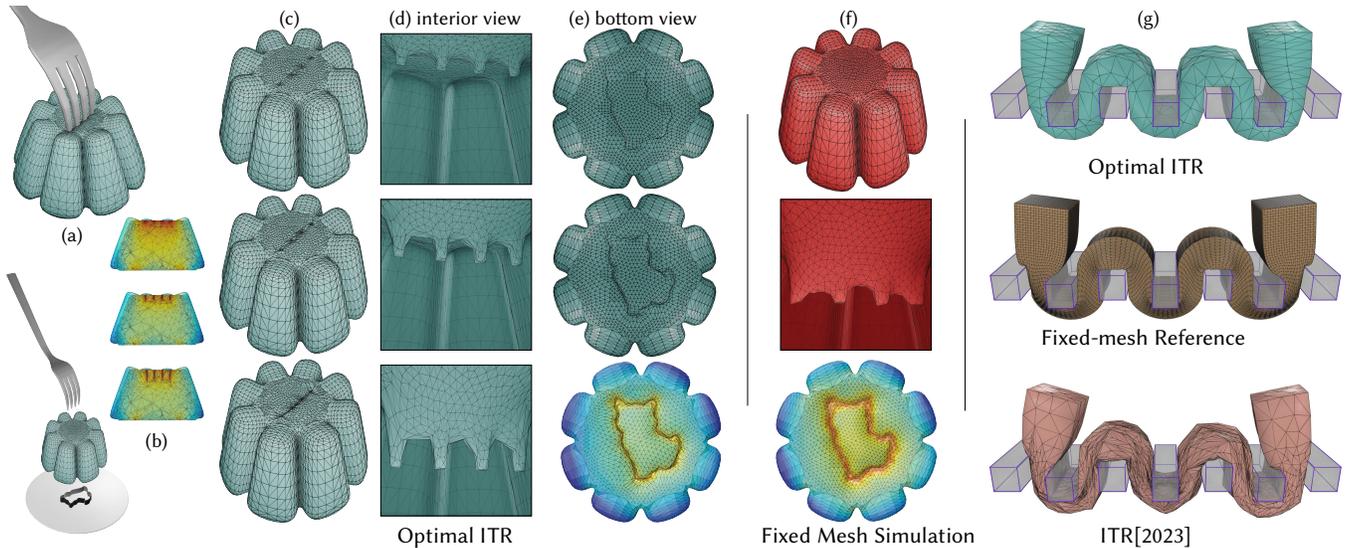


# Optimal r-Adaptive In-Timestep Remeshing for Elastodynamics

JIAHAO WEN, Adobe, USA and University of Southern California, USA  
 JERNEJ BARBIČ, University of Southern California, USA  
 DANNY M. KAUFMAN, Adobe, USA



**Fig. 1. Poke and Chew with Optimal r-Adaptive In-Timestep Remeshing:** In (a) we stick a stiff jelly treat, lying on a jello mold, with a metal fork. Optimal In-Timestep Remeshing (ITR) jointly adapts the reference mesh, (b-c), and solves the deformation, (d-e), over the simulation (top to bottom) to best model the squashing jelly’s physical solution at each timestep. In contrast, simulation with the original, unadapted mesh in (f), is unable to capture the sharp conforming contact of the fork tines, nor the creasing and tight stress concentrations generated by the pressing mold. In (g), masticating a rubber bar, our optimal ITR solution, top, generates an adapted coarse physical model that closely follows (Hausdorff distance 0.05) a corresponding fine-mesh solution, middle, while using orders of magnitude less DOF and memory (with a resultant 8.4X speedup) than the fine-mesh simulation. In contrast, bottom, Ferguson et al.’s [2023] original ITR model (with 8.5X more DOF and 73X slower than optimal ITR) diverges significantly (Hausdorff distance 0.16) from the fine model’s deformation behavior; see Figure 3 and our supplemental video for more details.

We propose a coupled mesh-adaptation model and physical simulation algorithm to jointly generate, per timestep, optimal adaptive remeshings and implicit solutions for the simulation of frictionally contacting elastodynamics. To do so, we begin with Ferguson et al.’s [2023] recently developed in-timestep remeshing (ITR) framework, which proposes an Incremental Potential based objective for mesh refinement, and a corresponding, locally greedy remeshing algorithm to minimize it. While this initial ITR framework demonstrates significant improvements, its greedy remeshing does not generate optimal meshes, and so does not converge to improving physical solutions with increasing mesh resolution. In practice, due to lack of optimality, the original ITR framework can and will find mesh and state solutions

with unnecessarily low-quality geometries and corresponding physical solution artifacts. At the same time, we also identify additional fundamental challenges to adaptive simulation in terms of both ITR’s original remeshing objective and its corresponding optimization problem formulation.

In this work, in order to extend the ITR framework to high-quality, optimal in-timestep remeshing, we first construct a new remeshing objective function built from simple, yet critical, updates to the Incremental Potential energy, and a corresponding *constrained* model problem, whose minimizers provide locally optimal remeshings for physical problems. We then propose a new in-timestep remeshing optimization that jointly solves, per-timestep, for a new locally optimal remeshing and the next physical state defined upon it.

To evaluate and demonstrate our extension of the ITR framework, we apply it to the optimal r-adaptive ITR simulation of frictionally contacting elasto-dynamics and statics. To enable r-adaptivity we additionally propose a new numerical method to robustly compute derivatives of the  $L^2$ -projection operator necessary for optimal mesh-to-mesh state mappings within solves, a constraint model to enable on-boundary node adaptivity, and an efficient Newton-type optimization method for practically solving each per-timestep r-adaptive ITR solution. We extensively evaluate our method on challenging large-deformation and frictionally contacting scenarios. Here we observe optimal r-adaptivity captures comparable and better accuracy than unadapted meshes orders-of-magnitude larger, with corresponding significant advantages in both computation speedup and decrease in memory usage.

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

Authors’ Contact Information: Jiahao Wen, Adobe, USA and University of Southern California, USA, [jiahaow@usc.edu](mailto:jiahaow@usc.edu); Jernej Barbič, University of Southern California, USA, [jnb@usc.edu](mailto:jnb@usc.edu); Danny M. Kaufman, Adobe, USA, [dannykaufman@gmail.com](mailto:dannykaufman@gmail.com).

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).

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
 ACM 1557-7368/2025/8-ART  
<https://doi.org/10.1145/3731204>

Additional Key Words and Phrases: Adaptive Meshing, Elastodynamics, Contact, Friction, r-Adaptivity

### ACM Reference Format:

Jiahao Wen, Jernej Barbič, and Danny M. Kaufman. 2025. Optimal r-Adaptive In-Timestep Remeshing for Elastodynamics. *ACM Trans. Graph.* 44, 4 (August 2025), 19 pages. <https://doi.org/10.1145/3731204>

## 1 Introduction

We propose a new coupled mesh-adaptation model and physical simulation algorithm to jointly evolve, per timestep, locally *optimal* adaptive remeshings and implicit solutions for the high-quality simulation of frictionally contacting elastodynamics and statics.

Large-deformation elastodynamic simulations generally require highly dense spatial discretizations to capture the critical and often transient features, such as shockwaves, localized deformations, contact compliances, and stress concentrations, that characterize these systems. At the same time, meshes dense enough to capture these behaviors are often prohibitively expensive in both runtime and memory for practical application. While practitioners can potentially hand-craft meshes suited for particular instants in time, no single mesh can generally be optimal for all timesteps of a simulation.

Instead, in order to parsimoniously capture evolving transients in dynamic systems, we focus on simulation meshes that correspondingly evolve via adaptive remeshing. Our goal is to adaptively concentrate limited simulation degrees of freedom (DOF) resources when and where they are most needed.

To do so, we begin with Ferguson et al.'s [2023] recently developed In-Timestep Remeshing (ITR) framework, which proposes an Incremental Potential (IP) [Li et al. 2020] based objective for mesh refinement, and a corresponding, locally greedy remeshing algorithm to minimize it within each timestep.

This first-proposed ITR objective thus directly considers physical criteria, rather than solely geometric or sizing field terms, in remeshing, and so demonstrates corresponding significant improvements in better capturing physical behaviors across a range of challenging scenarios. However, its greedy minimization algorithm does not generate optimal meshes, and so does not converge to improving physical solutions with increasing mesh resolution [Ferguson et al. 2023]. In practice, due to this lack of optimality, we observe that the original ITR framework can and will find mesh and state solutions with unnecessarily low-quality geometries and so corresponding physical solution artifacts (see e.g., Figure 1). At the same time, independent of the greedy minimization algorithm applied, we identify fundamental challenges to high-quality adaptive simulation in terms of the original ITR method's remeshing objective and its corresponding optimization problem formulation for modeling both dynamic and static elastica.

In this work, in order to extend the ITR framework to high-quality, optimal in-timestep remeshing, we revisit the construction of ITR with a focus on (1) the remeshing objective, (2) physical and geometric admissibility constraints, and (3) the underlying optimization problem itself.

**Objective.** We first construct a new remeshing objective function built from a set of simple, yet critical, updates to the Incremental

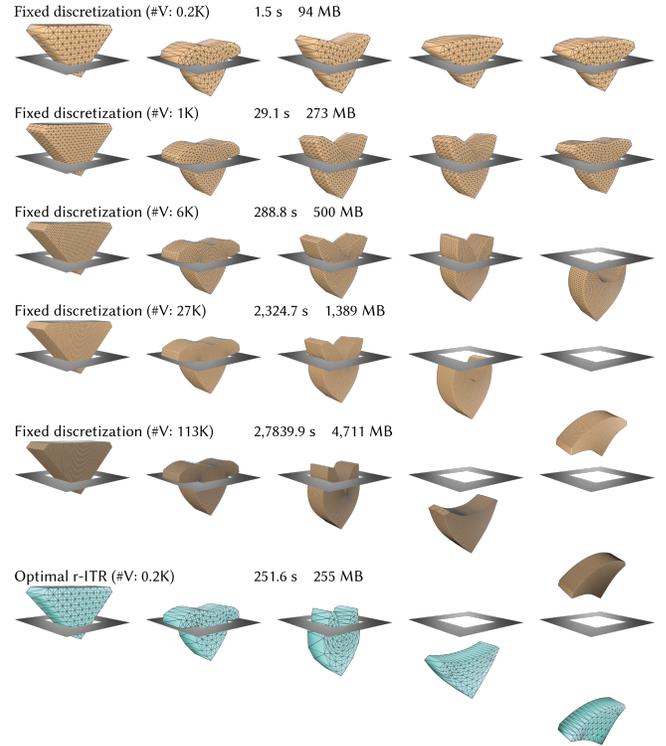


Fig. 2. **Dynamic drop through.** We drop an elastic wedge under gravity onto a tight, frictionless gap. Coarser, fixed-discretization based simulations are unable to pass the gap, while finer-mesh fixed-discretizations progressively avoid locking with more DOF. Bottom: Coarsest-resolution optimal r-ITR closely matches the dynamics of the corresponding finest 113K vertex fixed-mesh simulation. Please also see Figure 6 in our supplemental for trajectory overlays.

Potential objective energy. These changes, as we demonstrate in Section 4, avoid degenerate solutions in the original ITR objective, and enable ITR optimization to reach high-quality meshes *and* correspondingly accurate timestep solutions upon them.

**Admissibility and Optimization.** High-quality elastodynamics requires accurate resolution of contact and elasticity. Applying the recently proposed, Incremental Potential Contact (IPC) model [Li et al. 2020] for modeling frictionally contacting solids, we follow Ferguson et al. [2023] and ensure that invariants of non-intersection, and non-inverting deformation are preserved. We further construct admissibility criteria for ITR optimization (for dynamics *and* statics) in terms of both mesh geometry *and* physical model, to formulate a new *constrained* model problem for optimizing our IPC-based objective, whose *admissible* minimizers provide *locally optimal* remeshings for simulating elastica. We then propose a new in-timestep remeshing optimization that jointly solves our model problem, per-timestep, for each new locally optimal remeshing and the next physical state defined upon it.

**Application to r-Adaptivity.** While the above new ITR objective, constraints and optimization problem are general, numerically solving ITR optimization problems to find local minima is of course

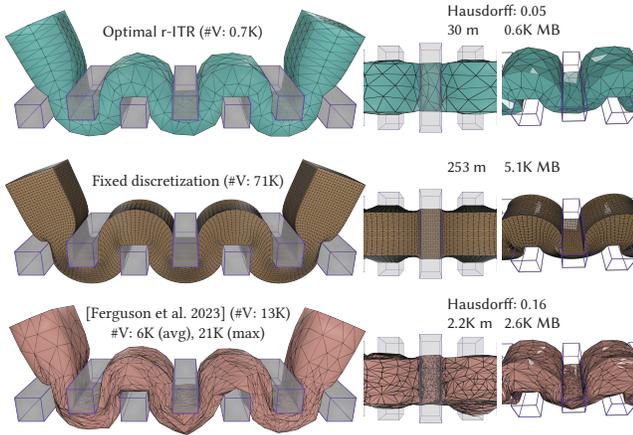


Fig. 3. **Masticating: ITR benchmark comparison.** We compare optimal ITR and Ferguson et al.’s [2023] original ITR model, with a finer-resolution fixed-mesh simulation baseline, evaluating timing, memory usage and Hausdorff distance to baseline.

challenging. In this work, to evaluate and demonstrate our extension of the ITR framework, we focus on applying Newton-type solvers to build the necessary algorithmic contributions to enable the numerical solution of our ITR optimization problem for the optimal *r-adaptive* (optimizing mesh geometry) simulation of frictionally contacting dynamic and static elastica. We leave optimal *h-adaptive* (optimizing mesh connectivity) ITR simulation, via non-smooth optimization or differentiable mesh connectivity for future extension.

**Contributions.** Computing optimal *r-adaptive* ITR simulations then further requires additional new numerical methods to carefully resolve and evaluate changing state mappings and on-boundary adaptivity, in-optimization, along with a carefully constructed Newton-type minimization algorithm to solve optimal ITR mesh and deformation evolution across time steps. In summary, our technical contributions include

- A new ITR objective, constraint model and optimization model problem, that give locally optimal timestep solutions for in-time-step remeshing simulations of frictionally contacting elastodynamics and elastostatics.
- A new method to robustly compute derivatives of the  $L^2$ -projection operator necessary for optimal mesh-to-mesh state mappings;
- A boundary maintenance constraint model to enable on-boundary node adaptivity; and
- A new truncated Newton-type optimization method for efficiently solving accurate, per-timestep, optimal *r-adaptive* ITR solutions.

Taken together this gives high-quality, *r-adaptive* ITR simulation with user-controllable accuracy in terms of solution optimality. This complements prior work in *h-adaptive* ITR and significantly improves the ITR model formulation. We extensively evaluate our method on challenging large-deformation and frictionally contacting scenarios. Here we observe optimal ITR *r-adaptivity* captures comparable and better accuracy than both the original ITR formulation, and unadapted meshes orders-of-magnitude larger, with the

corresponding significant advantages in both computation speedup and decrease in memory usage.

## 2 Related Work

We focus on the high-quality simulation of frictionally contacting elastodynamics, modeled by finite elements and solved by implicit numerical time integration. Improving simulation accuracy, and hence quality, requires increasing mesh resolution. However, increasing mesh resolution of course entails significant computational cost. This natural tension, between simulation quality and mesh resolution cost, has motivated extensive exploration of adaptive remeshing methods that seek to allocate mesh degree-of-freedom (DOF) resources where they are most needed [Manteaux et al. 2017].

