

A Linear and Angular Momentum Conserving Hybrid Particle/Grid Iteration for Volumetric Elastic Contact

ALAN MARQUEZ RAZON, University of California Los Angeles, USA

YIZHOU CHEN, University of California Los Angeles, USA

YUSHAN HAN, University of California Los Angeles, USA

STEVEN GAGNIERE, University of California Los Angeles, USA

MICHAEL TUPEK, Sandia National Laboratories, USA

JOSEPH TERAN, University of California Davis, USA

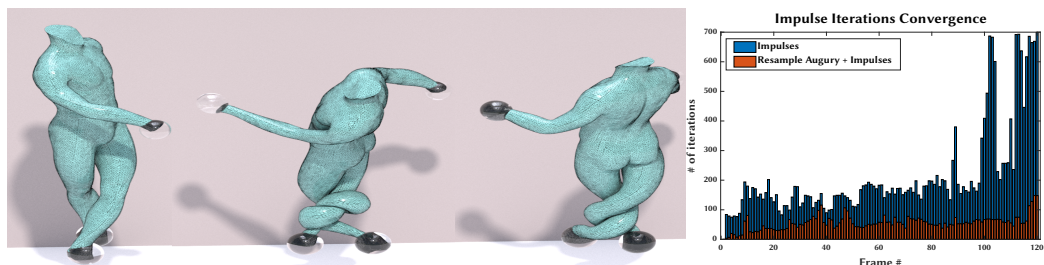


Fig. 1. (Left) A body mesh is twisted by constraints at the arms and legs. Our method robustly and efficiently resolves the collision driven deformation. (Right) Impulse iteration requirements with our hybrid approach (red) are significantly reduced compared to geometric impulses alone (blue), decreasing the total simulation run time by 25 percent.

We present a momentum conserving hybrid particle/grid iteration for resolving volumetric elastic collision. Our hybrid method uses implicit time stepping with a Lagrangian finite element discretization of the volumetric elastic material together with impulse-based collision-correcting momentum updates designed to exactly conserve linear and angular momentum. We use a two-step process for collisions: first we use a novel grid-based approach that leverages the favorable collision resolution properties of Particle-In-Cell (PIC) techniques, then we finalize with a classical collision impulse strategy utilizing continuous collision detection. Our PIC approach uses Affine-Particle-In-Cell momentum transfers as collision preventing impulses together with novel perfectly momentum conserving boundary resampling and downsampling operators that prevent artifacts in portions of the boundary where the grid resolution is of disparate resolution. We combine this with a momentum conserving augury iteration to remove numerical cohesion and model sliding friction. Our collision strategy has the same continuous collision detection as traditional approaches, however our hybrid particle/grid iteration drastically reduces the number of iterations required. Lastly, we develop a novel symmetric positive semi-definite Rayleigh damping model that increases the convexity of the nonlinear

Authors' addresses: Alan Marquez Razon, University of California Los Angeles, USA, ae.alan.marquez@gmail.com; Yizhou Chen, University of California Los Angeles, USA, chenyzhou@ucla.edu; Yushan Han, University of California Los Angeles, USA, yushanh1@math.ucla.edu; Steven Gagniere, University of California Los Angeles, USA, sgagniere@alumni.ucla.edu; Michael Tupek, Sandia National Laboratories, USA, mrtupek@gmail.com; Joseph Teran, University of California Davis, USA, jteran@math.ucdavis.edu.

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).

2577-6193/2023/8-ART44

<https://doi.org/10.1145/3606924>

systems associated with implicit time stepping. We demonstrate the robustness and efficiency of our approach in a number of collision intensive examples.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; • **Mathematics of computing** → **Discretization**.

ACM Reference Format:

Alan Marquez Razon, Yizhou Chen, Yushan Han, Steven Gagniere, Michael Tupek, and Joseph Teran. 2023. A Linear and Angular Momentum Conserving Hybrid Particle/Grid Iteration for Volumetric Elastic Contact. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 44 (August 2023), 25 pages. <https://doi.org/10.1145/3606924>

1 INTRODUCTION

The deformation of volumetric elastic solids is a fundamental aspect of computer graphics and related disciplines. Inertia-driven elastic deformation (particularly in response to collision and contact) adds rich detail and realism to animations. This is particularly true for animating characters with biomechanical soft tissues like muscle, fat, skin, etc. [Brunel et al. 2021; McAdams et al. 2011; Milne et al. 2016; Smith et al. 2018]. Indeed, effectively conveying the transition from squash to stretch is one of the key principles of animation [Thomas and Johnston 1995]. Since the pioneering work of Terzopoulos et al. [1987], researchers have developed approaches to generate elastic deformations with numerical simulation and the governing physics of elasticity. Graphics researchers often borrow and adapt techniques from the computational mechanics literature in this regard, however, despite many decades of research by both communities, the simulation of large-strain elastic solids with contact and collision remains an active area of research.

The governing physics of large strain elastic solids are comprised primarily of two factors: stress-based internal forces arising to resist deformation (often from distortion energies) [Gonzalez and Stuart 2008] and contact forces at the interface between colliding bodies [Belytschko et al. 2013; Kikuchi and Oden 1988]. In both the graphics and broader computational mechanics literature, the numerical treatment of stress-based forces is relatively settled. The finite element method (FEM) discretization of spatial stress gradients is used by the vast majority of researchers due to its support of complex geometries and its generally favorable numerical properties (e.g. symmetric discretization, spectral reproduction, etc.) [Belytschko et al. 2013; Hughes 2000]. In contrast, the numerical treatment of contact and collision is far less settled. Contact forces happen at such fast time scales that they are effectively discontinuous [Larionov et al. 2021; Li et al. 2020a] and as a consequence, their numerical discretization is far more delicate.

There are many existing options for resolving these terms and each of them has its relative pros and cons. Penalties, barriers, and repulsive forces effectively regularize the collision response, often idealizing it in terms of potential energy increasing with material overlap [Baraff and Witkin 1998; Barbič and James 2007; Gast et al. 2015; Moore and Wilhelms 1988; Spillmann et al. 2007; Teng et al. 2014; Teran et al. 2005]. Iterative discrete contact resolution between mesh facets like points, triangles, and edges has a long history in computer graphics and can provide strong collision-free assurances [Bridson et al. 2002; Cundall and Strack 1979; Harmon et al. 2009; Müller et al. 2007; Provot 1997; Volino et al. 1995; Wu et al. 2020]. Methods that formulate the problem in terms of constrained optimization are also very powerful, often resulting in Linear Complementary Problems (LCP) [Baraff and Witkin 1992; Larionov et al. 2021; Li et al. 2021; Otaduy et al. 2009; Wriggers and Laursen 2006; Wriggers et al. 1990]. Hybrid Lagrangian/Eulerian methods that make use of the natural collision resolution induced by numerical discretization in an Eulerian view are increasingly popular [Fan et al. 2013; Han et al. 2019; Jiang et al. 2017a; Levin et al. 2011; McAdams et al. 2009; Yue et al. 2018].

We develop an approach that uses a hybrid Lagrangian/Eulerian method combined with an iterative discrete contact model to attain the positive benefits of both approaches while minimizing their respective drawbacks. Particle-In-Cell (PIC) is a hybrid Lagrangian/Eulerian (particle/grid) technique originally developed for compressible flow applications [Harlow 1964]. Recently, generalizations of PIC techniques have been used to resolve collisions with a diverse range of materials in graphics applications. Particles in a discrete deformable object on a collision trajectory have velocities that become discontinuous at the moment of collision, however, the grid transfers in PIC techniques have a regularizing effect that prevents collision by preventing these discontinuities (see [Marquez et al. 2023]). However, as noted by Han et al. [2019] and Fei et al. [2021], these techniques suffer from numerical cohesion and friction by the same mechanism. We define a conservative Augury Affine-Particle-In-Cell (APIC) [Jiang et al. 2015; Tupek et al. 2021] technique that does not suffer from numerical cohesion or friction and that unlike Han et al. [2019] is perfectly linear and angular momentum conserving. We show that a notable benefit of this conservation is increased stability with large-time steps in practical simulations. PIC techniques generally require particle

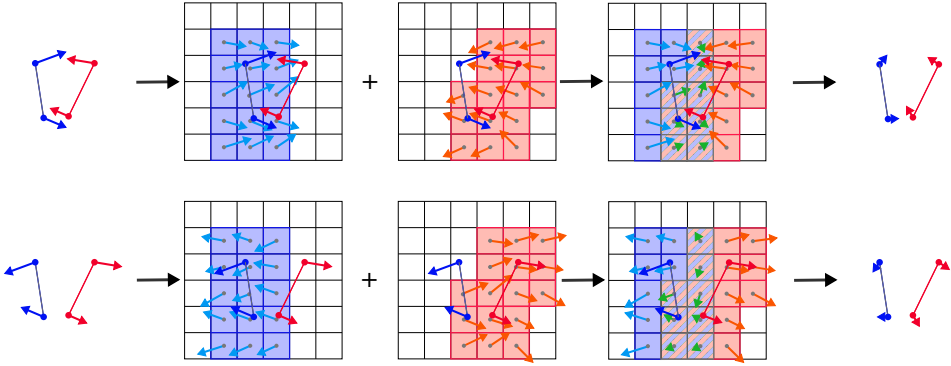


Fig. 2. (Top) Grid transfers resolve impact velocities. (Bottom) Numerical cohesion: opposing velocities are smoothed by the grid.

sampling to be comparable to grid resolution. Disregarding this constraint degrades the quality of their ability to prevent collisions (see [Marquez et al. 2023]). When viewing these transfers as operators on the boundary of a deformable object mesh, as we do, this requirement can be unrealistic. For example, when the unstructured mesh representing the deformable object has varying particle density on its boundary, the coarsest portion will determine acceptable grid resolution. This is undesirable since an overly coarse background grid leads to less effective collision resolution in practice. We design novel conservative resampling and subsequent downsampling operators for the boundary of the mesh to prevent these issues.

While our conservative hybrid approach is free from numerical cohesion and friction, maintaining this assurance can degrade its ability to prevent all collisions in practice. To resolve this, we augment our approach with the discrete contact model of Bridson et al. [2002]. We show that our combined approach drastically reduces the number of iterations that would be required from their approach alone; furthermore, we attain their strong collision-free assurances. We incorporate this hybrid collision model into a novel predictor/corrector implicit time stepping scheme. Backward Euler time integration is the primary building block of our approach and we add a novel Rayleigh damping model to regulate its inherent numerical damping. Our Rayleigh damping model has a symmetric positive semi-definite linearization which increases the convexity of the nonlinear backward Euler

systems and prevents the need for definiteness fixes [Teran et al. 2005] of finite element Hessians in practice. We summarize our technical contributions below:

- A resampling technique that conserves the mass, the center of mass, the linear momentum, and the angular momentum of a collection of particles with APIC mass and momentum state.
- A downsampling technique with analogous conservation properties are used to gather grid momenta associated with a resampled collection of particles into mass and momentum APIC state for the unresampled, original particles.
- A momentum conserving augury iteration for removing numerical friction and cohesion artifacts in APIC transfers.
- An implicit predictor/corrector time stepping scheme tailored to our collision response.
- A symmetric positive semi-definite Rayleigh damping model that increases the convexity of the hyperelastic backward Euler minimization problem.

