

Variational Elastodynamic Simulation

LETICIA MATTOS DA SILVA, Massachusetts Institute of Technology, USA

SILVIA SELLÁN, Massachusetts Institute of Technology, USA and Columbia University, USA

NATALIA PACHECO-TALLAJ, Massachusetts Institute of Technology, USA

JUSTIN SOLOMON, Massachusetts Institute of Technology, USA



Fig. 1. It’s flying rubber! Our method allows for the simulation of elastic objects, such as this rubbery bouncy blob.

Numerical schemes for time integration are the cornerstone of dynamical simulations for deformable solids. The most popular time integrators for isotropic distortion energies rely on nonlinear root-finding solvers, most prominently, Newton’s method. These solvers are computationally expensive and sensitive to ill-conditioned Hessians and poor initial guesses; these difficulties can particularly hamper the effectiveness of *variational* integrators, whose momentum conservation properties require reliable root-finding. To tackle these difficulties, this paper shows how to express variational time integration for a large class of elastic energies as an optimization problem with a “hidden” convex substructure. This hidden convexity suggests uses of optimization techniques with rigorous convergence analysis, guaranteed inversion-free elements, and conservation of physical invariants up to tolerance/numerical precision. In particular, we propose an Alternating Direction Method of Multipliers (ADMM) algorithm combined with a proximal operator step to solve our formulation. Empirically, our integrator improves the performance of elastic simulation tasks, as we demonstrate in a number of examples.

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

Additional Key Words and Phrases: Time integration methods, convex optimization, ADMM

Authors’ Contact Information: Leticia Mattos Da Silva, Massachusetts Institute of Technology, Cambridge, USA, leticiam@mit.edu; Silvia Sellán, Massachusetts Institute of Technology, Cambridge, USA and Columbia University, New York City, USA, silviasellan@cs.columbia.edu; Natalia Pacheco-Tallaj, Massachusetts Institute of Technology, Cambridge, USA, nataliap@mit.edu; Justin Solomon, Massachusetts Institute of Technology, Cambridge, USA, jsolomon@mit.edu.

SIGGRAPH Conference Papers ’25, Vancouver, BC, Canada

© 2025 Copyright held by the owner/author(s).

This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers ’25), August 10–14, 2025, Vancouver, BC, Canada*, <https://doi.org/10.1145/3721238.3730726>.

ACM Reference Format:

Leticia Mattos Da Silva, Silvia Sellán, Natalia Pacheco-Tallaj, and Justin Solomon. 2025. Variational Elastodynamic Simulation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers ’25), August 10–14, 2025, Vancouver, BC, Canada*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3721238.3730726>

1 Introduction

An elegant property of classical mechanics is that it admits a wide variety of formulations and interpretations. Most prominently, Newton’s laws determine the state of a system by evolving a second-order system of differential equations. An equivalent *variational* formulation provided by Lagrangian mechanics shows that the equations of motion can be understood as critical points of an *action functional*, which maps entire trajectories to scalar values. This formulation helps derive conserved quantities of physical interest like the Hamiltonian and momenta.

In the discrete setting, myriad time integration methods have been proposed to solve the equations of motion, with the goal of evolving the system’s state over discrete time steps while maintaining stability and accuracy. For instance, implicit Euler integration is used to take large time steps stably, with first-order accuracy. Explicit Runge-Kutta methods offer higher accuracy under moderate time step restrictions for stability. These well-known integrators, however, do not conserve physical invariants: implicit integrators like Euler’s introduce numerical damping, losing energy over time even in the absence of dissipative forces, while explicit integrators like Runge-Kutta can increase the energy of the system over time.

Inspired by these shortcomings, variational time integrators adapt the Lagrangian formulation, exploiting discrete Hamilton principles, i.e., discrete principles of stationary action, to conserve momenta



Fig. 2. A comparison between our approach and other time integrators. A method with approximate energy conservation is needed to realistically capture the periodic behavior of an elastic ball bouncing off of a rigid floor.

and approximately conserve energy. Implicit variational integrators typically rely on iterative nonlinear solvers such as Newton’s method to advance the state of the system. Thus, despite their conservation properties, these methods can be computationally expensive, be sensitive to the choice of initial conditions for the root-finding procedure, and lack convergence analysis.

In this paper, we propose an alternative to classical nonlinear solvers used in variational integrators for elastodynamic simulation. In particular, rather than expressing time integration using a generic root-finding or nonconvex optimization problem, we derive an equivalent optimization-based formulation that is jointly convex in all of its variables except the rotational component of the unknown deformation gradient. Leveraging this reformulation, we provide an Alternating Direction Method of Multipliers (ADMM) algorithm combined with a simple proximal operator step to solve our proposed formulation. By leveraging this hidden convexity, our approach comes with rigorous convergence analysis and faster convergence. Empirically, it provides increased scalability with the possibility of parallelization, as well as robustness to ill-conditioning.

Our optimization procedure employs convex subproblems that guarantee locally inversion-free deformation, using convex per-element semidefinite constraints. Thanks to our use of variational integration, it preserves important physical invariants like momenta up to the tolerance of the solver; other quantities like energy that are not preserved exactly by variational integration still are approximately preserved in our experiments. Our method applies to several tasks from physics-based simulation and animation, including dynamical deformation with nonlinear distortion energies and penalty-based external conservative and non-conservative forces. The main contributions of our work are as follows:

- We reformulate time integration for elastodynamics as an optimization problem, leveraging convexity properties of the elastic potential and kinetic energy functions. Our formulation is jointly convex in all motion variables except the rotational component;
- Based on our reformulation, we propose an algorithm for time integration using the ADMM technique;
- We propose a closed-form proximal operator approach to incorporate external forces into our algorithm that is computationally cheap and does not change the energy/momentum conservation.
- We apply our solver to a number of isotropic distortion energies used in elastic simulation, demonstrating stability and efficiency.

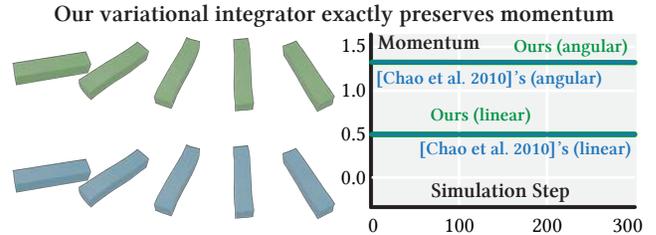


Fig. 3. Our method shares the same conservation of momenta property as the variational integrator in [Chao et al. 2010].

2 Related Work

Geometric distortion energies, such as the as-rigid-as-possible (ARAP) [Sorkine and Alexa 2007] and symmetric Dirichlet [Schreiner et al. 2004] energies, are among the most common constitutive models incorporated into the simulation of 2D and 3D shapes.

Many of these energies have their history in geometry processing. In particular, minimizing them yields appealing results in a number of static tasks from mesh parametrization [Liu et al. 2008; Rabinovich et al. 2017] to volume correspondence [Abulnaga et al. 2023] and shape deformation [Jacobson et al. 2012; Smith et al. 2019].

Distortion energies are typically nonlinear and nonconvex, which yield numerical challenges in optimization. To address this issue, several recent works on quasi-static problems [Brown and Narain 2021; Stein et al. 2022] optimize distortion energies via ADMM, a local-global algorithm that is relatively simple to implement and easy to parallelize. Most relevant, our dynamical formulation leverages the hidden convexity property of isotropic distortion energies observed by Stein et al. [2022] for static geometry processing problems. We show that a similar convexity property can be found in symplectic integrators for dynamical problems, after reshaping these integrators into a particular optimization-based form.

Beyond quasi-static problems, geometric distortion energies can be used as elastic potentials in models for time-evolving deformable volumes. Time integration of these physics problems often yields optimization and root-finding problems similar to those encountered in the quasi-static setting. A multitude of such methods have been proposed to time step with stability and efficiency. Smith et al. [2019], for instance, integrate their ARAP energy using the semi-implicit scheme introduced by Baraff and Witkin [1998]. This and other methods based on implicit Euler are unconditionally stable, at least in the linear case. They require solving large nonlinear root-finding problems, however, hindering performance; these implicit methods can also introduce artificial damping (see Figure 2). Closely related to

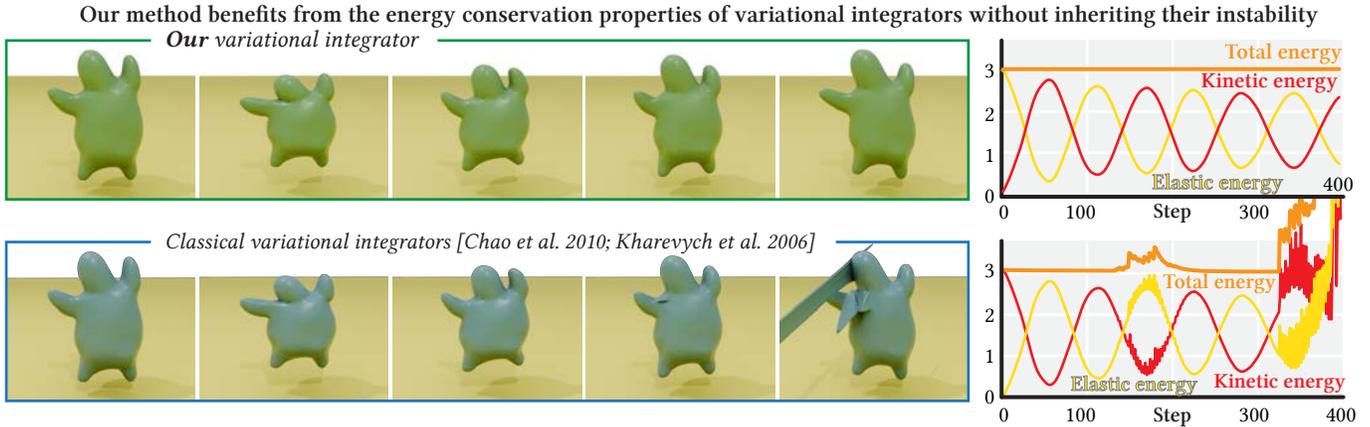


Fig. 4. We can simulate this bouncy animated character using ARAP energy. Our simulation is stable and obtains almost exact energy preservation.

implicit Euler integration is position-based dynamics (PBD) [Müller et al. 2007] and its extension XPBD [Macklin et al. 2016].

Another line of work is projective dynamics [Bouaziz et al. 2014], which proposes a fully-implicit time integration scheme. Their formulation solves problems with energy potentials satisfying specific properties that allow for a local-global solve. While more efficient than Newton’s and extremely popular in graphics, these methods require particular potentials, e.g., corotated elasticity, for optimal performance and struggle with numerical damping. To address some of these limitations, Liu et al. [2017] apply L-BFGS in the projective dynamics framework to enable faster convergence for hyperelastic materials. Narain et al. [2016] and Overby et al. [2017] tackle general elastic potentials via a local-global strategy and provide an ADMM scheme to solve for the state of the system. Similarly to projective dynamics, however, they do not approximately conserve energy; as noted in §5.1, their ADMM splitting also differs substantially from ours.

Other time integration methods include variational ones, which use Lagrangian principles to derive integrators with exact conservation of momenta and excellent energy behavior. We refer the reader to [Hairer et al. 2006] and [Marsden and West 2001] for an introduction to this class of integrators. In graphics, a number of variational integrators have been proposed [Chao et al. 2010; Kharevych et al. 2006; Martin et al. 2011; Stern and Grinspun 2009].

Variational principles also appear in contact dynamics, where potential terms are added to penalize constraint noncompliance [Kaufman and Pai 2012]. Incremental Potential Contact (IPC) [Li et al. 2020] has been widely used to handle collisions in the simulation of large deformations. A common approach to time integration for IPC is using implicit Newmark, which was shown to be symplectic by Kane et al. [2000]. Their approach to incorporating constraints, however, breaks the symplecticity of the time integrator. Furthermore, while IPC provides fast and robust performance in complex contact scenarios, it does not preserve physical invariants, including total energy (see Figure 13). To improve robustness and frictional accuracy in contact-rich scenarios, Larionov et al. [2024] replace IPC’s lagged friction formulation with an implicit model based on

soft constraints. Several papers build upon the barrier-function formulation introduced in IPC to provide efficient GPU-accelerated solvers [Chen et al. 2024c; Lan et al. 2023, 2022].

Perhaps most relevant to our work is the integrator by Kharevych et al. [2006], which was subsequently used by Chao et al. [2010] to perform simulations using their modified ARAP energy. Their implicit formulation recasts time stepping as an energy minimization, as opposed to root-finding. Their approach suffers from the drawbacks of Newton’s method, which lacks guarantees of global convergence and is sensitive to ill-conditioned Hessians.

Our work leverages the advantages of variational techniques to derive a time integrator with strong momentum and energy conservation, circumventing a disadvantage of classical integration methods. While past variational integrators depend on generic iterative solvers, our reformulation allows us to leverage algorithms typical for convex or nearly-convex optimization with rigorous convergence analysis and enforcement of element-wise inversion-free properties via semidefinite constraints.

3 Background

Our work focuses on the dynamical simulation of objects with isotropic distortion energies. Here, we establish notation and the mathematical constructions needed in our approach to this problem.

3.1 Spatial Discretization

Suppose $\mathcal{M} \subset \mathbb{R}^3$ is a volume at a reference configuration, and suppose we discretize \mathcal{M} as a tetrahedral mesh Ω , with vertices whose positions are stored in a matrix $q \in \mathbb{R}^{n \times 3}$ and m tetrahedra T . We can discretize vector valued functions $\mathcal{M} \rightarrow \mathbb{R}^3$ on the mesh, such as velocity v , using one vector in \mathbb{R}^3 per vertex.

In physics-based simulation, the typical discretization is finite element method (FEM). Take $M \in \mathbb{R}^{n \times n}$ to be the mass matrix and $D : \mathbb{R}^{n \times 3} \rightarrow (\mathbb{R}^{3 \times 3})^m$ be the deformation gradient operator. The Jacobian of the mapping from the reference configuration \tilde{q} to a deformed configuration q , restricted to each tetrahedron i , is given by $J_i = D_i q$, where $D_i q$ is the i^{th} component of Dq .

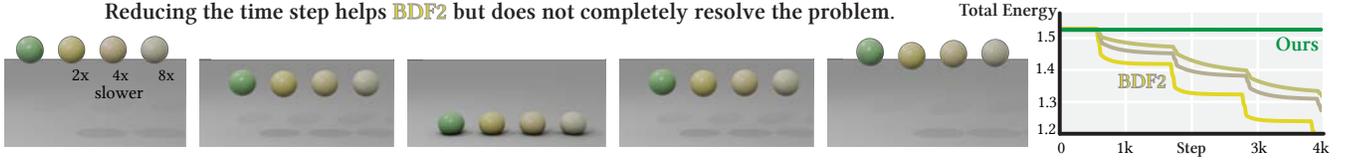


Fig. 5. Second-order backward differential formula (BDF2) preserves energy better than other classical methods, but still includes damping. Here, we decrease the time step size used with our method ($h = 0.004$) by 2, 4 and 8 \times to show the effect on total energy using BDF2 over the same time interval.

From Lagrangian mechanics, we can understand the dynamics of the deformable volume Ω as a critical point of the action functional with respect to an unknown $q(t)$, which describes the configuration of the mesh as a continuous function of the time t . We define the action functional as

$$\mathcal{S}[q] = \int_0^1 [K(\dot{q}(t)) - E(Dq(t))] dt, \quad (1)$$

where E is a potential, typically nonlinear, that depends only on the deformation gradient $Dq(t)$ and K is the kinetic energy $K(v) = \frac{1}{2}v^T Mv$. The first variation of \mathcal{S} yields Newton’s second law

$$M\ddot{q}(t) = -\nabla_q E(Dq(t)). \quad (2)$$

Below, we outline typical methods of solving for (2) in time.

3.2 Numerical Integration

We can rewrite equation (2) as a coupled pair of first-order equations:

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= -M^{-1}\nabla_q E(Dq). \end{aligned} \quad (3)$$

Suppose we are given a reference configuration $\tilde{q} \in \mathbb{R}^{n \times 3}$ as well as the initial velocity $\tilde{v} \in \mathbb{R}^{n \times 3}$, and we wish to advance the configuration forward in time by approximating the solution to (3). Given a time step $h > 0$, we use q^k and v^k to denote our approximations of the configuration and velocity, respectively, at time hk for $k \in \mathbb{N}$.

Explicit Euler. The simplest integration technique is forward Euler, which imposes a strict upper bound on h for stability. Higher-order variations of this integrator, such as Runge-Kutta, improve its stability and accuracy, but almost all require many small time steps to avoid instability; even with small time steps, underlying physical properties like total energy can diverge (see Figure 2).

Implicit Euler. This integrator is unconditionally stable and advances the solution of (3) with first-order accuracy in h , which suffices for small h . Accuracy in approximating the physical properties of the system, however, is traded-off for stability, as these solvers introduce noticeable numerical damping. When higher accuracy is required, a typical choice of implicit integration is the second-order backward differentiation formula (BDF2), which is also unconditionally stable and exhibits less dissipation (see Figures 2 and 5).

Variational Integration. The integrators above violate a key property of dynamics: physical invariants of a dynamical system, such as energy and momenta, are not conserved. Variational integrators correct the behavior of these invariants by deriving a time stepping scheme directly from the principle of least action, which corresponds to critical points of the action functional in (1). To be

Our method avoids instabilities

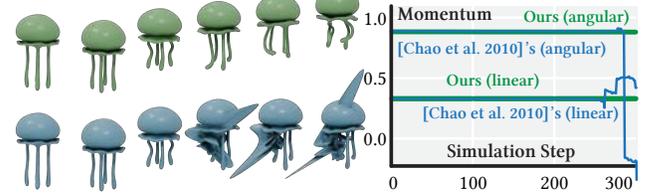


Fig. 6. A jellyfish moves and spins with exact preservation of momenta, avoiding the instability of prior methods.

concrete, variational integrators often discretize (1) using one-point quadrature as follows:

$$S^d(q^{k+1}, q^k, h) = \sum_{k+1} h \left[K(v^{k+1}) - E(D(\alpha q^{k+1} + (1-\alpha)q^k)) \right], \quad (4)$$

where $v^{k+1} = (q^{k+1} - q^k)/h$ and $\alpha \in [0, 1]$. Updates for q and v can be obtained by differentiating with respect to q^k for each k , with fixed endpoints. If $\alpha = 1/2$, then this approach yields a second-order integrator. These schemes often lead to expensive root-finding problems.

3.3 Polar Decomposition of the Jacobian

In the isotropic case, the potential energy of a deformable volume can be expressed as a function of the singular values of the Jacobian of the mapping from \tilde{q} to q . To be concrete, suppose we apply polar decomposition to each Jacobian matrix J_i via $J_i = U_i P_i$, where $U_i \in \text{SO}(3)$ is a rotation matrix and $P_i \in \mathcal{S}_+^3$ is a symmetric positive semidefinite (SPD) matrix. Then, as described by Stein et al. [2022], the potential energy of the deformed volume can be written entirely in terms of the matrices P_i :

$$E(Dq) = \sum_i w_i f(P_i),$$

where the w_i are per-tetrahedron weights and f is a per-tetrahedron distortion energy; many typical choices of f are convex over the set \mathcal{S}_+^3 of SPD matrices. These isotropic energies are dubbed P -centric invariants by Smith et al. [2019], where the corresponding eigensystems—which can be used to recover the Hessian of the energy—are derived in closed form.

The above observation suggests that exploiting convexity of f over \mathcal{S}_+^3 , which we refer to as “hidden convexity,” could yield a time integrator benefiting from convex optimization strategies. It is not obvious, however, that the hidden convexity of the distortion energy will be helpful to derive a time integrator because the discrete action S^d has one term that is convex and another that is non-convex. In

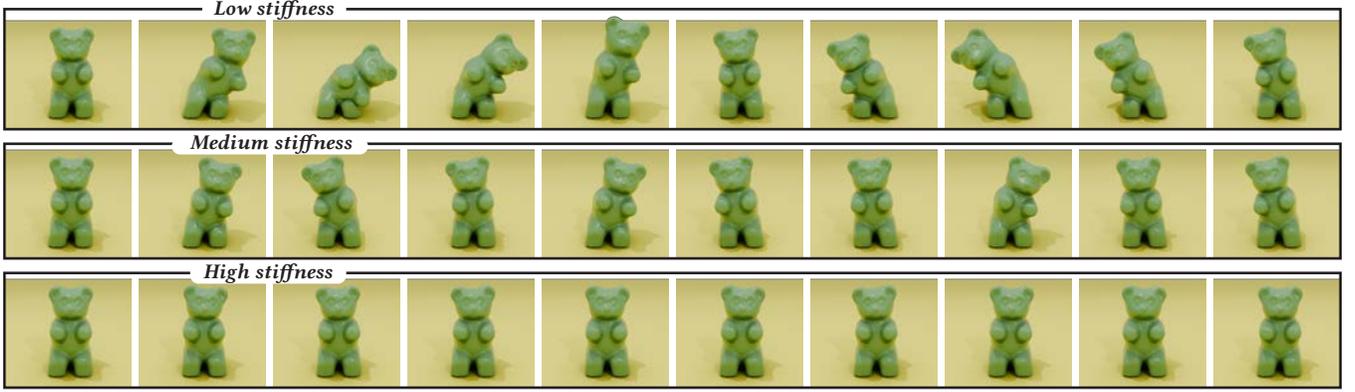


Fig. 7. By globally scaling the weights w_i , our simulation changes from modeling very flexible materials to more rigid.

what follows, we reshape the problem to reveal and leverage the hidden convexity.

4 Method

In what follows, we will show how to derive an elastodynamic time integrator that, although not jointly convex in all of its variables, still enjoys key benefits of convex optimization thanks to its joint convexity in the vertex positions and principal stretches.

4.1 Hidden Convexity in the Action Functional

Following the notation of §3.2, let $q^k \in \mathbb{R}^{n \times 3}$ be our estimate of $q(t)$ at time hk . Again, let $v^{k+1} = (q^{k+1} - q^k)/h$ be the velocity, and define $q_{\text{mid}}^{k+1} = (q^{k+1} + q^k)/2$ as the midpoint approximation. We estimate q^{k+1} by applying Hamilton's principle to the action functional combined with a constraint for the polar decomposition of the Jacobian.

In particular, we define a constrained action functional using midpoint discretization as follows:

$$S^d = \sum_k h \left[K(v^{k+1}) - E(P^{k+1}) \right], \quad (5)$$

subject to $D_i q_{\text{mid}}^{k+1} - U_i^{k+1} P_i^{k+1} = 0 \forall i$.

We note that the above formulation parametrizes the mesh by discretizing both configuration space and velocity on a staggered grid. Midpoint discretization is chosen to obtain second-order accuracy.

Critical points of the constrained action functional in (5) preserve the symplectic form associated to the system and thus preserve the geometry of the phase space. Thus, the resulting integrator satisfies approximate energy conservation. Moreover, by Noëther's theorem, continuous symmetries of the discrete Lagrangian yield exactly-conserved quantities. Because we use a midpoint discretization, the resulting integrator also enjoys second-order accuracy [Fetecau et al. 2003; West 2004]. We record these observations below:

PROPOSITION 4.1. *The integrator that arises by applying Hamilton's principle to (5) is second-order accurate and symplectic. If the discrete Lagrangian exhibits translational (resp., rotational symmetry), then linear (resp., angular) momentum is conserved.*

By the principle of least action, the equations of motion are given by critical points of the functional in (5). We show that these critical points minimize an optimization problem that is jointly convex in q and P (but not U):

THEOREM 4.2. *Let z^k be defined by*

$$z^k = (2q^k - q^{k-1}) - M^{-1} \frac{h^2}{2} \sum_i w_i D_i^\top U_i^k \nabla f(P_i^k). \quad (6)$$

Then, critical points of the constrained action functional in (5) are solutions to the following minimization problem:

$$\begin{aligned} \arg \min_{q, P \in S_+^3, U \in \text{SO}(3)} & K\left(\frac{q - z^k}{h}\right) + E(P), \\ \text{subject to} & D_i q_{\text{mid}} - U_i P_i = 0 \forall i. \end{aligned} \quad (7)$$

This formulation yields an optimization-based integration scheme that is jointly convex in q and P , with the nonconvexity entirely contained in U —which lives in the compact space $\text{SO}(3)$. The numerical challenges arising from the nonconvexity are now isolated in one bounded variable. This is a key insight of our work: the nonconvexity of the time variational integrator is hiding exclusively in the rotational component. This observation is shown in the proposition below.

PROPOSITION 4.3. *Let $U \in \text{SO}(3)$ be held fixed. Then, the minimization problem in (7) is jointly convex in q and P .*

Thus far, we have shown that we can uncover hidden convexity in the minimization problem in (7), whose minimizers satisfy the least action principle. As noted before, this yields a time integrator that preserves momenta and approximately preserves energy. In §5, we propose an algorithm to find these minimizers with convergence guarantees under mild assumptions.

REMARK. *If one wanted to use another optimization-based method to solve our time integration scheme, the update rule for vertex positions would be fully implicit with q^{k+1} obtained via solving the problem in (7), and the update for velocities would be given by $v^{k+1} = (q^{k+1} - q^k)/h$.*

4.2 Adding External Forces

We can incorporate conservative forces into our formulation, e.g., gravity or force functions to handle collisions of the volume with the ground (see Fig. 8), by adding a potential energy term $g(q_{\text{mid}})$ corresponding to these external forces into the Lagrangian.

Nonconservative forces, e.g., force functions to handle fixed point constraints (see Fig. 9), can also be incorporated into our formulation by including terms corresponding to the external forces in the action functional. We refer the reader to the supplemental material for the derivation of the formulations accommodating each kind of external force.

5 Algorithm

To solve the formulation in (7), we derive an optimization method based on the Alternating Direction Method of Multipliers (ADMM) [Boyd et al. 2011]. Our optimization problem has the following augmented Lagrangian

$$\begin{aligned} \Lambda = & K \left(\frac{q - z^k}{h} \right) + E(P) + \sum_i \text{Tr} (Y_i^\top (D_i q_{\text{mid}} - U_i P_i)) \\ & + g(q_{\text{mid}}) + \sum_i \frac{\rho_i}{2} \|D_i q_{\text{mid}} - U_i P_i\|_F^2 \end{aligned} \quad (8)$$

Our ADMM algorithm cycles the following steps:

- (1) Minimize Λ w.r.t. q (linear solve).
- (2) Minimize Λ w.r.t. U with added proximal term (closed form).
- (3) Minimize Λ w.r.t. P (closed form).
- (4) Update Y_i via gradient descent of step size ρ_i (closed form).
- (5) *Optional*: Rescale the dual variables ρ_i and Y_i (closed form).

Below, we detail each step in our algorithm.

Minimization with respect to q . We differentiate Λ with respect to q to obtain an update for q . In the absence of the term g , the update amounts to solving the linear system $Aq = B$, where

$$A = \frac{1}{h^2} M + \sum_i \frac{\rho_i}{4} D_i^\top D_i$$

$$\text{and } B = -\frac{1}{h^2} Mz^k - \frac{1}{2} \sum_i D_i^\top Y_i + \sum_i \frac{\rho_i}{2} D_i^\top U_i P_i - \frac{\rho_i}{4} D_i^\top D_i q^k. \quad (9)$$

The matrix A is precomputed and factored using Cholesky decomposition at the beginning and need not be updated, so long as the ρ_i are not rescaled. In this step, we also include any linear conservative forces or constraints that can be expressed entirely in q (see the supplemental material).

Minimization with respect to U . Because (5) is nonconvex in U , we must adopt an update procedure for this variable such that the ADMM algorithm still converges. To this end, we add a proximal term, as proposed by Stein et al. [2022] for a similar optimization problem, and then minimize the augmented Lagrangian with respect to U . In particular, we solve the following Procrustes problem:

$$\arg \min_{U_i \in \text{SO}(3)} \|U_i - (J_i + \rho_i^{-1} Y_i) P_i - c_i U_i^{(j)}\|_F^2, \quad (10)$$

which is decoupled for each i -th tetrahedron, and where $J_i = D_i q_{\text{mid}}$. Here, we use superscript (j) to denote the values of the variables

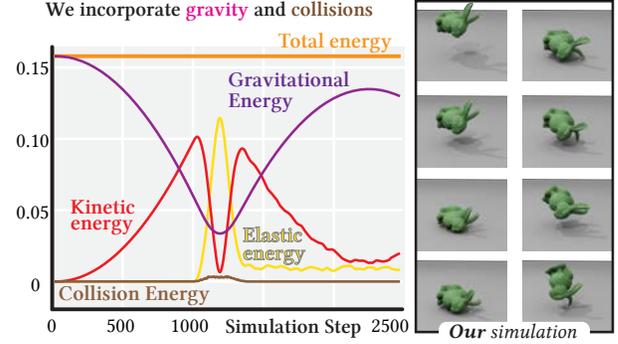


Fig. 8. An elastic bunny will exhibit internal vibrations after collision with the ground as seen by variation in the elastic energy curve. Our method’s conservation property is robust to these scenarios, maintaining the total energy approximately constant throughout the simulation.

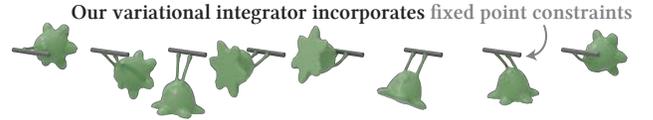


Fig. 9. Our method can effortlessly accommodate fixed point constraints, allowing a slug to swing under gravity.

at the j -th ADMM iteration. The minimizer of this problem can be computed from the signed SVD of $(J_i + \rho_i^{-1} Y_i) P_i + c_i U_i^{(j)}$.

Minimization with respect to P . Minimization of the augmented Lagrangian Λ with respect to P can be decoupled per tetrahedron i . It corresponds to the following convex problem:

$$\arg \min_{P_i \geq 0} w_i f(P_i) + \frac{\rho_i}{2} \|J_i - U_i P_i + \rho_i^{-1} Y_i\|_F^2, \quad (11)$$

where $J_i = D_i q_{\text{mid}}$. The convexity in $P \in \mathcal{S}_+^3$ implies that (11) admits a unique minimizer. Typically we can solve this optimization problem using closed-form formulas; see the supplemental material for examples.

Minimization with respect to q' . If any nonlinear terms arising from the potential term g , e.g. forces used to handle collisions with the ground, were incorporated directly into the q update, then the minimization with respect to q would require an iterative solver. Instead, we apply an optimization trick to handle the nonlinearity via a cheap (typically closed-form) proximal operator step.

In particular, we introduce a new optimization variable q' along with a constraint $q = q'$ and replace q with q' in the input to the nonlinear function g . We add the constraint and dual terms into the augmented Lagrangian and minimize with respect to q' , which amounts to solving the following proximal operator step:

$$\arg \min_{q'} g(q'_{\text{mid}}) + \frac{\mu}{2} \|q - q'\|_F^2 + \text{Tr} (W^\top (q - q')), \quad (12)$$

We refer the reader to the supplemental material for a closed-form solution to (12) when g is a Hookean penalty-based potential for collisions.

Dual variable update. The update for each of the dual variables Y_i employs the following closed form:

$$Y_i = Y_i^{(j)} + \rho_i(D_i q_{\text{mid}} - U_i P_i). \quad (13)$$

5.1 Convergence Analysis

In contrast to our proposed algorithm, iterative methods like Newton’s method, used as the standard solver for variational integrators, lack guarantees of convergence for nonconvex problems. In this section, we show that by extracting the hidden convexity of the nonconvex problem, we are able to provide a guarantee of global convergence for our algorithm under mild conditions.

In particular, we assume conditions (A1)–(A3) detailed in the supplemental material hold true. Key to obtaining a convergence result for our algorithm is a *sufficient decrease property*, which states that the gaps between values of the augmented Lagrangian Λ across iterates must decrease. The assumptions above are sufficient to verify that $\Lambda^{(j+1)} - \Lambda^{(j)}$ is bounded. We record the further conditions required to establish decrease (a direct consequence of [Stein et al. 2022, Proposition 5.4]) as follows:

PROPOSITION 5.1. *Assume conditions (A1)–(A3) are fulfilled. Then the matrix A in (9) having positive eigenvalues and each of the dual penalty parameters being sufficiently large are sufficient conditions for sufficient decrease.*

With these assumptions/sufficient conditions in hand, we state our convergence result, inherited from Stein et al. [2022, Theorem 5.5]:

THEOREM 5.2. *Assume conditions (A1)–(A3) and the sufficient conditions in Proposition 5.1 hold. Then Algorithm 1 in the supplemental material converges.*

REMARK. *Assumptions (A1) and (A2) hold so long as we follow the explicit bounds on parameters derived by Stein et al. [2022]. Indeed, these assumptions hold in our experiments (see Figure 1 in the supplemental material). The first sufficient condition in Proposition 5.1 is easily verified since A is SPD in our formulation. The second sufficient condition can be met either by deriving explicit lower bounds for each one of the dual parameters, which could include variables corresponding to the q' variable, or in practice by simply rescaling them. The only remaining requirement is assumption (A3), which can be verified analytically for a given choice of elastic potential energy.*

5.2 Potential Energy Functions

We apply our algorithm to the simulation of elastodynamics with four choices of potential energy function $f : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}$:

As-rigid-as-possible (ARAP). First introduced by Sorkine and Alexa [2007], ARAP energy measures the stretch of elements without considering locally rigid motions. Formally, the potential energy function is given by

$$f(P) = \frac{\kappa}{2} \|P - I\|_F^2 \quad (14)$$

where κ is the stiffness parameter.

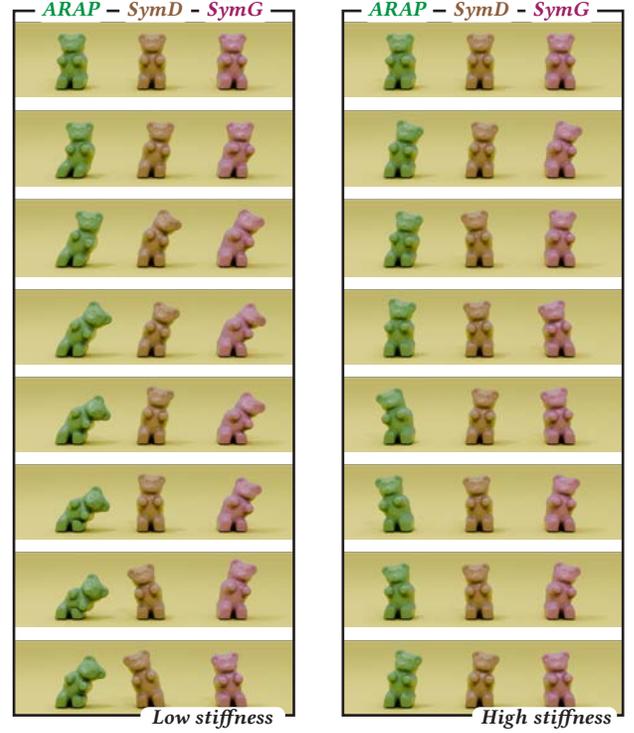


Fig. 10. Animators may choose their preferred flavor of distortion energy to incorporate into our algorithm.

Symmetric Dirichlet. Symmetric Dirichlet is the rotationally invariant distortion energy, with potential energy function given by

$$f(P) = \frac{\kappa}{2} (\|P\|_F^2 + \|P^{-1}\|_F^2) \quad (15)$$

This energy quantifies distortion by combining stretch and compression in a symmetric way and enforces flip-freeness via a singularity on the locus of zero-determinant PSD matrices.

Symmetric Gradient. Stein et al. [2022] propose a new energy, known as symmetric gradient energy, for parametrization and deformation. It is given by

$$f(P) = \frac{\kappa}{2} \|P\|^2 - \kappa \log \det(P) \quad (16)$$

This energy encourages local rigidity while remaining flip-free. In this paper, we demonstrate that the symmetric gradient energy is well-suited for physics-based animation.

Neo-Hookean. Neo-Hookean elasticity, which is often used in physics-based simulation of hyperelastic materials, is given by

$$f(P) = \frac{\kappa}{2} (\|P\|_F^2 - 3) - \kappa \log \det(P) + \frac{\lambda}{2} (\log \det(P))^2 \quad (17)$$

where κ , the stiffness parameter, and λ are the Lamé constants. The terms with a logarithm are volume-preserving penalty terms.

REMARK. *While neo-Hookean energies do not technically satisfy the assumptions of our convergence theory (except when $\lambda = 0$, in which case they do), they fit into our optimization algorithm; all that*

is needed is a modification of the proximal operator in (11). See the supplemental material for details.

6 Experiments and Applications

Implementation. The vast majority of our method’s runtime is spent iterating through the ADMM subproblems described in §5, where we leverage CPU parallelization in the local updates as well as Cholesky precomputation in the global q update. This makes the computational cost of each ADMM iteration scale linearly with the number of mesh elements, which we validate experimentally (see inset). In practice, we find that the number of iterations required for ADMM to converge varies (from single-digits to the thousands) and depends mostly on the geometric complexity of the model and its deformation, and less so on its level of refinement.

Experiments. When the Lagrangian is invariant under translations (resp., rotations), our integrator should exactly conserve total linear (resp., angular) momentum. In our setup, both of these symmetries occur, at least in the absence of gravity and anchors/ground constraints, because we choose our elastic potential energy to be isotropic. Figures 3 and 6 give two different examples of this exact conservation. The first example is of a spinning elastic bar. The second example is of a jellyfish, i.e., a more complicated geometry.

Our integrator also demonstrates long-term energy stability in scenarios where other variational integrators seem to lose stability. For instance, in Figure 4, we animate a rubber character wobbling under its own elastic potential energy. In this example, our method demonstrates long-term energy conservation, whereas the optimization procedure used in [Chao et al. 2010; Kharevych et al. 2006] becomes unstable after a few hundred time steps and eventually blows up, despite initially conserving energy. We implement their method using a Newton’s solver and ARAP energy. In Figure 11, we provide another comparison experiment against the variational integrator in [Chao et al. 2010], in which we visualize their method becoming unstable after a few hundred steps.

A similar occurrence is observed in the example in Figure 6. Here, our method exhibits long-term momentum conservation, whereas the method used in [Chao et al. 2010] becomes unstable after a

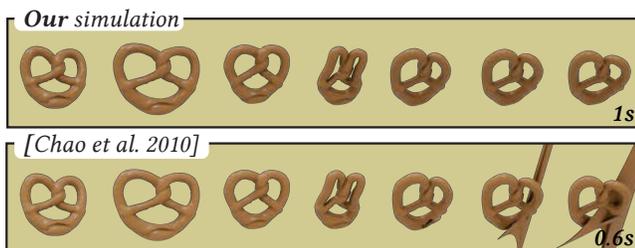


Fig. 11. Another example similar to Fig. 4 and Fig. 6, where our method outperforms another variational integrator in terms of stability. Runtimes shown are average, per simulation step.

just few hundred time steps for the jellyfish model. Preservation of physical invariants in the examples in Figure 4 and Figure 6 is a feature that should be shared by other symplectic variational integrators. Thus, to some degree, these instabilities reflect the difficulty of root-finding and Newton’s method solvers rather than the time integration scheme itself.

We also compare our method to XPBD (see Figure 12) and IPC (see Figure 13) in terms of energy conservation for the test case of an elastic cube bouncing off of the ground. While IPC provides a

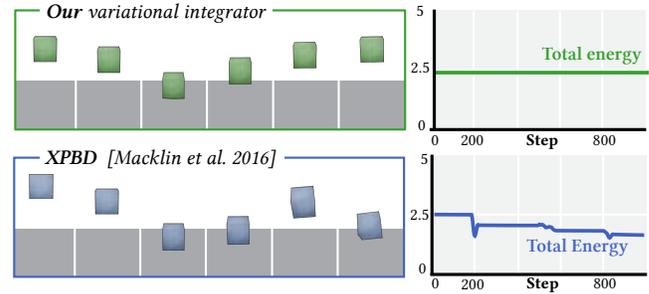


Fig. 12. In this test case, our variational integrator outperforms extended position-based dynamics (XPBD) in terms of energy conservation. In this comparison, we use ARAP energy with $\kappa = 200.0$ and time step size $h = 0.01$. The graph (bottom right) shows that XPBD includes numerical dissipation.

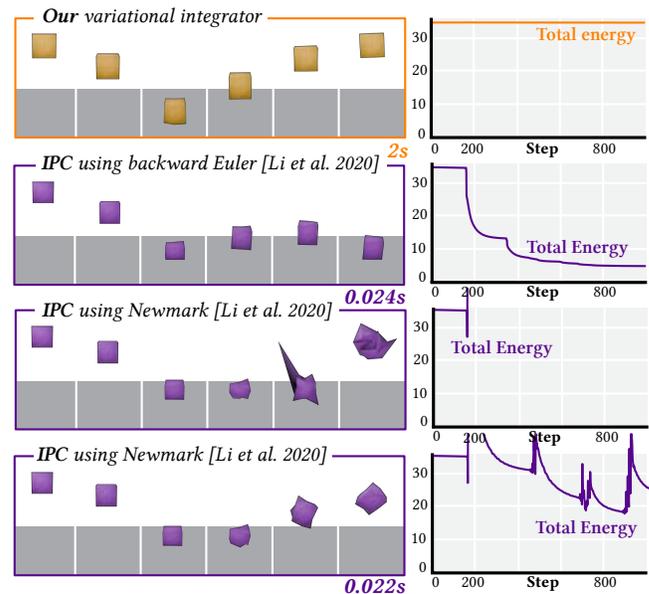


Fig. 13. Another example of a test case similar to Fig. 12. Here, we show that our method outperforms IPC in terms of total energy conservation across different time integration methods. Here, we compare to backward Euler with no friction (top), Newmark without (middle) and with lagged Rayleigh damping (bottom). We use $\kappa = 200.0$ and $\lambda = 0$ with the step size recommended for Newmark stability ($h = 0.005$). Runtimes reported are average, per time step. We use IPC’s official implementation for the comparison against our code; the latter is not fully optimized for speed.

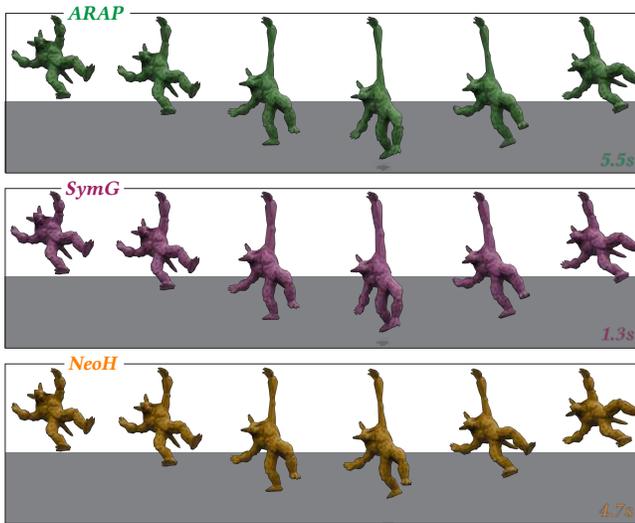


Fig. 14. We show the effectiveness of our integrator for a larger deformation than in other experiments. Here, the armadillo arm is stretch to up to $3\times$ its rest length under the force of gravity. We use stiffness $\kappa = 35.0$ and $\lambda = 4\kappa$. Runtimes shown are average, per time step.

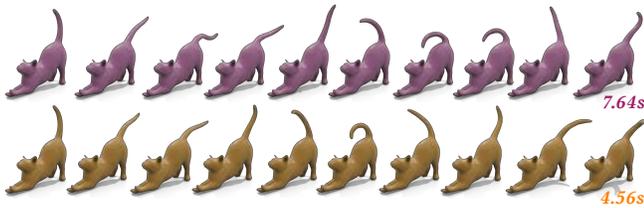


Fig. 15. An experiment demonstrating that the energies included in our method create compelling animations. Here, we show a cat wagging its tail with symmetric gradient (top) and neo-Hookean (bottom). We use $\kappa = 35.0$ and $\lambda = 4\kappa$. The neo-Hookean energy creates a more realistic motion.

significant runtime advantage, it exhibits noticeable damping (when using backward Euler integration) and a combination of damping and spurious energy spikes (when using Newmark integration). In the comparison to XPBD, we observe their method introduces dissipation, which is to be expected given its close connection to implicit Euler integration.

Applications. Our method can be used in a variety of examples of interest to animators. Crucially, its performance is not affected by making different physical choices or by using different input parameters. In Figure 10, we simulate a gummy bear stuck to the ground, oscillating elastically from side to side under three different potentials: ARAP, symmetric Dirichlet (SymD), and symmetric gradient (SymG). A key advantage of our method is that it can be easily applied to simulate elastic objects governed by other potential energies (so long as they are convex in P). While neo-Hookean (NeoH) energies do not satisfy the convexity assumption except when $\lambda = 0$, empirically, our algorithm achieves convergence for a number of larger λ ; we provide examples in Figures 14, 15, 16 and 17.

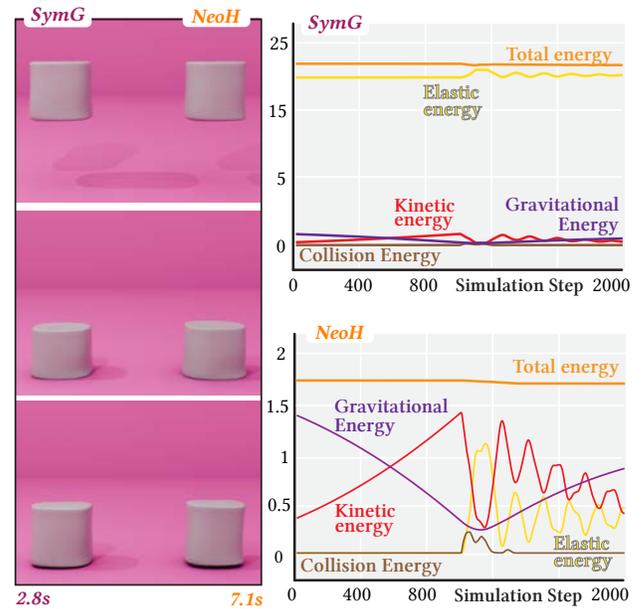


Fig. 16. The symmetric gradient and neo-Hookean energies share one volume-preserving penalty term in their formulation, but the neo-Hookean energy has an additional term that encourages further volume-preservation. We visualize this key difference in a collision experiment with two marshmallows, each modeled with one of these energies. Runtimes shown are average, per time step.

A key parameter in elastic simulation is the stiffness or rigidity of the mesh, set by the coefficient of the potential energy term. Ideally, one would like to have a method that is robust to a wide range of stiffness values. In Figure 7, we consider the same example of the gummy bear with three different values of stiffness, ranging from low to high. Our method remains stable in each case, allowing us to simulate elastic materials from flexible to rigid. A similar experiment is shown in Figure 17, where we visualize a beam bending for various choices of potential energy and stiffness.

External Collision Handling. Recall that we choose to model collisions using a penalty potential term in the Lagrangian. Consequently, our method should enjoy approximate energy conservation even in scenarios involving external collisions, so long as we include the “collision energy” in our total energy computation. In practice, we find that our method exhibits strong energy conservation in simulations involving collisions for a diverse set of meshes.

In addition to the simpler example presented in Figure 2, we consider a bunny bouncing on the ground under the influence of gravity in Figure 8. The bunny maintains its total energy, even at the point of collision, even though the derivatives of the kinetic and elastic energy change dramatically in the neighborhood of the collision. Moreover, total energy continues to be exactly conserved after the collision, even though the bounce induces secondary oscillations on the mesh, causing the elastic and kinetic energy graphs to “wiggle.” As another example, in Figure 18 we simulate a human skull bouncing elastically off of a wall (with no gravity). While these

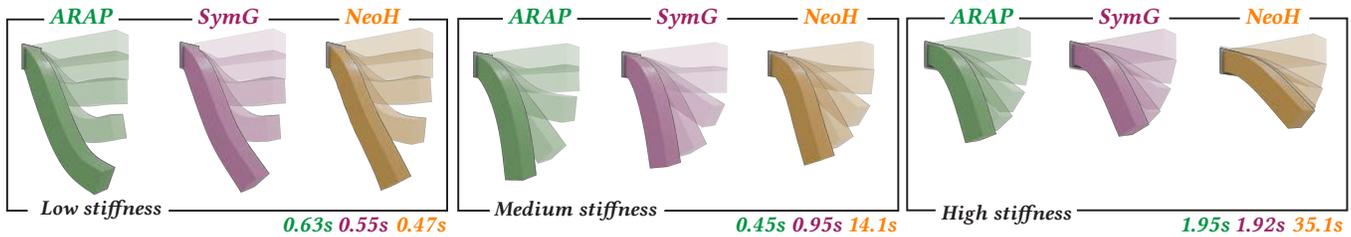


Fig. 17. We demonstrate the effectiveness of our method with a number of stiffness values. Here, we simulate an elastic beam bending and stretching under gravity using $\kappa = 35.0$ (left), $\kappa = 350.0$ (middle) and $\kappa = 1400.0$ (right). Our method converges consistently for a variety of energies, including neo-Hookean with $\lambda = 4\kappa$ (left, middle) and $\lambda = 2\kappa$ (right). Runtimes reported are **average, per time step**.

experiments use axis-aligned planes, the collision energy functional can be easily modified to handle planes that are not axis-aligned and, more generally, a spring-like penalty can be defined pointwise to handle non-planar collisions.

7 Conclusion

Hidden convexity in a class of elastic problems can be exploited to derive an implicit variational integrator that avoids nonlinear root-finding solvers (e.g., Newton’s). Our method exhibits several critical advantages, including convergence analysis, greater stability than previous works, strong energy conservation, and relative efficiency.

While we explored many applications of our framework, we would like to pursue a number of extensions. Most notably, self-collisions are not included in our simulations. Very flexible meshes, however, can experience self-collisions during simulation, in which case our method looks unrealistic (see Figure 19). Our approach to collisions is designed to approximately preserve total energy *even during* collisions. In future work, we would like to improve the array of collisions we can handle by integration rules. While existing approaches to contact mechanics like IPC robustly handle self-collisions and other multiple-collision scenes, their approach breaks the symplecticity of the integrator, losing conservation laws.

Our departure from iterative root-finding methods contributes to a wider use of variational integrators in physics-based simulation. In particular, we observed that instabilities of past solvers derived from a discrete principle of least action often arose from the numerical challenges of the iterative solver rather than the integration scheme itself. We hope that our work inspires revived interest in solvers derived from Hamilton’s principle and, in this context, further study into ways of extracting convexity from nonlinear and nonconvex time-evolving problems.

Acknowledgments

We thank Mirela Ben-Chen and Chris Wojtan for insightful discussions. We are grateful to Ahmed Mahmoud for suggestions, Zachary Banks for help debugging, and Oded Stein for proofreading and feedback. We acknowledge the authors of the 3D models used in this work: armadillo [Stanford Graphics Laboratory 1996], gummy bear [Engel 2013], bunny [Stanford Graphics Laboratory 1994], cat [Bill 2016], jellyfish [Romyblox1234 2013], pretzel [Quixel 2024], slug [Nudds 2021] and skull [YahooJAPAN 2013].

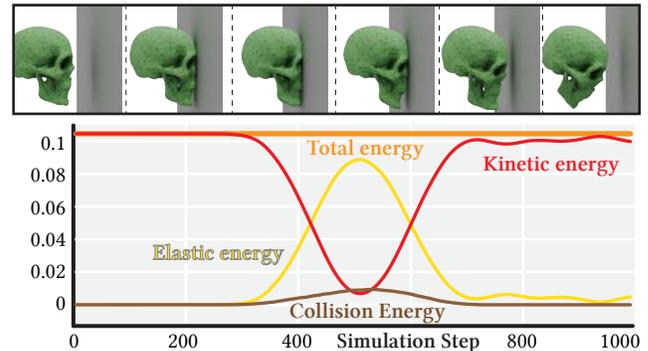


Fig. 18. A perfectly elastic headache!

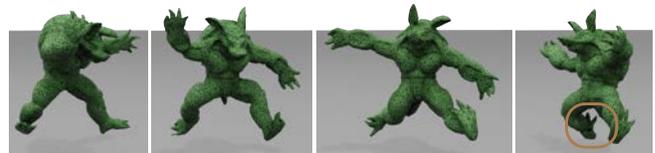


Fig. 19. Self-collisions can occur when the armadillo is too jumpy and too flexible. In this example our method remains stable, but the the motion in the self-collision region is unrealistic.

Leticia Mattos Da Silva acknowledges the generous support of the MathWorks Engineering Fellowship. The MIT Geometric Data Processing Group acknowledges the generous support of Army Research Office grants W911NF2010168 and W911NF2110293, of National Science Foundation grant IIS2335492, from the CSAIL Future of Data program, from the MIT-IBM Watson AI Laboratory, from the Wistron Corporation, and from the Toyota-CSAIL Joint Research Center. Natalia Pacheco-Tallaj was supported by the National Science Foundation grant DGE-2141064.

References

- S. Mazdak Abulnaga, Oded Stein, Polina Golland, and Justin Solomon. 2023. Symmetric Volume Maps: Order-invariant Volumetric Mesh Correspondence with Free Boundary. *ACM Trans. Graph.* 42, 3, Article 25 (April 2023), 20 pages. doi:10.1145/3572897
- David Baraff and Andrew Witkin. 1998. *Large Steps in Cloth Simulation* (1 ed.). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3596711.3596792

- Bill. 2016. Cat Stretch. <https://www.thingiverse.com/thing:1565405> Asset licensed under CC BY 4.0.
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Trans. Graph.* 33, 4, Article 154 (July 2014), 11 pages. doi:10.1145/2601097.2601116
- Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* 3 (2011), 1–122. <https://api.semanticscholar.org/CorpusID:51789432>
- George E. Brown and Rahul Narain. 2021. WRAPD: weighted rotation-aware ADMM for parameterization and deformation. *ACM Trans. Graph.* 40, 4, Article 82 (July 2021), 14 pages. doi:10.1145/3450626.3459942
- Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph.* 29, 4, Article 38 (July 2010), 6 pages. doi:10.1145/1778765.1778775
- Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. 2024c. Vertex Block Descent. *ACM Trans. Graph.* 43, 4, Article 116 (July 2024), 16 pages.
- Honglin Chen, Hsueh-Ti Derek Liu, Alec Jacobson, David I.W. Levin, and Changxi Zheng. 2024a. Trust-Region Eigenvalue Filtering for Projected Newton. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) (SA '24). Association for Computing Machinery, New York, NY, USA, Article 120, 10 pages.
- Honglin Chen, Hsueh-Ti Derek Liu, David I.W. Levin, Changxi Zheng, and Alec Jacobson. 2024b. Stabler Neo-Hookean Simulation: Absolute Eigenvalue Filtering for Projected Newton. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) (SIGGRAPH '24). Association for Computing Machinery, New York, NY, USA, Article 90, 10 pages.
- Honglin Chen, Changxi Zheng, and Kevin Wampler. 2023. Local Deformation for Interactive Shape Editing. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 13, 10 pages.
- Perry Engel. 2013. Gummy Bear. <https://www.thingiverse.com/thing:233643> Asset licensed under CC BY-NC 4.0.
- R. C. Fetecau, J. E. Marsden, M. Ortiz, and M. West. 2003. Nonsmooth Lagrangian Mechanics and Variational Collision Integrators. *SIAM Journal on Applied Dynamical Systems* 2, 3 (2003), 381–416. doi:10.1137/S11111102406038
- Ernst Hairer, Christian Lubich, and Gerhard Wanner. 2006. *Geometric numerical integration* (second ed.). Springer Series in Computational Mathematics, Vol. 31. Springer-Verlag, Berlin. xviii+644 pages. Structure-preserving algorithms for ordinary differential equations.
- Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, and Olga Sorkine. 2012. Fast automatic skinning transformations. *ACM Trans. Graph.* 31, 4, Article 77 (July 2012), 10 pages. doi:10.1145/2185520.2185573
- C. Kane, J. E. Marsden, M. Ortiz, and M. West. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *Internat. J. Numer. Methods Engrg.* 49, 10 (2000), 1295–1325. doi:10.1002/1097-0207(20001210)49:10<1295::AID-NME993>3.0.CO;2-W
- Danny M. Kaufman and Dinesh K. Pai. 2012. Geometric Numerical Integration of Inequality Constrained, Nonsmooth Hamiltonian Systems. *SIAM J. Comput.* 34, 5 (Jan. 2012), A2670–A2703. doi:10.1137/100800105
- L. Kharevych, Weiwei Yang, Y. Tong, E. Kanso, J. E. Marsden, P. Schröder, and M. Desbrun. 2006. Geometric, variational integrators for computer animation. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vienna, Austria) (SCA '06). Eurographics Association, Goslar, DEU, 43–51.
- Lei Lan, Minchen Li, Chenfanfu Jiang, Huamin Wang, and Yin Yang. 2023. Second-order Stencil Descent for Interior-point Hyperelasticity. *ACM Trans. Graph.* 42, 4, Article 108 (July 2023), 16 pages.
- Lei Lan, Guanqun Ma, Yin Yang, Changxi Zheng, Minchen Li, and Chenfanfu Jiang. 2022. Penetration-free projective dynamics on the GPU. *ACM Trans. Graph.* 41, 4, Article 69 (July 2022), 16 pages.
- Egor Larionov, Andreas Longva, Uri M. Ascher, Jan Bender, and Dinesh K. Pai. 2024. Implicit Frictional Dynamics With Soft Constraints. *IEEE Transactions on Visualization and Computer Graphics* 30, 12 (Dec. 2024), 7776–7787.
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Trans. Graph.* (SIGGRAPH) 39, 4, Article 49 (2020).
- Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A local/global approach to mesh parameterization. In *Proceedings of the Symposium on Geometry Processing* (Copenhagen, Denmark) (SGP '08). Eurographics Association, Goslar, DEU, 1495–1504.
- Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-Newton Methods for Real-Time Simulation of Hyperelastic Materials. *ACM Trans. Graph.* 36, 4, Article 116a (July 2017), 16 pages. doi:10.1145/3072959.2990496
- Miles Macklin, Matthias Müller, and Nuttapon Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games* (Burlingame, California) (MIG '16). Association for Computing Machinery, New York, NY, USA, 49–54. doi:10.1145/2994258.2994272
- J. Marsden and Matthew West. 2001. Discrete Mechanics and Variational Integrators. *Acta Numerica* 10 (10 2001). doi:10.1017/S096249290100006X
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based elastic materials. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) (SIGGRAPH '11). Association for Computing Machinery, New York, NY, USA, Article 72, 8 pages. doi:10.1145/1964921.1964967
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (April 2007), 109–118. doi:10.1016/j.jvcir.2007.01.005
- Rahul Narain, Matthew Overby, and George E. Brown. 2016. ADMM \subseteq projective dynamics: fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Zurich, Switzerland) (SCA '16). Eurographics Association, Goslar, DEU, 21–28.
- Scott Nudds. 2021. Brain Slug. <https://www.thingiverse.com/thing:5175008> Asset licensed under CC0 1.0 Universal.
- Matthew Overby, George E. Brown, Jie Li, and Rahul Narain. 2017. ADMM \subseteq Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Transactions on Visualization and Computer Graphics* 23, 10 (Oct. 2017), 2222–2234. doi:10.1109/TVCG.2017.2730875
- Quixel. 2024. Pretzel. <https://www.fab.com/listings/30c55d87-86b2-4489-9a0d-460202ccea0f>.
- Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Transactions on Graphics* 36, 2 (April 2017), 16:1–16:16.
- Romyblox1234. 2013. Jellyfish. <https://www.sketchfab.com/3d-models/jellyfish-spongebob-bfbb8-08812895b2314344bf4535e284ee7ac2> Asset licensed under CC BY 4.0.
- John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. 2004. Inter-surface mapping. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 870–877. doi:10.1145/1015706.1015812
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (March 2018), 15 pages. doi:10.1145/3180491
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (Feb. 2019), 15 pages. doi:10.1145/3241041
- Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Barcelona, Spain) (SGP '07). Eurographics Association, Goslar, DEU, 109–116.
- Stanford Graphics Laboratory. 1994. Stanford Bunny. The Stanford 3D Scanning Repository <https://graphics.stanford.edu/data/3Dscanrep/>.
- Stanford Graphics Laboratory. 1996. Armadillo. The Stanford 3D Scanning Repository <https://graphics.stanford.edu/data/3Dscanrep/>.
- