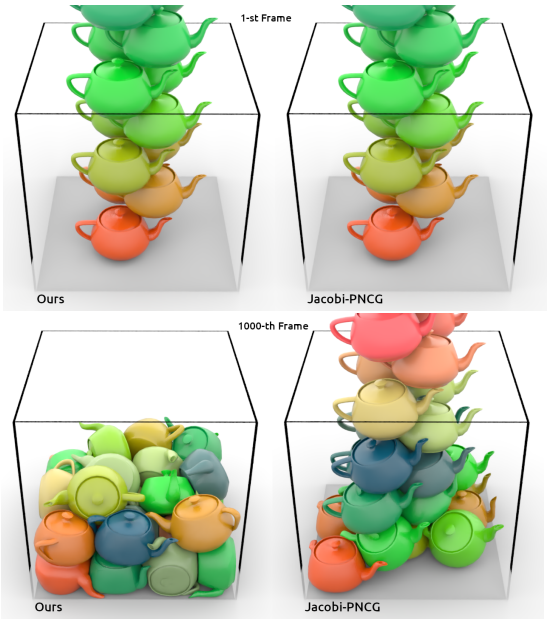


Fig. 10. **Teapots.** 32 elastic teapots are dropped into a box. We benchmark the proposed MAS preconditioner against the classical Jacobi preconditioner. For the same wall-clock budget, MAS attains markedly higher accuracy than Jacobi: MAS delivers smooth, physically plausible dynamics, whereas Jacobi exhibits excessive damping and severe distortion—clear symptoms of poor convergence. In fact, to match the visual fidelity of MAS, Jacobi must run for more than an order of magnitude longer and still occasionally diverges.

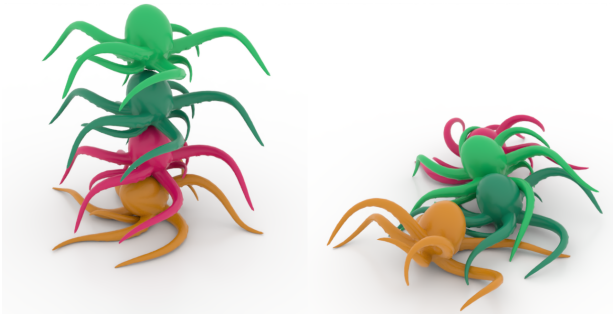


Fig. 11. **Octopus Stack.** Four octopuses fall to the ground. The frame on the left shows that the octopuses first come into mutual contact. We record the average *maximum* number of CCD iterations required per NCG step over several successive frames. The results are plotted in Figure 4.

Fig. 12. **Two bunnies.** One bunny falls on top of another bunny. Our method runs in 0.21 s per frame compared with GIPC's 1.18 s and StiffGIPC's 0.43 s. Despite this substantial runtime reduction, the visual results obtained by our faster MAS-PNCG solver are virtually comparable to those produced by the full Newton-based GIPC method.

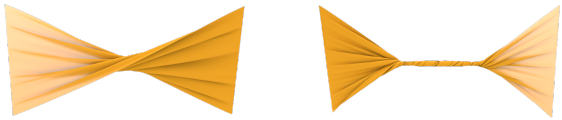


Fig. 13. **Cloth Twisting.** A narrow cloth strip is twisted through four full turns. Jacobi-PNCG frequently diverges or permits interpenetration in this setting. Conversely, MAS-PNCG converges reliably to a small error, preserves a penetration-free state.

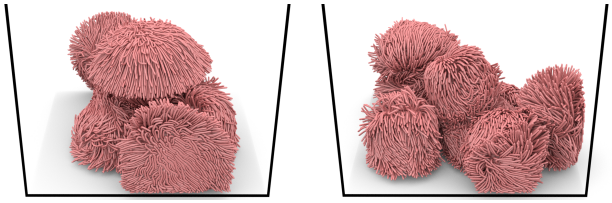


Fig. 14. **Puffer Balls.** A few frames from the same simulated scene as the Figure 1.

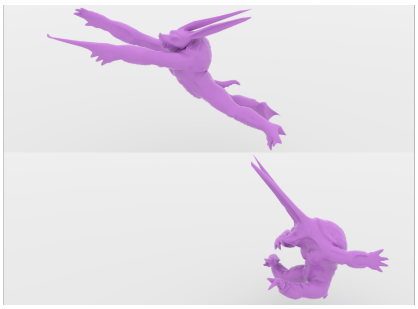


Fig. 15. **Stretching Armadillo.** (1) shows armadillo is gradually stretched by a large external force. (2) shows that the armadillo rebound within a few frames when the stretching force is suddenly removed.

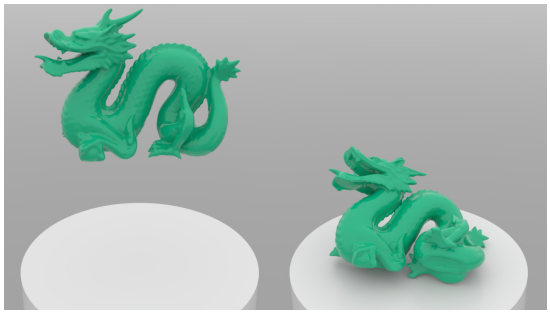


Fig. 16. **Dragon High.** A free-fall scene of a high-resolution dragon model (270k tetrahedral elements).

