

# MiNNIE: a Mixed Multigrid Method for Real-time Simulation of Nonlinear Near-Incompressible Elastics

LIANGWANG RUAN, School of Computer Science, Peking University, China

BIN WANG, State Key Laboratory of General Artificial Intelligence, BIGAI, China

TIANTIAN LIU\*, Taichi Graphics, China

BAOQUAN CHEN\*, State Key Laboratory of General Artificial Intelligence, Peking University, China

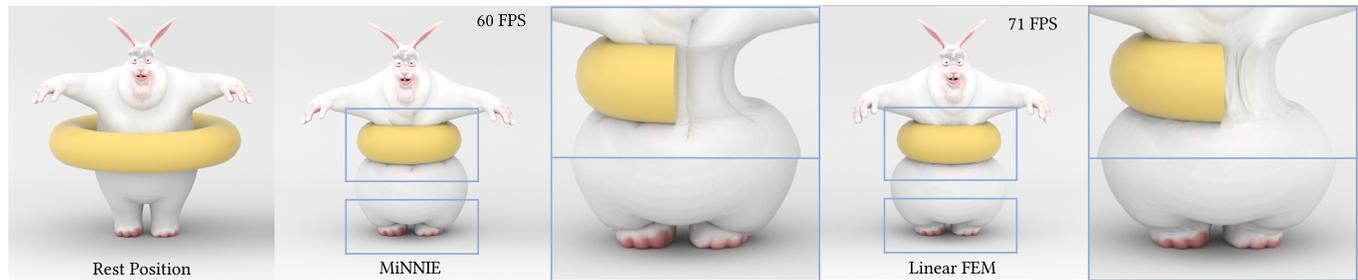


Fig. 1. A near-incompressible bunny with Poisson's ratio  $\nu = 0.49$ , consisting of 47k vertices and 223k elements, is squeezed by a gradually shrinking torus. Our solver, MiNNIE, generates vivid fat folds near the torso and ankles as shown in the close-up views (where the torus is intentionally rendered in half). In contrast, linear FEM solved with the GPU multigrid solver [Xian et al. 2019] fails to produce these details due to the volumetric locking artifact. The volume loss of MiNNIE and linear FEM are 0.8% and 1.2% respectively. In terms of efficiency, MiNNIE takes an average of 16.8 ms per frame. It is only 20% more expensive than [Xian et al. 2019] which takes on average 14.1 ms to produce a frame. Model courtesy of ©Blender.

We propose MiNNIE, a simple yet comprehensive framework for real-time simulation of nonlinear near-incompressible elastics. To avoid the common volumetric locking issues at high Poisson's ratios of linear finite element methods (FEM), we build MiNNIE upon a mixed FEM framework and further incorporate a pressure stabilization term to ensure excellent convergence of multigrid solvers. Our pressure stabilization strategy injects bounded influence on nodal displacement which can be eliminated using a quasi-Newton method. MiNNIE has a specially tailored GPU multigrid solver including a modified skinning-space interpolation scheme, a novel vertex Vanka smoother, and an efficient dense solver using Schur complement. MiNNIE supports various elastic material models and simulates them in real-time, supporting a full range of Poisson's ratios up to 0.5 while handling large deformations, element inversions, and self-collisions at the same time.

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

Additional Key Words and Phrases: Mixed FEM, Locking, Multigrid, Real-time simulation

\*corresponding authors

Authors' addresses: Liangwang Ruan, School of Computer Science, Peking University, Beijing, China, ruanliangwang@pku.edu.cn; Bin Wang, State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China, binwangbuaa@gmail.com; Tiantian Liu, Taichi Graphics, Beijing, China, ltt1598@gmail.com; Baoquan Chen, State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China, baoquan@pku.edu.cn.

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

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0730-0301/2024/12-ART258 \$15.00

<https://doi.org/10.1145/3687758>

## ACM Reference Format:

Liangwang Ruan, Bin Wang, Tiantian Liu, and Baoquan Chen. 2024. **MiNNIE: a Mixed Multigrid Method for Real-time Simulation of Nonlinear Near-Incompressible Elastics**. *ACM Trans. Graph.* 43, 6, Article 258 (December 2024), 15 pages. <https://doi.org/10.1145/3687758>

## 1 INTRODUCTION

Incompressibility is a common property for many deformable materials such as human tissue, fat, rubber, and hydrogel. This volume-preserving property is essential for various applications, including entertainment, virtual surgery, fabrication, robotics, and safety analysis. As the Poisson's ratio  $\nu$  of those near-incompressible materials approaches to 0.5, the bulk modulus  $\frac{E}{3(1-2\nu)}$  that measures the resistance of volume change goes towards infinity. This results in extremely ill-conditioned problems for classic finite element methods, causing notable locking and instability artifacts [Irving et al. 2007]. One example to show this locking artifact can be seen in Fig. 2.

One solution to this high-stiffness problem is using the mixed Galerkin finite element (mixed FEM) formulation. The mixed FEM treats the volume preservation constraints as Lagrange multipliers upon auxiliary pressure terms  $p$  instead of vertex positions  $x$ , turning this constrained deformation problem a saddle-point problem. In theory, the mixed FEM imposes stringent requirements on choosing solution spaces for  $x$  and  $p$ . Specifically, high-order elements that adhere to the Ladyzhenskaya–Babuška–Brezzi (LBB) or inf-sup condition ensure convergence and unique solutions [Boffi et al. 2008]. But they are complicated to implement and slow to solve. Low-order elements, on the other hand, are much more efficient to solve. However, the P1/P0 elements (featuring a linear



Fig. 2. A dancing bunny with  $\nu = 0.49$  whose back neck is constrained by a periodically-moving controller. The 0th, 10th, 18th, 27th, 38th, and 44th frames are shown here. MiNNIE (top) generates vivid dynamics while Linear FEM with [Xian et al. 2019] (bottom) produces much stiffer movements due to locking. Model courtesy of ©Blender.

displacement field and element-wise constant pressure field) are notorious for their volumetric locking issues [Frâncu et al. 2021], while increasing the order of the pressure field to the same as displacements (P1/P1 elements) introduces oscillatory pressure modes, often referred to as the *checkerboard* artifact [Sani et al. 1981]. The locking-proof tetrahedra [Frâncu et al. 2021] is the state-of-the-art near-incompressible elastic material simulator based on the P1/P1 elements. However, its performance, solved by either direct solvers or its proposed nonlinear Uzawa solver, is still insufficient to reach the real-time requirement. We provide the full derivation of mixed FEM with P1/P1 elements and explain the problem of applying well-studied solvers to it in Sec. 3.

Naive adoption of efficient multigrid solvers [Liu et al. 2018; Shao et al. 2022; Xian et al. 2019] does not help to improve the performance of the P1/P1-element-based solvers. We observe that this is caused by the oscillatory high-frequency spurious pressure mode, also known as the *checkerboard* artifact. This oscillatory pressure mode produces high-frequency errors in the near-kernel space. Meanwhile, multigrid algorithms are designed to eliminate high-frequency far-kernel errors by smoothers, and handle low-frequency near-kernel errors by passing them to coarser levels [Tamstorf et al. 2015]. This high-frequency near-kernel error cannot be quickly mitigated using smoothers, nor corrected by coarser grids [Zhu et al. 2010]. That is why multigrid solvers typically behave with poor convergence or even numerical instability in solving the P1/P1 elements.

We propose MiNNIE: a Mixed Multigrid Method for Real-time Simulation of Nonlinear Near-Incompressible Elastics. MiNNIE is designed upon a key observation that simple pressure stabilization [Dohrmann and Bochev 2004] which directly penalizes the pressure difference for each node and its neighbors, can greatly mitigate the checkerboard artifact. This pressure stabilization strategy thus enables the possibility of further accelerations using multigrid schemes. We study the influence of pressure stabilization on

nodal displacement under different strengths and element scales, and show this influence has a theoretical upper bound. We also provide a quasi-Newton option if optimal positional accuracy is needed. We discuss the pressure stabilization strategy in detail in Sec. 4. Given the stabilized pressure, we further design a specially tailored multigrid framework that includes a modified interpolation scheme, a novel smoother for indefinite systems, and a Schur-complement dense solver. Our multigrid solver uses the skinning space coordinates [Xian et al. 2019] which allows us to reduce the coarse resolution more aggressively for both position and pressure DoFs. To support the indefinite nature of the mixed FEM system in multi-resolution grids, we propose a novel vertex Vanka smoother and show its superior performance among all competing smoothers on GPU. We also take advantage of the constant compliant matrix and stabilization matrix and design a fast dense solver using pre-factorized matrix components. All details of our multigrid method can be found in Sec. 5. At last, to apply MiNNIE to various materials, we derive the split form of common constitutive models such as the stable Neo-Hookean, Mooney-Rivlin, and Corotation model in Sec. 6 and discuss the method to evaluate the explicit eigensystem of positional Hessian in the mixed form for general materials.

As a result, MiNNIE demonstrates remarkable robustness accommodating a wide range of Poisson’s ratio up to 0.5. MiNNIE does not suffer from locking artifacts and delivers vivid visual effects as shown in Fig. 1 and Fig. 2. MiNNIE supports large-scale models with stiff materials efficiently. For instance, it simulates the model with 105k vertices and 371k elements in 26 ms on average per frame, supporting self-collisions and degenerated or even inverted elements, preserving volumes after recovering from inversions as shown in Fig. 14. We release the official implementation of MiNNIE at <https://github.com/LwRuan/MiNNIE>.

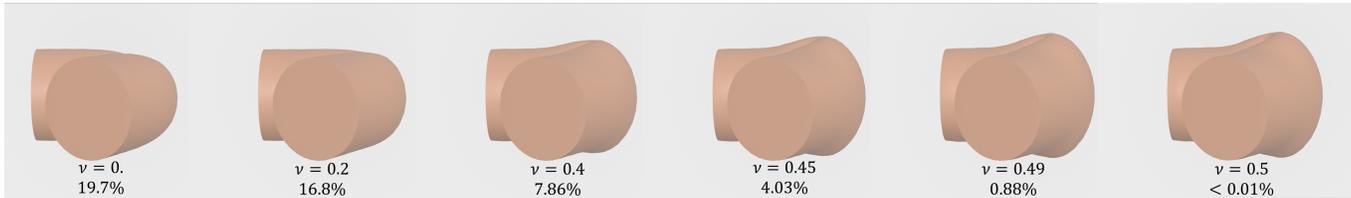


Fig. 3. Bulging. A cylinder with 10k vertices and 57k elements is bent with internal skeletons with positional constraints. We show different bulging effects and total volume loss for different Poisson's ratios  $\nu$ .

## 2 RELATED WORK

*Near-incompressible elastic simulation.* Incompressibility is a well-known concept in fluid simulations, often addressed through pressure projection to enforce divergence-free velocity constraints [Bridson 2015]. In the context of elastics, Irving et al. [2007] applied similar principles, incorporating a post-projection step to correct the velocities of elastic bodies. Dziol et al. [2011] enforce incompressibility by imposing a global volume constraint solely on the surface mesh. In contrast, Sheen et al. [2021] introduced region-based volume constraints for simulating human tissues. However, these methods can only simulate the incompressible limit ( $\nu = 0.5$ ), fail to capture the true Poisson's ratio's behaviors in physical materials. Alternatively, various methods have been developed to simulate elastic materials with volume penalties, including co-rotated [McAdams et al. 2011], Neo-Hookean [Liu et al. 2017], stable Neo-Hookean [Smith et al. 2018], StVK with volume terms [Kikuuwe et al. 2009; Teschner et al. 2004; Wang et al. 2019]. However, directly simulating these materials with linear tetrahedral elements often leads to noticeable locking artifacts [Frâncu et al. 2021; Irving et al. 2007]. Using higher-order elements like quadratic elements [Bargteil and Cohen 2014] and the discontinuous Galerkin method [Kaufmann et al. 2009], avoids the locking issue, at a cost of low run-time performance. Another approach to mitigate locking issues is the mixed FEM method [Boffi et al. 2008], which we adopt in this paper. Additionally, incompressibility can be incorporated into extended position-based dynamics (XPBD) [Macklin and Muller 2021], where both the distortion and volumetric energy are formulated into constraints. However, strictly preserving volume in XPBD often requires numerous iterations. Additionally, constraint-level parallelism using graph coloring limits its ability to handle tetrahedral meshes in complex scenarios, possibly requiring hundreds of colors to colorize all the elements.