### 2.1 Adaptivity Objectives

A wide range of remeshing and basis update operations have been applied and considered for adaptivity. We review these below. However, irrespective of which mesh-adapting operations are considered, a fundamental question remains on how to decide when and where to apply these operations. For modeling both statics and dynamics, no single mesh is generally a priori best-suited, without already having taken on the expense of computing the solution. In turn, when it comes to elastodynamics, with rapidly evolving transients, many different meshes may be needed across different phases of a modeled trajectory. Motivated by these challenges, a wide range of objectives, criteria and heuristics have been developed to guide adaptive remeshing.

Geometry derived information is a common guidepost for constructing adaptivity objectives. Constructed from either base reference meshing [Bargteil et al. 2007], or additionally deformed mesh state [Dunyach et al. 2013], evaluations of strains and stresses from mesh geometry can be applied to determine remeshing criteria [Bargteil et al. 2007; Debonne et al. 2001; Spillmann and Teschner 2008; Wicke et al. 2010]. For thin shells, additional in-plane stretch and out-of-plane bending measures are similarly processed [Li and Volkov 2005; Narain et al. 2013, 2012; Simnett et al. 2009; Villard and Borouchaki 2005].

Optimal Delaunay Triangulations and related methods [Alliez et al. 2005; Chen and Xu 2004; Du and Wang 2003] find optimal remeshings that minimize *geometric* objectives. While meshes generated by these geometric objectives can provide optimally well-shaped elements, they are oblivious to any physical quantities defined upon the meshes, that may be solved for during simulation. Thus, they do not account for the spatially varying and possibly anisotropic [Shewchuk 2002] accuracy needs (e.g., in capturing shockwaves, stress concentrations, and impacts) that solving static and dynamic *physical* problems impose. In particular, in order to parsimoniously capture evolving transients in dynamic systems, meshes should correspondingly evolve their adaptation. Generally, no single mesh can be optimal for all time steps of a simulation, and correspondingly no single geometric objective can define optimal meshing criteria for the solution of a physical problem.

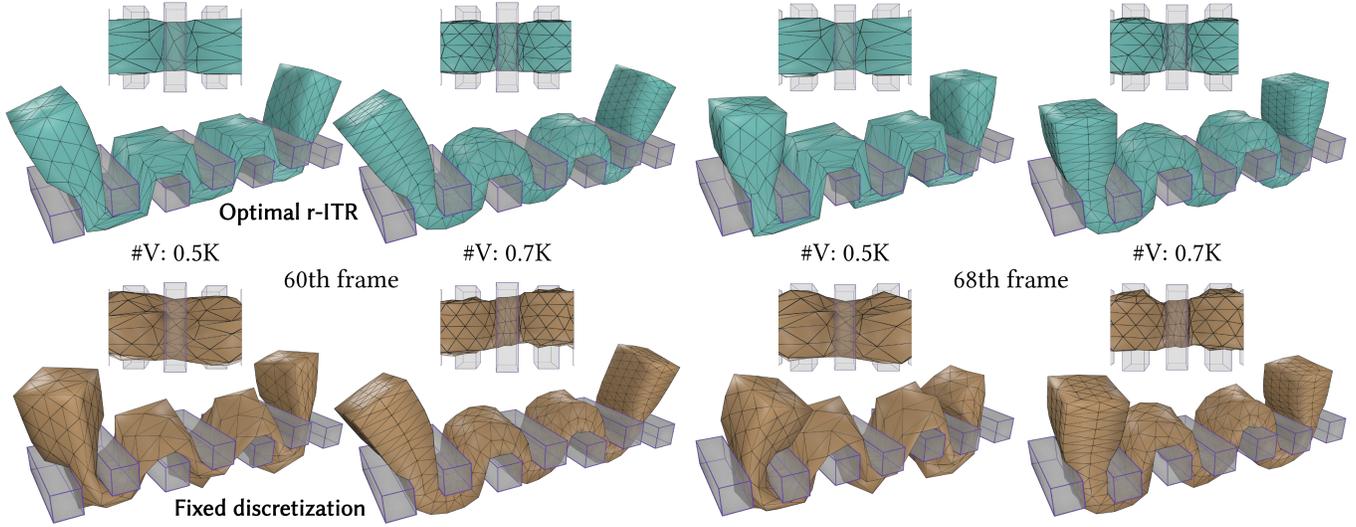


Fig. 4. **Masticating: fixed-discretization comparison.** We compare optimal ITR with corresponding fixed-discretizations at two coarse resolutions. As we coarsen optimal ITR’s available resolution, its ability to capture curvature and details reduces, while overall simulation behavior remains close to the finer-resolution optimal ITR model. In contrast, corresponding fixed-resolution mesh discretizations (please see our supplemental videos for animations) each exhibit different simulation artifacts, and so likewise different overall deformation behaviors.

Along with the above measures, surface-to-surface evaluations have been considered and applied to drive mesh adaptation in contact processing. For contact-based forces, collision-detected distances [Bender and Deul 2013; Erhart et al. 2006; Simnett et al. 2009], enriched by continuity criteria [Spillmann and Teschner 2008], and curvature measures [Li et al. 2018; Narain et al. 2013, 2012; Pfaff et al. 2014] are applied to determine whether mesh boundaries close-together are likely to require refinement for detailed contact resolution. As carefully analyzed by Ferguson et al. [2023], while such measures can be necessary criteria for adaptation, they overlook physical properties, e.g. whether a material is soft enough to conform to a contact, and so aggressively over-adapt in many scenarios, while missing opportunities for adaptation in others.

In concert with the above geometric and physical quantities, Eulerian on Lagrangian (EoL) methods [Sánchez-Banderas et al. 2020; Sueda et al. 2011; Weidner et al. 2018], insert and drive “Eulerian” reference coordinate locations to capture sharp contact transitions. With adaptivity driven by referential coordinate dynamics, EoL methods can be challenged by instabilities [Wen et al. 2020] inherent in the formulation. Closely related to EoL methods is Zielonka et al.’s [2008] variational r-adaptive method, which likewise formulates dynamic equations of motion for both deformation and referential coordinates (via Eshelbian-type balance equations [Braun 1997]). Adaptivity is then driven by simulation of reference coordinate dynamics with similar stability challenges to EoL.

Most closely related to our approach are methods that consider physical energy measures, including elasticity [Demkowicz 2006; Mitchell and McClain 2014], as remeshing objectives. Mosler and Ortiz introduce the *optimization* of the Incremental Potential energy to drive mesh-adaptivity for contact-free elastostatics and plasticity with r-adaptive [2006] and h-adaptive [2007] remeshing (see the next section for descriptions of h- and r-adaptivity). Ferguson et

al. [2023] extend this strategy to contacting elastodynamics via in-timestep remeshing (ITR). ITR proposes using the timestep-based Incremental Potential [Li et al. 2020] as a remeshing objective, with a greedy descent-based optimization for combined h-adaptivity and mesh-relaxation.

Both Mosler and Ortiz [2006; 2007] and Ferguson et al. [2023] propose *unconstrained* minimizations of the Incremental Potential. As we cover in Section 4.4 and in our supplemental, this unconstrained minimization can and will lead to solutions with nonphysical artifacts and degenerate meshes (see e.g., Figure 8). Interestingly, we note that Mosler and Ortiz [2006] report convergence failures in their remeshing solves (presumably due to the degeneracies inherent in the underlying optimization problems). They propose addressing these failures by applying an iteratively reweighted damped Newton solver. As this does not change the underlying optimization problem, this damping only decreases the solver’s progress towards, but does not prevent, simulation artifacts; see e.g., Figure 1 in the supplemental.

In this work, we extend the ITR framework to high-quality, optimal in-timestep remeshing, by constructing a new remeshing objective function built from critical updates to Ferguson et al.’s energy, a constrained optimization model problem whose minimizers provide locally optimal remeshings for simulating elastica, and a corresponding optimization method that jointly solves for the optimal remeshing and the physical state defined upon it.

## 2.2 Mesh-Adaptive Operations

With the above objectives in mind, a wide range of meshing operations can be applied. In this work we focus on the application of *r-adaptive* or “moving-mesh” operations [Budd et al. 2009; Huang and Russell 2010], which adapt mesh geometry, leaving mesh connectivity and resolution unchanged. When carefully treated, see Sections 4 and 5 below, r-adaptive operations generally offer smooth

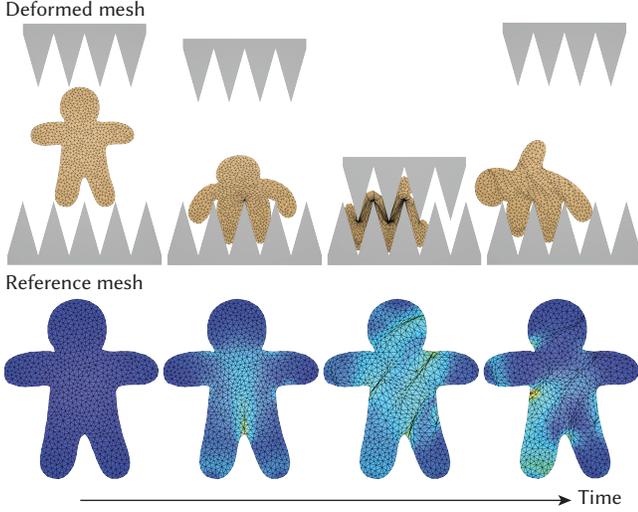


Fig. 5. **Gingerbread Squish.** We stress out an elastic “gingerbread” by tightly squishing it between compressing sharp teeth. Here we visualize both the deformation and the reference domain meshing and strain (ranging from red, high strain, to blue, low strain). Adaptivity initially focuses DOF to localized contacts. As compression increases, adaptivity further concentrates DOF along internal stress bands. As the teeth release, adapted meshing regions relax, aside from persistent contact regions.

problem updates and so enable us to construct Newton-type optimization methods to solve our ITR timesteps. In contrast, *h-adaptive* operations adapt mesh resolution and topology by insertion, deletion, and topology updates [Hu et al. 2018]. Across codimensions, ranging from the simulation of rods [Spillmann and Teschner 2008], shells [Li et al. 2018; Narain et al. 2013, 2012], and surfaces [Brochu and Bridson 2009], to volumetric deformables [Ferguson et al. 2023; Wicke et al. 2010], *h-adaptivity* has been applied to reduce remeshing objectives, better capture local contacts, preserve mesh quality, and decrease numerical error. These operations can be applied both locally [Ferguson et al. 2023; Wicke et al. 2010] or with a full reconstruction [Jiang et al. 2017; Klingner et al. 2006; Skouras et al. 2014; Stein et al. 2004].

Irrespective of how or what operations are applied, fundamental challenges remain for all remeshing operations in terms of both preserving invariants, and so *admissibility* [Brochu and Bridson 2009; Jiang et al. 2022], and mapping state between mesh updates with low error [Wicke et al. 2010]. In this work we observe the critical importance of including both mapping, and admissibility enforcement within remeshing optimization. We use this to construct optimal ITR’s constrained model problem and r-adaptive timestep solver.

For mapping, while many local and efficient approaches are possible, including closest-point [Molinari and Ortiz 2002] and barycentric interpolation [Spillmann and Teschner 2008; Wicke et al. 2010], we follow Ferguson et al. [2023] and use the  $L^2$ -projection [Léger et al. 2014; Vavourakis et al. 2013] to optimally map simulated physical quantities across mesh changes. Critically, we construct our physical remeshing objective with the  $L^2$ -projection directly embedded within. This ensures solutions that avoid simulation artifacts from mapping degeneracies (see e.g., Figure 6), but, at the same time,

requires us to develop algorithms that enable robust minimization with the projection operator included.

### 3 Background

We focus on the solution of large-deformation elastodynamics with frictional contact in  $d$ -dimensional space. With a reference material domain  $\Omega \subset \mathbb{R}^d$  ( $d = 2, 3$ ), deformation over time is defined globally by the deformation map  $\varphi : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}^d$ . Locally, for each time,  $t$ , and material point,  $X \in \Omega$ , we then have deformed positions,  $x(X, t) = \varphi(X, t)$ , and material velocities,  $v(X, t) = \dot{\varphi}(X, t)$ .

#### 3.1 Semi-Discrete Setting

Following Ferguson et al.[2023], we begin by discretizing *in time* (remaining continuous in space) to construct the solution of each timestep’s problem in semi-discrete optimization form as

$$x^{t+1} = \underset{x}{\operatorname{argmin}} E_t(x), \quad (1)$$

with a spatially continuous Incremental Potential (IP),

$$\begin{aligned} E_t(x) &= \int_{\Omega} \rho K(x(X), \tilde{x}^t(X)) dV \\ &+ \alpha h^2 \int_{\Omega} \Psi(x(X)) - x(X)^T f(X) dV \\ &+ \alpha h^2 \int_{\partial\Omega} B(x(X)) + D(x(X)) dA. \end{aligned} \quad (2)$$

Here  $h$  is timestep size,  $\Psi$  is a hyperelastic deformation-energy density (e.g. neo-Hookean),  $f$  contains the sum of body forces and (when ranging over boundary regions) any applied tractions,  $\rho$  is density, and  $B$  and  $D$  are the spatially-continuous analogs of the IPC energies [Ferguson et al. 2023] for, respectively, contact barrier and friction pseudo-potential. The energy density  $K$  is an inertial term for the discrete momentum contributions from “predictor” position,  $\tilde{x}^t$ . We cover our specific choice of construction for  $K$ , and its importance for adaptivity, below in Section 4.3.

In turn, choice of predictor  $\tilde{x}^t$  (an explicit update function of prior deformed position, velocity, and possibly acceleration fields,  $x^t, x^{t-1}, \dots, v^t, v^{t-1}, \dots, a^t, a^{t-1}, \dots$ ), scaling term  $\alpha \in \mathbb{R}^+$ , and corresponding explicit update equation for velocity (also acceleration as needed) from optimal solution  $x^{t+1}$ , jointly define the specific choice of numerical time-integration method. Here, in the main text, for simplicity, we will use implicit Euler with

$$\tilde{x}^t = x^t + hv^t, \quad v^{t+1} = \frac{1}{h}(x^{t+1} - x^t), \quad \alpha = 1, \quad (3)$$

while a range of alternate time integration methods are similarly covered [Li et al. 2020].

#### 3.2 Piecewise-Linear Spatial Discretizations

We build piecewise-linear discretizations of our spatial domain using simplicial meshes  $\mathcal{T} = (u, e)$  with  $n$  nodal reference coordinates<sup>1</sup>,  $u_i \in \Omega$  (defined at mesh vertices for the piecewise linear elements we cover here) stored in vector  $u = (u_1^T, \dots, u_n^T)^T \in \mathbb{R}^{dn}$ , and a corresponding mesh topology,  $e$ , defining mesh connectivity (triangles in 2D, tetrahedra in 3D) between mesh vertices. Discrete

<sup>1</sup>We emphasize that here and throughout we reserve  $u$  to always denote reference coordinates and never displacements.

vector fields are then correspondingly defined at nodes and stored as vectors  $x, v, a \in \mathbb{R}^{dn}$ .