2 RELATED WORK

Simulating contact between solids remains one of the more challenging aspects of finite deformation continuum mechanics simulations. We briefly discuss the most relevant works from the computer graphics and computational mechanics literature. There are many decades of research on the subject of contact and collision with deforming elastic objects. We refer the reader to reviews of the state-of-the-art in computer graphics [Ascher et al. 2021; Nealen et al. 2006] as well as the course notes in [Andrews et al. 2022; Kim and Elberle 2022]. In the engineering literature, early important contributions to Lagrangian finite element contact algorithms include [Belytschko and Neal 1991; Campos et al. 1982; Martins and Oden 1983; Simo and Laursen 1992; Wriggers et al. 1990]. Mortar contact methods are now often preferred for their accuracy and stability [Popp et al. 2010; Puso and Laursen 2004], but can have high computational expense due to additional Lagrange multiplier unknowns, typically do not exactly conserve angular momentum, and not all contact collisions are guaranteed to be detected. Generally, techniques for resolving collisions with deformable bodies fall into three categories: penalties/repulsive forces [Baraff and Witkin 1998; Barbič and James 2007; Gast et al. 2015; Moore and Wilhelms 1988; Spillmann et al. 2007; Teng et al. 2014; Teran et al. 2005], iterative discrete contact [Bridson et al. 2002; Cundall and Strack 1979; Harmon et al. 2009; Müller et al. 2007; Provot 1997; Volino et al. 1995; Wu et al. 2020] and constrained optimization [Baraff and Witkin 1992; Larionov et al. 2021; Li et al. 2021; Otaduy et al. 2009; Wriggers and Laursen 2006; Wriggers et al. 1990]. However, hybrid approaches are also possible [Daviet 2020; Fan et al. 2013; Han et al. 2019; Jiang et al. 2017a; Levin et al. 2011; McAdams et al. 2009; Yue et al. 2018]. Our approach utilizes a combination of hybrid PIC [Harlow 1964; Jiang et al. 2015] and discrete contact [Bridson et al. 2002] models. We discuss techniques most relevant to these. Our approach is inspired by the Material Point Method (MPM), which is an extension of PIC. MPM was first proposed in [Sulsky et al. 1994] and [Sulsky et al. 1995], with the first strategy for contact introduced shortly thereafter in [Bardenhagen et al. 2000]. Various improvements to MPM contact can also be found in [Homel and Herbold 2017; Huang et al. 2011; Nairn and Guo 2005; Xiao et al. 2021]. A strategy for introducing discontinuities into the MPM was recently suggested in [Moutsanidis et al. 2019] and shows advantages for sliding and separating contact. Hegemann et al. [2013] make use of PIC techniques in graphics applications of ductile fracture. Here we utilize the affine particle-in-cell (APIC) method from [Jiang et al. 2015, 2017b]. The APIC augury iterations utilized in this work were first suggested in [Tupek et al. 2021], with limited examples. The initial version proved to be unstable in some situations and was unable to accurately account for large size discrepancies between foreground and background meshes. Our method successfully addresses these issues. Our approach is directly related to a few recently proposed hybrid PIC/discrete geometric collision

techniques. Han et al. [2019] develop a PIC-based contact algorithm for FEM meshes, including augmentation with Bridson et al. [2002] for thin strands. McAdams et al. [2009] and Yue et al. [2018] similarly use a hybrid PIC/MPM technique for hair and granular material simulations respectively. McAdams et al. [2009] augment their approach with Bridson et al. [2002], and Yue et al. [2018] combine a discrete element method (DEM) [Cundall and Strack 1979] with a continuum model [Klar et al. 2016; Sulsky et al. 1994] for granular materials.

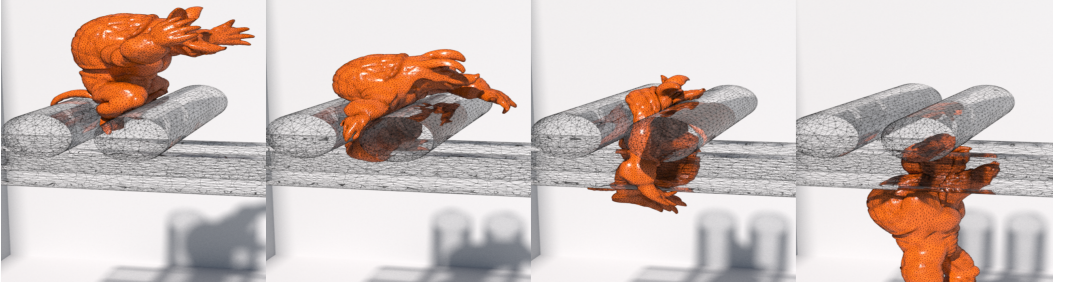


Fig. 3. As shown here, our method is robust to extreme deformation.

3 BACKGROUND AND NOTATION

We represent volumetric deformable objects as a collection of particles $\mathbf{x} \in \mathbb{R}^{dN}$ and a simplex mesh $\mathcal{M} \in \mathbb{N}^{N^e \times (d+1)}$ connecting them together. Our approach is designed for volumetric simulation in 3D, however, we use 2D examples for illustration. We use $d = 2, 3$ in our exposition to represent the case of 2D or 3D simulations respectively. We denote the boundary mesh of the volumetric object as $\mathcal{M}_b \in \mathbb{N}^{N_b^e \times d}$. Here N refers to the number of particles, N^e refers to the number of elements in the volumetric mesh (tetrahedra for $d = 3$, triangles for $d = 2$) and N_b^e refers to the number of elements in the boundary mesh (triangles for $d = 3$, segments for $d = 2$). We also store a velocity $\mathbf{v} \in \mathbb{R}^{dN}$ for each particle in the volumetric mesh. Our approach requires an APIC state over the particles in the boundary of the volumetric mesh. We use $\mathbf{A} \in \mathbb{R}^{d^2 N_b}$ to denote the vector of affine velocities associated with particles on the boundary of the volumetric mesh. We use N_b to denote the number of particles on the boundary. Next, we describe details of our method related to the spatially discretized governing physics of Lagrangian hyperelasticity and backward Euler integration of their temporal dynamics. These aspects of our approach are standard in the graphics and mechanics literature and we refer the reader to Bonet and Wood [2008] and Sifakis and Barbič [2012] for more detail. The contents of the following subsections are primarily intended for a brief review and establishing notation. We use Table 3 for a quick reference to the various symbols used in the exposition.

3.1 Elasticity

We discretize and solve the partial differential equations (PDEs) of motion for volumetric hyperelastic materials [Gonzalez and Stuart 2008] with Rayleigh damping [Belytschko et al. 2013] and frictional collision constraints [Bridson et al. 2002]. We use the finite element method (FEM) with linear interpolation functions over simplex meshes to integrate the PDEs [Hughes 2000]. This converts spatial terms in the PDE to discrete elastic forces $\mathbf{f}^E : \mathbb{R}^{dN} \rightarrow \mathbb{R}^{dN}$. These forces are the gradient of the discrete potential energy of the system $\text{PE} : \mathbf{x} \in \mathbb{R}^{dN} \rightarrow \mathbb{R}$. The potential energy increases as the mesh is deformed from a reference configuration (defined in terms of reference positions $\mathbf{X} \in \mathbb{R}^{dN}$) to any other configuration $\mathbf{x} \in \mathbb{R}^{dN}$. It is defined in terms of a hyperelastic

strain energy density function $\Psi : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$ which increases with non-rigidity of the unique affine transformation that relates the reference configuration of a mesh element to that defined by \mathbf{x} [Teran et al. 2003]. This is typically referred to as the deformation gradient in the element \mathbf{F}^e :

$$\text{PE}(\mathbf{x}) = \sum_{e \in \mathcal{M}} \Psi(\mathbf{F}^e(\mathbf{x})) V_e^0$$

$$\mathbf{f}_p^E(\mathbf{x}) = -\frac{\partial \text{PE}}{\partial \mathbf{x}_p}(\mathbf{x}) = \sum_{e \in \mathcal{M}} \mathbf{P}(\mathbf{F}^e(\mathbf{x})) \left(\frac{\partial N_p^L(\mathbf{X}^e)}{\partial \mathbf{X}} \right)^T V_e^0, \quad 0 \leq p < N.$$

Here V_e^0 is the measure of the element in the reference configuration, $\mathbf{P}(\mathbf{F}) = \frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F})$ is the first Piola-Kirchhoff stress [Bonet and Wood 2008], \mathbf{X}^e is the centroid of mesh element e in the reference configuration, \mathbf{x}_p is the position of particle p (which make up \mathbf{x}) and N_p^L are the piecewise linear interpolation functions associated with the FEM space. Furthermore, $\mathbf{f}_p^E(\mathbf{x}) \in \mathbb{R}^d$ are the elastic forces on volumetric particles p that comprise the vector $\mathbf{f}^E(\mathbf{x})$. Lastly, we note that we use the fixed-Corotated model of Stomakhin et al. [2012] for the energy density Ψ .

3.2 Mass

We define mass for each volumetric element $e \in \mathcal{M}$ from mass density ρ as $m^e = \rho V_e^0$. We define the mass for each particle in the volumetric mesh by taking a portion of the mass from each element that it belongs to $m_p = \sum_{e \in \mathcal{I}^p} \frac{m^e}{d+1}$. Here \mathcal{I}^p is the one ring of mesh elements $e \in \mathcal{M}$ that contains the particle p . We note that in a Lagrangian FEM discretization of the governing physics, conservation of mass implies that element and particle masses m_p do not change with the configuration of the mesh \mathbf{x} . We use $\mathbf{m} \in \mathbb{R}^N$ to denote the vector of particle masses in the volumetric mesh. To facilitate our conservative boundary element resampling, we also need a notion of boundary element mass $m_{b_i^e}^e$ for each particle $0 \leq i^e < d$ in boundary element $e \in \mathcal{M}_b$. We define the $m_{b_i^e}^e$ to partition the m_p such that

$$m_p = \sum_{e \in \mathcal{I}_b^p} \sum_{i^e=0}^{d-1} m_{b_i^e}^e, \quad m_{b_i^e}^e = \frac{m_p}{|\mathcal{I}_b^p|}$$

(where \mathcal{I}_b^p is the one ring of elements in the boundary mesh that contain the particle p) and $|\mathcal{I}^p|$ is the number of elements in the set.

3.3 Backward Euler (BE)

Given volumetric mesh positions \mathbf{x}^n and velocities \mathbf{v}^n at discrete time t^n , we approximate the trajectories of the mesh vertices under the elastodynamics at the next time $t^{n+1} = t^n + \Delta t$ with implicit time integration. Here, Δt is the time step. We adopt the approach of Gast et al. [2015] and characterize this via the minimization

$$\mathbf{x}^{n+1} = \underset{\mathbf{y} \in \mathbb{R}^{dN} \text{ s.t. } \mathbf{B}\mathbf{y}=\mathbf{c}}{\text{argmin}} E(\mathbf{y}), \quad (1)$$

$$E(\mathbf{y}) = (\mathbf{y} - \mathbf{x}^n - \Delta t \mathbf{v}^n)^T \mathbf{M}(\mathbf{y} - \mathbf{x}^n - \Delta t \mathbf{v}^n) + \Delta t^2 \text{PE}(\mathbf{y}) +$$

$$\Delta t (\mathbf{y} - \mathbf{x}^n)^T \mathbf{K}^{RD}(\mathbf{x}^n)(\mathbf{y} - \mathbf{x}^n) - \Delta t^2 \mathbf{y}^T \mathbf{M} \mathbf{g}$$

Here \mathbf{x}^{n+1} is the backward Euler solution at time t^{n+1} . $\mathbf{M} \in \mathbb{R}^{dN \times dN}$ is a lumped-mass diagonal matrix with entries equal to particle masses m_p . $\mathbf{K}^{RD} \in \mathbb{R}^{dN \times dN}$ is the Rayleigh damping matrix which we discuss in more detail in Section 8. $\mathbf{B} \in \mathbb{R}^{dN_c \times dN}$ and $\mathbf{c} \in \mathbb{R}^{dN_c}$ express linear constraints over vertices in the volumetric mesh and N_c is the number of constrained vertices. $\mathbf{g} \in \mathbb{R}^{dN}$ is

gravity. The backward Euler velocity can be obtained after the minimization in Equation (1) from $\mathbf{v}^{n+1} = \frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t}$. We will henceforth use the notation

$$(\mathbf{x}^{n+1}, \mathbf{v}^{n+1}) = BE(\mathbf{x}^n, \mathbf{v}^n, \Delta t, \mathbf{m}, \mathcal{M}, \mathbf{B}, \mathbf{c})$$

to denote the function that returns the backward Euler positions and velocities associated with the solution of the minimization in Equation (1). Lastly, we note that we solve the minimization problem in Equation (1) using Newton's method. We refer the reader to [Sifakis and Barbic 2012; Stomakhin et al. 2012; Teran et al. 2005] for more details. However, we use standard conjugate gradients to solve the linear systems. As a consequence of the increased convexity from our Rayleigh damping model, in practice we do not need to perform the definiteness fixes of the stress derivatives used Teran et al. [Teran et al. 2005]. We discuss this in more detail in Section 8.

4 METHOD OVERVIEW

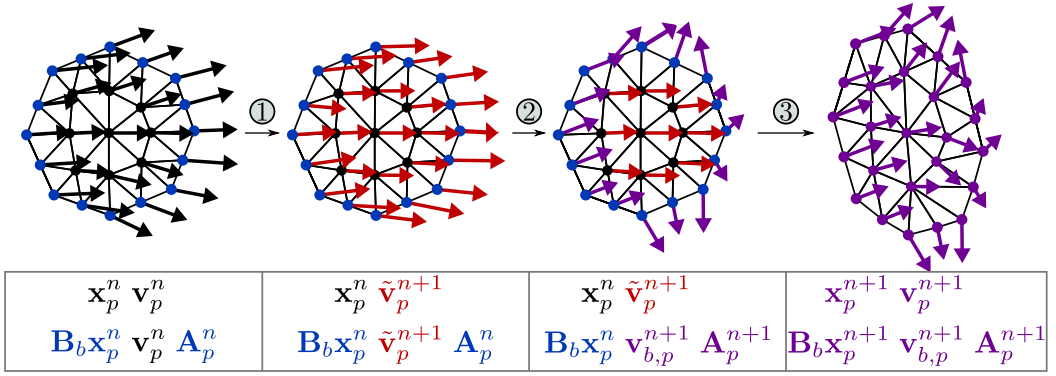


Fig. 4. Method overview: (1) collision-unaware backward Euler positions rewound to time t^n (2) collision processing for the boundary velocities (3) final backward Euler step with interior state reset to t^n and the boundary state at t^{n+1} used as Dirichlet boundary conditions.