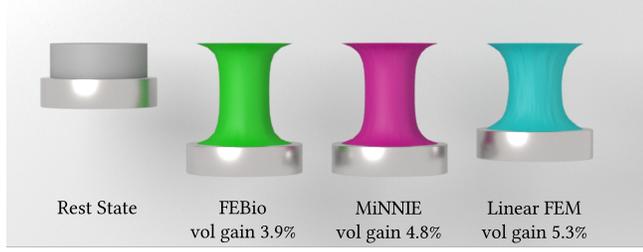
*Mixed FEM.* The mixed FEM has been widely adopted in engineering for simulating various phenomena since the last century [Brezzi and Fortin 1991]. In the field of computer graphics, Roth et al. [1998] were among the first to utilize mixed FEM to simulate linear elasticity for facial animations. Many previous works in computer graphics have focused on applying mixed FEM to regular grids. For instance, Zhu et al. [2010] introduced the mixed formulation on regular grids to simulate linear or co-rotated linear elasticity. Patterson et al. [2012] extended this approach to simulate nonlinear elastic bodies using a lattice deformer, achieving higher-order accuracy. Stomakhin et al. [2014] developed a mixed formulation for the material point method to simulate incompressible objects. Setaluri et al. [2015] also used regular grids in mixed form to warp 2D images.

While regular grids offer computational and storage efficiency, they are not conformal to underlying meshes. This raises challenges such as boundary handling, embedding artifacts, and limitations in handling complex geometries. For tetrahedral meshes, the work by Irving et al. [2007] can be considered as a variation of P1/P1 mixed FEM, but solved with operator splitting, as commonly done in fluid simulation [Bridson 2015]. Misztal et al. [2012] simulated multi-phase fluid using a P1/P0 mixed formulation. Frâncu et al. [2021] applied P1/P1 mixed FEM to simulate near-incompressible elastic bodies, but their solver was not sufficiently fast for real-time applications and left the checkerboard artifacts unsolved. More recently, another line of work by Trusty et al. [2023, 2022] explored the application of mixed FEM to address the convergence problems of large rotations, rather than near-incompressibility.

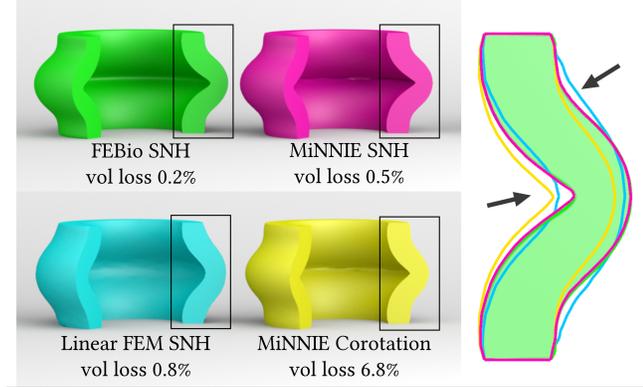
*Multigrid.* Multigrid methods are famous for their rapid convergence, particularly when dealing with large meshes and detail-rich deformations [Georgii and Westermann 2006]. These techniques have found applications in a variety of graphic tasks involving diverse geometric structures, such as regular grids [Liu et al. 2018; McAdams et al. 2011; Shao et al. 2022], triangle meshes [Liu et al. 2021; Tamstorf et al. 2015; Wang et al. 2018], tetrahedral meshes [Sacht et al. 2015; Xian et al. 2019]. However, most previous research primarily focuses on solving symmetric positive definite (SPD) problems. In the case of indefinite systems, particularly those arising from mixed FEM like the one employed in our work, there exists a certain foundation of research in the industry. Benzi et al. [2005] conducted an extensive survey of various multigrid approaches, as did by Oosterlee and Gaspar [2008]. However, in the realm of computer graphics, related research is still relatively rudimentary. Zhu et al. [2010] developed a geometric multigrid for regular grids to simulate linear or co-rotated elastic materials, and Setaluri et al. [2015] built a geometric multigrid on 2D regular grids for image warping. In contrast, our approach involves a Galerkin multigrid method capable of effectively handling 3D unstructured meshes and nonlinear materials, which is a substantial complement to existing methods.

## 3 MOTIVATION & BACKGROUND

In this section, we provide the motivation and a brief overview of the mixed finite element method (mixed FEM) for simulating nearly incompressible solids with linear tetrahedral elements. We use italic letters like  $\mathbf{x}$ ,  $p$  for continuous field, and roman letters like  $\mathbf{x} \in \mathbb{R}^{3n}$ ,  $p \in \mathbb{R}^n$  for discrete values, where  $n$  is the number of vertices.



(a) A stable Neo-Hookean (SNH) elastic cylinder with  $\nu = 0.49$  is stretched by a suspended weight.



(b) An elastic tube with  $\nu = 0.49$  is compressed to 70% of its height.

Fig. 4. We compare the results of 1. second-order FEM using FEBio [Maas et al. 2012], 2. MiNNIE, 3. Linear FEM using [Xian et al. 2019] and report their volume change. In both cases (a) and (b), linear FEM tends to be stiffer due to volumetric locking, while MiNNIE produces more similar results with second-order FEM and preserves volume better. We also show the result of the corotation model simulated by MiNNIE in (b), it uses a linearized volume-preserving constraint and hence does not preserve volume well.

For comprehensive derivations, please refer to the supplemental material.

### 3.1 Problem Formulation

Following [Fráncu et al. 2021; Patterson et al. 2012], we assume common constitutive models adhere to the split form:

$$\Psi(\mathbf{x}) = \Psi_d(\mathbf{x}) + \Psi_v(\mathbf{x}), \quad \Psi_v(\mathbf{x}) = \frac{\kappa}{2} \Phi(\mathbf{F})^2. \quad (1)$$

Here,  $\Psi_v(\mathbf{x})$  denotes a quadratic penalty enforcing the volume preservation constraint  $\Phi(\mathbf{F}) = 0$ , where  $\mathbf{F}$  is the deformation gradient and  $\kappa$  is the penalty stiffness.  $\Psi_d(\mathbf{x})$  constitutes the remaining part in  $\Psi(\mathbf{x})$  apart from  $\Psi_v(\mathbf{x})$ , primarily penalizing distortion. Many existing constitutive models align well with this split form. For instance, the Neo-Hookean [Bonet and Wood 2008] model has  $\Phi(\mathbf{F}) = \log J = \log(\det \mathbf{F})$  and  $\kappa = \frac{E\nu}{(1+\nu)(1-2\nu)}$  where  $E$  is the Young's modulus and  $\nu$  is the Poisson's ratio; the Mooney-Rivlin [Barbič et al. 2012] model has  $\Phi(\mathbf{F}) = J - 1 = \det \mathbf{F} - 1$  and  $\kappa = \frac{E}{3(1-2\nu)}$ ; and the corotation [McAdams et al. 2011] model has  $\Phi(\mathbf{F}) = \text{tr}(\mathbf{S} - \mathbf{I})$  and  $\kappa = \frac{E\nu}{(1+\nu)(1-2\nu)}$ , where  $\mathbf{S}$  is the symmetric part of  $\mathbf{F}$ 's polar decomposition, etc.. Note that the  $\Phi$  term in the corotation model is a linearized volume preservation constraint near

rest configurations [Patterson et al. 2012], it does not preserve exact volume under large deformation. In all examples above,  $\kappa$  goes to infinity as Poisson's ratio approaches 0.5 to enforce either the exact or the approximated volume preservation constraint  $\Phi(\mathbf{F}) = 0$ . Further elaboration on constitutive models will be provided in Section 6.

A common way to simulate elastodynamics is to incorporate Eq. 1 into a time integration problem which can be formulated using incremental potential optimization [Kane et al. 2000]:

$$\mathbf{x}^{n+1} = \arg \min_{\mathbf{x}} \mathcal{I}(\mathbf{x}), \quad (2)$$

$$\mathcal{I}(\mathbf{x}) = \int_{\Omega} [T(\mathbf{x}) + \Psi_d(\mathbf{x}) + \Psi_v(\mathbf{x})] d\Omega.$$

Here,  $\Omega$  is the simulation domain in rest configuration,  $T(\mathbf{x}) = \frac{\rho}{2h^2} \|\mathbf{x} - \mathbf{x}^*\|^2$  is the inertia term, where  $\mathbf{x}^* = \mathbf{x}^n + \nu^n h + \frac{1}{\rho} \mathbf{f}_{\text{ext}} h^2$ .  $\mathbf{f}_{\text{ext}}$ ,  $\rho$  and  $h$  correspond to the external force density, the mass density and the time step size, respectively. Eq. 2 is usually solved using the finite element method which discretizes  $\Omega$  with linear elements (P1 elements). However, the discretized system can be over-constrained when simulating near-incompressible materials with enormously stiff volume constraints. In a tetrahedral mesh with  $N_v$  vertices and  $N_e$  elements, the number of volume penalties ( $N_e$ ) is often comparable to the kinematic degrees of freedom ( $3N_v$ ), as noted in Tab. 2. Consequently, these volumetric penalties can eliminate most DoFs, resulting in an elastic body that is much stiffer than it should be. It is usually referred to as the *volumetric locking phenomena*. We demonstrate the locking phenomena under  $\nu = 0.49$  in Fig. 4, and more severe artifacts at larger Poisson's ratios in Fig. 10 and Fig. 8. The locking problem always occurs in linear FEM, and cannot be resolved by increasing mesh resolution (i.e. h-refinement). Higher-order FEM (i.e. p-refinement) mitigates this problem at the cost of employing much more complicated systems and less efficient solvers.

To address this issue, the mixed FEM introduces an auxiliary pressure field  $p$ , transforming Eq. 2 into a saddle point problem:

$$(\mathbf{x}^{n+1}, p^{n+1}) = \arg \min_{\mathbf{x}} \max_p \mathcal{L}(\mathbf{x}, p), \quad (3)$$

$$\mathcal{L}(\mathbf{x}, p) = \int_{\Omega} [T(\mathbf{x}) + \Psi_d(\mathbf{x}) + p\Phi - \frac{1}{2\kappa} p^2] d\Omega.$$

$\mathcal{L}(\mathbf{x}, p)$  is called the *perturbed Lagrangian* [Zienkiewicz et al. 2013]. When  $\kappa \rightarrow \infty$ ,  $\frac{1}{\kappa} = 0$  and  $p$  functions as a Lagrangian multiplier to enforce complete volume constraint  $\Phi = 0$ ; When  $\kappa < \infty$ , we have  $\max_p (p\Phi - \frac{1}{2\kappa} p^2) = \frac{\kappa}{2} \Phi^2 = \Psi_v(\mathbf{x})$ , where the maximum is reached at  $p = \kappa\Phi$ , thus  $\max_p \mathcal{L}(\mathbf{x}, p) = \mathcal{I}(\mathbf{x})$ , and the solution of Eq. 3 is the same as Eq. 2. The introduced pressure field isolates the discretization of the volume constraint  $\Phi$  from the discretization of displacement  $\mathbf{x}$ , thus limiting the number of constraints to avoid locking. The simplest non-locking discretization uses linear elements (P1/P1 elements) for both primal ( $\mathbf{x}$ ) and dual ( $p$ ) variables. The discretized Eq. 3 can be formulated as:

$$L(\mathbf{x}, p) = \frac{1}{2h^2} \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{M}}^2 + U_d(\mathbf{x}) + p^T \phi - \frac{1}{2} p^T \mathbf{C} p. \quad (4)$$

Here  $\mathbf{x} \in \mathbb{R}^{3n}$ ,  $p \in \mathbb{R}^n$ ,  $U_d \in \mathbb{R}$ ,  $\phi \in \mathbb{R}^n$ ,  $\mathbf{C} \in \mathbb{R}^{n \times n}$  are the discrete correspondences of  $\mathbf{x}$ ,  $p$ ,  $\Psi_d$ ,  $\Phi$ ,  $1/\kappa$  from Eq. 3, with their definitions listed in Tab. 1. Particularly for discrete volume constraint  $\phi$  we

have:

$$\phi_i = \sum_e \int_{\Omega_e} \Phi(\mathbf{F}_e) N_i(X) d\Omega = \frac{1}{4} \sum_{e \in \mathcal{N}_i} \Phi(\mathbf{F}_e) V_e, \quad (5)$$

where  $N_i(X)$  is the linear shape function of vertex  $i$ ,  $\mathcal{N}_i$  is the neighboring elements of vertex  $i$ ,  $V_e$  is the rest volume of element  $e$ . With the P1/P1 discretization, the number of discrete volume constraints is reduced to  $\dim(\phi) = n$ , which is much smaller than the system DoF  $\dim(\mathbf{x}) = 3n$ , thus avoiding the locking artifacts.