Per discretization, each term in our Incremental Potential is now expressed as a weighted sum of energy functions over mesh element stencils,  $s$  (tetrahedral, triangle, edge, point, or pairings thereof depending on energy and dimension) in  $\mathcal{T}$ ,

$$\sum_{s \in \mathcal{T}} w_s(u) W_s(x),$$

where  $w_s$  is the volume, area, or length-weighted scaling of the rest shape element  $s$ , and  $W_s$  is the respective energy density function of each potential restricted to this element's stencil.

#### 4 Optimal In-Timestep Remeshing

If we restrict ourselves to use a single *unadapted* mesh throughout simulation, solutions for each timestep in Equation (2), then reduce (by spatial discretization) to simply solving the local minimization of the standard fully discrete Incremental Potential [Li et al. 2020] for the next timestep's nodal deformations  $x$ . In contrast, for *optimal* in-timestep remeshing we instead evolve both mesh *and* discrete physical state defined on it, to best capture the solution of our simulated system at each time step.

We advance our simulation domain through time by incrementally updating the tetrahedralization (respectively triangulation) of the material domain,  $\mathcal{T}(t)$ , together with fields for deformations,  $x(t)$ , and velocities,  $v(t)$ , defined at  $\mathcal{T}(t)$ 's vertices. Inputs for each solve to advance the simulation from time  $t$  to  $t+1$  are the necessary fields, e.g.,  $x^t, v^t, \tilde{x}^t$ , and the current applied forces (body and external),  $f^{t+1}$ . These quantities are all defined on the current mesh  $\mathcal{T}^t$  with a guaranteed invariant that the *deformed* mesh ( $x^t, \mathcal{T}^t$ ) from the last timestep solve, is penetration-free and (as required) inversion-free. Our timestep solves then maintain this invariant.

To perform each timestep solve, rather than minimizing the standard fully discrete IP, we instead minimize our spatially continuous IP (Equation 2) over all deformation fields realizable by admissible piecewise linear  $\mathcal{T}$ , and corresponding admissible deformation fields defined upon them.

##### 4.1 In-Optimization $L^2$ -Projections

Our remeshing process thus evolves our mesh within each timestep's solve. However, each time we update our mesh, we must correspondingly also remap all fields used to compute our continuous IP objective, to the new mesh. If we were to instead ignore this necessary remapping and leave quantities (e.g.,  $\tilde{x}^t$ ) as-is in-optimization, we would distort our timestep problem's discretization, and so end up with unusable simulation artifacts and biased meshes. See Figure 6.

This means that in order to find optimal remeshings, we must incorporate continual remappings of all discrete nodal quantities *within* our optimized objective function. To do so we apply the  $L^2$ -Projection [Ferguson et al. 2023; Léger et al. 2014],

$$\pi(y, \mathcal{T}^c, \mathcal{T}^p), \quad (4)$$

to continuously remaps discrete fields,  $y$ , on a prior mesh,  $\mathcal{T}^p$ , to discrete fields on current mesh,  $\mathcal{T}^c$ , with minimal  $L^2$  error between finite-element spaces. In contrast to Ferguson et al. [2023] we must then also compute the corresponding derivatives of this remapping

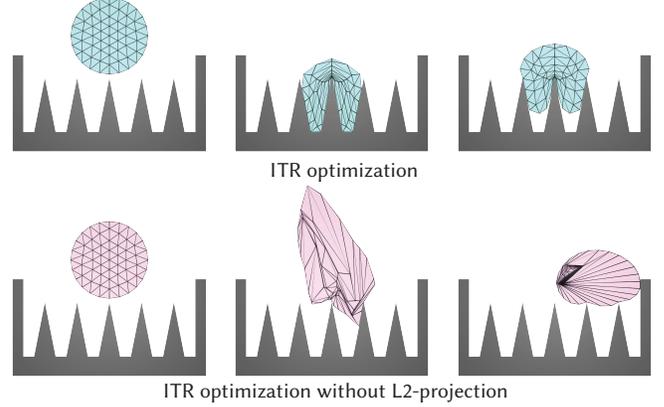


Fig. 6. **Necessity of  $L^2$ -Projections.** We drop an elastic circle onto rigid spikes. Top: optimal ITR adapts interior vertices to capture the large deformation and sharp contact geometry robustly. Bottom: applying  $L^2$ -projection only after optimization, velocities and positions are polluted during mesh adaptation, leading to unnatural deformation and instability.

across meshing variations. We cover this in detail, along with necessary background on the computation of the  $L^2$ -Projection below in Section 5.

While each applied remapping via the projection,  $\pi$ , is optimally error reducing, we seek to avoid unnecessary projections of fields, and so unnecessary accumulation of residuals. We do so during in-timestep optimization solves by solely (re-)computing projections from start-of-step mesh quantities to each newly evaluated mesh. As such it's useful to specialize these  $L^2$  projections as

$$\pi_t(\cdot, \mathcal{T}) = \pi(\cdot, \mathcal{T}, \mathcal{T}^t). \quad (5)$$

Repeatedly applying  $\pi_t$  to our start-of-step quantities,  $\tilde{x}^t, f^{t+1}$ , during optimizations (at each newly evaluated mesh), ensures no accumulation of consecutive projection errors over the course of our optimization solve. At the same time, at each solver iteration,  $i$ , position unknowns,  $x^i$ , are implicitly updated on each new corresponding mesh,  $\mathcal{T}^i$ , so that we can safely skip their re-mapping altogether. See Section 5.5 below for details.

##### 4.2 Incremental Potential Objective for Remeshing

We now construct our per-timestep IP objective for each new mesh candidate,  $\mathcal{T}$ , and corresponding deformed nodal positions,  $x$ , as,

$$\begin{aligned} E_t(x, \mathcal{T}) &= E_t(x, \mathcal{T}, \tilde{x}^t) \\ &= \rho K(x, \pi_t(\tilde{x}^t, \mathcal{T}), \mathcal{T}) \\ &\quad + \alpha h^2 (\Psi(x, \mathcal{T}) + B(x, \mathcal{T}) + D(x, \mathcal{T}) \\ &\quad - x^T \pi_t(f^{t+1}, \mathcal{T})). \end{aligned} \quad (6)$$

Here, energies  $K(\cdot, \cdot, \mathcal{T})$ ,  $\Psi(\cdot, \mathcal{T})$ ,  $B(\cdot, \mathcal{T})$  and  $D(\cdot, \mathcal{T})$  are the total resultant potentials generated, respectively, by our piecewise linear discretizations (recall Section 3.2) of corresponding inertial, deformation, contact barrier, and friction energy densities from the spatially continuous IP in (2), on mesh  $\mathcal{T}$ , with deformed nodal positions  $x$ .

*Admissible* meshes,  $\mathcal{T}$ , with corresponding *admissible* deformation solutions,  $x$ , that *together* give locally lowest values of (6), are *the locally optimal remeshed solutions to timestep  $t+1$* . We cover

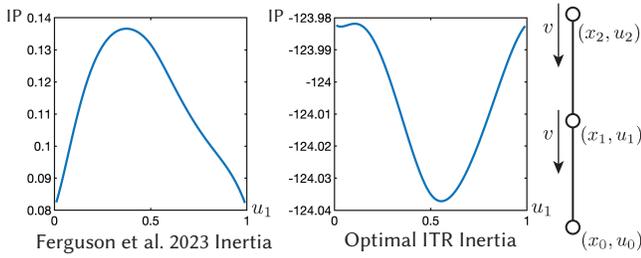


Fig. 7. **Choice of inertial energy.** We construct a simple 1D example to illustrate source of errors in the standard IP’s inertial term. Here (see far right) we discretize a rod with three free vertices. Second and third vertices have non-zero downwards velocities, velocity for the first vertex is zero, with  $x_0 = u_0 = 0$ ,  $x_2 = u_2 = 1$ . In left and right plots we compute Incremental Potentials with, respectively, standard inertial energy (left), and our corrected inertial energy (right), across adaptations of  $u_1$ . Left: minimization with the uncorrected inertial energy biases the middle vertex towards the boundary, and so generates degenerate elements. Right: our modified inertial energy avoids this issue, here with a well defined basin.

admissibility criteria for both meshes and deformations below in Section 4.4.

### 4.3 Inertial Contribution

In constructing their inertial term, Ferguson et al. [2023] mirror prior, fully discrete optimization-based time-stepping work [Gast et al. 2015; Li et al. 2020], and so employ a bilinear inertial contribution (for  $K$ ) for their spatially continuous Incremental Potential,

$$\frac{1}{2} \|x(X) - \tilde{x}^t(X)\|^2. \quad (7)$$

This gives, per discretization mesh  $\mathcal{T}$ , a satisfyingly positive-definite quadratic term for the inertial energy potential

$$\frac{1}{2} (x - \tilde{x}^t)^T M(\mathcal{T}) (x - \tilde{x}^t). \quad (8)$$

Here  $M(\mathcal{T})$  is the discretization’s consistent mass matrix. This standard “mass-metric” form is enticing. Positive-definiteness of the energy contribution helps downstream numerical optimization solves, and the metric form provides geometric intuition for the IP energy’s landscape.

However, once we optimize with meshing parameters as free variables, we observe that this prior formulation pollutes the objective with terms that bias solutions away from physical accuracy and so generate unnecessary simulation errors and artifacts (see e.g., Figure 7). To see the source of these errors, we expand Ferguson et al.’s [2023] inertial term in (8), noting the final term  $\frac{1}{2} (\tilde{x}^t)^T M(\mathcal{T}) \tilde{x}^t$ . Under *fixed* discretizations this is a constant term and so can safely be added to IPs without changing minima. However, when we jointly optimize over position and meshing parameters this term is no longer constant. Instead, it changes with meshing and non-physically biases our objective to solutions that unnecessarily decrease the mass-metric magnitude of our *fixed* initial predictor field,  $\tilde{x}^t$ , given by the previous time step.

To address these errors, we instead construct our inertial energy density as

$$K(x(X), \tilde{x}^t(X)) = \frac{1}{2} \|x(X)\|^2 - x(X)^T \tilde{x}^t(X). \quad (9)$$

This term, after spatial discretization, then defines our discrete inertial energy contribution to our objective (6) as

$$K(x, \tilde{x}^t, \mathcal{T}) = \frac{1}{2} x^T M(\mathcal{T}) x - x^T M(\mathcal{T}) \pi_t(\tilde{x}^t, \mathcal{T}). \quad (10)$$

Note that this change now removes an extraneous term from the IP. While, (as covered above) in fixed-mesh simulation this term is safely constant, when optimizing ITR it is not, and so adds a non-physical energy that distorts minimizers. All other terms in the IP, including those with  $\tilde{x}^t$  (which is remapped), are not extraneous for momentum balance and so do not distort. With our ITR objective now fully specified, we next define admissibility criteria for adaptivity below and then construct our corresponding general-form ITR optimization.

### 4.4 Admissibility

While tempting, we *can not* directly optimize our above-defined ITR objective over all possible deformations,  $x$ , and meshes,  $\mathcal{T}$ . We require *admissible* solutions.

*Geometric Admissibility.* We begin with the *geometric admissibility* constraints, that each mesh variation,  $\mathcal{T}$ , should be

- (1) Injective, and so provide a non-overlapping and locally injective parametrization of the reference domain of integration; and
- (2) Domain Preserving, and so maintain the shape (and thus boundary) of the reference domain.

Beginning with an initial, non-overlapping domain we can ensure full geometric admissibility of both conditions, if each mesh,  $\mathcal{T}$ , maintains the reference domain boundary and element non-inversion.

*Physical Admissibility.* In turn, for each considered mesh variation,  $\mathcal{T}$ , corresponding deformations defined upon it, should be *physically admissible* solutions of the discrete equations of motion defined by our selected numerical time-integration method. Applying our ITR objective in Equation (6), physical admissibility then reduces to requiring,

$$x = \underset{y}{\operatorname{argmin}} E_t(y, \mathcal{T}). \quad (11)$$

### 4.5 ITR Optimization Model

*Physical Admissibility Constraint.* If we ignore geometric admissibility, we note that unconstrained optimality of our objective,  $E_t(x, \mathcal{T})$ , satisfies physical admissibility. However, this energy is strongly non-convex in  $(x, \mathcal{T})$  with unbounded and degenerate solutions. To address this challenge, we minimize  $E_t(x, \mathcal{T})$  subject to physical admissibility,  $\nabla_x E_t(x, \mathcal{T}) = 0$ . This provides well-posed, constrained minimizers with correspondingly nondegenerate physical solutions. In Figure 8 we demonstrate exactly this difference in behavior between physical-admissibility constrained and unconstrained remeshing optimizations. Here, even in simple 1D (left) and 2D (right) elastostatic examples (issues only worsen for dynamics) we see that the unconstrained solutions obtain fully degenerate remeshings with collapsed or inverted elements (for solutions respectively with and without geometric admissibility additionally

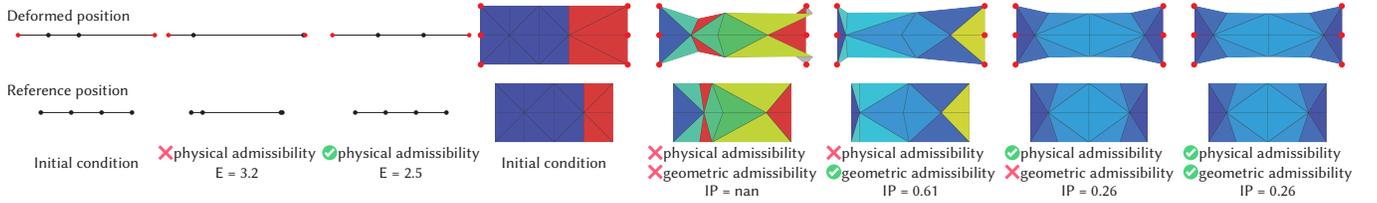


Fig. 8. **Adaptively computing static equilibrium with and without physical admissibility.** Evaluating the need for physical admissibility constraints in ITR optimization, we compare on the right (rendering per-element strain, ranging from red, high strain, to blue, low strain) in 2D ITR solves, progressing left-to-right in order, with no-admissibility, just geometric admissibility, just physical admissibility and finally our full optimal ITR model problem solved with both geometric and physical admissibility constraint. On the left we correspondingly demonstrate the same ablation with a simple 1D rod where geometric admissibility is automatically satisfied. Please see Section 4.5 for detailed discussion of these results and their implications for variational adaptivity solves.

enforced). In contrast, the same optimizations, with physical admissibility now added as constraint, generate low-energy, well-shaped deformations. Here the result is the same for with and without geometric constraint, as geometric admissibility is naturally enforced by optimal ITR in this case (see Section 5.3.2).

Our full optimization problem is then

$$\min_{x, \mathcal{T}} E_t(x, \mathcal{T}) \text{ s.t. } x = \underset{z}{\operatorname{argmin}} E_t(z, \mathcal{T}) \text{ and } \mathcal{T} \in \mathcal{G}, \quad (12)$$

where  $\mathcal{G}$  is the set of geometrically admissible meshes. In the absence of other criteria, *admissible* joint minimizers  $(x, \mathcal{T})$  of  $E_t$  in (6) can be solved via (12) to give locally optimal piecewise-linear solutions to the minimization of our spatially continuous IP (2).