Our method is comprised of three main components: backward Euler steps (Section 3.3), augury APIC hybrid particle/grid transfers (Section 5), and discrete geometric impulses (Section 7). While our discrete geometric impulse operation is the standard approach of Bridson et al. [2002], our augury APIC technique has many novel aspects. Of particular importance are our resampling/downsampling operators (Section 6) and our iterative resolution of numerical cohesion and friction (Section 5).

ALGORITHM 1: Method Overview: Time Stepping

1. $(\tilde{\mathbf{x}}^{n+1}, \tilde{\mathbf{v}}^{n+1}) = BE(\mathbf{x}^n, \mathbf{v}^n, \Delta t, \mathbf{m}, \mathcal{M}, \mathbf{B}(t^{n+1}), \mathbf{c}(t^{n+1}))$;
 2. $(\mathbf{v}_b^A, \mathbf{A}^{n+1}) = \mathcal{L}^A(\mathbf{B}_b \tilde{\mathbf{v}}^{n+1}, \mathbf{A}^n, \mathbf{B}_b \mathbf{m}, \mathcal{M}_b)$;
 $\mathbf{v}_b^{n+1} = \mathcal{L}^I(\mathbf{v}_b^A, \mathbf{B}_b \mathbf{m}, \mathcal{M}_b)$;
 - $\mathbf{B}^{IA}(t^{n+1}) = \begin{bmatrix} \mathbf{B}(t^{n+1}) \\ \mathbf{B}_b \end{bmatrix}, \mathbf{c}^{IA}(t^{n+1}) = \begin{bmatrix} \mathbf{c}(t^{n+1}) \\ \mathbf{B}_b \mathbf{x}^n + \Delta t \mathbf{v}_b^{n+1} \end{bmatrix}$;
 3. $(\mathbf{x}^{n+1}, \mathbf{v}^{n+1}) = BE(\mathbf{x}^n, \mathbf{v}^n, \Delta t, \mathbf{m}, \mathcal{M}, \mathbf{B}^{IA}(t^{n+1}), \mathbf{c}^{IA}(t^{n+1}))$;
-

We first take a backward Euler step to compute tentative time t^{n+1} positions and velocities $(\tilde{\mathbf{x}}^{n+1}, \tilde{\mathbf{v}}^{n+1})$. This update is collision unaware so we rewind the positions to time t^n and process

the tentative velocities $\tilde{\mathbf{v}}^{n+1}$ for collision. We do this in a two-step process. In the first step, we use our resampled augury APIC impulse operator $\mathcal{L}^A : \mathbb{R}^{dN_b} \times \mathbb{R}^{d^2N_b} \times \mathbb{R}^{N_b} \times \mathbb{N}^{N_b^c \times d} \rightarrow \mathbb{R}^{dN_b} \times \mathbb{R}^{d^2N_b}$ to efficiently apply conservative and grid-resolution-independent impulses that prevent collision without numerical cohesion or friction (see Section 5). This defines a new boundary APIC state $(\mathbf{v}_b^A, \mathbf{A}^{n+1})$. Note that after this process, the linear part of the APIC state (\mathbf{A}^{n+1}) is finalized for the time step, while the constant part (\mathbf{v}_b^A) will be further processed in the second step.

In the second step, we apply the collision impulse model of Bridson et al. [2002] (see Section 7). We denote this process with the operator $\mathcal{L}^I : \mathbb{R}^{dN_b} \times \mathbb{R}^{N_b} \times \mathbb{N}^{N_b^c \times d} \rightarrow \mathbb{R}^{dN_b}$. This defines the final collision processed boundary velocities \mathbf{v}_b^{n+1} .

Lastly, once we have finalized the collision response on the boundary, we redo the backward Euler time step from time t^n to t^{n+1} where the boundary is constrained to follow the linear trajectories determined by the finalized velocities in the previous two steps. This propagates the collision response to the interior, which allows for increased stability and larger time steps (see [Marquez et al. 2023]). We note that our collision processing has the continuous collision detection (CCD) assurances of Bridson et al. [2002] that linear trajectories of boundary points (based on their collision-processed velocities) will guarantee a collision-free state on the boundary over the time step.

We summarize this process in Fig 4 and Algorithm 1. Note that we make use of the matrix $\mathbf{B}_b \in \mathbb{R}^{dN_b \times dN}$ whose rows are associated with boundary particles and whose columns are associated with volumetric particles. There is only one non-zero entry per row in the column associated with the boundary particle's location in the volumetric particle array. For convenience, we use the notation $\mathbf{B}_b \mathbf{m}$ to denote the masses associated with boundary particles, although this is a slight abuse of notation.

5 AUGURY ITERATION: \mathcal{L}^A

We leverage the natural collision prevention tendencies of PIC techniques to define our collision operator $\mathcal{L}^A : \mathbb{R}^{dN_b} \times \mathbb{R}^{d^2N_b} \times \mathbb{R}^{N_b} \times \mathbb{N}^{N_b^c \times d} \rightarrow \mathbb{R}^{dN_b} \times \mathbb{R}^{d^2N_b}$. This operator maps the elastic object boundary affine momentum state to itself to resolve portions of the mesh on collision trajectories. It is composed of three components: APIC [Jiang et al. 2015] particle/grid mass/momentum transfers, novel resampling/downsampling strategies, and an augury iteration designed to prevent numerical cohesion and friction. We adopt the standard APIC transfers and refer the reader to Jiang et al. [2015] for details. Our resampling and downsampling strategies are designed to prevent the degradation of collision prevention abilities that arise with disparate boundary mesh and background grid resolutions (see Section 6). Here we discuss our conservative approach to removing numerical cohesion and friction inherent in the naive APIC transfers.

We first define the conservative operator $\mathcal{L}^{\text{RSA}} = \mathcal{L}^{\text{G2P}} \circ \mathcal{L}^{\text{DS}} \circ \mathcal{L}^{\text{P2G}} \circ \mathcal{L}^{\text{RS}}$ which maps the affine velocity state of the boundary particles to itself via composition of the APIC particle-to-grid (P2G) and (G2P) operators (\mathcal{L}^{P2G} and \mathcal{L}^{G2P} respectively) together with the resampling and downsampling operations (\mathcal{L}^{RS} and \mathcal{L}^{DS}) outlined in Section 6. We summarize this process as

$$(\mathbf{v}_b, \mathbf{A}) \xrightarrow{\mathcal{L}^{\text{RS}}} (\hat{\mathbf{v}}_b, \hat{\mathbf{A}}) \xrightarrow{\mathcal{L}^{\text{P2G}}} (\hat{\mathbf{m}}_g, \hat{\mathbf{p}}_g) \xrightarrow{\mathcal{L}^{\text{DS}}} (\mathbf{m}_g, \mathbf{p}_g) \xrightarrow{\mathcal{L}^{\text{G2P}}} \mathcal{L}^{\text{RSA}}((\mathbf{v}_b, \mathbf{A})). \quad (2)$$

Here we use $(\mathbf{v}_b, \mathbf{A})$ to denote the APIC state of boundary (where $\mathbf{v}_b \in \mathbb{R}^{dN_b}$ and $\mathbf{A} \in \mathbb{R}^{d^2N_b}$ are vectors containing the constant and linear parts of the APIC velocity for particles in the boundary) and $(\mathbf{m}_g, \mathbf{p}_g)$ to denote the grid mass and momentum state after an APIC particle-to-grid (P2G) transfer (where $\mathbf{m}_g \in \mathbb{R}^{N_{b,g}}$ and $\mathbf{p}_g \in \mathbb{R}^{dN_{b,g}}$ are vectors containing the mass and linear momentum on grid nodes and $N_{b,g}$ is the number of grid node \mathbf{x}_i whose interpolation functions N_i are non-zero on some boundary particle). Here, $N_i(\mathbf{x})$ are quadratic B-spline interpolating functions associated

with grid node \mathbf{x}_i used to transfer particle quantities to grid quantities [Jiang et al. 2015; Stomakhin et al. 2013]. Similarly, we use $(\hat{\mathbf{v}}_b, \hat{\mathbf{A}})$ and $(\hat{\mathbf{m}}_g, \hat{\mathbf{p}}_g)$ to denote their resampled counterparts (where $\hat{\mathbf{v}}_b \in \mathbb{R}^{d\hat{N}_b}$, $\hat{\mathbf{A}} \in \mathbb{R}^{d^2\hat{N}_b}$, $\hat{\mathbf{m}}_g \in \mathbb{R}^{\hat{N}_{b,g}}$, $\hat{\mathbf{p}}_g \in \mathbb{R}^{d\hat{N}_{b,g}}$, \hat{N}_b and $\hat{N}_{b,g}$ have meanings analogous to their downsampled counterparts).

As shown in Figure 2, the APIC transfers naturally regularize particles on collision trajectories to prevent impact. However, they also prevent separation and relative sliding by the same mechanism. This was noted in Han et al. [2019] where cohesive and frictional responses were discarded in a simple and non-conservative per-boundary-particle manner. We adopt the approach of Tupek et al. [2021] to leverage the conservative-smoothing/low-pass-filter nature of the APIC operator to conservatively damp out regions where it is cohesive and/or overly frictional. We do this by iteratively applying the operator to the change it induced in the previous iteration, but only where it is deemed overly frictional or cohesive with the aim of damping the unwanted cohesion and friction to zero. By design, this process retains the momentum conservation properties of APIC techniques. We refer the reader to Jiang et al. [2015] for discussion of the conservative-smoothing/low-pass-filter properties of PIC.

We define the conservatively resampled augury APIC mapping \mathcal{L}^A by first initializing its output to be equal to the resampled APIC operator \mathcal{L}^{RSA} . We then interpret its output as defining impulsive changes in the boundary grid node momenta and check if these impulses are cohesive. If they are not deemed cohesive, we accept the normal component of the impulse. Next, we increment the APIC output by the difference between the conservatively resampled APIC operator \mathcal{L}^{RSA} applied to the cohesion processed impulse and itself. This increment does not change the momentum state since both \mathcal{L}^{RSA} and the identity operator are linear and angular momentum conserving. We repeat this process until we hit a maximum number of iterations or the impulses converge to a tolerance.

We reiterate that the low pass filter nature of \mathcal{L}^{RSA} and its repeated application to the normal component of the impulse is designed to dampen any cohesion or friction to zero. Technically, for a frictionless response, the tangential component of the impulse must be filtered along with the cohesion. However, we found that while this effectively limited these effects it degraded collision prevention. If we instead keep the tangential component whenever a change is deemed non-cohesive, we observed stronger collision prevention abilities, albeit at the expense of some numerical friction.

We outline this process in Algorithm 2. Note that the normal \mathbf{n}_p is used for determining numerical cohesion for each particle on the boundary via area-weighted normalization. Further, note that we denote the option of accepting some numerical friction for the benefit of improved collision prevention as NF .

6 BOUNDARY ELEMENT RESAMPLING: \mathcal{L}^{RS} , \mathcal{L}^{DS}

As shown in [Marquez et al. 2023], when the resolution of the boundary mesh is too coarse relative to the background grid, the collision resolution abilities of PIC techniques degrade. In extreme cases, collisions can be completely missed altogether. We design a novel resampling strategy to remove this limitation as well as a novel downsampling strategy to return the boundary to its original resolution.