### 3.2 Numerical Solver

We employ the Newton-Raphson method to solve the Karush-Kuhn-Tucker (KKT) system derived from Eq. 4. This involves solving the following linearized symmetric indefinite system at each iteration:

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{K} & \mathbf{G}^T \\ \mathbf{G} & -\mathbf{C} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \quad (6)$$

with each quantity defined in Tab. 1. Though Eq. 6 involves a larger matrix of size  $4n \times 4n$  compared to the  $3n \times 3n$  matrix in classic linear FEM, the sub-blocks  $\mathbf{C}$ ,  $\mathbf{G}$ , and  $\mathbf{K}$  maintain the beneficial sparsity property of linear FEM.  $\mathbf{C}$  is a constant diagonal matrix encoding the compliance.  $\mathbf{G}$  is the Jacobian of  $\phi$ ,  $G_{ij} = \partial\phi_i/\partial x_j$  is nonzero only when vertex  $i$  and  $j$  belong to the same tetrahedron.  $\mathbf{K}$  is the positional Hessian matrix following the same sparsity pattern with linear FEM. This mixed FEM formulation has been widely used in mechanical engineering, but designing an efficient numerical solver for this problem remains challenging. The system in Eq. 6 is indefinite, preventing the application of most high-performance SPD solvers, such as DOT [Li et al. 2019], preconditioned Conjugate Gradient [Wu et al. 2022], and multigrid [Xian et al. 2019]. A straightforward solution is to turn this problem into a positive definite problem using Schur complement. We provide two options here:

$$\begin{aligned} \text{option 1: } & (\mathbf{K} + \mathbf{G}^T \mathbf{C}^{-1} \mathbf{G}) \delta\mathbf{x} = \mathbf{f} + \mathbf{G}^T \mathbf{C}^{-1} \mathbf{g}, \\ \text{option 2: } & (\mathbf{C} + \mathbf{G} \mathbf{K}^{-1} \mathbf{G}^T) \delta\mathbf{p} = -\mathbf{g} + \mathbf{G} \mathbf{K}^{-1} \mathbf{f}, \end{aligned} \quad (7)$$

However, neither of these options is efficient enough. For option 1,  $\mathbf{G}$  shares the same sparsity pattern as  $\mathbf{K}$ , thus making  $\mathbf{G}^T \mathbf{C}^{-1} \mathbf{G}$  much denser than  $\mathbf{K}$ : each vertex has nonzero coefficients to all its second-ring neighboring vertices. Moreover,  $\mathbf{C}^{-1}$  is proportional to the stiffness  $\kappa$ , which goes to infinity when  $\nu \rightarrow 0.5$ . For option 2, we need to deal with the non-trivial  $\mathbf{K}^{-1}$  sandwiched in the matrix, which requires nested sparse solvers. For these reasons, we decide to develop a monolithic GPU multigrid solver specially designed for mixed FEM utilizing P1/P1 elements (Eq. 6).

## 4 PRESSURE STABILIZATION

Before diving into our multigrid solver, we would first like to emphasize an oscillatory pressure problem inherited from the P1/P1 elements, as it undermines the stability of multigrid solvers. For simplicity, let us assume a case where boundary conditions are ignored and the Poisson's ratio is exactly 0.5. In this case,  $\kappa$  goes infinity and  $\mathbf{C} = 0$ , the only pressure term in the discrete saddle point in Eq. 4 is the  $\mathbf{p}^T \phi$  term. We expand this term using Eq. 5:

$$\mathbf{p}^T \phi = \sum_i p_i \phi_i = \sum_i p_i \frac{1}{4} \sum_{e \in \mathcal{N}_i} \Phi(\mathbf{F}_e) V_e. \quad (8)$$

Table 1. The definition of notations in P1/P1 mixed FEM.

Definition	Shape	Meaning
$V_i = 1/4 \sum_{e \in \mathcal{N}_i} V_e$	scalar	rest volume of vertex $i$
$\mathbf{M} = \text{Diag}(\rho V_i \mathbf{I}_3)$	$3n \times 3n$	mass matrix
$U_d(\mathbf{x}) = \sum_e \Psi_d(\mathbf{F}_e) V_e$	scalar	distortion energy
$\phi = [\phi_1, \dots, \phi_n]^T$	$n \times 1$	vertex volume constraint
$\mathbf{C} = \text{Diag}(V_i/\kappa)$	$n \times n$	inverted volumetric stiffness matrix
$\mathbf{G} = \partial\phi/\partial\mathbf{x}$	$n \times 3n$	Jacobian of constraint
$\mathbf{f}_d = -\partial U_d/\partial\mathbf{x}$	$3n \times 1$	distortion force
$\mathbf{f} = -\partial L/\partial\mathbf{x}$ $= \mathbf{M}/h^2(\mathbf{x}^* - \mathbf{x}) + \mathbf{f}_d - \mathbf{G}^T \mathbf{p}$	$3n \times 1$	$\mathbf{f} = 0$ is the KKT condition for $\mathbf{x}$
$\mathbf{g} = -\partial L/\partial\mathbf{p}$ $= \mathbf{C}\mathbf{p} - \phi$	$n \times 1$	$\mathbf{g} = 0$ is the KKT condition for $\mathbf{p}$
$\mathbf{K} = \partial^2 L/\partial\mathbf{x}^2$	$3n \times 3n$	projected Hessian of position

This summation first iterates over each vertex  $i$ , then over each neighboring element  $e$  of vertex  $i$  ( $e \in \mathcal{N}_i$ ), which can be reordered by first iterating over each element  $e$ , then over each vertex  $i$  in element  $e$  ( $i \in \mathcal{E}_e$ ):

$$\mathbf{p}^T \phi = \sum_e \Phi(\mathbf{F}_e) V_e \frac{1}{4} \sum_{i \in \mathcal{E}_e} p_i = \sum_e \Phi(\mathbf{F}_e) V_e \bar{p}_e. \quad (9)$$

where  $\bar{p}_e := \frac{1}{4} \sum_{i \in \mathcal{E}_e} p_i$  is the average pressure in element  $e$ . From Eq. 9, we can see that only element-wise averaged pressure  $\bar{p}_e$ , not the individual  $p_i$ , contributes to  $\mathbf{p}^T \phi$ . As long as  $\bar{p}_e = 0$ , even if the pressure at each vertex  $p_i$  exhibits large alternating positive and negative values (i.e. oscillated pressures or the "checkerboard" pattern), the final equilibrium state of positions remains unaffected. In other words, this oscillated pressure mode is a zero mode of the saddle point problem in this case. In less contrived cases where  $\nu < 0.5$  and proper boundary conditions are applied, this oscillated pressure mode corresponds to small eigenvalues of the system. This means we can inject a large amount of oscillated pressure values into the system with negligible impact on the positional DoFs, as shown in the upper-left of Fig. 5. This problem may be safely disregarded in applications where the accuracy of only positional DoFs is primarily concerned, if solved using direct solvers [Fráncu et al. 2021]. However, this claim does not hold for multigrid solvers.

In a typical multigrid setup, smoothers (e.g. stationary iterative solves like Jacobi or Gauss-Seidel) can efficiently smooth out local high-frequency errors, corresponding to large eigenvalues of the system. Conversely, global low-frequency errors, corresponding to error modes with small eigenvalues, are transferred to coarser grids where they appear as high-frequency errors for coarse level smoothers. However, in our case, the oscillated pressure mode has small eigenvalues so it cannot be efficiently smoothed out by stationary smoothers. Moreover, the oscillated pressure mode is a high-frequency mode and thus can not be effectively transferred to coarser levels as well. As a result, multigrid solvers can neither eliminate the oscillatory pressure errors using their smoothers, nor pass those errors into coarser grids. Directly applying multigrid solvers to the P1/P1 mixed FEM systems often results in divergence.

The root cause of this problem lies in using the same order of discretization for both the position and pressure DoF. A similar issue arises in fluid simulations when the same discretization is applied to both velocity and pressure variables [Bridson 2015]. One effective solution is to use a staggered grid [Zhu et al. 2010], but this approach is limited to regular grids. For unstructured meshes, more complex discretization is often required to eliminate this artifact, for instance, the Taylor-Hood (P2/P1) element, the MINI element, the Crouzeix-Raviart element (also used in [English and Bridson 2008]), etc. [Boffi et al. 2008]. In our case, for computational efficiency, we adopt another well-established solution, known as the local projection stabilization method [Dohrmann and Bochev 2004]. This technique aims to suppress the oscillated pressure modes by projecting nodal pressure DoF to a lower order of discretization using an extra penalty term in the saddle point formulation of Eq. 3:

$$E_p = -\frac{\alpha}{2\mu} \sum_e \int_{\Omega_e} (N(X)p - \bar{p}_e)^2 d\Omega. \quad (10)$$

Here  $N(X) = [N_0(X), \dots, N_{n-1}(X)]$  are the linear shape functions of vertices,  $\bar{p}_e$  is the average pressure in element  $e$ ,  $\mu = \frac{E}{2(1+\nu)}$  is the shear modulus, and  $\alpha$  is a tunable strength parameter, which is set to 1 in [Dohrmann and Bochev 2004]. The negative sign is necessary because the pressure  $p$  maximizes the Lagrangian  $\mathcal{L}$ . This term tends to minimize the difference between nodal pressures  $p_i$  and element mean pressures  $\bar{p}_e$ , effectively targeting the unwanted oscillated pressure mode as previously analyzed.

Inevitably, since  $\mathbf{x}$  and  $\mathbf{p}$  are strongly coupled, this extra pressure penalty also affects the equilibrium state of positions, and the greater the penalty, the larger the position error. As shown in Fig. 5, for  $\alpha = 1$ , the oscillated pressure distribution is smoothed out, and the position deviation is not noticeable; however for  $\alpha = 30$ , the pressure field is excessively smoothed, leading to undesired position artifacts. Fortunately, regardless of the choice of pressure stabilization strength  $\alpha$ , the solutions always converge to the ground truth as the mesh resolution increases. This is established in Theorem 3.14 in Elman et al. [2014], which demonstrates that the error between the stabilized discrete solution and the true solution is bounded by a constant multiplied by the maximum edge length of the tetrahedrons. We also validate this theorem experimentally, as shown at the bottom of Fig. 5. Both the theoretical and experimental results confirm the correctness of our stabilization method. In cases where positional accuracy is of greater concern, a quasi-Newton solution can be employed to enforce the undisturbed final solution. Specifically, we apply the stabilization term from Eq. 10 only to the left-hand side of Eq. 6, leaving the right-hand side unchanged, as if no pressure stabilization were applied. This approach ensures convergence to equilibrium without any pressure stabilization, which results in smooth displacement and oscillatory pressure, albeit with slightly slower convergence speed. In practice, we apply full pressure stabilization in Eq. 10 for real-time dynamic simulations to achieve optimal dynamics, and the quasi-Newton strategy for quasi-static cases for optimal accuracy. We find a moderate amount of penalty ( $\alpha = 1$ ) works well for all our experiments, except for the extreme inversion test in Fig. 12 where we set  $\alpha = 20$ .

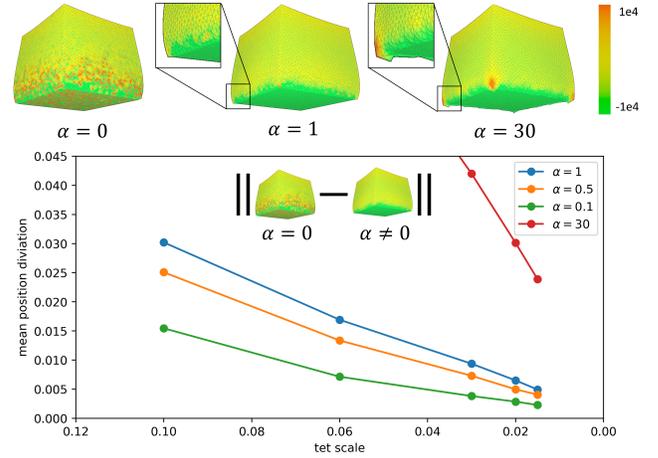


Fig. 5. Top: A cube with 14k vertices and 70k elements deforms under gravity with the bottom vertices fixed. The cubes are colored according to the pressure distribution at their equilibrium states solved by MINRES under three configurations: without pressure stabilization ( $\alpha = 0$ ), with  $\alpha = 1$ , and with  $\alpha = 30$  as defined in Eq. 10. A notable pressure checkerboard pattern appears without pressure stabilization. Pressure stabilization effectively mitigates this issue, although excessive stabilization can lead to small undesired visual artifacts. Bottom: Y-axis is the mean vertex position deviation between the equilibrium states of the same mesh with or without pressure stabilization, X-axis is the max scale of tetrahedrons specified in the meshing tools from Houdini [SideFX 2023]. The deviation linearly converges to near zero as the tetrahedron scale decreases.

For each individual element, the penalty described in Eq. 10 introduces a constant Laplacian penalty to the pressures of its four vertices, represented by the following  $4 \times 4$  constant matrix:

$$S_e = \frac{\alpha V_e}{80\mu} \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}. \quad (11)$$

Assembling all the  $S_e$  yields the penalty matrix  $\mathbf{S}$ , which is then incorporated into  $\mathbf{C}$  in Eq. 4 and Eq. 6. This form offers two significant advantages: the sparsity pattern aligns with  $\mathbf{K}$  and  $\mathbf{G}$ , and it's a constant matrix added solely to the pressure part. These features are leveraged in the design of our multigrid solver, as detailed in the following section.