Note that, as we cover above and illustrate in Figure 8, solely enforcing geometric feasibility,  $\mathcal{T} \in \mathcal{G}$ , is insufficient. As in Equation (12), we require physical admissibility conditions applied as *constraint* when minimizing  $E_t$  in our ITR optimization. To do so, we find it generally useful to apply our physical admissibility constraint *implicitly*, and so reformulate our optimization of the ITR timestep solves solely over meshing variables as

$$\min_{\mathcal{T}} E_t(\underset{x}{\operatorname{argmin}} E_t(x, \mathcal{T}), \mathcal{T}) \text{ s.t. } \mathcal{T} \in \mathcal{G}. \quad (13)$$

At the same time, we may also wish to consider criteria such as geometric meshing metrics, penalties, or adaptivity inertias to regularize and/or constrain our physical metric,  $E_t$ , in our adaptive meshing process. We thus construct our final objective function for optimal ITR as

$$R(x, \mathcal{T}) = E_t(x, \mathcal{T}) + \sum_i \beta_i C_i(\mathcal{T}), \quad (14)$$

where functions  $C_i$  and corresponding weights  $\beta_i$  add regularizations and/or penalties<sup>2</sup>. Our corresponding full ITR optimization problem is then

$$\begin{aligned} \mathcal{T}^{t+1} &= \underset{\mathcal{T}}{\operatorname{argmin}} W(\mathcal{T}), \text{ s.t. } \mathcal{T} \in \mathcal{G} \text{ with} \\ W(\mathcal{T}) &= R(\underset{x}{\operatorname{argmin}} E_t(x, \mathcal{T}), \mathcal{T}). \end{aligned} \quad (15)$$

<sup>2</sup>As we will demonstrate in Section 5, functions  $C_i$  can also be applied as penalties for geometric admissibility, in which case our side constraint for  $\mathcal{T} \in \mathcal{G}$  drops out from our final optimization problem.

A final timestep solution is then delivered by a locally optimal mesh,  $\mathcal{T}^{t+1}$ , and updates<sup>3</sup>,

$$\begin{aligned} x^{t+1} &= \underset{x}{\operatorname{argmin}} E_t(x, \mathcal{T}^{t+1}), \\ v^{t+1} &= \frac{1}{h}(x^{t+1} - \pi_t(x^t, \mathcal{T}^{t+1})). \end{aligned} \quad (16)$$

#### 4.6 Example Systems

Putting this all together we next consider here the action of the above ITR optimization contributions with simple, illustrative systems.

*Optimality.* We begin with a unit length 1D compressing elastic rod with uniform density  $\rho$ , an energy density of  $\Psi(x') = k(x' - 1)^2$  and initial inward-pointing velocities with magnitudes  $v$  at rod boundaries. Discretizing in time, our spatially continuous IP from Equation (6) reduces to

$$E_t(x(u), x'(u), u) = \int_0^1 \frac{1}{2} \rho x^2 - \rho x \dot{x}^t + \frac{h^2}{2} k_s (x' - 1)^2 du, \quad (17)$$

where we have  $\tilde{x}^t(u) = hv + u(1 - 2hv)$  from symmetry. Next, beginning with an initially uniform spatial discretization of the rod with four vertices spaced along the rod, in Figure 9, we compare our ITR optimized re-meshed solution with the ground-truth optimal timestep update given by the above continuous IP's Euler-Lagrange

<sup>3</sup>The velocity update is demonstrated here with our running implicit Euler example.

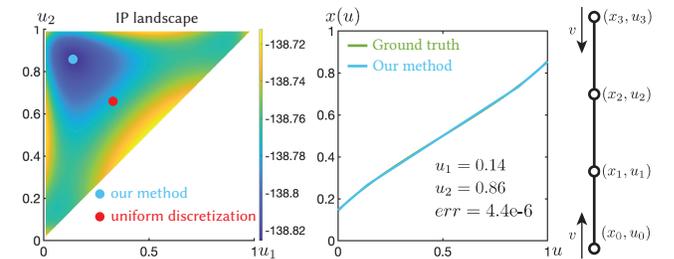


Fig. 9. **Optimality.** We evaluate all spatially continuous IP values across a simple 1D elastic rod simulation step (see setup on far right). Left: starting from a uniform discretization (red point), optimal ITR computes a remeshing and state solution at the energy's minimum (blue point). Right: plotting the continuous  $x(u)$  solution curve for the timestep update with optimal ITR's solution overlaid, we see a corresponding small error of  $4.4e-6$ .

equation,

$$h^2 k x'' - \rho(x - \tilde{x}^t) = 0. \quad (18)$$

In Figure 9 right, we see that overlaying the ground-truth solution and our ITR-optimized four-node solution curve<sup>4</sup> gives a low-error solution with closely matching energy. In Figure 9 left we then plot the IP energy over enumerated node discretizations confirming that our solution obtains the minimum energy solution on the landscape.

*In-Optimization  $L^2$ -projection.* In Figure 6 we drop a 2D elastic ball on spikes using our new IP objective with the  $L^2$  projection in Equation 6 (top), compared with (bottom)  $L^2$ -projection remappings performed after IP optimization. Even in this simple example, solving with unmapped fields quickly introduces warped mesh solutions and extreme simulation artifacts (please also see the supplemental video).

## 5 Optimal r-adaptive ITR Simulation

In the last section we built a general model and optimization problem that, when minimized, gives locally optimal timestep solutions for adapted mesh and physical state. We have not yet covered how such an optimization problem can be numerically solved for practical simulation application.

In this section we now focus on building the necessary algorithmic contributions to enable the numerical solution of our ITR optimization problem for the optimal *r-adaptive* simulation of frictionally contacting elasto-dynamics, statics, and quasi-statics. For r-adaptive ITR we optimize the simulation mesh's reference nodal positions,  $u$ , while leaving connectivity parameters,  $e$ , fixed. Focusing on r-adaptive simulation allows us to construct a smooth, Newton-type optimization method. We leave optimal h-adaptive ITR simulation (optimizing mesh connectivity) for future work, and briefly cover challenges and future opportunities for it, e.g., via differentiable mesh connectivity, in Section 7.

### 5.1 $L^2$ -Projections for r-adaptive Updates

In order to compute our  $L^2$ -projections for r-adaptivity, we begin by recalling (Section 4.1) that we can restrict ourselves to computing projections from start of timestep, to new r-adaptive updates,  $\pi_t(\cdot, u) = \pi_t(\cdot, (u, e))$ .

Each iteration of our ITR optimization modifies the simulation mesh  $(u^t, e) \rightarrow (u^i, e)$ , and so, in turn, these changes in mesh update the underlying function space of our FE model. For each such referential update  $u^t \rightarrow u^i$  we have corresponding function spaces,  $V^t$ , with basis  $\{\psi_p^t | 1 \leq p \leq n\}$ , and  $V^i$ , with basis  $\{\psi_q^i | 1 \leq q \leq n\}$ , for our start of step,  $t$ , and current iterate,  $i$ , meshes respectively. For functions  $f^t \in V^t$ , their  $L^2$ -projections onto  $V^i$ , given as  $f^i$ , minimize the  $L^2$  residual  $\frac{1}{2} \|f^i - f^t\|^2$  [Léger et al. 2014]. Optimality of this least-squares problem then gives, for discrete vector fields  $y^t$ , defined on  $\mathcal{T}^t = (u^t, e)$ , our r-adaptive projection operator,

$$\pi_t(y^t, u^i) = \bar{M}(u^i)^{-1} A_t(u^i) y^t. \quad (19)$$

Here  $\bar{M}(u^i)$  is the density-normalized consistent mass matrix on the mesh  $\mathcal{T}^i = (u^i, e)$ , and  $A_t(u^i)$  is the transfer matrix between bases so that submatrix  $A_{t,p,q} = \int_{\Omega} \psi_p^i \psi_q^t dV \otimes I_d$ .

<sup>4</sup>For solving we use  $k = 2e5$ ,  $\rho = 1e3$ , and  $h = 1e-2$ .

We then require a quadrature to compute the corresponding energy, gradient, and Hessians of the terms in our objective that include projections,  $\pi_t$ . To begin with, considering Equation (6), we can pre-multiply all appearances of  $\pi_t$  in our objective, with  $x^T M(u)$  and so can focus (when simulating with constant density bodies) on the differentiable computation of terms

$$b_t(y^t, x, u) = x^T M(u)^{-1} \bar{M}(u) A_t(u) y^t = \rho x^T A_t(u) y^t, \quad (20)$$

for discrete vector fields,  $y^t \in \mathbb{R}^{dn}$ . This allows us to simplify evaluations and avoid additional computation of derivatives with respect to our mass matrix in our projection operator derivatives.

### 5.2 Computing the $L^2$ -Projection and its Derivatives

Prior work, including Ferguson et al. [2023], compute the necessary integrals to evaluate the transfer matrix by a quadrature arrangement [Krause and Zulian 2016] formed from the intersection cells of the two overlapping meshes of the material domain. I.e., computed via PolyClipper [Powell 2021] and further decomposed by triangulation. For *evaluation* of  $A_t$  this strategy is effective and efficient. However, for computation of the transfer matrix's *derivatives* we find this solution is no longer stable nor practical and so we can not apply Ferguson et al.'s [2023] numerical integration. This is primarily due to two connected issues. First, jumps in intersection cell structure, and so corresponding jumps in their quadrature arrangements and weights, can not be parameterized with respect to arbitrarily updating vertex coordinates,  $u^t$ , during optimization. Second, even if we restrict ourselves to small vertex changes that preserve cell-structure topology, we still observe rapid changes in cell geometries (and so the corresponding shape function we are integrating) that, in turn, can generate unstable evaluations of the integrals' derivatives.

To address these issues we instead compute the integration of  $A_t$  by direct quadrature on just the adapted mesh, so that our arrangement and weights are directly parameterized by  $u$ . Effectively, we are applying each new mesh  $\mathcal{T} = (u, e)$  to integrate the same sharply varying function given by the inner products of the two different meshes' shape functions. To efficiently compute our integrals, we construct an AABB-based BVH [Schneider et al. 2024] at start of each time step on the last mesh,  $\mathcal{T}^t$ , and use it, at each quadrature point evaluation inside the new mesh  $\mathcal{T}^i$ , to look up which element (and so basis function) in  $\mathcal{T}^t$  we are integrating against. Simply put, we compute and assemble the projector with quadrature sampling per-element of adapted mesh, a check per quadrature point for which element in the original mesh it overlaps with, and calculation of shape function multiplications. See our supplemental for details.

However, while our parameterization is now smooth, the underlying function we are integrating over still sharply varies with  $u$  over the domain. A low quadrature order (e.g., a three-point triangle stencil in 2D, as applied by Ferguson et al. [2023]) can and will miss sampling different values from smaller regions of the domain. In order to robustly capture and integrate piecewise changes, we apply Witherden and Vincent's [2015] 49-point and 95-point arrangements for 2D and 3D domains respectively [Schlömer et al. 2024]. We emphasize that, while we apply a higher-order *quadrature*

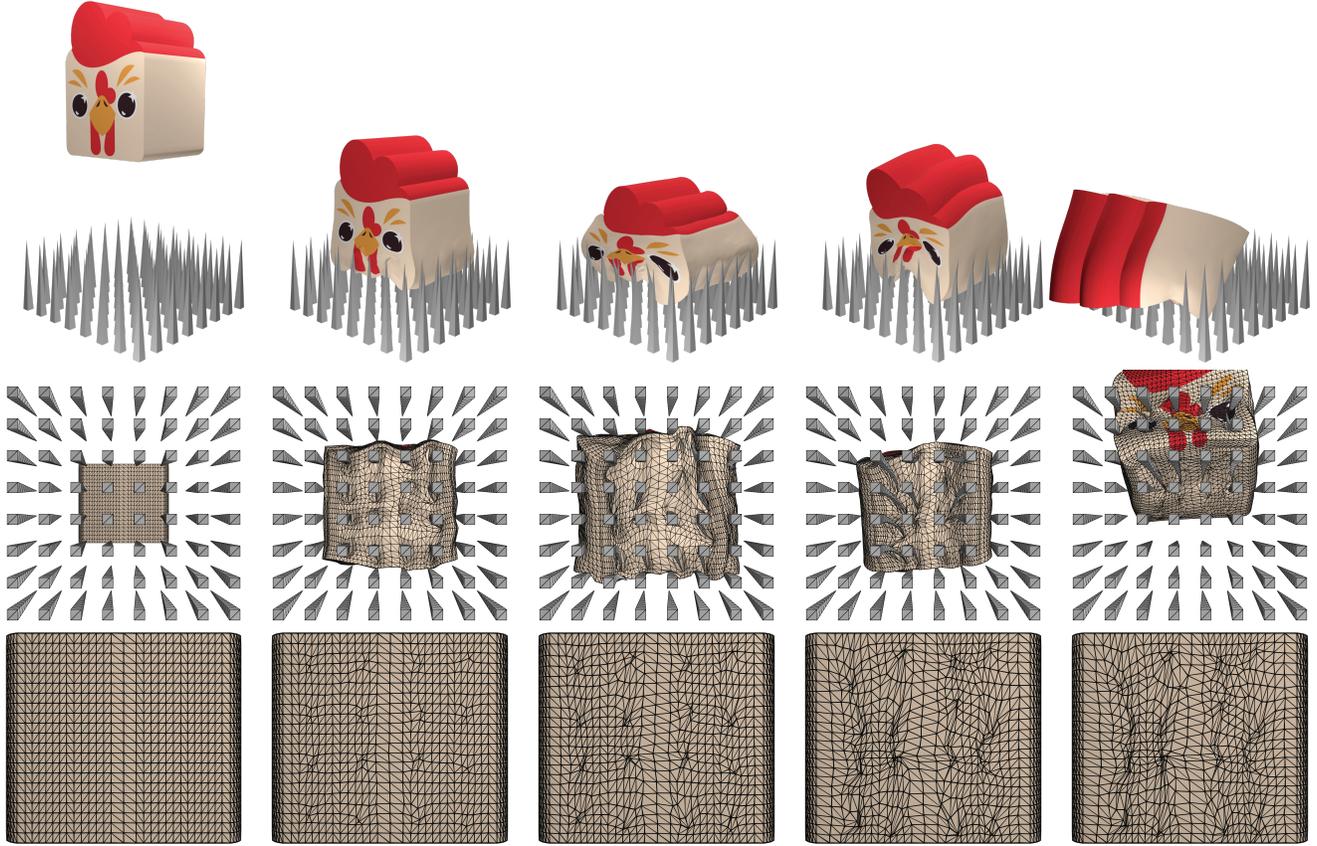


Fig. 10. **Chicken Drop!** We drop a soft marshmallow chicken, squashing onto stiff sharpened spikes, with (top to bottom) visualizations of the full scene, a bottom deformed meshed view, and a bottom view of the reference-mesh’s adaptivity.

for computation of the projection, the order of our underlying finite elements remains unchanged.

We find that these quadratures work well, generating tight numerical convergence to locally optimal solutions in all our experiments, see Section 6. However, we note that the underlying function we integrate can be, independent of sufficient sampling density, discontinuous *within* our simplices and so could still lead to discontinuous derivatives. In practice however, we do not encounter cases preventing solver convergence across our examples and note that this merits further investigation, see Section 7.