Our resampling strategy is designed on a per-boundary element manner. For boundary element $e \in \mathcal{M}_b$ with boundary positions $\mathbf{x}_{b_{ie}}^e$, masses $m_{b_{ie}}^e$, velocities $\mathbf{v}_{b_{ie}}^e$ and affine velocities $\mathbf{A}_{b_{ie}}^e$ where $0 \leq i^e < d$, we resample to create positions $\hat{\mathbf{x}}_{b_{je}}^e$, masses $\hat{m}_{b_{je}}^e$, velocities $\hat{\mathbf{v}}_{b_{je}}^e$ and affine velocities $\hat{\mathbf{A}}_{b_{je}}^e$ for $0 \leq j^e < N_r^e$. For each element local particle index i^e , there is a corresponding global boundary index p ; the velocity $\mathbf{v}_{b_{ie}}^e$ and affine velocity $\mathbf{A}_{b_{ie}}^e$ are then equal to \mathbf{v}_{bp} and \mathbf{A}_p respectively, where these are the vectors which constitute the input state $(\mathbf{v}_b, \mathbf{A})$ to \mathcal{L}^{RS} . Here we use N_r^e to denote the number of resample points in boundary element e (see Figure 5). Care must be taken to

ALGORITHM 2: Augury Iteration: $(\mathbf{v}, \mathbf{A})^A = \mathcal{L}^A((\tilde{\mathbf{v}}, \tilde{\mathbf{A}}))$

```

 $(\mathbf{v}, \mathbf{A})^A = \mathcal{L}^{\text{RSA}}((\tilde{\mathbf{v}}, \tilde{\mathbf{A}}));$ 
while  $it < \max\_it$  do
     $\delta \mathbf{v} = \mathbf{v}^A - \tilde{\mathbf{v}};$ 
    for  $(p = 0; p < N^b; p++)$  do
        if  $\delta \mathbf{v}_p$  is not cohesive ( $\delta \mathbf{v}_p \cdot \mathbf{n}_p \leq 0$ ) then
            if  $NF$  then
                 $\delta \mathbf{v}_p = \mathbf{0};$ 
            else
                 $\delta \mathbf{v}_p = \delta \mathbf{v}_p - (\delta \mathbf{v}_p \cdot \mathbf{n}_p) \mathbf{n}_p;$ 
            end
        else
            end
    end
     $(\mathbf{v}, \mathbf{A})^{A+} = \mathcal{L}^{\text{RSA}}((\delta \mathbf{v}, \delta \mathbf{A})) - (\delta \mathbf{v}, \mathbf{0});$ 
end

```

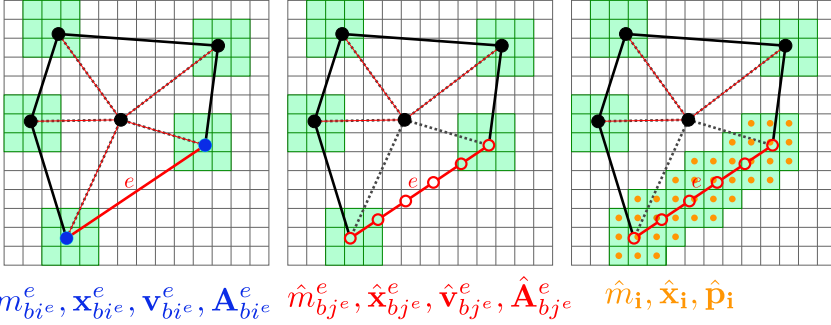


Fig. 5. (Left) Boundary element e has particle quantities (blue) with implied background grid state (green). (Middle) Resampling of the particle quantities into the resample points (red) conserves various element properties. (Right) Resample points have an analogous grid representation that better represents the element in finer grids.

conserve the total mass (m_b^e), the center of mass (\mathbf{x}_{bcom}^e), linear (\mathbf{p}_b^e), and angular momenta (\mathbf{l}_b^e) of the boundary element in the resampling process. While the element mass and center of mass are defined in the standard way, i.e.

$$m_b^e = \sum_{i^e} m_{bie}^e, \quad \mathbf{x}_{bcom}^e = \frac{1}{m_b^e} \sum_{i^e=0}^{d-1} m_{bie}^e \mathbf{x}_{bie}^e,$$

element momenta are defined with the APIC [Jiang et al. 2015] convention where boundary element particles \mathbf{x}_{bie}^e contribute based on their associated grid momenta

$$\mathbf{p}_i^{e,i^e} = m_{bie}^e N_i(\mathbf{x}_{bie}^e) (\mathbf{v}_{bie}^e + \mathbf{A}_{bie}^e (\mathbf{x}_i - \mathbf{x}_{bie}^e))$$

at grid node \mathbf{x}_i defined from an APIC transfer to the grid. Summing over the element points i^e defines the boundary element grid momentum distribution $\mathbf{p}_i^e = \sum_{i^e=0}^{d-1} \mathbf{p}_i^{e,i^e}$. The total linear

($\mathbf{p}_b^e = \sum_i \mathbf{p}_i^e$) and angular ($\mathbf{l}_b^e = \sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}^e) \times \mathbf{p}_i^e$) momenta of the element (computed about the element center of mass) are defined from the element grid momentum distribution.

The total momentum distribution of the boundary is defined from the sum of the element-wise contributions $\mathbf{p}_i = \sum_e \mathbf{p}_i^e$. The total mass and center of mass are defined similarly:

$$m_b = \sum_{e \in \mathcal{M}_b} m_b^e, \quad \mathbf{x}_{bcom} = \frac{1}{m_b} \sum_i \sum_{e \in \mathcal{M}_b} m_{b_{ie}}^e N_i(\mathbf{x}_{b_{ie}}^e) \mathbf{x}_i.$$

The total grid linear \mathbf{p}_b and angular \mathbf{l}_b momenta (computed about the boundary center of mass) of the boundary are related to the element-wise counterparts by the relations

$$\mathbf{p}_b = \sum_{e \in \mathcal{M}_b} \mathbf{p}_b^e, \quad \mathbf{l}_b = \sum_{e \in \mathcal{M}_b} \mathbf{l}_b^e + \sum_{e \in \mathcal{M}_b} (\mathbf{x}_{bcom}^e - \mathbf{x}_{bcom}) \times \mathbf{p}_b^e. \quad (3)$$

Equation (3) shows that the total linear momentum of the boundary is equal to the sum of the element-wise linear momenta. As illustrated in Figure 6, the total angular momentum of the boundary is equal to the sum of the angular momenta of each element (computed about its center of mass) plus the sum of the angular momenta induced by the total linear momentum of the element and the relation of its center of mass to the global center of mass. Thus, choosing a resampling strategy that preserves m^e , \mathbf{x}_{bcom}^e , \mathbf{p}_b^e , and \mathbf{l}_b^e gives the appropriate notion of local and global conservation.

$$\sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}) \times \mathbf{p}_i = \sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}^0) \times \mathbf{p}_i^0 + \sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}^1) \times \mathbf{p}_i^1 + \sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}^2) \times \mathbf{p}_i^2 + \sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}^3) \times \mathbf{p}_i^3 + \sum_e (\mathbf{x}_{bcom}^e - \mathbf{x}_{bcom}) \times \mathbf{p}_b^e$$

Fig. 6. The global angular momenta between the mesh and grid representations are equivalent. The details of the proof are in the supplemental technical document [Marquez et al. 2023], we illustrate the basis for the proof in this diagram.

6.1 Mass and Position Resampling: Partition of Unity and Linear Reproduction

We choose the resampled positions $\hat{\mathbf{x}}_{b_{je}}^e$ using random Poisson disc sampling within the element together with the original $\mathbf{x}_{b_{ie}}^e$. We use the convention that the first d resampled points $\hat{\mathbf{x}}_{b_{je}}^e$ coincide with the original $\mathbf{x}_{b_{ie}}^e$. Poisson disc sampling for the remaining points is chosen to prevent particle clumping while maintaining separation by no more than $\frac{\Delta x}{4}$ on average (Δx is the grid spacing). We require the resampled positions $\hat{\mathbf{x}}_{b_{je}}^e$ to be inside the element and we require that the resampled masses $\hat{m}_{b_{je}}^e$ be positive and conserve the element mass $\hat{m}_b^e = \sum_{j=0}^{N_r^e-1} \hat{m}_{b_{je}}^e = m_b^e$.

We also require conservation of the element center of mass \mathbf{x}_{bcom}^e . Furthermore, in order to allow for momentum conservation in our downsampling strategy (see Section 6.2), we must ensure that both the center of mass of the first d resample positions and the center of mass of the remaining resample positions ($d \leq j^e < N_r^e$) are equal to the element center of mass.

We preserve the center of mass of the first d resampling points by simply scaling their original masses $\hat{m}_{bj^e}^e = sm_{bj^e}^e$, $s \in (0, 1]$, $0 \leq j^e < d$

This associates a scaling of the total element mass sm_b^e to these particles and the remaining resampled masses $\hat{m}_{bj^e}^e$, $d \leq j^e < N_r^e$ must be chosen to partition the remaining element mass $(1-s)m_b^e$ in such a way that their center of mass is equal to that of the element. This can be achieved by defining these masses in terms of interpolation functions $\hat{N}_{j^e}^e$, $d \leq j^e < N_r^e$, defined over the remaining resample points as

$$\hat{m}_{bj^e}^e = (1-s)m_b^e \hat{N}_{j^e}^e(\mathbf{x}_{bcom}^e), \quad (4)$$

where the functions $\hat{N}_{j^e}^e$ satisfy partition of unity ($\sum_{j^e=d}^{N_r^e-1} \hat{N}_{j^e}^e(\mathbf{x}) = 1$) and linear reproduction ($\sum_{j^e=d}^{N_r^e-1} \hat{N}_{j^e}^e(\mathbf{x}) \hat{\mathbf{x}}_{bj^e}^e = \mathbf{x}$) properties. We adopt the approach of Arroyo and Ortiz [2006] to create interpolation functions that satisfy these requirements for unstructured particles. This choice allows for total element mass conservation and the conservation of the center of mass from the partition of unity and linear reproduction properties (respectively) of the interpolating functions $\hat{N}_{j^e}^e$ [Marquez et al. 2023].

6.1.1 Affine Velocity Resampling. We choose the resampled APIC velocity state $\hat{\mathbf{v}}_{bj^e}^e$ and $\hat{\mathbf{A}}_{bj^e}^e$ in a manner that preserves the total linear (\mathbf{p}_b^e) and angular (\mathbf{l}_b^e) momenta of the boundary element and that preserves the state of the first d resample points $\hat{\mathbf{x}}_{bke}^e = \mathbf{x}_{bke}^e$, $0 \leq k^e < d$ (those that coincide with the original boundary positions). Recall from Section 6 that the linear and angular momenta of the original points in the element are defined from the grid momentum distributions $\mathbf{p}_i^{e,i^e} = m_{bi^e}^e N_i(\mathbf{x}_{bi^e}^e) \left(\mathbf{v}_{bi^e}^e + \mathbf{A}_{bi^e}^e (\mathbf{x}_i - \mathbf{x}_{bi^e}^e) \right)$. As outlined in the supplementary material [Marquez et al. 2023], we can relate the total linear and angular momenta of this distribution to the APIC velocity state as $\mathbf{p}_b^e = \sum_{i^e=0}^{d-1} m_{bi^e}^e \mathbf{v}_{bi^e}^e$ and $\mathbf{l}_b^e = \sum_{i^e=0}^{d-1} \frac{\Delta x^2 m_{bi^e}^e}{4} \epsilon : \mathbf{A}_{bi^e}^e T + (\mathbf{x}_{bi^e}^e - \mathbf{x}_{bcom}^e) \times m_{bi^e}^e \mathbf{v}_{bi^e}^e$, where ϵ is the permutation tensor and the notation $\epsilon : \mathbf{A}$ denotes the vector \mathbf{b} with indices $b_\alpha = \epsilon_{\alpha\beta\gamma} A_{\beta\gamma}$ [Jiang et al. 2015]. We resample linear velocities $\hat{\mathbf{v}}_{bj^e}^e$ using linear interpolation: $\hat{\mathbf{v}}_{bj^e}^e = \sum_{i^e=0}^{d-1} \lambda_{jei^e}^e \mathbf{v}_{bi^e}^e$, where $\lambda_{jei^e}^e$ are the barycentric weights of the resampled positions $\hat{\mathbf{x}}_{bj^e}^e$ relative to $\mathbf{x}_{bi^e}^e$. This preserves the linear velocities of the original element points $\hat{\mathbf{v}}_{bke}^e = \mathbf{v}_{bke}^e$, $0 \leq k^e < d$ and preserves the total linear momentum. Here we use the linear reproduction property of the resampling interpolation functions $\hat{N}_{j^e}^e$ to equate the barycentric weights of the center of mass with the interpolated barycentric weights of the resample points

$$\lambda_{ie}^{e,com} = \sum_{j^e=d}^{N_r^e-1} \hat{N}_{j^e}^e(\mathbf{x}_{bcom}^e) \lambda_{jei^e}^e. \quad (5)$$

Also, the center of mass velocity is defined from the total linear momentum as $\mathbf{v}_{bcom}^e = \frac{1}{m_b^e} \mathbf{p}_b^e = \sum_{i^e=0}^{d-1} \frac{m_{bi^e}^e}{m_b^e} \mathbf{v}_{bi^e}^e = \sum_{i^e=0}^{d-1} \lambda_{ie}^{e,com} \mathbf{v}_{bi^e}^e$ since ratios of the element vertex masses $m_{bi^e}^e$ and the total element mass m^e are equal to the barycentric weights of the center of mass $\frac{m_{bi^e}^e}{m^e} = \lambda_{ie}^{e,com}$.