## 5 MULTIGRID

### ALGORITHM 1 two-level v-cycle Galerkin multigrid

**Require:**  $\mathbf{A}, \mathbf{b}, \mathbf{u}_1$  ▷  $\mathbf{u}_1$  is the initial guess

- 1:  $\mathbf{u}_1 \leftarrow \text{PRESMOOTH}(\mathbf{A}, \mathbf{b}, \mathbf{u}_1)$
- 2:  $\mathbf{r}_1 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{u}_1, \mathbf{r}_2 \leftarrow \mathbf{P}^T \mathbf{r}_1$  ▷ restriction
- 3: Solve  $(\mathbf{P}^T \mathbf{A} \mathbf{P}) \mathbf{e}_2 = \mathbf{r}_2$  ▷ can be nested into a deeper v-cycle
- 4:  $\mathbf{e}_1 \leftarrow \mathbf{P} \mathbf{e}_2, \mathbf{u}_1 \leftarrow \mathbf{u}_1 + \mathbf{e}_1$  ▷ prolongation
- 5:  $\mathbf{u}_1 \leftarrow \text{POSTSMOOTH}(\mathbf{A}, \mathbf{b}, \mathbf{u}_1)$

A classic two-level v-cycle Galerkin multigrid to solve  $\mathbf{A}\mathbf{u} = \mathbf{b}$  is shown in Alg. 1, which consists of three main components: the

smoother (step 1, 5), the interpolation operator  $\mathbf{P}$  (step 2, 4), and the direct solver at coarsest level (step 3). This two-level structure can be nested into a deeper v-cycle by consecutively coarsening  $\mathbf{P}^T \mathbf{A} \mathbf{P}$ . For classic SPD linear system, Xian et al. [2019] propose a skinning-space Galerkin multigrid framework that is easy to construct, efficient, and scalable. We reconfigure the essential components of their method, to accommodate our mixed FEM formulation, with minimal efficiency lost while avoiding locking artifacts. The multilevel multigrid structures and key configuration parameters for all our examples are detailed in Tab. 2 under the columns "grid set-up" and "solver set-up".

## 5.1 Interpolation

We first explain the position interpolation scheme proposed by Xian et al. [2019]. Instead of building a coarse grid, Xian et al. [2019] propose using Linear Blend Skinning (LBS) to interpolate displacement DoF at the finest level. Specifically, a number of random vertices are selected as handles, with each handle  $b$  having a local affine transformation frame  $\mathbf{T}_b \in \mathbb{R}^{3 \times 4}$  to control the displacement on each vertex  $\mathbf{x}_i$ :

$$\mathbf{x}_i = \sum_b \omega_{ib} \mathbf{T}_b \bar{\mathbf{X}}_i, \quad (12)$$

where  $\bar{\mathbf{X}}_i = (\mathbf{X}_i^T, 1)^T \in \mathbb{R}^4$  and  $\mathbf{X}_i$  is the rest position of vertex  $i$ . The standard LBS form in Eq. 12 can be rewritten using the Kronecker product  $\otimes$  to align with the interpolation matrix  $\mathbf{P}$ :

$$\mathbf{x}_i = \sum_b \mathbf{P}_{ib} \mathbf{q}_b, \quad \mathbf{P}_{ib} = \omega_{ib} \mathbf{I}_3 \otimes \bar{\mathbf{X}}_i^T, \quad \mathbf{q}_b = \mathit{vec}(\mathbf{T}_b) \in \mathbb{R}^{12}, \quad (13)$$

where  $\mathit{vec}(\mathbf{T}_b)$  is the row vectorization of  $\mathbf{T}_b$ , and  $\mathbf{P}_{ib} \in \mathbb{R}^{3 \times 12}$  is the sub-block matrix of  $\mathbf{P}$  corresponding to  $\mathbf{x}_i$  and handle  $b$ . To enhance the sparsity of coarse level matrix  $\mathbf{P}^T \mathbf{A} \mathbf{P}$ , Xian et al. [2019] employ a piece-wise constant weight  $\omega_{ib}$ , which ensures each vertex  $i$  only has nonzero weight for one nearest handle  $b_i$  ( $\omega_{ib_i} = 1$ ) and zero weight for all other handles. As a result, the interpolated displacement field  $\mathbf{x}$  is a discontinuous piece-wise affine field, with each handle fully controlling its neighboring region. The discontinuities between handle regions are mainly localized errors, which can be efficiently smoothed out using smoothers. At coarser levels, direct aggregation of neighboring handles is employed as the restriction operation. For more details please refer to the original paper.

We extend the LBS interpolation scheme from [Xian et al. 2019] to incorporate the pressure DoF within mixed P1/P1 elements. Our key insight is that a vertex's pressure  $p_i$  can be viewed as the fourth DoF, similar to the displacement DoF  $\mathbf{x}_i = (x_i^0, x_i^1, x_i^2)$ . If we consider the x-axis component of displacement separately in Eq. 12 with a piece-wise constant weight, we have:

$$\mathbf{x}_i^0 = \mathbf{T}_{b_i}^0 \bar{\mathbf{X}}_i = (\mathbf{g}_{b_i} \quad \mathbf{t}_{b_i}) \begin{pmatrix} \mathbf{X}_i \\ 1 \end{pmatrix} = \mathbf{g}_{b_i} \mathbf{X}_i + \mathbf{t}_{b_i}, \quad (14)$$

Here  $\mathbf{T}_{b_i}^0 \in \mathbb{R}^{1 \times 4}$  is the first row of  $\mathbf{T}_{b_i}$ , which can be further decomposed into  $\mathbf{g}_{b_i} \in \mathbb{R}^{1 \times 3}$  as the slope and  $\mathbf{t}_{b_i} \in \mathbb{R}$  as the offset. From Eq. 14, if we view each component of displacement as a scalar field in the rest configuration, the effect of LBS is a piece-wise linear interpolation of the scalar field. Since the pressure is also a scalar field, we can apply the same treatment as displacement, resulting in

a compact interpolation scheme:

$$\begin{pmatrix} \mathbf{x}_i \\ p_i \end{pmatrix} = \mathbf{T}'_{b_i} \bar{\mathbf{X}}_i = \mathbf{I}_4 \otimes \bar{\mathbf{X}}_i^T \mathit{vec}(\mathbf{T}'_{b_i}), \quad \mathbf{P}_{ib_i} = \mathbf{I}_4 \otimes \bar{\mathbf{X}}_i^T, \quad (15)$$

where  $\mathbf{T}'_{b_i} \in \mathbb{R}^{4 \times 4}$  is the extended affine transformation matrix that includes the slope and offset of pressure, and  $\mathit{vec}(\mathbf{T}'_{b_i}) \in \mathbb{R}^{16}$  is the DoF on handle  $b_i$ , with the first 12 DoF corresponding to displacements and the last 4 DoF corresponding to pressure. The new sub-block matrix  $\mathbf{P}_{ib_i}$  in the interpolation matrix  $\mathbf{P}$  is of size  $4 \times 16$  now. This form of interpolation also suggests an opportunity for optimizing the storage of the system matrix  $\mathbf{A}$ . Notice that  $\mathbf{K}$ ,  $\mathbf{G}$ ,  $\mathbf{G}^T$ , and the stabilized  $\mathbf{C}$  from Section 4 all exhibit the same sparsity pattern, allowing us to reorganize the system matrix  $\mathbf{A}$  from a structure of arrays (SoA) layout to an array of structures (AoS) layout, thereby transforming it into a 4x4-block sparse matrix:

$$\mathbf{A}_{ij} = \begin{pmatrix} \mathbf{K}_{ij} & \mathbf{G}_{ij}^T \\ \mathbf{G}_{ij} & -\mathbf{C}_{ij} \end{pmatrix} \in \mathbb{R}^{4 \times 4}. \quad (16)$$

Applying the interpolation in Eq. 15, the system matrix turns into a 16x16-block matrix, with each block aggregated from the finest level:

$$(\mathbf{P}^T \mathbf{A} \mathbf{P})_{b_i b_j} = (\mathbf{I}_4 \otimes \bar{\mathbf{X}}_i^T)^T \mathbf{A}_{ij} (\mathbf{I}_4 \otimes \bar{\mathbf{X}}_j^T) = \mathbf{A}_{ij} \otimes (\bar{\mathbf{X}}_i \bar{\mathbf{X}}_j^T). \quad (17)$$

The final coarse system  $\mathbf{P}^T \mathbf{A} \mathbf{P}$  is the sum of Eq. 17 for all the  $i - b_i$  and  $j - b_j$  pairs. For coarser levels, we restrict the system DoF by ordinary handle aggregation, i.e. adding coarse DoFs from multiple neighboring handles into one coarser handle, ensuing the 16x16-block structure is preserved. In this way, our interpolation scheme can be seen as a 4-dim extension of the 3-dim version in [Xian et al. 2019], inheriting all its implementation and efficiency benefits. We validate our interpolation scheme in the example in Fig. 11. We construct a two-level multigrid using 100 handles (1600 DoFs) for a cube with 31k vertices (93k DoFs). Our multigrid exhibit superior convergence speed compared to only using the fine-level smoother and reduces the volume gain much faster.

## 5.2 Smoother

Owing to the indefinite nature of the saddle point problem outlined in Eq. 6, traditional smoothing techniques such as Jacobi, Gauss-Seidel, and successive over relaxation exhibit poor convergence rate. Developing an effective smoother has been a central focus in previous works on multigrid methods for mixed FEM [Benzi et al. 2005; Oosterlee and Gaspar 2008]. Among these methods, the Vanka-type smoother [Vanka 1986], widely used as a domain decomposition smoother, serves as the basis for our approach.

The Vanka-type smoother partitions the computational domain  $\Omega$  into smaller, overlapping subdomains  $\{\Omega_m\}$ , and directly solves the small indefinite KKT system for each subdomain  $\Omega_m$  successively:

$$(\mathbf{R}_m \mathbf{A} \mathbf{R}_m^T) \mathbf{u}_m = \mathbf{R}_m (\mathbf{b} - \mathbf{A} \mathbf{u}), \quad \mathbf{u} \leftarrow \mathbf{u} + \omega \mathbf{R}_m^T \mathbf{u}_m. \quad (18)$$



Fig. 6. Vanka Subdomains. The pressure DoFs are illustrated using squares and the position DoFs are illustrated using dots. The cell-oriented Vanka (left) uses both position and pressure DoFs in a cell as a subdomain; The patch-based Vanka (middle) uses a single pressure DoF and all connected position DoFs as a subdomain. Our vertex-based Vanka (right) uses only the position and pressure DoFs associated with a single vertex as a subdomain.

Here  $\mathbf{R}_m$  is the DoF restriction operator for subdomain  $\Omega_m$ ,  $\mathbf{R}_m \mathbf{A} \mathbf{R}_m^T$  is the restricted KKT matrix, and  $\omega$  is the relaxation parameter. Previous engineering papers [Emami 2013; John and Tobiska 2000; Larin and Reusken 2008] identify two type of Vanka smoothers: the cell-oriented Vanka ('cVanka'), which defines  $\Omega_m$  to include all the pressure DoFs and displacement DoFs within a cell, and the patch-based Vanka ('pVanka'), which defines  $\Omega_m$  to include one pressure DoF and all the connected displacement DoFs, as demonstrated in Fig. 6. While these smoothers offer good convergence speed per iteration, they suffer from poor parallelism. To parallelize 'cVanka', element colorization is required, similar to XPBD [Macklin and Muller 2021], often requires many colors depending on the maximum degree of the element adjacency graph of the mesh. In a moderate case, such as the octopus with 40k vertices and 112k elements shown in Fig. 7, a total of 216 colors are needed. The situation is similar for the 'pVanka' smoother, as the second-order vertex neighbors must be colored differently.

To enhance parallelism on the GPU, we propose our vertex Vanka smoother ('vVanka'), where each subdomain encompasses only one vertex's displacement and pressure DoF, as shown in Fig. 6. This concept aligns with the approach proposed by [Wu et al. 2022] which utilizes non-overlapping subdomains in the Schwarz preconditioner for SPD systems. For parallelism, we color each vertex using the method described by Fratarcangeli et al. [2016], following the same principle as recent works by Chen et al. [2024b] and Chen et al. [2024a]. Moreover, if we use the AoS arrangement of the matrix from Section 5.1, the vVanka smoother can be equivalently viewed as a classic block Gauss-Seidel smoother, except the local 4x4 system is now an indefinite KKT system.