Additive Schwarz preconditioner with novel update strategies, including Sparse-Input Woodbury updates for fine-level components and Powell's restart for global rebuilds. Complemented by an optimal 2D subspace minimization strategy for search direction and step-size determination, and a fast, conservative CCD with a tighter relative-motion lower bound for penetration-free steps, MAS-PNCG offers a compelling balance of speed and robustness.

Experimental evaluations confirm that MAS-PNCG achieves accuracy comparable to state-of-the-art MAS preconditioning Newton-PCG solvers while demonstrating substantially improved computational efficiency. This advantage is particularly pronounced in large-scale simulations, positioning our method as a scalable alternative. The ablation studies systematically validated the positive contributions of each key component: the efficient Sparse-Input Woodbury update scheme, the optimal subspace descent, the conservative CCD, and the improved lower bound formulation. We plan to release our implementation as open-source code to benefit the research community.

References

Mehiddin Al-Baali. 2014. Damped techniques for enforcing convergence of quasi-Newton methods. *Optimization Methods and Software* 29, 5 (2014), 919–936.

David Belgrod, Bolun Wang, Zachary Ferguson, Xin Zhao, Marco Attene, Daniele Panozzo, and Teseo Schneider. 2021. Time of Impact Dataset for Continuous Collision Detection and a Scalable Conservative Algorithm. *arXiv preprint arXiv:2112.06300* (2021).

Yu Fang, Minchen Li, Yadi Cao, Xuan Li, Joshuah Wolper, Yin Yang, and Chenfanfu Jiang. 2023. Augmented incremental potential contact for sticky interactions. *IEEE Transactions on Visualization and Computer Graphics* 30, 8 (2023), 5596–5608.

Zachary Ferguson, Pranav Jain, Denis Zorin, Teseo Schneider, and Daniele Panozzo. 2023. High-order incremental potential contact for elastodynamic simulation on curved meshes. In *ACM SIGGRAPH 2023 conference proceedings*. 1–11.

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.

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

Florian Hecht, Yeon Jin Lee, Jonathan R Shewchuk, and James F O'Brien. 2012. Updated sparse cholesky factors for corotational elastodynamics. *ACM Transactions on Graphics (TOG)* 31, 5 (2012), 1–13.

Kemeng Huang, Floyd M Chitalu, Huancheng Lin, and Taku Komura. 2024a. GIPC: Fast and stable Gauss-Newton optimization of IPC barrier energy. *ACM Transactions on Graphics* 43, 2 (2024), 1–18.

Kemeng Huang, Xinyu Lu, Huancheng Lin, Taku Komura, and Minchen Li. 2025. StiffGIPC: Advancing GPU IPC for Stiff Affine-Deformable Simulation. *ACM Transactions on Graphics* (2025).

Zizhou Huang, Davi Colli Tozoni, Arvi Gjoka, Zachary Ferguson, Teseo Schneider, Daniele Panozzo, and Denis Zorin. 2024b. Differentiable solver for time-dependent deformation problems with contact. *ACM Transactions on Graphics* 43, 3 (2024), 1–30.

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

Lei Lan, Minchen Li, Chenfanfu Jiang, Huamin Wang, and Yin Yang. 2023. Second-order Stencil Descent for Interior-point Hyperelasticity. *ACM Trans. Graph.* 42, 4, Article 108 (jul 2023), 16 pages. doi:10.1145/3592104

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

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

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

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

Xuan Li, Yu Fang, Lei Lan, Huamin Wang, Yin Yang, Minchen Li, and Chenfanfu Jiang. 2023. Subspace-preconditioned gpu projective dynamics with contact for cloth simulation. In *SIGGRAPH Asia 2023 Conference Papers*. 1–12.

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.

Jorge Nocedal and Stephen J Wright. 1999. *Numerical optimization*. Springer.

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.

Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable neo-hookean flesh simulation. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–15.

Barry F. Smith, Petter E. Bjorstad, and William D. Gropp. 1996. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge, UK.

Pengbin Tang, Stelian Coros, and Bernhard Thomaszewski. 2023. Beyond Chainmail: Computational Modeling of Discrete Interlocking Materials. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–12.

Andrea Toselli and Olof Widlund. 2004. *Domain decomposition methods-algorithms and theory*. Vol. 34. Springer Science & Business Media.

Bolun Wang, Zachary Ferguson, Xin Jiang, Marco Attene, Daniele Panozzo, and Teseo Schneider. 2022. Fast and exact root parity for continuous collision detection. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 355–363.

Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, and Daniele Panozzo. 2021. A large-scale benchmark and an inclusion-based algorithm for continuous collision detection. *ACM Transactions on Graphics (TOG)* 40, 5 (2021), 1–16.

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.

Botao Wu, Zhendong Wang, and Huamin Wang. 2022. A GPU-based multilevel additive schwarz preconditioner for cloth and deformable body simulation. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.

Tianyi Xie, Minchen Li, Yin Yang, and Chenfanfu Jiang. 2023. A contact proxy splitting method for Lagrangian solid-fluid coupling. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–14.

Cem Yuksel. 2022. A Fast & Robust Solution for Cubic & Higher-Order Polynomials. In *ACM SIGGRAPH 2022 Talks* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 28, 2 pages. doi:10.1145/3532836.3536266