We define resampled affine velocities $\hat{\mathbf{A}}_{bj^e}^e$ using linear interpolation, but with a corrective angular velocity $\Delta\omega^e$ for resample points $d \leq j^e < N_r^e$ chosen to allow for perfect conservation of

angular momentum

$$\hat{\mathbf{A}}_{bk^e}^e = \mathbf{A}_{bk^e}^e, 0 \leq k^e < d, \quad (6)$$

$$\hat{\mathbf{A}}_{bj^e}^e = \sum_{i^e=0}^{d-1} \lambda_{j^e i^e}^e \mathbf{A}_{bi^e}^e + \boldsymbol{\epsilon} : \Delta \boldsymbol{\omega}^e, d \leq j^e < N_r^e. \quad (7)$$

Here the notation $\boldsymbol{\epsilon} : \Delta \boldsymbol{\omega}^e$ refers to the second order tensor \mathbf{B} with indices $B_{\alpha\beta} = \epsilon_{\alpha\beta\gamma} \omega_\gamma^e$. The corrective angular velocity $\Delta \boldsymbol{\omega}^e$ affects the total angular momentum of the resampled element as

$$\begin{aligned} \hat{\mathbf{l}}_b^e &= \mathbf{l}_b^e + (1-s) \sum_{j^e=d}^{N_r^e-1} \sum_{i^e=0}^{d-1} m_b^e \hat{N}_{j^e}^e(\mathbf{x}_{b\text{com}}^e) \left(\hat{\mathbf{x}}_{bj^e}^e - \mathbf{x}_{b\text{com}}^e \right) \times \lambda_{j^e i^e}^e \mathbf{v}_{bi^e}^e \\ &\quad - (1-s) \sum_{j^e=0}^{d-1} \left(\mathbf{x}_{bj^e}^e - \mathbf{x}_{b\text{com}}^e \right) \times m_{j^e}^e \mathbf{v}_{bj^e}^e - (1-s) m_b^e \frac{\Delta x^2}{2} \Delta \boldsymbol{\omega}^e. \end{aligned} \quad (8)$$

Note that this shows that linearly interpolating the affine velocities $\mathbf{A}_{bi^e}^e$ perfectly conserves the angular momentum in the element resulting from those terms, but that linearly interpolating the linear velocities $\mathbf{v}_{bi^e}^e$ does generally change the angular momentum. We choose the corrective angular velocity $\Delta \boldsymbol{\omega}^e$ to account for this difference:

$$\Delta \boldsymbol{\omega}^e = \frac{2}{\Delta x^2 (1-s) m^e} \Delta \mathbf{l}_b^e, \quad (9)$$

$$\begin{aligned} \Delta \mathbf{l}_b^e &= (1-s) \sum_{j^e=d}^{N_r^e-1} \sum_{i^e=0}^{d-1} m_b^e \hat{N}_{j^e}^e(\mathbf{x}_{b\text{com}}^e) \left(\hat{\mathbf{x}}_{bj^e}^e - \mathbf{x}_{b\text{com}}^e \right) \times \lambda_{j^e i^e}^e \mathbf{v}_{bi^e}^e \\ &\quad - (1-s) \sum_{j^e=0}^{d-1} \left(\mathbf{x}_{bj^e}^e - \mathbf{x}_{b\text{com}}^e \right) \times m_{bj^e}^e \mathbf{v}_{bj^e}^e. \end{aligned} \quad (10)$$

The resampled velocities and affine velocities across all boundary elements together form the output state $(\hat{\mathbf{v}}_b, \hat{\mathbf{A}})$ in an analogous manner to the input state $(\mathbf{v}_b, \mathbf{A})$.

6.2 Downsampling Affine Velocity: \mathcal{L}^{DS}

The output $\hat{\mathbf{m}}_g$ and $\hat{\mathbf{p}}_g$ of the P2G operator \mathcal{L}^{P2G} are comprised of masses \hat{m}_i and momenta $\hat{\mathbf{p}}_i$, respectively, over the grid nodes \mathbf{i} . The grid node velocities are updated after the transfers as $\hat{\mathbf{p}}_i$ divided by \hat{m}_i . Resampled particles of the element can be seen as having a grid mass distribution $\hat{m}_i^e = \sum_{j^e=d}^{N_r^e-1} N_i(\mathbf{x}_{bj^e}^e) \hat{m}_{bj^e}^e$. Multiplying the grid masses by their corresponding updated grid velocities will provide the resampled element linear momentum distribution $\hat{\mathbf{p}}_i^e = \hat{m}_i^e (\hat{\mathbf{p}}_i / \hat{m}_i)$. This has the effect of conservatively partitioning the grid momentum distribution into element-wise counterparts.

We design a conservative procedure for condensing $\hat{\mathbf{p}}_i^e$ back to affine velocity state $\mathbf{v}_{bi^e}^e, \mathbf{A}_{bi^e}^e$, $0 \leq i^e < d$ defined over the original element. Note that the resampled grid momentum distribution is defined over many grid nodes that do not (in general) affect the original unresampled element (see Figure 7). That is, $\hat{\mathbf{p}}_i^e$ is defined over grid nodes $\hat{\mathbf{x}}_i$ with $N_i(\hat{\mathbf{x}}_{bj^e}^e) \neq 0$ for some j^e with $0 \leq j^e < N_r^e$. To conservatively define the APIC state over the unresampled original element, we create a grid momentum distribution \mathbf{p}_i^e defined over grid nodes \mathbf{x}_i with $N_i(\mathbf{x}_{bi^e}^e) \neq 0$ for some i^e with $0 \leq i^e < d$ and then leverage the conservative nature of the APIC grid-to-particle (G2P) transfers from this grid momentum distribution to the affine state $\mathbf{v}_{bi^e}^e, \mathbf{A}_{bi^e}^e$, $0 \leq i^e < d$. We first compute the total

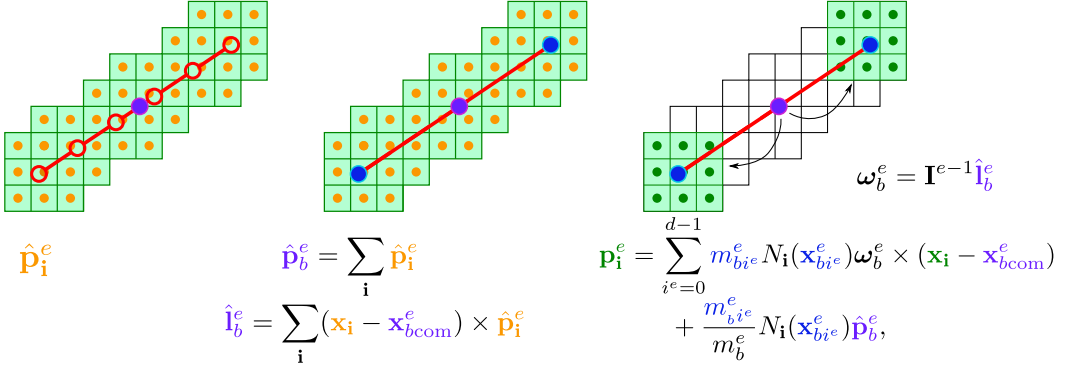


Fig. 7. (Left) The downsampling operator starts with the updated grid momenta $\hat{\mathbf{p}}_i^e$. (Middle) Linear and angular momentum of the resample points' grid distribution is merged to the element center of mass: $\hat{\mathbf{p}}_b^e$, $\hat{\mathbf{l}}_b^e$. (Right) Momenta is distributed to the original boundary element grid in a conservative manner: \mathbf{p}_i^e .

element linear and angular momenta from the grid distribution

$$\hat{\mathbf{p}}_b^e = \sum_i \hat{\mathbf{p}}_i^e, \quad \hat{\mathbf{l}}_b^e = \sum_i (\mathbf{x}_i - \mathbf{x}_{bcom}^e) \times \hat{\mathbf{p}}_i^e. \quad (11)$$

We add a portion of linear momentum $\hat{\mathbf{p}}_b^e$ to each grid node \mathbf{x}_i based on their associated interpolation function N_i and the mass ratio $\frac{m_{bi^e}^e}{m_b^e}$. We similarly distribute the angular momentum $\hat{\mathbf{l}}_b^e$ by computing the angular velocity ω_b^e associated with grid nodes \mathbf{x}_i^e assuming they have mass distribution $m_i^e = \sum_{i^e=0}^{d-1} m_{bi^e}^e N_i(\mathbf{x}_{bi^e}^e)$ and associated inertia tensor \mathbf{I}^e . We then add momenta to the grid nodes \mathbf{x}_i^e associated with an angular momentum state ω_b^e about the element center of mass

The total mass of each distribution $\sum_i m_i^e = \sum_i \sum_{i^e=0}^{d-1} m_{bi^e}^e N_i(\mathbf{x}_{bi^e}^e)$ and $\sum_i \hat{m}_i^e = \sum_i \sum_{j^e=0}^{N_f^e-1} \hat{m}_{bj^e}^e N_i(\hat{\mathbf{x}}_{bj^e}^e)$ is equal to the element mass m_b^e and their respective centers of mass are both equal to the element center of mass \mathbf{x}_{bcom}^e . Furthermore, the total linear momentum of the downsampled grid distribution \mathbf{p}_i^e is equal to $\hat{\mathbf{p}}_b^e$ and its total linear momentum is equal to $\hat{\mathbf{l}}_b^e$.

$$\sum_i \sum_{i^e=0}^{d-1} \mathbf{p}_i^{e,i^e} = \hat{\mathbf{p}}_b^e, \quad \sum_i \sum_{i^e=0}^{d-1} (\mathbf{x}_i - \mathbf{x}_{bcom}^e) \times \mathbf{p}_i^{e,i^e} = \hat{\mathbf{l}}_b^e. \quad (12)$$

The details of these calculations may be found in [Marquez et al. 2023]. The final output state $(\mathbf{m}_g, \mathbf{p}_g)$ thus consists of the grid masses $m_i = \sum_e m_i^e$ and momenta $\mathbf{p}_i = \sum_e \mathbf{p}_i^e$.

7 DISCRETE GEOMETRIC IMPULSES: \mathcal{L}^I

Although our augury operator \mathcal{L}^A prevents most collisions, in practice the iterative removal of cohesive/overly-frictional terms can degrade some of its ability to prevent the collision. In Figure 8, segment penetration can occur on forced collisions with large deformations that cause penetration through the tangential component if the timesteps are large. We, therefore, augment it with an iterative impulse-based post-process using conventional point/triangle and edge/edge pairs in the boundary mesh. We adopt the approach of Bridson et al. [2002] and iteratively apply these impulses based on proximity at the beginning of the time step and based on pairs determined to be colliding during the time step using continuous collision detection (CCD). Furthermore, we adopt their friction model since by design, our augury APIC operator \mathcal{L}^A is frictionless. Particles in the point/triangle and edge/edge pairs are assumed to take linear trajectories determined from locations

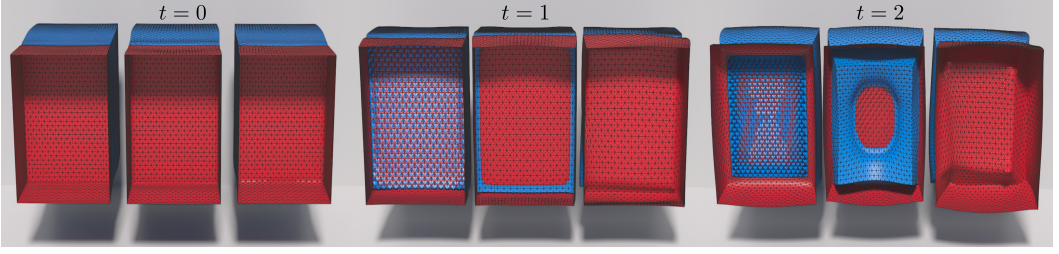


Fig. 8. (Left) Resample augury iteration that constrains tangential sliding has segment-segment penetration. (Middle) Resample augury that allows tangential sliding has penetration as the material slides tangentially around the corners and edges. (Right) \mathcal{L}^{RSA} resolves the frictionless collision with no penetration.

at the beginning of the time step and the linear part of their APIC velocity state. We use the approach of Brochu et al. [2012] for the CCD determinations. We consider this to be a process that operates on the linear part $\mathbf{v} \in \mathbb{R}^{dN^b}$ of the APIC velocity state. We use $\mathcal{L}^I : \mathbb{R}^{dN_b} \times \mathbb{R}^{N_b} \times \mathbb{N}^{N_b \times d} \rightarrow \mathbb{R}^{dN_b}$ to denote this process. For a detailed explanation of how impulses are calculated, as well as proof that they are linear and angular momentum conserving, we refer the readers to [Marquez et al. 2023].