We compare the performance of our vertex Vanka smoother with other popular smoothers. Here, we briefly explain the basic concepts of these smoothers. For implementation details, please refer to the supplemental material. The Kaczmarz smoother [Stefan 1993] transforms the indefinite system  $\mathbf{A}\mathbf{u} = \mathbf{b}$  into a SPD system by solving the normal equation  $\mathbf{A}^2\mathbf{y} = \mathbf{b}$ , with  $\mathbf{u} = \mathbf{A}\mathbf{y}$ , where parallel Jacobi iterations can be applied rather than parallel Gauss-Seidel constrained by the sparsity pattern of  $\mathbf{A}^2$ . The inexact Uzawa smoother [Elman and Golub 1994] takes advantage of the fact that  $\mathbf{K}$  and the stabilized  $\mathbf{C}$  in Eq. 6 are SPD. Therefore,  $\mathbf{x}$  and  $\mathbf{p}$  can be smoothed independently using traditional smoothers. We identify two variations of inexact Uzawa smoothers: the 'jUzawa' smoother that uses Jacobi iterations and the 'gUzawa' smoother that uses Gauss-Seidel iterations. We also compare an additive variation of the cell-oriented Vanka

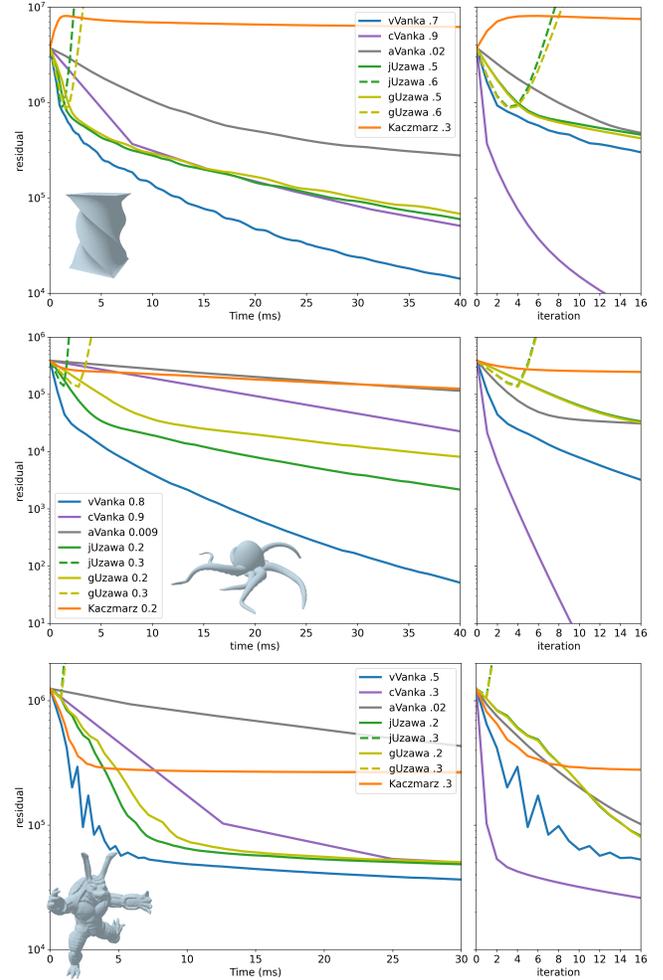


Fig. 7. Smoother Comparison. vVanka: Our vertex Vanka smoother. cVanka: Cell-oriented Vanka. aVanka: Additive cell-oriented Vanka. jUzawa: Inexact Uzawa using Jacobi for position update. gUzawa: Inexact Uzawa using Gauss-Seidel for positions update. Kaczmarz: Kaczmarz smoother using Jacobi. The number in the legend refers to the relaxation parameter of each smoother. Each figure demonstrate the convergence curve at a random frame from three different scenes: the cube (31k vertices and 156k elements,  $\nu = 0.4999$ ) from Fig. 11, the octopus (40k vertices and 112k elements,  $\nu = 0.499$ ) from Fig. 9, the armadillo (77k vertices and 340k elements,  $\nu = 0.49$ ) from Fig. 13 with real-time interaction.

smoother ('aVanka'), which updates all the subdomains simultaneously in a Jacobi-like manner, adding the contribution from multiple subdomains to each vertex. We fine-tune the relaxation parameters for all smoothers independently to achieve their best performance, the results are shown in Fig. 7.

Among all smoothers, our vertex Vanka smoother (shown as the blue curve) demonstrates the fastest residual reduction speed with respect to time. Although the 'cVanka' smoother (shown as the purple curve) shows the best convergence per iteration, its poor parallelism flattens its overall convergence speed in time. Switching to the additive 'avanka' smoother (shown as the gray curve)

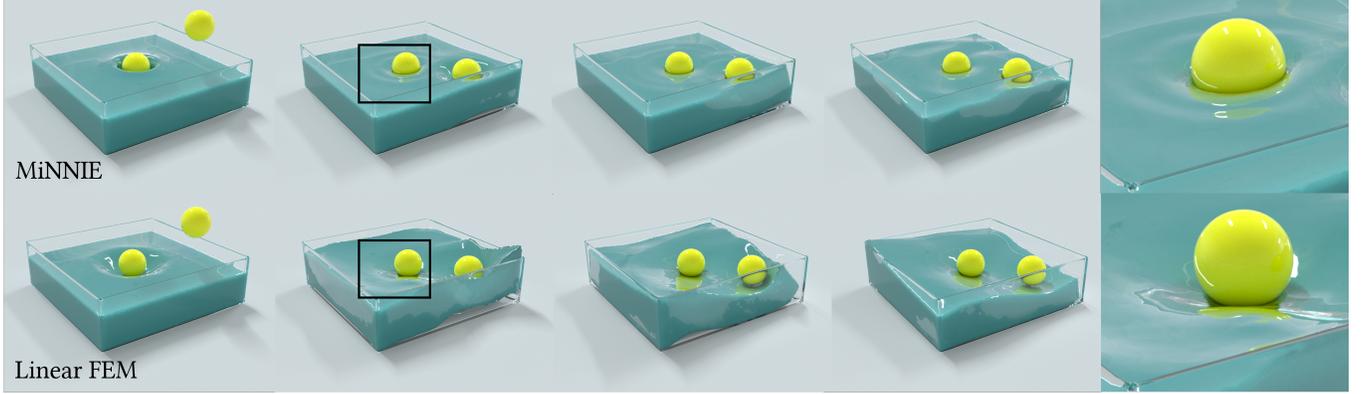


Fig. 8. Wave. Two kinematically-controlled spheres are dropped onto the hydrogel with Young’s module  $E = 3e5$  and Poisson’s ratio  $\nu = 0.4999$ . The top row is simulated using MiNNIE, the second row is simulated using linear FEM (multigrid solver by Xian et al. [2019] with the stable Neo-Hookean material [Smith et al. 2018]). We show the 35th, 104th, 160th, 275th frames of both simulations. MiNNIE accurately reproduces the dynamic wave-like response of extremely soft, incompressible material induced by large impacts, while linear FEM generates large unnatural deformations suffering from volumetric locking.

reduces per-iteration cost but scarifies too much convergence speed due to Jacobi iterations. The convergence speed of inexact Uzawa smoothers, ‘jUzawa’ and ‘gUzawa’ (shown as the green and yellow curves), are hindered by their asynchronous updates of  $\mathbf{x}$  and  $\mathbf{p}$ , making them more sensitive to relaxation parameters than our vertex Vanka smoother. The Kaczmarz smoother (shown as the orange curve), which squares the system matrix and its condition number, shows the poorest performance.

At the coarser levels, we can apply the same vertex Vanka strategy, with each subdomain containing all the 16 DoF on each handle, i.e. the  $vec(\mathbf{T}'_b)$  defined in Eq. 15. We adopt the MINRES [Paige and Saunders 1975] iteration to solve the  $16 \times 16$  indefinite matrix for each handle. To further accelerate the convergence of MINRES, we apply a  $4 \times 4$ -block diagonal preconditioner to the  $16 \times 16$  system, utilizing its  $4 \times 4$ -block structure as shown in Eq. 17. To ensure the positive definiteness of the preconditioner, we need to negate the last  $4 \times 4$  block since it corresponds to the  $-C$  block for pressures. Our block preconditioner can reduce the condition number of the  $16 \times 16$  system from  $1e4$  to  $1e2$ , while a direct diagonal preconditioner can only halve the condition number.

### 5.3 Dense Solver

At the coarsest level, the system matrix is usually small (several thousands dimension) and can be solved by a direct solver. We propose to use Schur complement to further accelerate this process:

$$\begin{aligned} (\mathbf{K}_l + \mathbf{G}_l^T \mathbf{C}_l^{-1} \mathbf{G}_l) \delta \mathbf{x}_l &= \mathbf{f}_l + \mathbf{G}_l^T \mathbf{C}_l^{-1} \mathbf{g}_l, \\ \delta \mathbf{p}_l &= \mathbf{C}_l^{-1} (\mathbf{G}_l \delta \mathbf{x}_l - \mathbf{g}_l). \end{aligned} \quad (19)$$

In this formulation,  $\mathbf{C}_l^{-1}$  can be precomputed and cached at the beginning of the simulation, since the stabilized  $\mathbf{C}$  at the finest level (Section 4) and the interpolation matrix (Section 5.1) are all constant matrices. This allows us to solve a smaller system  $\mathbf{K}_l + \mathbf{G}_l^T \mathbf{C}_l^{-1} \mathbf{G}_l$  at run time, which is also SPD, leveraging the well-established parallel Cholesky (LLT) factorization. By taking this strategy, our dense solver typically takes less than 2 ms per solve, as exemplified in Fig. 9,

which is significantly faster than directly solving the original indefinite system. Additionally, in the extreme case where  $\nu = 0.5$  and  $\mathbf{C}_l$  is not invertible, we add a small regularization term ( $1e-6$ ) to the diagonal of  $\mathbf{C}_l$ . This is equivalent to using an approximate Hessian matrix, having no effect on the final solution since the right-hand side remains the same.

## 6 CONSTITUTIVE MODELS

MiNNIE accommodates various constitutive models in graphics. Their split forms that conform to Eq. 1 are:

- Neo-Hookean [Bonet and Wood 2008]:  
 $\Psi_d = \frac{\mu}{2} (I_c - 3) - \mu \log J$ ,  $\mu = \frac{E}{2(1+\nu)}$ ,  
 $\Psi_v = \frac{\kappa}{2} (\log J)^2$ ,  $\Phi(J) = \log J$ ,  $\kappa = \frac{E\nu}{(1+\nu)(1-2\nu)}$ .
- stable Neo-Hookean [Smith et al. 2018]:  
 $\Psi_d = \frac{\mu}{2} (I_c - 3) - \mu (J - 1)$ ,  $\mu = \frac{E}{2(1+\nu)}$ ,  
 $\Psi_v = \frac{\kappa}{2} (J - 1)^2$ ,  $\Phi(J) = J - 1$ ,  $\kappa = \frac{E}{2(1+\nu)(1-2\nu)}$ .
- Mooney-Rivlin [Barbič et al. 2012]:  
 $\Psi_d = \mu_{10} (J^{-\frac{2}{3}} I_c - 3) + \frac{\mu_{01}}{2} (J^{-\frac{4}{3}} (I_c^2 - II_c) - 6)$ ,  
 $\mu_{01} + \mu_{10} = \frac{E}{4(1+\nu)}$ ,  
 $\Psi_v = \frac{\kappa}{2} (J - 1)^2$ ,  $\Phi(J) = J - 1$ ,  $\kappa = \frac{E}{3(1-2\nu)}$ .
- Corotation [McAdams et al. 2011]:  
 $\Psi_d = \mu \|\mathbf{F} - \mathbf{R}\|^2$ ,  $\mu = \frac{E}{2(1+\nu)}$ ,  
 $\Psi_v = \frac{\kappa}{2} \text{tr}^2(\mathbf{S} - \mathbf{I})$ ,  $\Phi(\mathbf{F}) = \text{tr}(\mathbf{S} - \mathbf{I})$ ,  $\kappa = \frac{E\nu}{(1+\nu)(1-2\nu)}$ .
- StVK [Barbič et al. 2012]:  
 $\Psi_d = \mu \text{tr}(\mathbf{E})^2$ ,  $\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I})$ ,  $\mu = \frac{E}{2(1+\nu)}$ ,  
 $\Psi_v = \frac{\kappa}{2} \text{tr}^2(\mathbf{E})$ ,  $\Phi(\mathbf{F}) = \text{tr}(\mathbf{E})$ ,  $\kappa = \frac{E\nu}{(1+\nu)(1-2\nu)}$ .

In these expressions,  $I_c = \text{tr}(\mathbf{F}\mathbf{F}^T)$ ,  $II_c = \text{tr}(\mathbf{F}\mathbf{F}^T \mathbf{F}\mathbf{F}^T)$ ,  $J = \det \mathbf{F}$ ,  $\mathbf{F} = \mathbf{R}\mathbf{S}$  is the polar decomposition,  $E$  is the Young’s modulus,  $\nu$  is the Poisson’s ratio. Although these splitting forms cause  $\kappa$  and  $\Psi_v$  to approach infinity as  $\nu \rightarrow 0.5$ , their ability to preserve volume depends on the specific definition of  $\Phi$ .  $\Phi$  in Corotation and StVK can be viewed as linearized versions of the volume constraint near

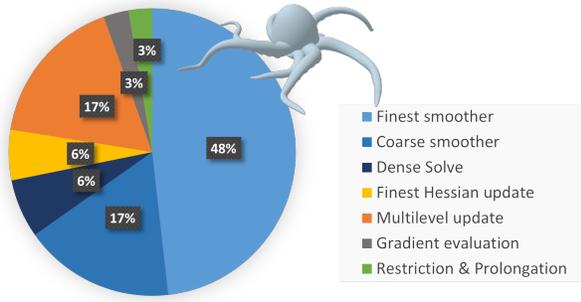


Fig. 9. Multigrid time breakdown. An octopus with 40k vertices, 112k elements, and a Poisson's ratio of  $\nu = 0.499$  requires 11.2 ms per frame. The time consuming of each component is shown in the figure. Most of the time is spent on smoothers, followed by the update of multi-level matrices.

rest configurations [Patterson et al. 2012], but not as a true volume constraint  $J = \det \mathbf{F} = 1$ . Thus, although we can enforce  $\Phi = 0$  by setting a large  $\nu$ , the volume is not conserved as shown in Fig. 4. Among these constitutive models, we favor the stable Neo-Hookean model in our experiments for its stability and robustness to element inversions. Moreover, the  $\kappa$  in stable Neo-Hookean is  $\frac{E}{2} \neq 0$  when  $\nu = 0$ , so we can safely deal with the  $\frac{1}{\kappa}$  term in Eq. 3 for all Poisson's ratios.