While we initially prototyped our gradient and Hessian computation via automatic differentiation (AD), the resulting code was significantly bottlenecked by the AD computation. In our final implementation, we use hand-derived analytic derivatives, which give a  $\sim 2X$  speedup. See our supplemental for the analytic derivatives of the  $L^2$ -projector.

### 5.3 Geometric Admissibility for r-Adaptive Solves

To enable efficient Newton-type optimization for r-adaptive solves (see Section 5.5) we apply our geometric admissibility conditions via penalty terms,  $C_i$ , folded into our final objective  $W$  in Equation (14).

**5.3.1 Boundary Maintenance.** To preserve our piecewise-linear reference domain we enable boundary vertices to adapt by sliding along boundary facets (lines in 2D, planar regions in 3D) and along straight edges (in 3D). For exact enforcement, facet-sliding membership in both 2D and 3D simply amounts to comparing the normals of incident boundary faces to each boundary vertex. If normals agree these vertices can be enabled to adapt in-facet. If a 3D boundary vertex then *does not* lie in a one-ring facet, it is suitable to check for edge-sliding membership. If the directions of just two boundary edges in the vertex’s one ring are collinear the vertex can be enabled to adapt in-line.

Rather than requiring the above hard membership criteria, we loosen membership tolerances and so also correspondingly weaken sliding constraints to be approximate. This allows on-boundary sliding on perturbed geometries and approximately planar regions. For each reference boundary vertex,  $u_i$ , we apply its area-weighted (respectively length-weighted) vertex normal,  $n_i = 1/d \sum_{t \in \mathcal{N}(i)} w_t n_t$ , to define tentative sliding directions. We assign facet-sliding membership for the vertex to this approximated sliding facet, if incident boundary face normals are all closer than a user prescribed tolerance,  $\epsilon_b$ , to the proposed sliding direction,  $n_i$ , so that  $1 - n_i^T n_t \leq \epsilon_b, \forall t \in \mathcal{N}(i)$ . We then add the corresponding “loose” boundary-facet sliding penalty  $C_i(u) = 1/2\kappa_s ((u_i - u_i^0) / \|u_j - u_j^0\| \cdot n_i)^2$  to our objective  $W$ , with  $\kappa_s$  set to match the IPC contact barrier stiffness [Li

et al. 2020]. If a 3D boundary vertex,  $u_i$ , is not so assigned, we then correspondingly check collinearity of non-adjacent one-ring edge pairs  $c_{j,k} = (u_j - u_i)^T(u_k - u_i) + 1$ ,  $j, k \in \mathcal{N}(i)$ . If just one pairing  $j, k$  satisfies user tolerance with  $c_{j,k} \leq \epsilon_b$ , then we assign edge-sliding membership for the vertex to with an approximated sliding direction  $t_i = 1/2((u_j - u_i)/\|u_j - u_i\| + (u_k - u_i)/\|u_k - u_i\|)$ . We then add the corresponding “loose” boundary-edge sliding penalty  $C_i(u) = 1/2\kappa_s((u_i - u_i^0)/\|u_j - u_i^0\| \times t_i)^2$  to our objective  $W$ . Resulting constraints are tightly enforced at converged solutions, with violations less than  $10^{-8}$  across all examples in Section 6.

**5.3.2 Non-Inversion.** In practice, we observe that ITR optimization, with just the physical admissibility constraint, generally produces injective parameterizations naturally, without requiring explicit application of geometric non-inversion constraints. See e.g., Figure 8. However, we observe that r-adaptivity will sometimes seek to compress elements towards a flattened (but not inverting) shape. Upon close examination we then confirm that, in these configurations, this flattening effectively is, variationally, the better solution. The removal of these elements, if applied in these timesteps, would indeed give better, lower energy, solutions. This is a promising hint for the potential utility of the ITR optimization in future work combining *h-adaptivity* with r-adaptivity which would allow these identified elements to be removed and added in a numerically robust way. However, for our current, purely r-adaptive application, we wish to avoid these degeneracies while at the same time, not strongly penalizing anisotropic elements from forming when advantageous, e.g., to better capture locally anisotropic strain distributions [Shewchuk 2002]. To balance between these needs we add a small volume (respectively area) barrier [Ferguson et al. 2023] to our objective  $W$  (with stiffness again matching the IPC contact barrier’s), for each element  $t \in \mathcal{T}$ ,

$$C_t(u) = \begin{cases} -(w_t(u) - \hat{w})^2 \ln(\frac{w_t(u)}{\hat{w}}), & 0 < w_t(u) < \hat{w}, \\ 0, & w_t(u) \geq \hat{w}. \end{cases} \quad (21)$$

Here  $w_t$  is the volume (respectively area) function of the reference element, and  $\hat{w} > 0$  is the minimum element volume threshold. In practice, we set  $\hat{w}$  to 1/10 of the smallest element volume in the our starting mesh,  $\mathcal{T}^0$ .

#### 5.4 Truncated System for r-adaptive ITR Newton Solves

We build an efficient Newton-type solver to r-adaptively minimize our ITR energy from Equation (15),

$$u^{t+1} = \min_u W_t((u, e)), \quad (22)$$

to tight tolerances.

To solve each iterate  $i + 1$ , we first build a *truncated* Hessian approximation from  $(u_i, x_i)$ , by the positive semi-definite projection of the reference-coordinate Hessian,

$$H_i = \text{Proj}^+ \left( \frac{\partial^2 R(x_i, u_i)}{\partial u^2} \right). \quad (23)$$

We then compute a next descent direction in reference coordinates,  $u$ , by solving the linear system

$$H_i \Delta u = -g_i, \quad (24)$$

where  $g_i = \partial R(x_i, u_i)/\partial u$ .

We choose these terms by first expanding the ITR energy’s gradient and Hessian terms wrt  $u$  as

$$\frac{\partial W_t}{\partial u} = \frac{\partial R}{\partial u} + \frac{\partial R}{\partial x} \frac{\partial x}{\partial u}, \quad (25)$$

and

$$\frac{\partial^2 W_t}{\partial u^2} = \frac{\partial^2 R}{\partial u^2} + \frac{\partial^2 R}{\partial u \partial x} \frac{\partial x}{\partial u} + \frac{\partial x^T}{\partial u} \frac{\partial^2 R}{\partial x^2} \frac{\partial x}{\partial u} + \frac{\partial x^T}{\partial u} \frac{\partial^2 R}{\partial x \partial u} + \frac{\partial R}{\partial x} \frac{\partial^2 x}{\partial u^2}. \quad (26)$$

With  $x(u) = \text{argmin}_z E(z, u)$  in  $W_t$ , local optimality gives  $\partial E/\partial x = 0$ . Observing that  $\partial R/\partial x = \partial E/\partial x$ , the last term in both the gradient and Hessian then drop out. Next, expanding the optimality conditions, we have

$$\frac{\partial^2 E}{\partial x^2} \frac{\partial x}{\partial u} + \frac{\partial^2 E}{\partial x \partial u} = 0. \quad (27)$$

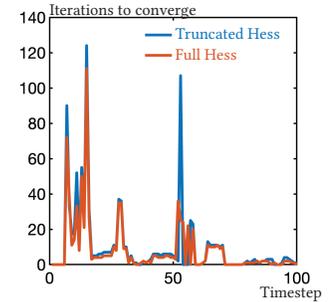
The third and fourth terms in the Hessian then respectively simplify to

$$\frac{\partial x^T}{\partial u} \frac{\partial^2 R}{\partial x^2} \frac{\partial x}{\partial u} = \frac{\partial x^T}{\partial u} \frac{\partial^2 E}{\partial x^2} \frac{\partial x}{\partial u} \quad \text{and} \quad \frac{\partial x^T}{\partial u} \frac{\partial^2 R}{\partial x \partial u} = -\frac{\partial x^T}{\partial u} \frac{\partial^2 E}{\partial x^2} \frac{\partial x}{\partial u}, \quad (28)$$

and so cancel.

For the remaining terms in the ITR Hessian, we directly compute the sparse reference-coordinate Hessian,  $\partial^2 R/\partial u^2$ . To build the final remaining term in the ITR Hessian, we could implicitly solve for  $\partial x/\partial u$  from Equation (27) above. However, the resulting Jacobian is dense, requires expensive computation to generate, and, once computed, results in dense fill-in contributions to our final Hessian.

To address this potential bottleneck we explored two possible strategies. For the first we observed that most entries in the dense Jacobian are relatively small (corresponding to the influence of reference node positions on far away DOF  $x$ ). To leverage this we tried applying a filtered Jacobian, zeroing out all terms below a filtering threshold. This removes dense fill-in at the cost of reduced accuracy. Our second, strategy was to simply and directly apply a truncated



**Fig. 11. Truncated System Convergence.** Full Hessian ITR solves converge faster than our truncated Hessian solver. However, total cumulative improvement in convergence rate, for the full Hessian, generally concentrate in just a few timesteps, and does not outweigh its significant compute costs over the truncated Hessian.

Hessian (see Equation 23) that drops the second term. In the end, this simplest strategy converges much faster (than either full or filtered Hessian strategies) in wall-clock time (albeit taking more iterations) and so is, by far, most effective. See Figure 11.

#### 5.5 Algorithm for r-adaptive ITR Simulation

We parameterize the target accuracy of our ITR solver with two tolerances. The first,  $\epsilon_e$ , defines the accuracy to which each numerical timestep is solved on its optimal mesh. The second,  $\epsilon_r$ , controls how

tightly we solve for the optimal mesh. Lower values of  $\epsilon_r$  indicate a tighter solve of the ITR problem, and so more adaptivity, while higher values indicate looser solves and so less adaptivity in regions where a mesh is already reasonably effective, see e.g., Figure 12.

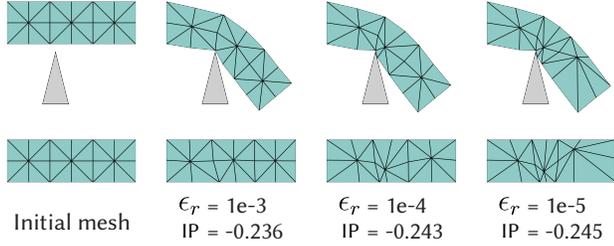


Fig. 12. **Static equilibrium under different  $\epsilon_r$ .** We fix the left boundary of an elastic beam and calculate the static equilibrium under different  $\epsilon_r$ . The beam bends under gravity and has contact with the sharp rigid spike obstacle. Optimal r-ITR automatically adapts boundary vertices to the contact point, and cluster vertices to capture the curvature and bulging due to bending deformation. As users decrease the  $\epsilon_r$ , tolerance optimal ITR generates increasingly aggressive adaptivity.

To timestep with r-adaptive ITR we begin each simulation with an initial admissible starting mesh,  $\mathcal{T}^0 = (u^0, e)$ , and initial admissible state,  $x^0, v^0$ . We start each new timestep solve with the admissible ITR state  $x^t, v^t, a^t, u^t \in \mathbb{R}^{dn}$  from the last step. We then explicitly compute the predictor term,  $\tilde{x}^t$  (e.g., to  $x^t + hv^t$  for implicit Euler), on the current mesh,  $\mathcal{T}^t = (u^t, e)$ .

Our Newton-type solver then effectively follows a standard projected-Newton algorithm pipeline with our truncated Hessian, with one

---

#### Algorithm 1 Optimal r-adaptive ITR

---

```

1: procedure R-ITR( $x^t, v^t, u^t, e$ )
2:    $\tilde{x}^t \leftarrow x^t + hv^t$ 
3:    $x \leftarrow \operatorname{argmin}_x E_t(x, (u^t, e), \tilde{x}^t)$ 
4:    $u \leftarrow u^t$ 
5:   loop
6:      $g \leftarrow \partial R(x, u) / \partial u$ 
7:      $H \leftarrow \operatorname{Proj}^+(\partial^2 R(x, u) / \partial u^2)$ 
8:      $\Delta u \leftarrow -H^{-1}g$ 
9:     if  $\|\Delta u\| \leq \epsilon_r$  break
10:     $\alpha \leftarrow \min(1, \operatorname{StepFilter}(u, \Delta u))$ 
11:     $R_0 \leftarrow R(x, u)$ 
12:    do
13:       $u' \leftarrow u + \alpha \Delta u$ 
14:       $x \leftarrow \operatorname{argmin}_x E_t(x, (u', e), \tilde{x}^t)$ 
15:       $\alpha \leftarrow \alpha / 2$ 
16:    while  $R(x, u') > R_0$ 
17:     $u \leftarrow u'$ 
18:  end loop
19:   $u^{t+1} \leftarrow u$ 
20:   $x^{t+1} \leftarrow x$ 
21:   $v^{t+1} \leftarrow 1/h(x^{t+1} - \pi_t(x^t, (u^{t+1}, e)))$ 
22:  return  $x^{t+1}, v^{t+1}, u^{t+1}$ 
23: end procedure

```

---

key caveat. This being that evaluations of our minimized energy are nested. Specifically, the trade-off for our not solving a problem twice the size of system DOF (i.e., in both  $x$  and  $u$ ) is that we solve a standard implicit timestep solve within each energy evaluation of  $W$ . See Algorithm 1.

In future work it remains a to-be-explored acceleration opportunity to evaluate our energy with nested, fast approximate timestep solves. However, in this work we've first focused on end-to-end accurate solves. Even so, as we evaluate next in Section 6, we see that with optimal r-adaptivity, we can capture comparable and better accuracy than with unadapted meshes orders-of-magnitude larger and so with up to 27,000X speedup and 31X decrease in memory usage (see Figure 17).

## 6 Evaluation

Our method is implemented in C++, parallelizing assembly and evaluations with Intel TBB, using Eigen [Guennebaud et al. 2010] for basic linear-algebra routines and Eigen's Cholesky ( $LL^T$ ) factorization for the Newton solves. We run all the experiments on an Apple M3 pro machine with 18 GB memory. We use uniform refinement with per edge subdivision to create increasing resolution meshes, for our refinement studies. Please see our supplemental document for simulation and material parameters used for all examples, as well as corresponding and timing breakdown and memory statistics.

### 6.1 Dynamics Benchmarks

Here we evaluate quantitatively and qualitatively the trajectories generated by low-resolution optimal r-adaptive ITR simulations with much-higher resolution fixed-discretization simulated meshes.

*High-speed impact and transients:* We first consider the high-speed impact of a 2D elastic bar fired at high-speed at the ground, with an initial uniform downward velocity of 80m/s. We consider the behavior of a low-resolution, 0.2K vertices, optimal ITR simulation with respect to fixed-mesh simulations over increasingly fine resolutions. In Figures 13 and 14 we compare, respectively the deformation (also meshing for ITR) and strain for each timestep moving through the initial collision, compression, and restoration phases of the impact. Over progressive fixed-mesh resolution simulations (please see Figures 4 and 5 in our supplemental for the remainder of fixed resolution results and strain distribution on the reference mesh) we reach a comparable trajectory, with a simulation-averaged Hausdorff distance of  $8e-3$  and IP error of  $9e-2$ , to fixed discretization solution with 7.1K vertex mesh.