8 RAYLEIGH DAMPING

Our Rayleigh damping model serves two purposes. First, it damps non-rigid modes so that backward Euler does not suffer from purely numerical damping that reduces with smaller time steps. This adds predictive control of the amount of damping in simulations, independent of time step size. Second, it increases the convexity of the backward Euler system allowing for more rapid convergence. Rayleigh damping for FEM discretization of hyperelasticity is analogous to adding damping to a one-dimensional spring where the damping coefficient is proportional to the stiffness of the spring [Belytschko et al. 2013]. However, in this case, the stiffness is given by the force Jacobian matrix $\frac{\partial \mathbf{f}^E}{\partial \mathbf{x}}$ evaluated at the previous time step. As such our Rayleigh damping model is of the form

$$\mathbf{f}^{RD}(\mathbf{x}^n, \mathbf{v}^{n+1}) = -\mathbf{K}^{RD}(\mathbf{x}^n) \mathbf{v}^{n+1}$$

where $\mathbf{K}^{RD}(\mathbf{x}^n)$ is proportional to a modified stiffness matrix. The stiffness matrix, although symmetric, will generally have negative, zero, and positive eigenvalues. In Teran et al. [2005], an element-wise definiteness fix was used to compute an always symmetric semi-definite counterpart to admit the use of conjugate gradients for the solution of the linear systems in quasistatic problems. However, this requires the solution of a 3X3 eigensystem and three 2X2 eigensystems. Although this cost is negligible since these can be computed rapidly and in parallel, a simpler and satisfactory strategy for the Rayleigh damping model is to compute element strain energy Hessians at the polar decomposition of the deformation gradient

$$\begin{aligned} \mathbf{K}_{i\alpha j\beta}^{RD}(\mathbf{x}^n) &= -a^{RD} \sum_{e \in \mathcal{M}} \sum_{\gamma=0}^{d-1} \sum_{\epsilon=0}^{d-1} \frac{\partial \mathbf{P}_{\alpha\gamma}}{\partial \mathbf{F}_{\beta\epsilon}}(\mathbf{R}^e(\mathbf{x}^n)) \frac{\partial N_i^L(\mathbf{X}^e)}{\partial \mathbf{X}_\gamma} \frac{\partial N_j^L(\mathbf{X}^e)}{\partial \mathbf{X}_\epsilon} V_e^0 \\ &\approx -a^{RD} \sum_{e \in \mathcal{M}} \sum_{\gamma=0}^{d-1} \sum_{\epsilon=0}^{d-1} \frac{\partial \mathbf{P}_{\alpha\gamma}}{\partial \mathbf{F}_{\beta\epsilon}}(\mathbf{F}^e(\mathbf{x}^n)) \frac{\partial N_i^L(\mathbf{X}^e)}{\partial \mathbf{X}_\gamma} \frac{\partial N_j^L(\mathbf{X}^e)}{\partial \mathbf{X}_\epsilon} V_e^0 \\ &= a^{RD} \frac{\partial \mathbf{f}_{i\alpha}^E}{\partial \mathbf{x}_{j\beta}}(\mathbf{x}^n). \end{aligned}$$

Here $a^{RD} \geq 0$ is the damping coefficient where $a^{RD} < 1$, $a^{RD} = 1$ and $a^{RD} > 1$ give behaviors analogous to underdamped, critically damped, and overdamped springs respectively. Note that in practice we found $a^{RD} = .1$ to give a qualitatively appropriate amount of damping in our examples. $\mathbf{F}^e = \mathbf{R}^e \mathbf{S}^e$ is the polar decomposition of the deformation gradient. We use the approximation $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{F}^e) \approx \frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{R}^e)$ because the matrix $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{R}^e)$ is positive semi-definite yet still rotationally consistent. The semi-definiteness of $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{R}^e)$ can be seen readily from the derivations in [Teran et al. 2005]. From Teran et al. [2005], we know that $\delta \mathbf{F} : \frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{R}^e) : \delta \mathbf{F} = (\mathbf{U}^{eT} \delta \mathbf{F} \mathbf{V}^e) : \frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{I}_d) : (\mathbf{U}^{eT} \delta \mathbf{F} \mathbf{V}^e)$ since $\mathbf{R}^e = \mathbf{U}^e \mathbf{V}^{eT}$, so it suffices to show the positive semi-definiteness of $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{I}_d)$ where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is the identity matrix. Also from Teran et al. [2005], we know that for isotropic materials, the matrix $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{I}_d)$ is a block diagonal matrix with diagonal components $\mathbf{A}, \mathbf{B}_{12}, \mathbf{B}_{13}, \mathbf{B}_{23}$ such that $\mathbf{A} = \begin{bmatrix} 2\mu + \lambda & \lambda & \lambda \\ \lambda & 2\mu + \lambda & \lambda \\ \lambda & \lambda & 2\mu + \lambda \end{bmatrix}, \mathbf{B}_{ij} = \begin{bmatrix} \mu & \mu \\ \mu & \mu \end{bmatrix}$ where λ and μ are Lamé parameters. Since each matrix on the diagonal is positive semi-definite, the matrix $\frac{\partial \mathbf{P}}{\partial \mathbf{F}}(\mathbf{R}^e)$ is positive semi-definite.

As shown in Equation (1), the Δt scaling before \mathbf{K}^{RD} is one order lower than that of the potential energy PE . So in practice the mass matrix \mathbf{M} and Rayleigh matrix \mathbf{K}^{RD} dominate the backward Euler Hessian $\mathbf{H}^{BE} = \mathbf{M} + \Delta t \mathbf{K}^{RD} + \Delta t^2 \frac{\partial \mathbf{f}^E}{\partial \mathbf{x}}$ relative to the stiffness matrix $\frac{\partial \mathbf{f}^E}{\partial \mathbf{x}}$. We found that this made the \mathbf{H}^{BE} symmetric positive definite in practice, without performing the definiteness fix on $\frac{\partial \mathbf{f}^E}{\partial \mathbf{x}}$. Note that without our symmetric semi-definite Rayleigh damping model, this is not the case, e.g. as was shown in McAdams et al. [2011].

9 EXAMPLES

9.1 Augury Iterations and Resampling

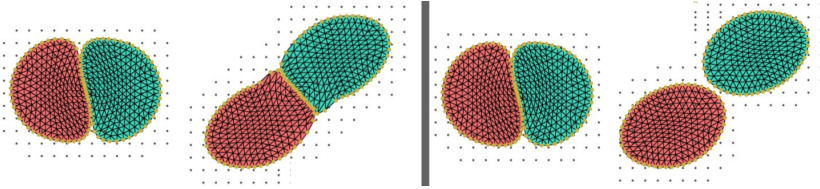


Fig. 9. (Left) Two circles collide using APIC transfers on a coarse grid where cohesion prevents separation. (Right) Using five augury iterations [Tupek et al. 2021] the cohesion is removed.

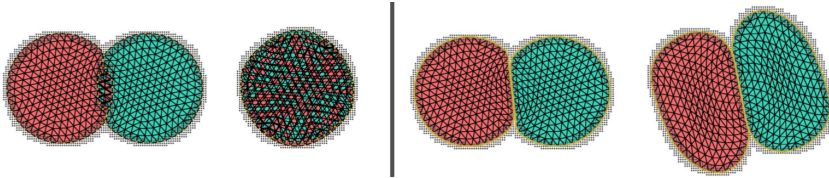


Fig. 10. (Left) On a finer background grid the APIC transfers with augury iterations [Tupek et al. 2021] miss the collision. (Right) The resampling method manages to capture the collision.

The collision of two circular meshes is simulated with only APIC collision responses (no \mathcal{L}^I impulses) to illustrate the grid-dependent collision resolution degradation and the numerical cohesion that arises without the augury iteration method from Tupek et al. [2021]. A coarse

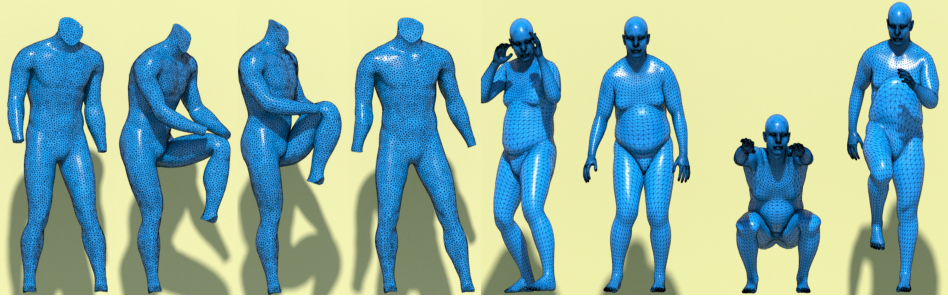


Fig. 11. Our method naturally resolves contact in flesh deformation simulations.

background grid will experience numerical cohesion from grid transfers as shown in the left simulation in Figure 9 where the circles stick together. The cohesion is removed from the coarse grid with only five augury iterations. This is shown in the right simulation in Figure 9 where the circles separate after colliding. In Figure 10, we refine the background grid to highlight the importance of the resampling/downsampling strategy. Note that in the left simulation, we do not use the resampling/downsampling operators only the approach of [Tupek et al. 2021] to highlight the degradation in the collision response that occurs. The degradation arises from the coarsely spaced surface nodes relative to the background grid. In the bottom simulation, our resampling/downsampling strategy increases the foreground particle density to be high-enough relative to the background grid to capture the collision and prevent the spheres from overlapping. Thus, the collision is resolved on finer grids than would otherwise be possible.

9.2 Two Blocks Collision

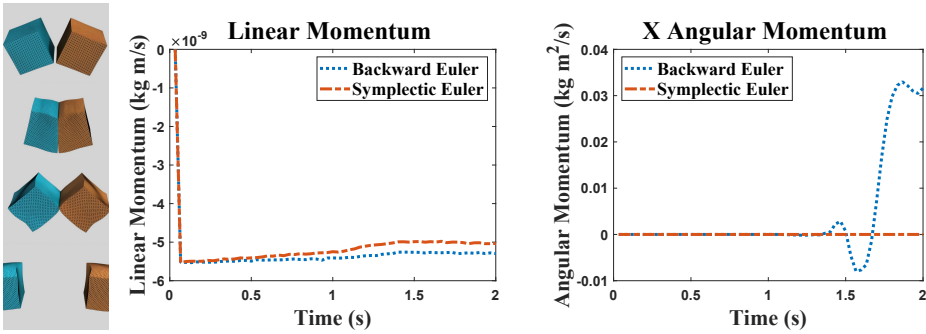


Fig. 12. (Left) Collision of two rotating blocks. (Right) Linear and angular momentum plots for symplectic and backward Euler timesteps.

In Figure 12 two rotating elastic blocks collide to demonstrate the linear and angular momentum conservation from the resample/downsampling operators. Linear momentum is conserved through the collision for both timestep methods, however, angular momentum changes only when using backward Euler due its lack of an angular momentum conservation property [Jiang et al. 2017b]. In Figure 8, two elastic blocks are compressed by a moving boundary condition. The limitations of the augury APIC method \mathcal{L}^A alone, without the geometric impulses \mathcal{L}^I are explored. The three simulations use: (left) \mathcal{L}^A alone with $NF = true$, (middle) \mathcal{L}^A alone with $NF = false$, and (right)

\mathcal{L}^A with $NF = true$ and \mathcal{L}^I . Without \mathcal{L}^I (left and middle), as compression increases, the corners and edges of the block penetrate the other block from the tangential direction. This is because the augury estimation of friction is less separated from the estimation of cohesion at regions of high curvature. The right simulation resolves the issue with the addition of \mathcal{L}^I .

As shown in Figure 13, our approach has the additional benefit of improving the convergence and overall run time of the impulse iterations \mathcal{L}^I . The initial frames of the collision only require a few iterations to converge as the collision occurs primarily in the normal direction, thus \mathcal{L}^A captured most of the collision. More \mathcal{L}^I impulse iterations are required when penetration in the tangential direction is allowed by \mathcal{L}^A , but significantly less than a method based purely on \mathcal{L}^I . The overall run time, including the overhead of \mathcal{L}^A is also reduced compared to \mathcal{L}^I alone.

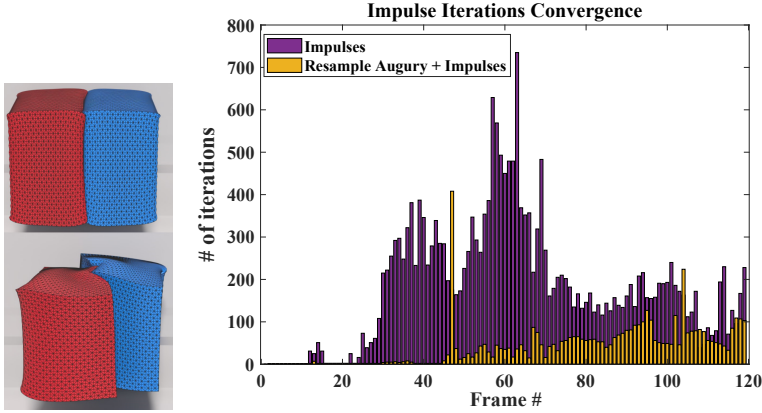


Fig. 13. (Left) Side view of two blocks being pushed into frictionless contact by elastic supports. (Right) Augury iteration reduces Bridson et al. [2002] impulse iterations. This decreases the run time from 5319s to 4680s.

9.3 Comparison with other methods

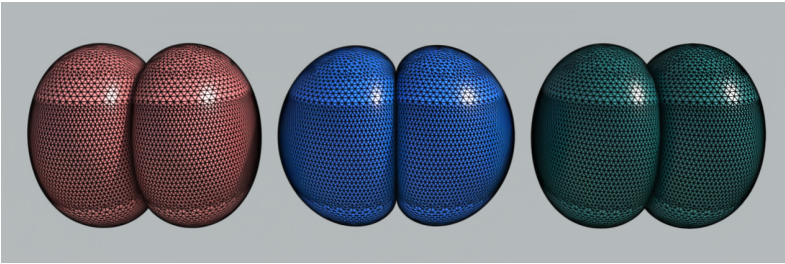


Fig. 14. Three methods are compared for two colliding spheres with 5K elements (Left) [Hegemann et al. 2013], (Mid) IPC, and (Right) ours. Each method resolves collision successfully.

We compare our approach with similar methods used in graphics applications. The incremental potential contact (IPC) approach of Li et al. [2020b] uses a barrier potential to perfectly resolve collisions and has been shown to handle many challenging contact problems with volumetric solids. Hegemann et al. [2013] use a similar PIC-style approach to resolving contact. We compare the

performance of our approach with both approaches using a colliding spheres example (see Figure 14) over multiple spatial resolutions and with the same time step sizes (see Table 2). The PIC-style approach of Hegemann et al. [2013] performs faster than our method as the PIC/FLIP transfers are only done once as opposed to multiple times from our approach, however, this method does not conserve linear or angular momentum once collisions occur (see Figure 15). IPC performs similarly to ours, however, it is more costly at higher spatial resolution than our approach. Unlike Hegemann et al. [2013], IPC and our approach both conserve linear momentum in all cases. However, backward Euler time stepping does not conserve angular momentum (see e.g. [Jiang et al. 2017b]). When explicit (symplectic) Euler is used with our approach, conservation of the angular momentum is achieved.

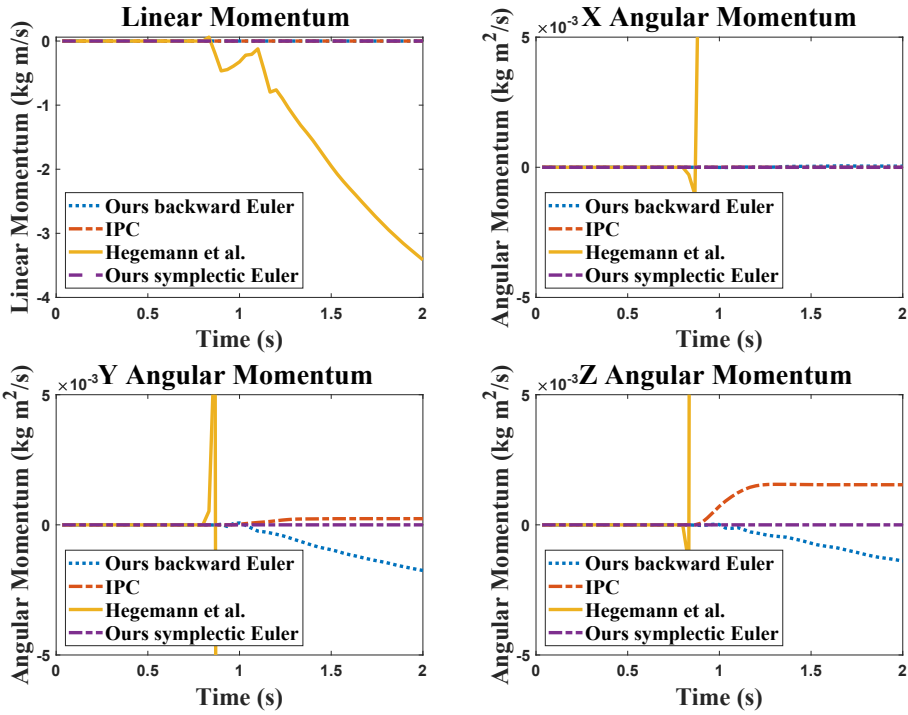


Fig. 15. We compare the linear and angular momentum conservation properties of our method using backward Euler (BE) and symplectic Euler with Li et al. [2020b] (BE) and Hegemann et al. [2013] (BE).

9.4 Twisting Legs

In Figure 1, a body mesh is put through large deformation by imposing rotational motion constraints on the leg and arms. The collision that the twisting motion generates has a significant tangential sliding component that needs to be captured without penetration. The convergence when using only \mathcal{L}^I impulse iterations compared to using our combined approach is shown in Figure 1. The iterations required to converge are significantly reduced with our combined approach, as the overall runtime is reduced from 21082s to 15510s.

9.5 Moving Body

Our algorithm can be used to simulate collisions on animated mesh models. The motion of the interior points in a body tetrahedral mesh is fully prescribed, forcing the surface mesh to collide and respond. In Figure 11, the collisions on the surface that occur from the motion of the interior points are captured accurately with our method.

9.6 Roller Examples

In Figure 3, an armadillo mesh experiences large deformations under compression by having gravity and friction drive it through rotating cylinders. The rollers have constrained velocities and positions for each time step, the roller masses are larger than the armadillo to drive the motion. Self-collision in the armadillo and collisions between the roller and the armadillo are robustly captured.

10 PERFORMANCE

Table 3 shows average per-timestep runtime details for several of our examples. For this table, all experiments were run on a desktop with an Intel Core i9-10920X 3.5 GHz LGA 2066 12-Core processor equipped with 32GB RAM.

11 DISCUSSION AND FUTURE WORK

Our hybrid particle/grid iteration improves the state-of-the-art in simulation of volumetric elastic contact. As shown in Section 9.3, our approach improves in conservation and speed over IPC [2020b] and Hegemann et al. [2013]. Furthermore, our approach drastically decreases the impulse iteration count required with the technique in Bridson et al. [2002] (see Figure 1). Note that our time step limitations are similar to Bridson et al. [2002], however, our APIC grid transfers require that the magnitude of position changes does not exceed one cell length of the background grid per time step. The APIC grid transfers were leveraged for their conservation of angular momentum, but other transfers with similar properties can be used with augury iterations. Note also that our conservative resampling strategy addresses grid-resolution restrictions on the FEM that would be present in Tupek et al. [2021] (see [Marquez et al. 2023]). In future works, the resampling and downsampling strategies similar to the one presented can be expanded to volumetric elements instead of contact surfaces. These techniques could be used in more general MPM and PIC techniques. Lastly, we mention that our use of MPM/APIC transfers can allow for the natural coupling of unstructured FEM techniques with general MPM simulations of various materials.

REFERENCES

- S. Andrews, K. Erleben, and Z. Ferguson. 2022. Contact and friction simulation for computer graphics. , 172 pages.
- M. Arroyo and M. Ortiz. 2006. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *Int J Num Meth Eng* 65, 13 (2006), 2167–2202. <https://doi.org/10.1002/nme.1534>
- U. Ascher, E. Larionov, S. Sheen, and D. Pai. 2021. Simulating deformable objects for computer animation: A numerical perspective. *Journal of Computational Dynamics* (2021).
- D. Baraff and A. Witkin. 1992. Dynamic Simulation of Non-Penetrating Flexible Bodies. In *Proc 19th Ann Conf Comp Graph Int Tech (SIGGRAPH '92)*. Association for Computing Machinery, New York, NY, USA, 303–308. <https://doi.org/10.1145/133994.134084>
- D. Baraff and A. Witkin. 1998. Large Steps in Cloth Simulation. In *Proc ACM SIGGRAPH (SIGGRAPH '98)*. 43–54.
- J. Barbič and D. James. 2007. Time-Critical Distributed Contact for 6-DoF Haptic Rendering of Adaptively Sampled Reduced Deformable Models. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '07)*. Eurographics Association, Goslar, DEU, 171–180.
- S. Bardenhagen, J. Brackbill, and D. Sulsky. 2000. The material-point method for granular materials. *Comp Meth App Mech Eng* 187, 3–4 (2000), 529–541.
- T. Belytschko, W. Liu, B. Moran, and K. Elkhodary. 2013. *Nonlinear finite elements for continua and structures*. John Wiley and sons.

- T. Belytschko and M. Neal. 1991. Contact-impact by the pinball algorithm with penalty and Lagrangian methods. *Internat. J. Numer. Methods Engrg.* 31, 3 (1991), 547–572.
- J. Bonet and R. Wood. 2008. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press.
- R. Bridson, R. Fedkiw, and J. Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans Graph* 21, 3 (2002), 594–603.
- T. Brochu, E. Edwards, and R. Bridson. 2012. Efficient Geometrically Exact Continuous Collision Detection. *ACM Trans Graph* 31, 4 (2012), 96:1–96:7.
- C. Brunel, P. Bénard, and G. Guennebaud. 2021. A Time-Independent Deformer for Elastic Contacts. *ACM Trans. Graph.* 40, 4, Article 159 (jul 2021), 14 pages. <https://doi.org/10.1145/3450626.3459879>
- L. Campos, J. Oden, and N. Kikuchi. 1982. A numerical analysis of a class of contact problems with friction in elastostatics. *Computer Methods in Applied Mechanics and Engineering* 34, 1 (1982), 821 – 845. [https://doi.org/10.1016/0045-7825\(82\)90090-1](https://doi.org/10.1016/0045-7825(82)90090-1)
- P. Cundall and O. Strack. 1979. A discrete numerical model for granular assemblies. *Geotech* 29, 1 (1979), 47–65.
- Gilles Daviet. 2020. Simple and scalable frictional contacts for thin nodal objects. *ACM Trans Graph* 39, 4 (2020), 547–572.
- Y. Fan, J. Litven, D. Levin, and D. Pai. 2013. Eulerian-on-Lagrangian Simulation. *ACM Trans Graph* 32, 3 (2013), 22:1–22:9.
- Y. Fei, Q. Guo, R. Wu, L. Huang, and M. Gao. 2021. Revisiting Integration in the Material Point Method: A Scheme for Easier Separation and Less Dissipation. *ACM Trans. Graph.* 40, 4, Article 109 (jul 2021), 16 pages. <https://doi.org/10.1145/3450626.3459678>
- T. Gast, C. Schroeder, A. Stomakhin, C. Jiang, and J. Teran. 2015. Optimization Integrator for Large Time Steps. *IEEE Trans Vis Comp Graph* 21, 10 (2015), 1103–1115.
- O. Gonzalez and A. Stuart. 2008. *A first course in continuum mechanics*. Cambridge University Press.
- X. Han, T. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran. 2019. A Hybrid Material Point Method for Frictional Contact with Diverse Materials. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (Jul 2019), 1–24.
- F. Harlow. 1964. The particle-in-cell method for numerical solution of problems in fluid dynamics. *Meth Comp Phys* 3 (1964), 319–343.
- D. Harmon, E. Vouga, B. Smith, R. Tamstorf, and E. Grinspun. 2009. Asynchronous Contact Mechanics. *ACM Trans Graph* 28, 3 (2009), 87:1–87:12.
- J. Hegemann, C. Jiang, C. Schroeder, and J. M. Teran. 2013. A Level Set Method for Ductile Fracture. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim*. 193–201.
- M. Homel and E. Herbold. 2017. Field-gradient partitioning for fracture and frictional contact in the material point method. *Int. J. Numer. Methods Eng.* 109 (2017), 1013–1044. Issue 7.
- P. Huang, X. Zhang, S. Ma, and X. Huang. 2011. Contact algorithms for the material point method in impact and penetration simulation. *Int J Num Meth Eng* 85, 4 (2011), 498–517. <https://doi.org/10.1002/nme.2981>
- T. Hughes. 2000. *The finite element method : linear static and dynamic finite element analysis*. Mineola, NY : Dover Publications.
- C. Jiang, T. Gast, and J. Teran. 2017a. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans Graph* 36, 4 (2017), 152.
- C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015. The Affine Particle-In-Cell Method. *ACM Trans Graph* 34, 4 (2015), 51:1–51:10.
- C. Jiang, C. Schroeder, and J. Teran. 2017b. An angular momentum conserving affine-particle-in-cell method. *J Comp Phys* 338 (2017), 137 – 164.
- N. Kikuchi and J. Oden. 1988. *Contact problems in elasticity: a study of variational inequalities and finite element methods*. SIAM.
- T. Kim and D. Elberle. 2022. Dynamic deformables: implementation and production practicalities. , 259 pages.
- G. Klar, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran. 2016. Drucker-Prager Elastoplasticity for Sand Animation. *ACM Trans Graph* 35, 4 (July 2016).
- E. Larionov, Y. Fan, and D. Pai. 2021. Frictional contact on smooth elastic solids. *ACM Transactions on Graphics* 40, 2 (2021), 1–17.
- D. Levin, J. Litven, G. Jones, S. Sueda, and D. Pai. 2011. Eulerian Solid Simulation with Contact. *ACM Trans Graph* 30, 4 (2011), 36:1–36:10.
- M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin, D. Panozzo, C. Jiang, and D. Kaufman. 2020a. Incremental potential contact. *ACM Transactions on Graphics* 39, 4 (2020).
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020b. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Trans. Graph. (SIGGRAPH)* 39, 4, Article 49 (2020).
- M. Li, D. Kaufman, and C. Jiang. 2021. Codimensional incremental potential contact. *ACM Transactions on Graphics* 40, 4 (2021), 1–24.
- A. Marquez, Y. Chen, H. Yushan, S. Gagniere, M. Tupek, and J. Teran. 2023. *Supplementary Technical Document* (2023).