To ensure the descent direction of Newton's iteration and to robustly handle element inversions that occur during large deformations, we also need to perform SPD projection of  $\mathbf{K} = \partial^2 L / \partial \mathbf{x}^2$  in Eq. 6 at runtime. According to the definition in Eq. 4, the positional Hessian  $\mathbf{K}$  of  $L(\mathbf{x}, \mathbf{p})$  is only related to the constant mass matrix  $\mathbf{M}/h^2$  and  $U_d(\mathbf{x}) + \mathbf{p}^T \phi$ , the latter of which can be expended using Eq. 9:

$$\begin{aligned} U_d(\mathbf{x}) + \mathbf{p}^T \phi &= \sum_e \Psi_d(\mathbf{F}_e) V_e + \sum_e \bar{p}_e \Phi(\mathbf{F}_e) V_e \\ &= \sum_e (\Psi_d(\mathbf{F}_e) + \bar{p}_e \Phi(\mathbf{F}_e)) V_e. \end{aligned} \quad (20)$$

Comparing Eq. 20 with the standard linear FEM, the only difference is to change the summation of  $\Psi_e(\mathbf{F}_e)$  to  $\bar{p}_e \Phi(\mathbf{F}_e)$ . In other words, we can assume the mixed form  $\Psi_d(\mathbf{F}) + \bar{p} \Phi(\mathbf{F})$  is the "constitutive model" of  $\mathbf{x}$ . For instance, the mixed form of stable Neo-Hookean material is  $\Psi_d(\mathbf{F}) + \bar{p} \Phi(\mathbf{F}) = \frac{H}{2} (I_c - 3) + (\bar{p} - \mu)(J - 1)$ , whose explicit eigensystem is given in [Smith et al. 2018]. The  $\bar{p}$  term can be interpreted as the geometric stiffness of the constraints as explained by Tournier et al. [2015]. For all the isotropic distortion energies we used in this paper, we adopt the positive definiteness fix technique described in [Smith et al. 2019] to ensure the projected positional Hessian  $\mathbf{K}$  is SPD.

## 7 RESULTS

### 7.1 Implementation

We implement MiNNIE on GPU using the CUDA [NVIDIA 2024]. All the examples are tested on an AMD Ryzen 9 7950X 16-Core CPU and a Nvidia RTX 4090 GPU. We scale the pressure DoF in Eq. 6 as in [Zarifi and Batty 2017] to mitigate the numerical scale difference between position and pressure DoF. The positional constraints and

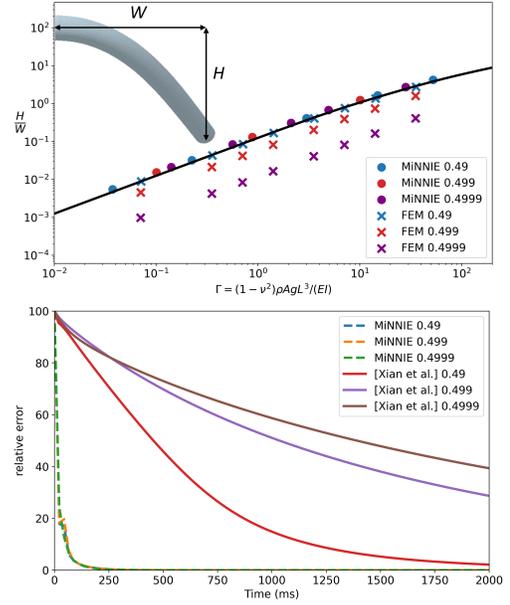


Fig. 10. Comparisons with theoretical solutions [Romero et al. 2021] on the cantilever test. MiNNIE produces physical correct results for all Poisson's ratios, while classic linear FEM suffers from locking under large Poisson's ratio. Additionally, MiNNIE converges much faster than linear FEM with [Xian et al. 2019] under all Poisson's ratios.

collisions are handled by adding quadratic penalties only to position DoFs [Xian et al. 2019]. For self-collision detection, we apply the spatial hashing method from [Teschner et al. 2003] for its simplicity. Due to the use of P1/P1 elements, all the previous self-collision treatments for tetrahedral meshes can be applied to our method. We follow the semi-implicit time integration without line-search [Baraff and Witkin 1998] to simulate all dynamics cases. We also use the quasi-Newton solution for pressure stabilization to handle all quasi-static cases as described in Sec. 4. The parameter table and time statistics for all our examples are listed in Tab. 2. A time breakdown in a single multigrid solve of the octopus example is shown in Fig. 9. We manually adjust the number of iterations and other parameters for each example to optimize performance. For most examples, we apply 6 smoother iterations and set the relaxation parameter  $\omega$  to 0.4. However, for the cube-random (Fig. 12) and wave (Fig. 8) examples, we use more conservative parameters due to their extreme initialization or material parameters. We could further reduce the number of smoother iterations or increase  $\omega$  for better performance. However, as discussed in the limitations in Section 8, too aggressive settings may lead to instabilities.

### 7.2 Validations

*Physical validation & Comparison with FEM.* To validate the physical correctness of MiNNIE, we set up the cantilever protocol test described in [Romero et al. 2021]: a slender horizontal cylinder (radius  $r = 0.3$ , length  $L = 6$ ) with one end fixed deforms under gravity, as shown in Fig. 10. If correctly solved, the dimensionless gravito-bending parameter  $\Gamma = (1 - \nu^2) \frac{\rho A g L^3}{EI} = (1 - \nu^2) \frac{4 \rho g L^3}{E r^2}$  and the

Table 2. Parameter Table. "V(n)" stands for n times of our vertex Vanka smoother, "D" stands for direct solver.  $\omega$  is the relaxation parameter for smoothers.  $E$  is the Young's modulus,  $\nu$  is the Poisson's ratio.  $\alpha$  is the pressure stabilization constant.  $h$  is the timestep size, "/" stands for quasi-static simulation. FPS stands for frame per second. For all the examples, we set the density  $\rho = 1e3$ , and  $g = 9.8$  if gravity is applied.

Examples	#verts	#elems	grid set-up	solver set-up	$\omega$	$E$	$\nu$	$\alpha$	$h$	FPS
bunny-torus (Fig. 1)	47k	223k	100/200/all	D/V(3)/V(6)	0.4	2e6	0.49	1	1e-2	60
bunny-dance (Fig. 2)	47k	223k	100/200/all	D/V(3)/V(6)	0.4	2e6	0.49	1	3e-2	60
cylinder (Fig. 4a)	13k	67k	100/all	D/V(6)	0.4	3e5	0.49	1	/	160
tube (Fig. 4b)	15k	79k	100/all	D/V(6)	0.5	3e6	0.49	1	/	158
cube-gravity (Fig. 5)	14k	70k	100/all	D/V(6)	0.4	1e5	0.4999	1	/	160
cube-twist (Fig. 11)	31k	156k	100/all	D/V(6)	0.4	1e7	0.4999	1	/	110
cube-random (Fig. 12)	4k	25k	100/all	D/V(10)	0.2	1e7	0.4999	20	/	155
octopus (Fig. 9)	40k	112k	50/200/all	D/V(6)/V(6)	0.4	1e6	0.499	1	1e-2	88
Cantilever (Fig. 10)	8k	34k	100/all	D/V(3)	0.7	1e7	0.4999	1	/	229
bulging (Fig. 3)	10k	57k	100/all	D/V(3)	0.5	1e7	0~0.5	1	/	210
wave (Fig. 8)	30k	159k	100/300/all	D/V(8)/V(10)	0.7	3e5	0.4999	1	5e-3	72
letters (Fig. 14)	105k	371k	80/400/all	D/V(3)/V(6)	0.5	1e6	0.49999	1	1e-2	38
armadillo (Fig. 13)	20k	88k	100/all	D/V(6)	0.4	5e6	0.49	1	1e-2	142
armadillo (Fig. 13)	35k	187k	100/all	D/V(6)	0.4	5e6	0.49	1	1e-2	107
armadillo (Fig. 13)	77k	340k	100/300/all	D/V(6)/V(6)	0.4	5e6	0.49	1	1e-2	53
armadillo (Fig. 13)	117k	598k	100/200/400/all	D/V(6)/V(6)/V(6)	0.4	5e6	0.49	1	1e-2	35
armadillo (Fig. 13)	182k	1017k	100/400/800/1600/all	D/V(6)/V(6)/V(6)/V(6)	0.4	5e6	0.49	1	1e-2	21

height-width ratio of the equilibrium state  $\frac{H}{W}$  follow a parameter-free curve, called the "master curve" [Romero et al. 2021], plotted as the black curve in Fig. 10. We sample different  $\Gamma$  by changing the density  $\rho$ , the Young's module  $E$ , the gravity  $g$ , and check the final height-width ratio  $\frac{H}{W}$  using MiNNIE, compared with the classic linear FEM using the same stable Neo-Hookean model [Smith et al. 2018]. As shown in Fig. 10, MiNNIE fits the master curve for all Poisson's ratios, while classic linear FEM only satisfies the master curve under  $\nu = 0.49$ , and gradually deviates from it as Poisson's ratio increases. This deviation highlights the locking issue of classic FEM under large Poisson's ratio, i.e. the height-width ratio  $\frac{H}{W}$  is much smaller than expected, indicating a stiffer-than-expected material behavior. We also compare the convergence speed of MiNNIE and linear FEM using the fast GPU solver [Xian et al. 2019] by tracking the relative error of the height-width ratio  $\frac{H}{W}$  as shown at the bottom of Fig. 10. MiNNIE (shown as the dash curves) converges to the final solution in a few iterations regardless of Poisson's ratios, while linear FEM (shown as the solid curves) requires many more iterations and slows down as  $\nu$  increases.

When direct compressing or stretching is applied, significant locking can be observed even under  $\nu = 0.49$ , as shown in Fig. 4. In the top row, Fig. 4a shows a stable Neo-Hookean elastic cylinder with  $E = 3e5$ ,  $\nu = 0.49$  being stretched by a metal block (weight  $1.2e5$  kg) attached at the bottom. The results show that linear FEM exhibits locking artifact, resulting in less stretch of the elastic cylinder, while MiNNIE produces results almost identical to those of second-order FEM. At the bottom, Fig. 4b shows a tube with  $E = 3e6$ ,  $\nu = 0.49$  being compressed to 70% of its height by fixing the top and bottom surfaces. MiNNIE behaves similarly to the second-order FEM, including the sharp crease on the inner surface, while the result of linear FEM using [Xian et al. 2019] is much flatter. MiNNIE also

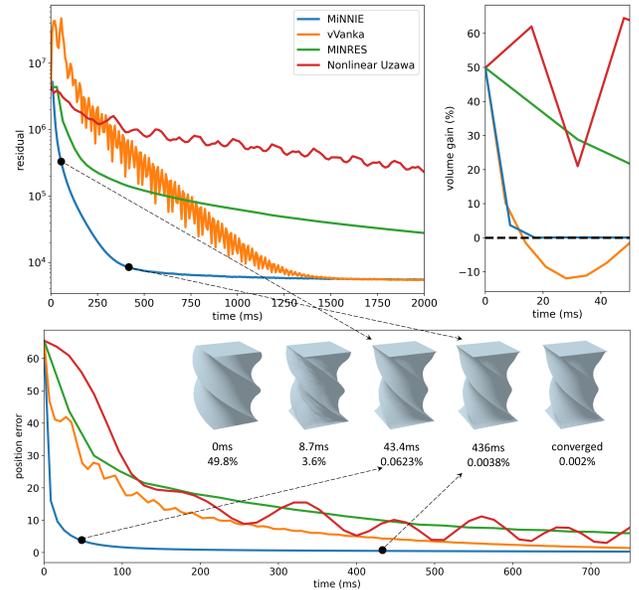


Fig. 11. Convergence of different solvers. A cube with 31k vertices, 156k elements, and Poisson's ratio  $\nu = 0.4999$  recovers to an equilibrium state after being twisted and stretched. The figure shows the convergence of residuals (top-left), volume gains (top-right), and position errors (bottom). MiNNIE demonstrates the fastest convergence, and can restore the volume gain to less than 1% in just two iterations.

supports the Corotation model without locking, though the volume is less preserved as explained in Section 6.