*Dynamic drop through:* In Figure 2 we next drop a 3D neo-Hookean elastic wedge through a tight frictionless gap obstacle, again comparing a coarse-mesh (0.2K vertices) optimal r-ITR simulation with successively finer fixed-mesh simulations. Lower-resolution fixed mesh solutions, with numerical stiffening and geometric locking near contacts (please see our video and Figure 6 in our Supplemental), are unable to pass through the gap. While finer mesh simulations pass through, it's not until a 113K vertex simulation (with a resultant 110X slower runtime and a 18.5X increase in memory usage) that a fixed-mesh simulation's motion obtains a close match, per-step, to the optimal r-ITR simulation's dynamics.

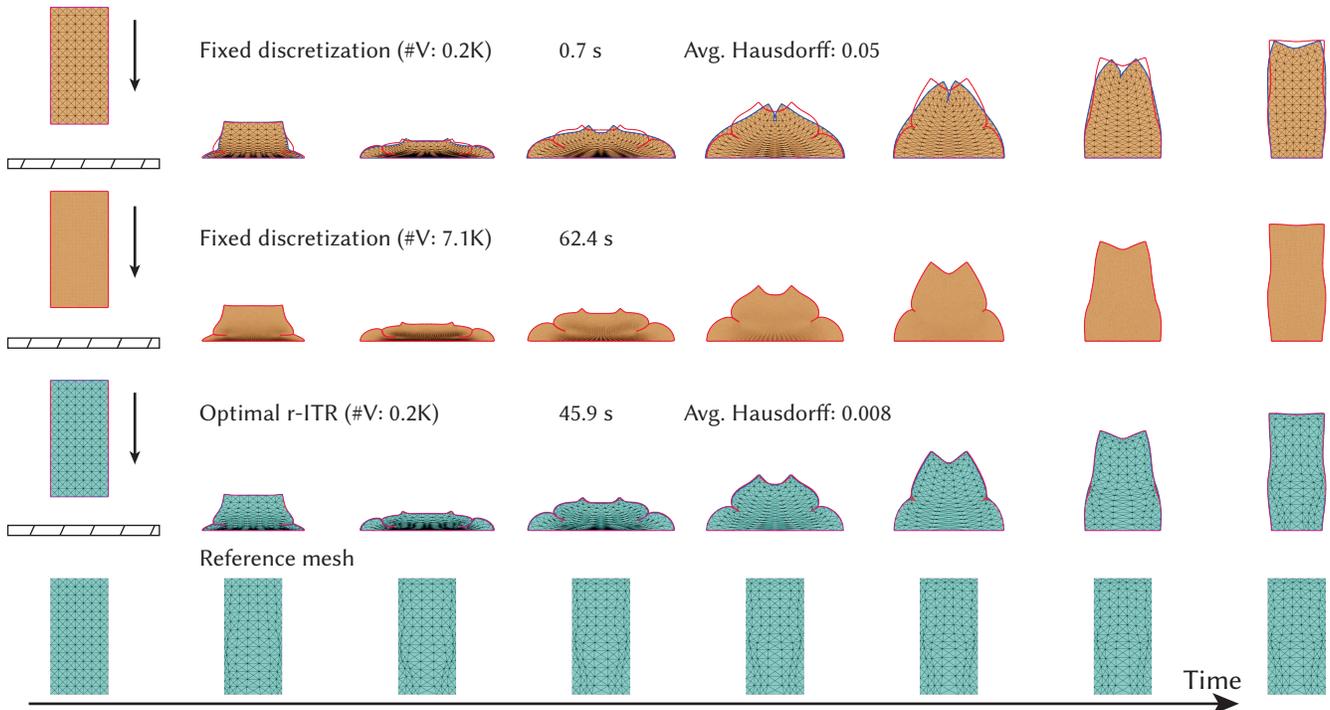


Fig. 13. **High-speed impact, deformation and meshing.** We fire a 2D neo-Hookean elastic bar at high-speed towards the ground. Simulated (bottom green) with optimal r-ITR using a coarse 0.2K vertex mesh we see both the evolving ITR deformation and reference mesh per timestep. Middle orange: over progressive fixed-mesh resolution simulations (please see our supplemental for the remainder of evaluated fixed-resolution simulation results) we reach a comparable trajectory (trajectory averaged Hausdorff distance of  $8e-3$ ) to our optimal ITR solution with 7.1K vertex fixed mesh (red contour). Top orange: A comparable fixed-mesh simulation with the same coarse resolution as the bottom optimal ITR solution, generates a distinctly different per timestep trajectory and strain (see Figure 14).

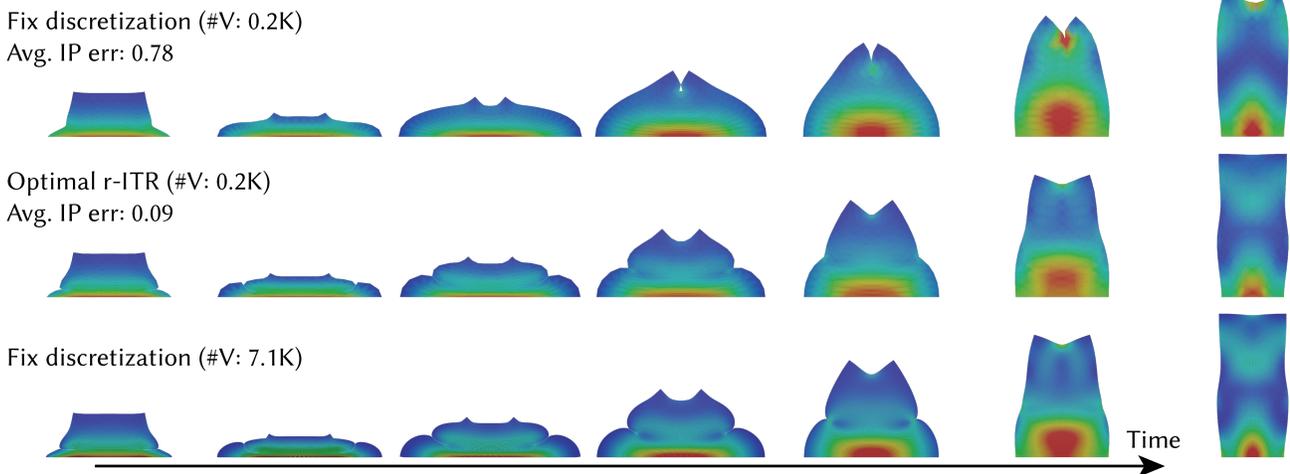


Fig. 14. **High-speed impact, strain and deformation** A coarse-resolution 0.2K vertex optimal r-ITR simulation's deformation and strain closely match (see Figure 13 and the text for details) a finest-resolution 7.1K vertex fixed-mesh simulation, (please see Figures 4 and 5 in our supplemental for strain distributions on additional reference mesh resolutions).

*Varying friction:* In Figure 21, we consider adaptivity for a 2D neo-Hookean wedge dropped through a conforming angled gap with r-ITR's adaptivity responding to increasing coefficients of friction. At lower coefficients the wedge slides, catches and compresses with adaptivity enabling the sliding of the deformed wedge through the gap. Simulations correspondingly slide through faster as friction

lowers (respectively  $\mu = 0$  and  $0.1$ ). At higher friction ( $\mu = 0.5$ ) the wedge likewise compresses partially through the gap, but can not slide fully through, with adaptivity enabling the simulation to capture the final configuration of the wedge caught between gap corners. In contrast, without adaptivity even the frictionless wedge

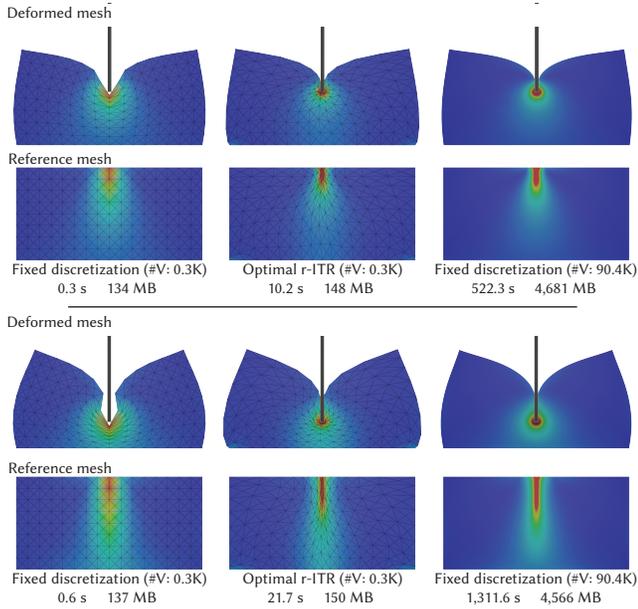


Fig. 15. **Deep elastic punch.** We incrementally punch a neo-Hookean elastic foam block with a rigid bar to different depths. We show the strain distribution and the wireframes on both deformed mesh and reference mesh. Red indicates higher strain, blue lower. We apply a uniform-refined 90K vertex mesh as a benchmark baseline. In both scenarios, our method produces a very similar outline and strain distribution to the benchmark with only 0.3K vertices (300X less vertices), while comparable-resolution fixed discretization produces significant, obvious artifacts. We also show that at the same time cost and peak memory usage, our method obtains a much lower energy. See Figure 16 and our discussion for details.

is unable to pass through and simply ends at rest, without significant deformation, against the obstacle.

## 6.2 Statics Benchmarks

Next we consider corresponding evaluations of optimal r-adaptive ITR simulation on challenging contact-driven elastostatics problems.

*Deep elastic punch:* Scripting a bar obstacle, we incrementally punch it into a 2D elastic neo-Hookean foam block (width 1m, height 0.5m), fixed on its base, to two depths of 0.21m and 0.36m. We solve each scenario to equilibrium with optimal ITR on a coarse 0.3K vertex mesh and correspondingly over a range of successively increasing resolutions, see Figure 15. Deformation of each block is driven by frictionless contact with the bar. We see that we qualitatively, in terms of both deformation and strain distribution, begin obtaining a close match to our optimal r-ITR solution with fixed-mesh solution at 90.4K vertices, with a 60X slower runtime and 32X increase in memory. However, even here, at this much (300X) finer resolution we still observe (Figure 16) that the optimal r-ITR solution has a significantly lower IP energy. To better understand this we observe that, due to imposed boundary conditions on the base of the block, the fixed mesh is unable to obtain deformation freedom to relax near the base as optimal r-ITR is. As confirmation, in Figure 2 of our supplemental, we additionally *fix* optimal ITR's reference coordinates along the base and now find that we reach a closely matching solution to the 90.4K fixed discretization.

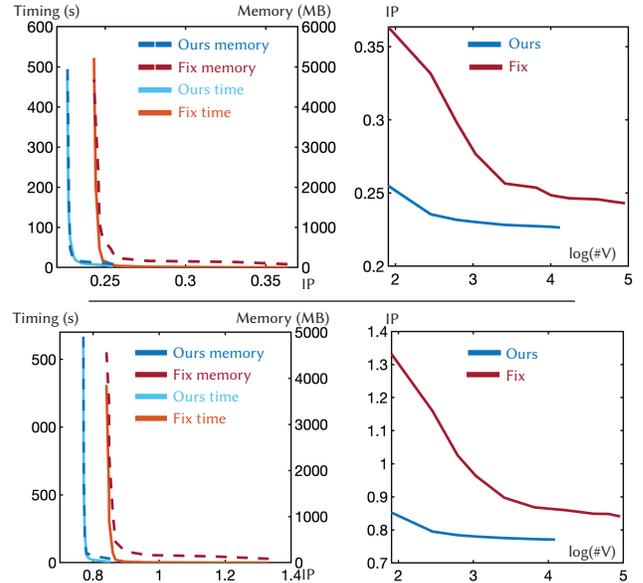


Fig. 16. **Memory and timing for deep elastic punch examples.** For same timing cost, peak memory usage, and mesh resolution, optimal ITR obtains much lower Incremental Potential energies than fixed-discretization simulations.

*Static drop-through:* Next, to continue examining the difference in locking behaviors between fixed-meshed simulations and optimal r-ITR, we consider a simplified 2D elastostatics variation of the above dynamic drop-through experiment. In Figure 17 we drop a 2D neo-Hookean elastic wedge, pinned along its left and right upper edges, through a tight frictionless gap obstacle and solve to equilibrium. As a baseline we first simulate with a coarse-mesh, 0.1K vertices, optimal r-ITR simulation. In Figure 17 bottom we visualize intermediate steps of the solve, and optimal r-ITR's final compliant equilibrium solution on the right. Next, successively solving the static drop-through across increasingly finer-resolution, fixed-mesh equilibrium solves, we again see improving results with lower-resolutions locking on top of the obstacle and finer resolutions, beginning to pass through. However, even with a simulation mesh of 400K vertices we still see that equilibrium solutions fail to match the compliance and low energy of our starting coarse resolution ITR simulation; see Figure 17 and Figure 18. To better understand this behavior, we zoom-in to the contact corners and observe that across resolutions, fixed-mesh equilibrium solves the interaction between sharp contact asperities and severely compressed elements introduce a repeating locking pattern in which elements rotate against each other and form seams; see Figure 19. This complex behavior is best seen in animation (please see our supplemental video) and remains in all fixed-discretization simulations up into the 400K mesh (finest we have run) and so restricts the compliance of each mesh's solution.

*Stamping.* In Figure 23 we press four Chinese letter stamps (*qí tiān dà shèng* for the famous mythological character Sun Wukong) into four neo-Hookean elastic foam blocks. Here optimal r-ITR adapts the reference mesh so that its deformation along mesh edges can tightly conform to the shape of the contacting block edges. In contrast,

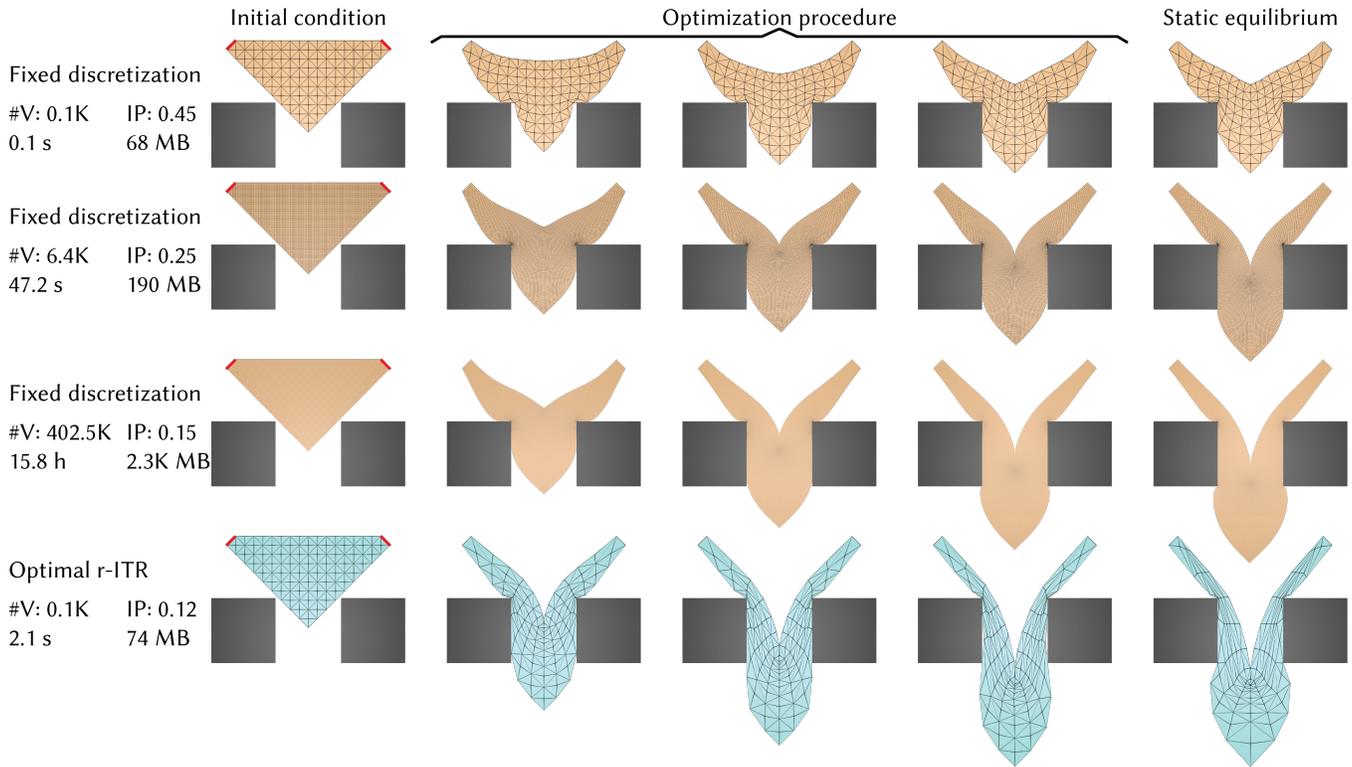


Fig. 17. **2D static drop-through.** We drop a 2D elastic wedge, pinned along its left and right upper edges (red lines), through a tight frictionless gap, and solve to static equilibrium and here show solver steps towards equilibrium. Here, fixed-discretization simulations lock on top of the obstacle with coarser meshes, and then begin to pass through as we go to finer meshes, but still remain far from a low energy compliant solution. In contrast, our optimal r-ITR passes the gap with the coarsest-mesh resolution, with a corresponding lower Incremental Potential than the 402.5K fixed-discretization which requires 27,000X longer compute time and 31X more memory usage.

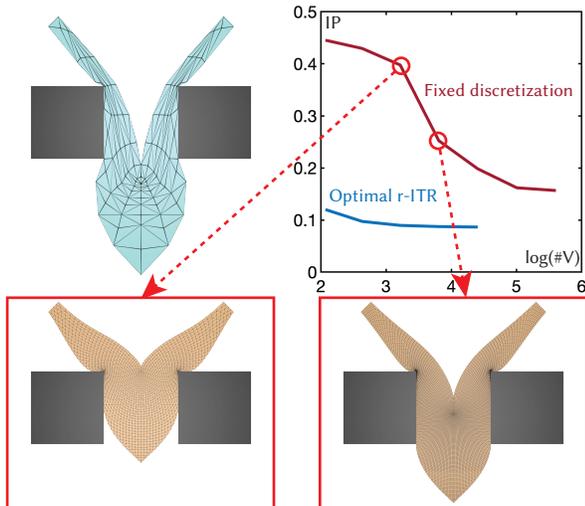


Fig. 18. **Plot of 2D static drop-through.** We plot the Incremental Potential curve with respect to the number of vertices. For the same mesh resolution, optimal r-ITR always converges to a lower Incremental Potential. We observe that there is a sudden jump on the curve of fixed-discretization simulation; this is because the fixed-discretization simulations are locked at lower resolution, and pass through at the higher.

the fixed discretization fuzzes around the contacting letter-press boundaries. Correspondingly, we see that the strain concentrations for the optimal r-ITR solution are much more sharply and accurately resolved, while energies are (as expected) significantly lower, see Figure 23 right.

### 6.3 ITR Comparisons

Next, we apply Ferguson et al.'s [2023] Masticator benchmark to compare optimal r-adaptive ITR with Ferguson et al.'s original ITR formulation, and fixed-mesh baselines.

In Figure 3 bottom, we first run the original Masticator benchmark from the ITR paper's code release and see that the crushed neo-Hookean elastic block simulates across timesteps with an average mesh size of 6K vertices, and ranges up to meshes with a maximum of 21K vertices. During simulation, please also see our accompanying videos, we observe that ITR generates a range of unnaturally sharp localized edges, dimples and folds during simulation. This is in contrast to a baseline 71K vertex fixed-mesh Masticator simulation, Figure 3 middle, which smoothly folds, curves, and bulges under compression of the Masticator's teeth. Correspondingly a 700 vertex optimal-ITR simulation, Figure 3 top, closely follows the 71K fixed-mesh simulation with generally a 3X smaller Hausdorff distance (e.g., 0.05m for optimal ITR and 0.16m for ITR in the Figure's current frame) than original ITR. Here optimal ITR uses orders of magnitude

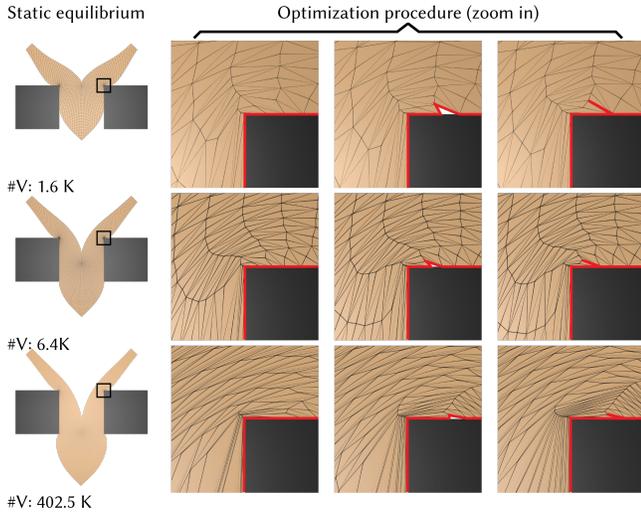


Fig. 19. **Locking behavior of fixed-discretization simulations in the 2D static drop-through.** Here, we zoom-in to sharp contact corners and demonstrate the locking behavior during static solve steps for fixed-discretization simulations. Boundary edges are highlighted in red. As Newton iteration proceeds, the fixed-discretization simulation suffers from a repeating locking pattern where elements rotate against each other, form tight contact seams and release (see our supplemental video for animation of this process). The locking mitigates but remains as we refine fixed-discretization simulations up into a 400K mesh simulation.

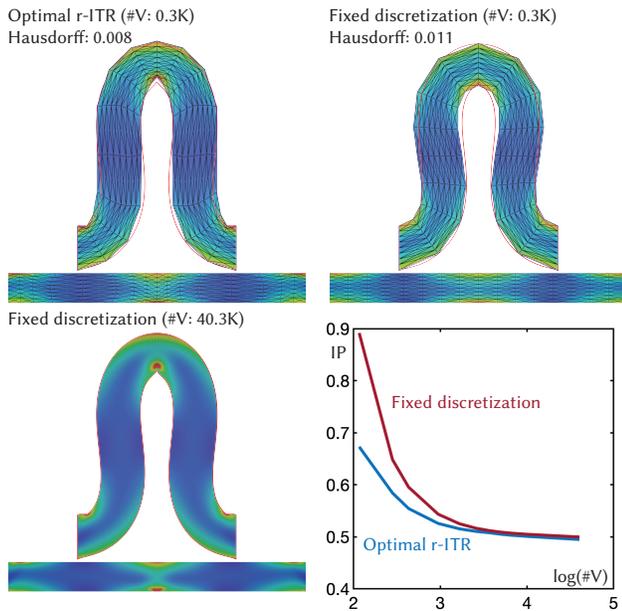


Fig. 20. **Bending beam convergence.** We bend a neo-Hookean elastic bar by moving its two ends together and solve for static equilibrium; bottom of each sub-figure is the corresponding reference mesh. We plot strain distribution ranging from high (red) at creases where optimal ITR concentrates fixed discretization DOF for significantly faster convergence under refinement.

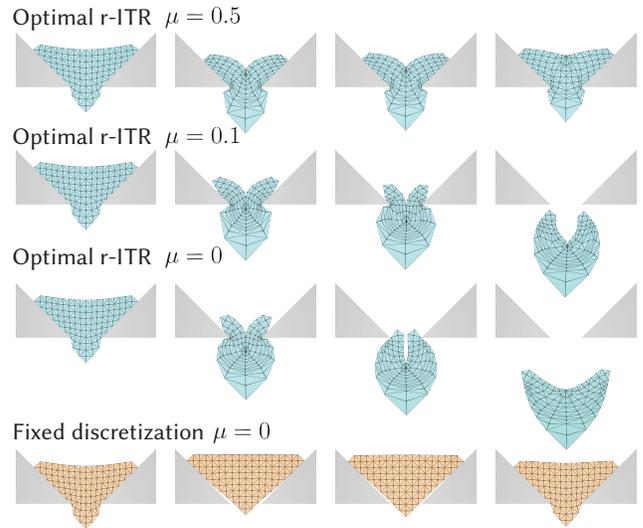


Fig. 21. **Adaptivity under varying friction.** We drop an elastic wedge through a tight angled gap. At lower friction, ITR captures the sliding of the deformed wedge through the gap at increasing speed as  $\mu$  decreases. At higher friction the wedge can not slide fully through, with adaptivity enabling the simulation to capture the final configuration of the wedge caught between gap corners. In contrast, without adaptivity, even the frictionless wedge is unable to pass through and simply ends at rest, without significant deformation, against the obstacle.

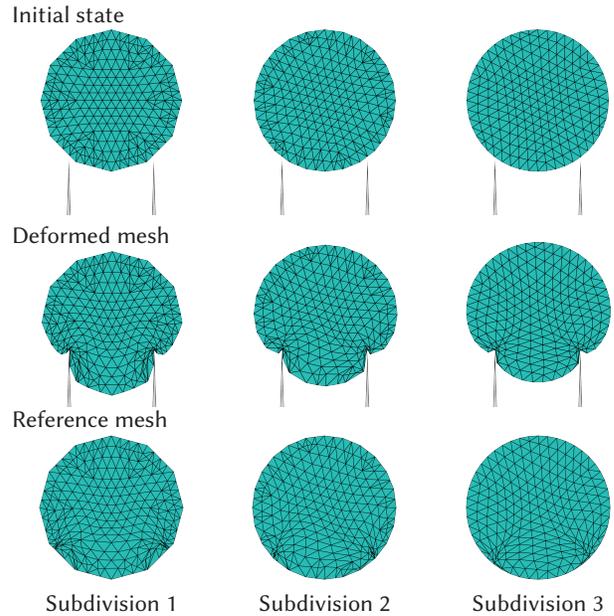


Fig. 22. **On-boundary adaptivity reduces with curvature complexity.** Here we drop onto spikes three similar shapes constructed to have the same number of boundary vertices and approximately the same number of total vertices. As curvature complexity increases we see that our boundary preserving assumption locks more boundary nodes and so increasingly limits the amount of r-adaptivity possible on the body’s surface.

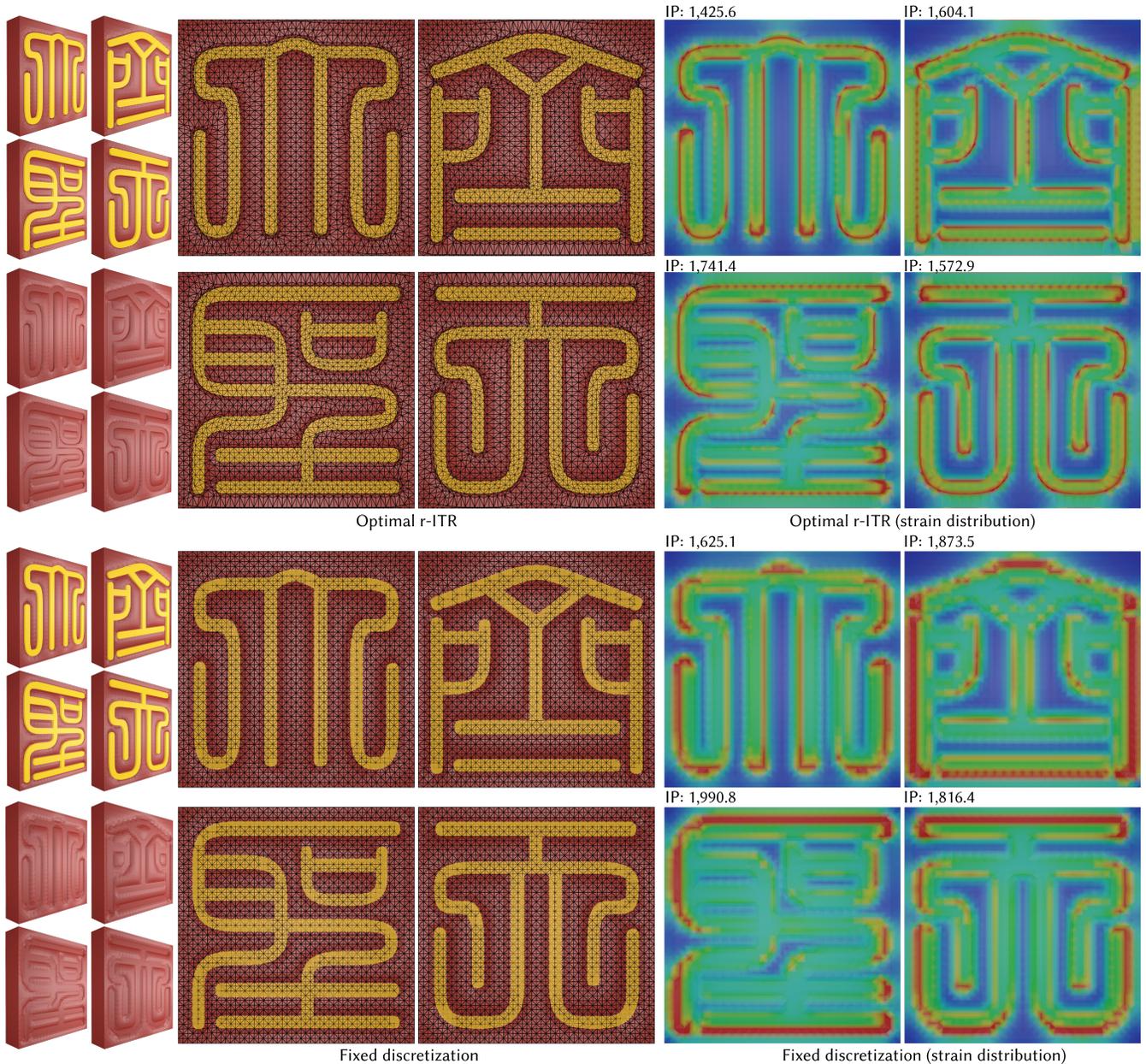


Fig. 23. **Stamping.** We press four rigid Chinese letter stamps (qí tiān dà shèng, the title of the famous Chinese mythological character Sun Wukong) into four elastic foams. Optimal r-ITR sharply captures the contour of the stamps’ strokes, while fixed-discretization simulation produces stampings with fuzzy boundaries. With the wireframe views we demonstrate that edges in the optimal r-ITR solutions’ meshes sharply align with the boundaries of the strokes (transparent yellow areas); in contrast, there are many non-conforming contacting edges in the fixed-discretization solutions. We also show the resulting strain distributions of all results; here we see that optimal ITR correspondingly resolves the sharp strain concentrations that the fixed-discretization simulations miss.

less DOF and memory (with a resultant 8.4X speedup) than the fine-mesh simulation, and 8.5X less DOF, with a 73X speedup over ITR.