- J. Martins and J. Oden. 1983. A numerical analysis of a class of problems in elastodynamics with friction. *Computer Methods in Applied Mechanics and Engineering* 40, 3 (1983), 327 – 360. [https://doi.org/10.1016/0045-7825\(83\)90105-6](https://doi.org/10.1016/0045-7825(83)90105-6)
- A. McAdams, A. Selle, K. Ward, E. Sifakis, and J. Teran. 2009. Detail Preserving Continuum Simulation of Straight Hair. *ACM Trans Graph* 28, 3 (2009), 62:1–62:6.
- A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans Graph* 30, 4 (2011), 37:1–37:12.
- A. Milne, M. McLaughlin, R. Tamstorf, A. Stomakhin, N. Burkard, M. Counsell, J. Canal, D. Komorowski, and E. Goldberg. 2016. Flesh, Flab, and Fascia Simulation on Zootopia. In *ACM SIGGRAPH 2016 Talks (SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 34, 2 pages. <https://doi.org/10.1145/2897839.2927390>
- M. Moore and J. Wilhelms. 1988. Collision Detection and Response for Computer Animation. In *Proc 15th Ann Conf Comp Graph Int Tech (SIGGRAPH '88)*. Association for Computing Machinery, 289–298. <https://doi.org/10.1145/54852.378528>
- G. Moutsanidis, D. Kamensky, D. Zhang, Y. Bazilevs, and C. Long. 2019. Modeling strong discontinuities in the material point method using a single velocity field. *Computer Methods in Applied Mechanics and Engineering* 345 (2019), 584–601.
- M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. 2007. Position based dynamics. *J Vis Comm Im Rep* 18, 2 (2007), 109–118.
- J. Nairn and Y. Guo. 2005. Material point method calculations with explicit cracks, fracture parameters, and crack propagation. *11th International Conference on Fracture 2005, ICF11 2* (01 2005).
- A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. 2006. Physically based deformable models in computer graphics. In *Comp Graph Forum*, Vol. 25. Wiley Online Library, 809–836.
- M. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. 2009. Implicit Contact Handling for Deformable Objects. *Comp Graph Forum* 28, 2 (2009).
- A. Popp, M. Gitterle, M. Gee, and W. Wall. 2010. A dual mortar approach for 3D finite deformation contact with consistent linearization. *Internat. J. Numer. Methods Engrg.* 83, 11 (2010), 1428–1465. <https://doi.org/10.1002/nme.2866> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2866>
- X. Provot. 1997. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation*. Springer, 177–189.
- M. Puso and T. Laursen. 2004. A mortar segment-to-segment frictional contact method for large deformations. *Computer Methods in Applied Mechanics and Engineering* 193, 45 (2004), 4891 – 4913. <https://doi.org/10.1016/j.cma.2004.06.001>
- E. Sifakis and J. Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses (SIGGRAPH '12)*. ACM, 20:1–20:50.
- J. Simo and T. Laursen. 1992. An augmented lagrangian treatment of contact problems involving friction. *Computers & Structures* 42, 1 (1992), 97 – 116. [https://doi.org/10.1016/0045-7949\(92\)90540-G](https://doi.org/10.1016/0045-7949(92)90540-G)
- B. Smith, F. De Goes, and T. Kim. 2018. Stable neo-hookean flesh simulation. *ACM Trans Graph (TOG)* 37, 2 (2018), 1–15.
- J. Spillmann, M. Becker, and M. Teschner. 2007. Non-iterative computation of contact forces for deformable objects. (2007).
- A. Stomakhin, R. Howes, C. Schroeder, and J. Teran. 2012. Energetically consistent invertible elasticity. In *Proc Symp Comp Anim.* 25–32.
- A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. 2013. A Material Point Method for snow simulation. *ACM Trans Graph* 32, 4 (2013), 102:1–102:10.
- D. Sulsky, Z. Chen, and H. Schreyer. 1994. A particle method for history-dependent materials. *Comp Meth App Mech Eng* 118, 1 (1994), 179–196.
- D. Sulsky, S. Zhou, and H. Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Comp Phys Comm* 87, 1 (1995), 236–252.
- Y. Teng, M. Otaduy, and T. Kim. 2014. Simulating Articulated Subspace Self-Contact. *ACM Trans. Graph.* 33, 4, Article 106 (jul 2014), 9 pages. <https://doi.org/10.1145/2601097.2601181>
- J. Teran, S. Blemker, V. Hing, and R. Fedkiw. 2003. Finite volume methods for the simulation of skeletal muscle. In *Proc 2003 ACM SIGGRAPH/Eurograph Symp Comp Anim.* Eurographics Association, 68–74.
- J. Teran, E. Sifakis, G. Irving, and R. Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proc 2005 ACM SIGGRAPH/Eurograph Symp Comp Anim.* 181–190.
- D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. 1987. Elastically Deformable Models. *SIGGRAPH Comput Graph* 21, 4 (1987), 205–214.
- F. Thomas and O. Johnston. 1995. *The illusion of life: Disney animation*. Hyperion New York.
- M. Tupek, J. Koester, and M. Mosby. 2021. A momentum preserving frictional contact algorithm based on affine particle-in-cell grid transfers. arXiv:[2108.02259](https://arxiv.org/abs/2108.02259) [cs.CE]
- P. Volino, M. Courchesne, and N. Magnenat-Thalmann. 1995. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proc 22nd Conference Comp Graph Int Tech (SIGGRAPH '95)*. ACM, 137–144.
- P. Wriggers and T. Laursen. 2006. *Computational contact mechanics*. Vol. 2. Springer.

- P. Wriggers, T. Vu Van, and E. Stein. 1990. Finite element formulation of large deformation impact-contact problems with friction. *Computers & Structures* 37, 3 (1990), 319 – 331. [https://doi.org/10.1016/0045-7949\(90\)90324-U](https://doi.org/10.1016/0045-7949(90)90324-U)
- L. Wu, B. Wu, Y. Yang, and H. Wang. 2020. A Safe and Fast Repulsion Method for GPU-Based Cloth Self Collisions. *ACM Trans. Graph.* 40, 1, Article 5 (dec 2020), 18 pages. <https://doi.org/10.1145/3430025>
- M. Xiao, C. Liu, and W. Sun. 2021. DP-MPM: Domain partitioning material point method for evolving multi-body thermal mechanical contacts during dynamic fracture and fragmentation. *Computer Methods in Applied Mechanics and Engineering* 385 (2021).
- Y. Yue, B. Smith, P. Chen, M. Chantharayukhonthorn, K. Kamrin, and E. Grinspun. 2018. Hybrid grains: adaptive coupling of discrete and continuum simulations of granular media. *ACM Trans Graph* 37, 6 (2018), 283:1–283:19.

Table 1. Summary of Notation.

Symbol	Definition
Ψ	Hyperelastic energy density function
\mathbf{F}^e	Element deformation gradient
$\mathbf{f}_p^E(\mathbf{x})$	Elastic force on mesh node p
$\frac{\partial N_p^L(\mathbf{x}^e)}{\partial \mathbf{x}}$	Linear interpolating functions of FEM mesh
m_p	Mass of mesh node p
$ \mathcal{I}_b^p $	Number of elements in the one ring (in boundary mesh) of node p
\mathbf{x}^n	Mesh node positions at time t^n
\mathbf{v}^n	Mesh node velocities at time t^n
\mathbf{M}	Lumped-mass diagonal matrix
\mathbf{K}^{RD}	Rayleigh damping matrix
$\mathbf{B}_b \mathbf{m}$	Boundary mesh nodal masses
$\mathbf{B}_b \mathbf{v}$	Boundary mesh nodal velocities
\mathbf{A}	APIC linear velocity state
\mathcal{L}^A	Augury APIC impulse operator
\mathcal{L}^I	Bridson et al. [2002] impulse operator
\mathcal{L}^{RSA}	Resample augury APIC impulse operator
\mathcal{L}^{G2P}	APIC grid to particle operator
\mathcal{L}^{DS}	Downsampling operator
\mathcal{L}^{P2G}	APIC particle to grid operator
\mathcal{L}^{RS}	Resample operator
$m_{b i^e}^e$	Boundary element nodal mass of element node i^e
$\mathbf{x}_{b i^e}^e$	Boundary element nodal position of element node i^e
$\mathbf{v}_{b i^e}^e$	Boundary element nodal velocity of element node i^e
$\mathbf{A}_{b i^e}^e$	Boundary element nodal linear velocity of element node i^e
$\hat{m}_{b j^e}^e$	Resampled boundary element nodal mass of resampled element node j^e
$\hat{\mathbf{x}}_{b j^e}^e$	Resampled boundary element nodal position of resampled element node j^e
$\hat{\mathbf{v}}_{b j^e}^e$	Resampled boundary element nodal velocity of resampled element node j^e
$\hat{\mathbf{A}}_{b j^e}^e$	Resampled boundary element nodal linear velocity of resampled element node j^e
\hat{m}_i	Resampled background grid mass for grid node i
$\hat{\mathbf{x}}_i$	Resampled element background grid position for grid node i
$\hat{\mathbf{p}}_i$	Resampled element background grid momentum for grid node i
$\mathbf{x}_{b \text{com}}^e$	Boundary element center of mass
N_i	Background grid interpolating function
\mathbf{p}_b^e	Boundary element linear momentum
\mathbf{l}_b^e	Boundary element angular momentum
s	Mass scaling coefficient
\hat{N}_{i^e}	Arroyo and Ortiz [2006] local maximum entropy interpolating functions
$\lambda_{j^e i^e}^e$	Barycentric weights of the resampled positions j^e with respect to original positions i^e
$\lambda_{i^e}^{e, \text{com}}$	Barycentric weights of the boundary element center of mass
$\Delta \omega^e$	Corrective angular velocity
\mathbf{I}^e	Element inertia tensor
a^{RD}	Damping coefficient
\mathbf{H}^{BE}	Backward Euler Hessian

Table 2. Performance measurements.

Example	E, ρ, ν	# nodes	Δt	run time (s)	avg t per frame (s)
1K Sphere BE IPC	50K, 1000, 0.3	3.5k	0.005	125	2.0833
1K Sphere BE [Hegemann et al. 2013]	50K, 1000, 0.3	3.5k	0.005	180	3
1K Sphere BE Our Method	50K, 1000, 0.3	3.5k	0.005	240	4
5K Sphere BE IPC	50K, 1000, 0.3	14k	0.005	993	16.55
5K Sphere BE [Hegemann et al. 2013]	50K, 1000, 0.3	14k	0.005	720	12
5K Sphere BE Our Method	50K, 1000, 0.3	14k	0.005	960	16
19K Sphere BE IPC	50K, 1000, 0.3	57K	0.005	11,420	190.3333333
19K Sphere BE [Hegemann et al. 2013]	50K, 1000, 0.3	57k	0.005	7,620	127
19K Sphere BE Our Method	50K, 1000, 0.3	57K	0.005	8,400	140

Table 3. Performance measurements.

Example (Figure)	E, ρ, ν	# nodes	average Δt	avg t substep (s)
Bunnies ([Marquez et al. 2023])	750, 100, 0.3	210k	0.005	5.3
Blocks (13)	1000, 100, 0.3	209k	0.005	3.6
Twisting legs (1)	7000, 1000, 0.4	145k	0.002	6.36
Moving body (11)	5000, 25, 0.3	50k	0.003	6.45
Armadillo (3)	5000, 1000, 0.3	68k	0.002	4.5