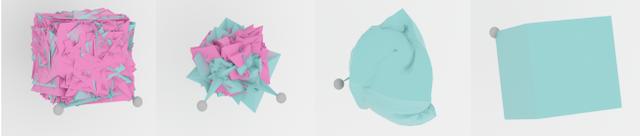


Fig. 12. Inversion Test. We randomize the vertices of a cube with Poisson’s ratio  $\nu = 0.4999$ . The inverted elements are colored in pink. MiNNIE can robustly recover the cube to its rest pose from states containing degenerated or inverted elements.

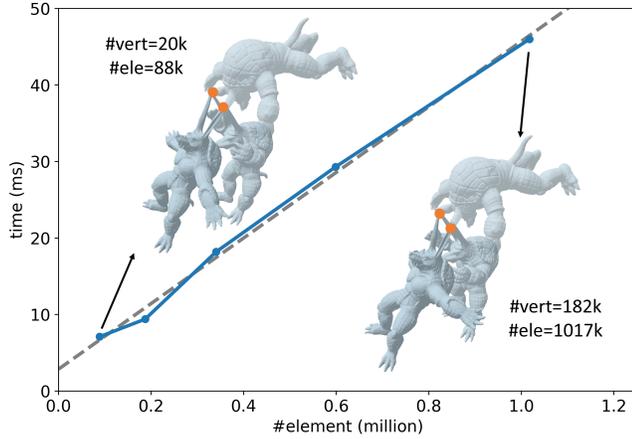


Fig. 13. MiNNIE’s time cost scales linearly with the increasing number of elements. Snapshots at the 0th, 200th, and 300th frames for both the coarsest mesh example (20k vertices and 88k elements, shown in the top left) and the finest mesh example (182k vertices and 1017k elements, shown in the bottom right) demonstrate similar trajectories.

**Convergence.** We set up a quasi-static example of a twisting cube to compare the convergence of different solvers on the same mixed FEM problem. Initially, a cube consisting of 31k vertices and 156k elements, with Poisson’s ratio  $\nu = 0.4999$  is linearly rotated by 180 degrees and vertically stretched by 1.5 times. Then we release the cube while fixing the top and the bottom vertices, and compute the residual of Eq. 6, the volume gain, and the position error to the converged configuration, as shown in Fig. 11. We compare four solvers: MiNNIE, vertex Vanka smoother, MINRES [Paige and Saunders 1975], the nonlinear Uzawa method used in [Frâncu et al. 2021]. We also tune the relaxation parameters for all solvers independently to reach their best performance. As shown in Fig. 11, MiNNIE has the best performance on all fronts. The gap between MiNNIE and pure vertex Vanka smoother shows the effectiveness of our multigrid scheme. The flattened tail of MiNNIE’s residual curve is due to the quasi-Newton approximation explained in Section 4, which does not affect the fast convergence of positions as shown in the error curve at the bottom. The nonlinear Uzawa method [Frâncu et al. 2021] has the worst convergence in this case. Unlike the other methods, the nonlinear Uzawa method updates the positions and pressures unsynchronized, so it needs more iterations to converge compared with the other methods that update positions and pressures simultaneously.

**Inversion test.** To test the robustness of MiNNIE, we fix the two corners of a cube (4k vertices, 25k elements, Poisson’s ratio  $\nu = 0.4999$ ), and start the simulation from a randomized pose. The initial state’s signed volume is 44.89% of the rest volume, and has a lot of inverted elements. MiNNIE can recover the cube to its rest configuration in real-time (155 fps). To handle this ill-conditioned problem, the pressure stabilization constant  $\alpha$  should be set higher and the relaxation parameter  $\omega$  should be set lower than other regular cases. We find  $\alpha = 20$ ,  $\omega = 0.2$  works well in this case. We present this extreme example as a robustness test, demonstrating that MiNNIE can handle infeasible initial states with reasonable boundary conditions. However, MiNNIE does not support boundary conditions that conflict with the incompressibility constraints, for example, compressing an incompressible material in all directions.

**Scalability.** An armadillo with  $E = 3e6$ ,  $\nu = 0.49$ , and both ear tips fixed is swinging under gravity. We gradually refine the model from the lowest level with 20k vertices and 88k elements to the highest level with 182k vertices and 1017k elements over five levels. For each level, we record the average computation time per frame using our multigrid solver. As shown by the solid curve in Fig. 13, with the system adequately solved, indicated by the roughly similar motion at the highest and lowest levels, the time cost of our multigrid solver scales linearly with the number of elements.

### 7.3 Examples

**Bunny.** MiNNIE avoids volumetric locking better than linear FEM under  $\nu = 0.49$  in dynamic case, as shown in Fig. 1 and Fig. 2. In the torus-squeezing example of Fig. 1, MiNNIE generates detailed fat folds more vividly than linear FEM. In the dancing example of Fig. 2, MiNNIE generates more dynamic and hilarious movements, while the result of linear FEM is much stiffer. In terms of efficiency, MiNNIE is only 20% slower than the GPU multigrid solver of linear FEM by Xian et al. [2019] in both examples.

**Bulging.** MiNNIE can demonstrate the bulging effect of bending elbows under different Poisson’s ratios  $\nu$  from 0 to 0.5, as shown in Fig. 3. As the Poisson’s ratio increases, more tissues bulge out at the joint due to self-collisions. MiNNIE preserves the total volume to a loss at only  $< 0.01\%$  when setting  $\nu = 0.5$ .

**Wave.** MiNNIE can simulate extremely soft yet volume-conserved elastic bodies such as hydrogel to generate fluid-like waves, as shown in Fig. 8. The hydrogel in the tank has 30k vertices and 159k elements, a Young’s module of  $E = 3e5$ , and a Poisson’s ratio of  $\nu = 0.4999$ . It takes only 14 ms per frame to simulate large impact waves and detailed secondary waves. We also compare our results with linear FEM which suffers from volumetric locking artifacts.

**Letters.** We show the excellent scalability of MiNNIE by dropping elastic letters to a box as shown in Fig. 14. The scene has 105k vertices and 371k elements, up to 30k self-collision pairs. It takes less than 20 ms per frame to update and solve the system and averages 26 ms in total, including collision detection and re-colorization for our vertex Vanka smoother. We set the Poisson’s ratio of the letters to  $\nu = 0.49999$  and find the volume fluctuation during the entire simulation is bounded under 0.05%. Note that MiNNIE does not



Fig. 14. Six sets of near-incompressible elastic "SIGGRAPH" letters with Poisson's ratio  $\nu = 0.49999$ , consisting of 105k vertices and 371k elements, are dropped into a box. These letters generate vivid deformations with volume fluctuation below 0.05%. Our method takes less than 20 ms per frame to solve the elasto-dynamics of the letters and an average of 26 ms to simulate an entire frame, including self-collision handling.

suffer from any volumetric locking issues and generates dynamic, squasy animations of the letters.

## 8 CONCLUSIONS, LIMITATIONS & FUTURE WORK

MiNNE simulates nonlinear near-incompressible elastics in real time. The point of departure of MiNNE is a mixed FEM framework using P1/P1 elements [Fráncu et al. 2021] that avoids locking issues. However, direct solvers and vanilla Krylov methods do not meet the real-time requirement for our mixed FEM problem. We observe that simple pressure stabilization can mitigate the checkerboard artifacts that used to hinder the convergence of multigrid solvers. We also use a quasi-Newton method to minimize the extra nodal displacement introduced by pressure stabilization. Once the checkerboard artifacts are gone, we design a specially tailored multigrid solver for mixed FEM and implement it on GPU. Our multigrid scheme consists of a modified skinning-space interpolation scheme [Xian et al. 2019], an efficient vertex Vanka smoother, and a prefactorized dense-level linear solver. We equip MiNNE with various elastic material models to simulate a full range of Poisson's ratios up to 0.5. As a result, it simulates large near-incompressible scenarios containing hundreds of thousands of elements in real time, producing vivid visual effects, as shown in Fig. 1, Fig. 2, and Fig. 14.

The major limitation of MiNNE is the introduction of extra tunable parameters such as the stabilization constant  $\alpha$  and the relaxation parameter  $\omega$  for smoothers. The choice of these hyperparameters can affect our system's convergence. Though we provide a quasi-Newton treatment and show the robustness with many difficult cases in this paper, inappropriate choice of hyperparameters may still crash the simulation. On the other hand, the MINRES method is guaranteed to converge for our cases, even without the pressure stabilization [Paige and Saunders 1975]. But MINRES converges much slower than our multigrid solver due to poor conditioning. An interesting future direction is to design a better preconditioner for MINRES, similar to the multilevel Schwarz preconditioner for the conjugate gradient method proposed by [Wu et al. 2022]. We are also interested in extending our mixed multigrid solver to other simulation tasks where the stiffness of some energy terms can go extremely high, for instance, the barrier energy in IPC [Li et al. 2020], the spring energy in inextensible cloth [Goldenthal et al. 2007], materials with dramatic heterogeneity [Chen et al. 2019], etc..

We look forward to building high-performance solvers for these difficult problems as well.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. Authors from Peking University acknowledge the funding support by National Key R&D Program of China 2022ZD0160802. We would like to acknowledge Libo Huang from miHoYo Co. Ltd., Ye Fan from Vital Mechanics Inc., Shuliang Lu and Xinlei Wang from Shenzhen Zenus Technology Co. Ltd. for their valuable insights and discussions. The Houdini Education license is credited for the video generations.

## REFERENCES

- David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54. <https://doi.org/10.1145/280814.280821>
- Jernej Barbič, Fun Shing Sin, and Daniel Schroeder. 2012. Vega FEM Library. <http://www.jernejbarbic.com/vega>.
- Adam W. Bargteil and Elaine Cohen. 2014. Animation of Deformable Bodies with Quadratic Bézier Finite Elements. *ACM Trans. Graph.* 33, 3, Article 27 (jun 2014), 10 pages. <https://doi.org/10.1145/2567943>
- Michele Benzi, Gene H. Golub, and Jörg Liesen. 2005. Numerical solution of saddle point problems. *Acta Numerica* 14 (2005), 1–137. <https://doi.org/10.1017/S0962492904000212>
- Daniele Boffi, Franco Brezzi, Leszek F. Demkowicz, Ricardo G. Durán, Richard S. Falk, Michel Fortin, and Lucia Gastaldi. 2008. *Mixed Finite Elements, Compatibility Conditions, and Applications*. <https://api.semanticscholar.org/CorpusID:124471984>
- Javier Bonet and Richard D. Wood. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis* (2 ed.). Cambridge University Press.
- Franco Brezzi and Michel Fortin. 1991. *Mixed and hybrid finite element methods*. Springer-Verlag, Berlin, Heidelberg.
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. CRC press.
- Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. 2024b. Vertex Block Descent. *ACM Trans. Graph.* 43, 4, Article 116 (jul 2024), 16 pages. <https://doi.org/10.1145/3658179>
- Jiong Chen, Max Budninskiy, Houman Owahdi, Hujun Bao, Jin Huang, and Mathieu Desbrun. 2019. Material-adapted refinable basis functions for elasticity simulation. *ACM Trans. Graph.* 38, 6, Article 161 (nov 2019), 15 pages. <https://doi.org/10.1145/3355089.3356567>
- Yizhou Chen, Yushan Han, Jingyu Chen, Zhan Zhang, Alex McAdams, and Joseph Teran. 2024a. Position-Based Nonlinear Gauss-Seidel for Quasistatic Hyperelasticity. *ACM Trans. Graph.* 43, 4, Article 115 (jul 2024), 15 pages. <https://doi.org/10.1145/3658154>
- R. Dziol, J. Bender, and D. Bayer. 2011. Robust Real-Time Deformation of Incompressible Surface Meshes. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Vancouver, British Columbia, Canada) (SCA '11)*. Association for Computing Machinery, New York, NY, USA, 237–246. <https://doi.org/10.1145/2019406.2019438>
- Clark R. Dohrmann and Pavel B. Bochev. 2004. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *International Journal*

- for Numerical Methods in Fluids 46, 2 (2004), 183–201. <https://doi.org/10.1002/flid.752> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/flid.752>
- Howard Elman, David Silvester, and Andy Wathen. 2014. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780199678792.001.0001>
- Howard C. Elman and Gene H. Golub. 1994. Inexact and Preconditioned Uzawa Algorithms for Saddle Point Problems. *SIAM J. Numer. Anal.* 31, 6 (1994), 1645–1661. <https://doi.org/10.1137/0731085> arXiv:<https://doi.org/10.1137/0731085>
- M Emami. 2013. Efficient Multigrid Solvers for the Stokes Equations using Finite Elements. *Lehrstuhl für Informatik 10, Systemsimulation* (2013).
- Elliot English and Robert Bridson. 2008. Animating developable surfaces using nonconforming elements. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–5. <https://doi.org/10.1145/1360612.1360665>
- Mihai Frăncu, Arni Asgeirsson, Kenny Erleben, and Mads J. L. Rønnow. 2021. Locking-Proof Tetrahedra. *ACM Trans. Graph.* 40, 2, Article 12 (apr 2021), 17 pages. <https://doi.org/10.1145/3444949>
- Marco Fratracangeli, Valentina Tivaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-Seidel Method for Stable Soft Body Dynamics. *ACM Trans. Graph.* 35, 6, Article 214 (dec 2016), 9 pages. <https://doi.org/10.1145/2980179.2982437>
- Joachim Georgii and Rüdiger Westermann. 2006. A Multigrid Framework for Real-Time Simulation of Deformable Bodies. *Comput. Graph.* 30, 3 (jun 2006), 408–415. <https://doi.org/10.1016/j.cag.2006.02.016>
- Rony Goldenthal, David Harmon, Kanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph.* 26, 3 (jul 2007), 49–es. <https://doi.org/10.1145/1276377.1276438>
- Geoffrey Irving, Craig Schroeder, and Ronald Fedkiw. 2007. Volume Conserving Finite Element Simulations of Deformable Models. In *ACM SIGGRAPH 2007 Papers* (San Diego, California) (SIGGRAPH '07). Association for Computing Machinery, New York, NY, USA, 13–es. <https://doi.org/10.1145/1275808.1276394>
- Volker John and Lutz Tobiska. 2000. Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 33, 4 (2000), 453–473. [https://doi.org/10.1002/1097-0363\(20000630\)33:4<453::AID-FLD15>3.0.CO;2-0](https://doi.org/10.1002/1097-0363(20000630)33:4<453::AID-FLD15>3.0.CO;2-0)
- C. Kane, J. E. Marsden, M. Ortiz, and M. West. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *Internat. J. Numer. Methods Engrg.* 49, 10 (2000), 1295–1325.
- Peter Kaufmann, Sebastian Martin, Mario Botsch, and Markus Gross. 2009. Flexible simulation of deformable models using discontinuous Galerkin FEM. *Graphical Models* 71, 4 (2009), 153–167. <https://doi.org/10.1016/j.gmod.2009.02.002> Special Issue of ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2008.
- Ryo Kikuuwe, Hiroaki Tabuchi, and Motoji Yamamoto. 2009. An edge-based computationally efficient formulation of Saint Venant-Kirchhoff tetrahedral finite elements. *ACM Trans. Graph.* 28, 1, Article 8 (feb 2009), 13 pages. <https://doi.org/10.1145/1477926.1477934>
- Maxim Larin and Arnold Reusken. 2008. A comparative study of efficient iterative solvers for generalized Stokes equations. *Numerical Linear Algebra with Applications* 15, 1 (2008), 13–34. <https://doi.org/10.1002/nla.561> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nla.561>
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy 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. (SIGGRAPH)* 39, 4, Article 49 (2020).
- Minchen Li, Ming Gao, Timothy Langlois, Chenfanfu Jiang, and Danny M. Kaufman. 2019. Decomposed Optimization Time Integrator for Large-Step Elastodynamics. *ACM Transactions on Graphics* 38, 4 (2019).
- Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. 2018. Narrow-Band Topology Optimization on a Sparsely Populated Grid. *ACM Trans. Graph.* 37, 6, Article 251 (dec 2018), 14 pages. <https://doi.org/10.1145/3272127.3275012>
- Hsueh-Ti Derek Liu, Jiayi Eris Zhang, Mirela Ben-Chen, and Alec Jacobson. 2021. Surface Multigrid via Intrinsic Prolongation. *ACM Trans. Graph.* 40, 4 (2021).
- Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-Newton Methods for Real-Time Simulation of Hyperelastic Materials. *ACM Trans. Graph.* 36, 3, Article 23 (may 2017), 16 pages. <https://doi.org/10.1145/2990496>
- Steve A. Maas, Benjamin J. Ellis, Gerard A. Ateshian, and Jeffrey A. Weiss. 2012. FEBio: finite elements for biomechanics. *Journal of biomechanical engineering* 134 1 (2012), 011005. <https://api.semanticscholar.org/CorpusID:2483033>
- Miles Macklin and Matthias Müller. 2021. A Constraint-Based Formulation of Stable Neo-Hookean Materials. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Virtual Event, Switzerland) (MIG '21). Association for Computing Machinery, New York, NY, USA, Article 12, 7 pages. <https://doi.org/10.1145/3487983.3488289>
- Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (jul 2011), 12 pages. <https://doi.org/10.1145/2010324.1964932>
- M. K. Misztal, K. Erleben, A. Bargteil, J. Fursund, B. Bunch Christensen, J. A. Bærentzen, and R. Bridson. 2012. Multiphase Flow of Immiscible Fluids on Unstructured Moving Meshes. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics Conference on Computer Animation* (Lausanne, Switzerland) (EUROSCA'12). Eurographics Association, Goslar, DEU, 97–106.
- NVIDIA. 2024. NVIDIA CUDA Toolkit Documentation. <https://developer.nvidia.com/cuda-toolkit> Accessed: 2024-01-01.
- C.W. Oosterlee and F.J. Gaspar. 2008. Multigrid relaxation methods for systems of saddle point type. *Applied Numerical Mathematics* 58, 12 (2008), 1933–1950. <https://doi.org/10.1016/j.apnum.2007.11.014> Special Issue in Honor of Piet Hemker.
- C. C. Paige and M. A. Saunders. 1975. Solution of Sparse Indefinite Systems of Linear Equations. *SIAM J. Numer. Anal.* 12, 4 (1975), 617–629. <https://doi.org/10.1137/0712047> arXiv:<https://doi.org/10.1137/0712047>
- Taylor Patterson, Nathan Mitchell, and Eftychios Sifakis. 2012. Simulation of Complex Nonlinear Elastic Bodies Using Lattice Deformers. *ACM Trans. Graph.* 31, 6, Article 197 (nov 2012), 10 pages. <https://doi.org/10.1145/2366145.2366216>
- Victor Romero, Mickaël Ly, Abdullah Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. 2021. Physical Validation of Simulators in Computer Graphics: A New Framework Dedicated to Slender Elastic Structures and Frictional Contact. *ACM Trans. Graph.* 40, 4, Article 66 (jul 2021), 19 pages. <https://doi.org/10.1145/3450626.3459931>
- S.H. Roth, Markus H. Gross, Silvio Turello, and Friedrich R. Carls. 1998. A Bernstein-Bézier Based Approach to Soft Tissue Simulation. *Computer Graphics Forum* 17, 3 (1998), 285–294. <https://doi.org/10.1111/1467-8659.00275> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00275>
- Leonard Sacht, Etienne Vouga, and Alec Jacobson. 2015. Nested Cages. *ACM Transactions on Graphics (TOG)* 34, 6 (2015).
- R. L. Sani, P. M. Gresho, R. L. Lee, and D. F. Griffiths. 1981. The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier-Stokes equations: Part 1. *International Journal for Numerical Methods in Fluids* 1, 1 (1981), 17–43. <https://doi.org/10.1002/flid.1650010104> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/flid.1650010104>
- Rajsekhar Setaluri, Yu Wang, Nathan Mitchell, Ladislav Kavan, and Eftychios Sifakis. 2015. Fast Grid-Based Nonlinear Elasticity for 2D Deformations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) (SCA '14). Eurographics Association, Goslar, DEU, 67–76.
- Han Shao, Libo Huang, and Dominik L. Michels. 2022. A Fast Unsmoothed Aggregation Algebraic Multigrid Framework for the Large-Scale Simulation of Incompressible Flow. *ACM Trans. Graph.* 41, 4, Article 49 (jul 2022), 18 pages. <https://doi.org/10.1145/3528223.3530109>
- Seung Heon Sheen, Egor Larionov, and Dinesh K. Pai. 2021. Volume Preserving Simulation of Soft Tissue with Skin. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 3, Article 32 (sep 2021), 23 pages. <https://doi.org/10.1145/3480143>
- SideFX. 2023. Houdini (Version 19.5). <https://www.sidefx.com>. [Computer software].
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (mar 2018), 15 pages. <https://doi.org/10.1145/3180491>
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (feb 2019), 15 pages. <https://doi.org/10.1145/3241041>
- Kaczmarz Stefan. 1993. Approximate solution of systems of linear equations†. *Internat. J. Control* 57, 6 (1993), 1269–1271. <https://doi.org/10.1080/00207179308934446> arXiv:<https://doi.org/10.1080/00207179308934446>
- Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for Phase-Change and Varied Materials. *ACM Trans. Graph.* 33, 4, Article 138 (jul 2014), 11 pages. <https://doi.org/10.1145/2601097.2601176>
- Rasmus Tamstorf, Toby Jones, and Stephen F. McCormick. 2015. Smoothed Aggregation Multigrid for Cloth Simulation. *ACM Trans. Graph.* 34, 6, Article 245 (nov 2015), 13 pages. <https://doi.org/10.1145/2816795.2818081>
- M. Teschner, B. Heidelberger, M. Müller, and M. Gross. 2004. A versatile and robust model for geometrically complex deformable solids. In *Proceedings Computer Graphics International, 2004*. 312–319. <https://doi.org/10.1109/CGI.2004.1309227>
- Matthias Teschner, Bruno Heidelberger, Matthias Müller, Danat Pomeranets, and Markus Gross. 2003. Optimized Spatial Hashing for Collision Detection of Deformable Objects. *VMV'03: Proceedings of the Vision, Modeling, Visualization* 3 (12 2003).
- Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. 2015. Stable constrained dynamics. *ACM Trans. Graph.* 34, 4, Article 132 (jul 2015), 10 pages. <https://doi.org/10.1145/2766969>
- Ty Trusty, Otman Benckekroun, Eitan Grinspun, Danny M. Kaufman, and David I.W. Levin. 2023. Subspace Mixed Finite Elements for Real-Time Heterogeneous Elastodynamics. In *SIGGRAPH Asia 2023 Conference Papers* (, Sydney, NSW, Australia.) (SA '23). Association for Computing Machinery, New York, NY, USA, Article 112, 10 pages. <https://doi.org/10.1145/3610548.3618220>

- Ty Trusty, Danny Kaufman, and David I.W. Levin. 2022. Mixed Variational Finite Elements for Implicit Simulation of Deformables. In *SIGGRAPH Asia 2022 Conference Papers* (, Daegu, Republic of Korea,) (SA '22). Association for Computing Machinery, New York, NY, USA, Article 40, 8 pages. <https://doi.org/10.1145/3550469.3555418>
- S.P Vanka. 1986. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J. Comput. Phys.* 65, 1 (1986), 138–158. [https://doi.org/10.1016/0021-9991\(86\)90008-2](https://doi.org/10.1016/0021-9991(86)90008-2)
- Bohan Wang, George Matcuk, and Jernej Barbič. 2019. Hand Modeling and Simulation Using Stabilized Magnetic Resonance Imaging. *ACM Trans. Graph.* 38, 4, Article 115 (jul 2019), 14 pages. <https://doi.org/10.1145/3306346.3322983>
- Zhendong Wang, Longhua Wu, Marco Fratarcangeli, Min Tang, and Huamin Wang. 2018. Parallel Multigrid for Nonlinear Cloth Simulation. *Computer Graphics Forum* 37, 7 (2018), 131–141. <https://doi.org/10.1111/cgf.13554> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13554>
- Botao Wu, Zhendong Wang, and Huamin Wang. 2022. A GPU-Based Multilevel Additive Schwarz Preconditioner for Cloth and Deformable Body Simulation. *ACM Trans. Graph.* 41, 4, Article 63 (jul 2022), 14 pages. <https://doi.org/10.1145/3528223.3530085>
- Zangyueyang Xian, Xin Tong, and Tiantian Liu. 2019. A Scalable Galerkin Multigrid Method for Real-Time Simulation of Deformable Objects. *ACM Trans. Graph.* 38, 6, Article 162 (nov 2019), 13 pages. <https://doi.org/10.1145/3355089.3356486>
- Omar Zarifi and Christopher Batty. 2017. A Positive-Definite Cut-Cell Method for Strong Two-Way Coupling between Fluids and Deformable Bodies. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) (SCA '17). Association for Computing Machinery, New York, NY, USA, Article 7, 11 pages. <https://doi.org/10.1145/3099564.3099572>
- Yongning Zhu, Eftychios Sifakis, Joseph Teran, and Achi Brandt. 2010. An Efficient Multigrid Method for the Simulation of High-Resolution Elastic Solids. *ACM Trans. Graph.* 29, 2, Article 16 (apr 2010), 18 pages. <https://doi.org/10.1145/1731047.1731054>
- O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu (Eds.). 2013. *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)* (seventh edition ed.). Butterworth-Heinemann, Oxford. <https://doi.org/10.1016/B978-1-85617-633-0.00024-1>