At the same time, in Figure 4 we see that, if we further coarsen optimal ITR’s available resolution to 500 vertices, the solution certainly degrades with less resolution to capture curvature. However, the overall simulation behavior remains close to the finer-resolution optimal-ITR model. This is in contrast with corresponding-resolution

fixed mesh discretizations in Figure 4 (please also see our supplemental videos for animations) which both exhibit different simulation artifacts, and so likewise different overall deformation behaviors.

#### 6.4 Scaling, Performance and Convergence

Here we summarize takeaways on the scalability and performance of optimal ITR and fixed-discretization simulation. In our small

2D high-speed impact experiment (Figures 13 and 14), optimal r-ITR generates contours and strain distributions closely matching the fixed-discretization solution with 35X finer resolution, with a 1.3X speedup. In the 3D drop through experiment (Figure 2), optimal ITR’s trajectory with 0.2K vertices closely matches the fixed discretization solution with 113K vertices, which at 565X finer resolution requires a 110X slower compute time and 18.5X more memory usage. The fixed discretization fails to match for all lower mesh resolutions evaluated, which implies good application of optimal ITR for predicting high-fidelity simulation results with coarse meshing. In the masticator benchmark (Figure 4), we demonstrate that under refinement, optimal ITR can generate consistent dynamics (by improved mesh allocation) across some resolutions.

*Convergence.* In the deep-punch experiment (Figure 16), we run both optimal ITR and fixed discretization simulations across increasing mesh resolutions. We evaluate their resulting Incremental Potential curve and see in Figure 16 that optimal ITR converges much faster than the fixed discretization meshes, with a significant advantage in terms of timing and memory. Even more extreme, in the static drop-through experiment (Figure 17), we see that optimal ITR with the coarsest mesh converges to a lower energy than all simulated fixed-discretizations up to a finest mesh with 4025X more vertices. We explore this phenomena, and see that fixed-mesh discretizations across resolutions can and will get partially or fully locked at sharp contacts. We see this clearly in the sudden jump in our energy plot in Figure 18. While this locking mitigates to some degree as we go to finer resolution, we also see that it remains with significant simulation error at very fine meshing (Figure 19). Finally, in Figure 20, we bend a neo-Hookean, 2D elastic straight beam by moving its two ends close to each other, and solve for static equilibrium. We see that optimal ITR, even at coarse resolution, automatically clusters vertices into large strain (red) regions, so that its strain distribution and contour are both closer to the finest-resolution solution. Under refinement we see that optimal ITR likewise converges at a faster rate than uniform refinement for the same allocated number of nodes.

## 6.5 Additional Examples:

*Gingerbread Squish:* In Figure 5 we stress test an elastic “gingerbread” shape by tightly squishing it between compressing sharp teeth. Here we visualize both the deformation and the reference domain meshing and strain. As the teeth begin to press, we see adaptivity initially focusing DOF to capture localized contact regions. Then, as compression increases, adaptivity further concentrates DOF along internal stress bands. Finally as the teeth release backwards we see adapted meshing regions relaxing to more regular shapes away from the persistent contacts.

*Chicken Drop.* In Figure 10, we drop a soft (neo-Hookean material,  $E = 4e4$  Pa) chicken marshmallow, at large timestep ( $h = 0.01s$ ), onto stiff ( $E = 2e8$  Pa) sharp spikes. As the chicken drops, compresses, vibrates, and finally bounces off, we observe optimal ITR dynamically adjusts the meshing to enable capturing tight compliant contacts with the spikes and elastic follow through as the spikes rebound.

*Jelly.* In Figure 1, we spike a jelly treat (neo-Hookean material,  $E = 2e6$  Pa) on top of a plate with cat-shaped jelly mold, with a metal fork. Comparing optimal ITR with fixed discretization we see accurate capture of contact crease lines and clear separation of contact into separate punctures by all four fork tines. This is in contrast to fixed discretization which is unable to separate the punctures nor capture the contact shapes. In cutaway we see the strain distribution on the cross-section where optimal ITR concentrates strain within the deforming contact regions, and so automatically clusters mesh vertices in those regions. Again, this in contrast to fixed discretizations blurring of both the contact and its stresses. We see a similar story on the bottom of the jelly, where optimal ITR captures the sharp cat-shaped contour and corresponding strain of the jelly mold, while the fixed discretization again only captures a blurred deformation and strain map.

## 7 Discussion, Limitations and Conclusion

We have presented a new objective, model problem and optimization, to enable, per timestep, joint adaptive remeshing and state updates for the simulation of frictionally contacting elastica. Applying this simulation pipeline to r-adaptive modeling of dynamics and statics, we demonstrated its ability to significantly improve in quality and accuracy over the original ITR method and, moreover, to capture comparable and often better simulation results than those generated by direct, unadapted simulations using orders-of-magnitude more mesh DOF. As covered in earlier sections, there also remains a range of opportunities and limitations to next address.

For robust computation of  $L^2$  evaluations and derivatives we propose new quadrature domains and higher-density arrangements. While we find that these quadratures work well, they may certainly be doing too much or, possibly in some cases, too little sampling depending on local deformation and meshing configuration. Preliminary experiments in *adpating* quadrature with meshing are promising, but significant challenges remain in order to maintain efficient convergence behavior while updating quadratures. Likewise, our on-boundary r-adaptivity currently seeks to preserve the piecewise-surface structure of our input triangulated reference-mesh boundaries. This increasingly limits the amount of possible on-boundary adaptivity for surface geometries with higher curvature complexity (see e.g., Figure 22). Future work in balancing boundary preservation with more flexible adaptivity, to refine locally to smoother surface representations is an exciting direction to extend boundary r-adaptivity. However, this comes with challenges in terms of preserving invariants.

We have so far focused on ensuring a first model and method to provide numerically accurate, converged ITR solutions for simulation. We currently do this, as covered above, at significant additional cost in inner iteration IP solves. This is quite likely overkill, as most iterations of our ITR optimization, especially early ones, do not generally require tight solutions of the inner IP solves. Early explorations on fast approximate inner IP solves in an inexact Newton fashion are promising, but more exploration is required here to further accelerate ITR timestep solutions. Despite these computational challenges we have seen that exceedingly low-resolution optimal ITR solves can now capture deformation behaviors of much finer

mesh simulations. As such it is also exciting to now consider extending the application of our optimal ITR model and framework from r-adaptivity, to joint rh-adaptivity with adaptive updates to mesh connectivity. Here initial steps and challenges for h-adaptivity, addressed in the original ITR model [Ferguson et al. 2023], and recent work on differentiable meshing [Rakotosaona et al. 2021], both point the way towards utilizing the optimal ITR model for fully general remeshing adaptivity.

## Acknowledgments

This research was sponsored in part by the NSF (IIS-1911224), the USC Annenberg Fellowship for Jiahao Wen, and Bosch Research.

## References

- Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. 2005. Variational tetrahedral meshing. In *ACM SIGGRAPH 2005 Papers*.
- Adam W Bargteil, Chris Wojtan, Jessica K Hodgins, and Greg Turk. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph.* 26, 3 (2007).
- Jan Bender and Crispin Deul. 2013. Adaptive cloth simulation using corotational finite elements. *Computers & Graphics* 37, 7 (2013).
- Manfred Braun. 1997. Configurational forces induced by finite-element discretization. *Proc. Estonian Acad. Sci. Phys. Math* 46, 1/2 (1997).
- Tyson Brochu and Robert Bridson. 2009. Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009).
- Chris J Budd, Weizhang Huang, and Robert D Russell. 2009. Adaptivity with moving grids. *Acta Numerica* 18 (2009).
- Long Chen and Jin-chao Xu. 2004. Optimal delaunay triangulations. *Journal of Computational Mathematics* (2004).
- Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H Barr. 2001. Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.
- Leszek Demkowicz. 2006. *Computing with hp-adaptive finite elements: volume 1 one and two dimensional elliptic and Maxwell problems*. Chapman and Hall/CRC.
- Qiang Du and Desheng Wang. 2003. Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations. *International journal for numerical methods in engineering* 56, 9 (2003).
- Mario Duniach, David Vanderhaeghe, Loïc Barthe, and Mario Botsch. 2013. Adaptive remeshing for real-time mesh deformation. In *Eurographics 2013*. The Eurographics Association.
- Tobias Erhart, Wolfgang A Wall, and Ekkehard Ramm. 2006. Robust adaptive remeshing strategy for large deformation, transient impact simulations. *International journal for numerical methods in engineering* 65, 13 (2006).
- Zachary Ferguson, Teseo Schneider, Danny M Kaufman, and Daniele Panozzo. 2023. In-Timestep Remeshing for Contacting Elastodynamics. *ACM Trans. Graph.* 42, 4 (2023).
- Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics* 21, 10 (2015).
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages.
- Weizhang Huang and Robert D Russell. 2010. *Adaptive moving mesh methods*. Vol. 174. Springer Science & Business Media.
- Zhongshi Jiang, Jiacheng Dai, Yixin Hu, Yunfan Zhou, Jeremie Dumas, Qingnan Zhou, Gurkirat Singh Bajwa, Denis Zorin, Daniele Panozzo, and Teseo Schneider. 2022. Declarative Specification for Unstructured Mesh Editing Algorithms. *ACM Trans. Graph.* 41, 6 (2022).
- Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial complex augmentation framework for bijective maps. *ACM Transactions on Graphics* 36, 6 (2017).
- Bryan M Klingner, Bryan E Feldman, Nuttapong Chentanez, and James F O'Brien. 2006. Fluid animation with dynamic meshes. In *ACM SIGGRAPH 2006 Papers*.
- Rolf Krause and Patrick Zulian. 2016. A parallel approach to the variational transfer of discrete fields between arbitrarily distributed unstructured finite element meshes. *SIAM Journal on Scientific Computing* 38, 3 (2016).
- S Léger, A Fortin, C Tibirna, and M Fortin. 2014. An updated Lagrangian method with error estimation and adaptive remeshing for very large deformation elasticity problems. *Internat. J. Numer. Methods Engrg.* 100, 13 (2014).
- Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E Brown, and Laurence Boissieux. 2018. An implicit frictional contact solver for adaptive cloth simulation. *ACM Trans. Graph.* 37, 4 (2018).
- Ling Li and Vasily Volkov. 2005. Cloth animation with adaptively refined meshes. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*.
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.* 39, 4 (2020).
- S. Léger, A. Fortin, C. Tibirna, and M. Fortin. 2014. An updated Lagrangian method with error estimation and adaptive remeshing for very large deformation elasticity problems. *Internat. J. Numer. Methods Engrg.* 100, 13 (2014).
- P-L Manteaux, Christopher Wojtan, Rahul Narain, Stéphane Redon, François Faure, and M-P Cani. 2017. Adaptive physically based models in computer graphics. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library.
- William F Mitchell and Marjorie A McClain. 2014. A comparison of hp-adaptive strategies for elliptic partial differential equations. *ACM Transactions on Mathematical Software (TOMS)* 41, 1 (2014).
- JF Molinari and M Ortiz. 2002. Three-dimensional adaptive meshing by subdivision and edge-collapse in finite-deformation dynamic-plasticity problems with application to adiabatic shear banding. *Internat. J. Numer. Methods Engrg.* 53, 5 (2002).
- J Mosler and M Ortiz. 2006. On the numerical implementation of variational arbitrary Lagrangian-Eulerian (VALE) formulations. *Internat. J. Numer. Methods Engrg.* 67, 9 (2006).
- J Mosler and M Ortiz. 2007. Variational h-adaption in finite deformation elasticity and plasticity. *Internat. J. Numer. Methods Engrg.* 72, 5 (2007).
- Rahul Narain, Tobias Pfaff, and James F O'Brien. 2013. Folding and crumpling adaptive sheets. *ACM Trans. Graph.* 32, 4 (2013).
- Rahul Narain, Armin Samii, and James F O'Brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.* 31, 6 (2012).
- Tobias Pfaff, Rahul Narain, Juan Miguel De Joya, and James F O'Brien. 2014. Adaptive tearing and cracking of thin sheets. *ACM Trans. Graph.* 33, 4 (2014).
- Devon Powell. 2021. *PolyClipper*. <https://github.com/LLNL/PolyClipper>
- Marie-Julie Rakotosaona, Noam Aigerman, Niloy J Mitra, Maks Ovsjanikov, and Paul Guerrero. 2021. Differentiable surface triangulation. *ACM Trans. Graph.* 40, 6 (2021).
- Rosa María Sánchez-Banderas, Alejandro Rodríguez, Héctor Barreiro, Miguel A Otaduy, et al. 2020. Robust eulerian-on-lagrangian rods. *ACM Trans. Graph.* 39, 4 (2020).
- Nico Schlömer, Nick Papior, Rasmus Zetter, and Jan Blechta. 2024. *quadpy*. <https://github.com/sigma-py/quadpy>
- Teseo Schneider, Zachary Ferguson, and Daniel Zint. 2024. *SimpleBVH*. <https://github.com/geometryprocessing/SimpleBVH>
- Jonathan Shewchuk. 2002. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley* 2002 (2002).
- Timothy JR Simmet, Stephen D Laycock, and Andy M Day. 2009. An Edge-based Approach to Adaptively Refining a Mesh for Cloth Deformation. In *TPCG*.
- Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing inflatable structures. *ACM Trans. Graph.* 33, 4 (2014).
- Jonas Spillmann and Matthias Teschner. 2008. An adaptive contact model for the robust simulation of knots. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library.
- Keith Stein, Tayfun E Tezduyar, and Richard Benney. 2004. Automatic mesh update with the solid-extension mesh moving technique. *Computer Methods in Applied Mechanics and Engineering* 193, 21-22 (2004).
- Shinjiro Sueda, Garrett L Jones, David IW Levin, and Dinesh K Pai. 2011. Large-scale dynamic simulation of highly constrained strands. In *ACM SIGGRAPH 2011 papers*.
- Vasileios Vavourakis, Dimitrios Loukidis, Dimos C Charnpis, and Panos Papanastasiou. 2013. Assessment of remeshing and remapping strategies for large deformation elastoplastic finite element analysis. *Computers & Structures* 114 (2013).
- Julien Villard and Houman Borouchaki. 2005. Adaptive meshing for cloth animation. *Engineering with Computers* 20 (2005).
- Nicholas J Weidner, Kyle Piddington, David IW Levin, and Shinjiro Sueda. 2018. Eulerian-on-lagrangian cloth simulation. *ACM Trans. Graph.* 37, 4 (2018).
- Jiahao Wen, Jiong Chen, Nobuyuki Umetani, Hujun Bao, and Jin Huang. 2020. Cosserrat Rod with rh-Adaptive Discretization. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library.
- Martin Wicke, Daniel Ritchie, Bryan M Klingner, Sebastian Burke, Jonathan R Shewchuk, and James F O'Brien. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph.* 29, 4 (2010).
- Fredie D Witherden and Peter E Vincent. 2015. On the identification of symmetric quadrature rules for finite element methods. *Computers & Mathematics with Applications* 69, 10 (2015).
- MG Zielonka, M Ortiz, and JE Marsden. 2008. Variational r-adaption in elastodynamics. *International journal for numerical methods in engineering* 74, 7 (2008).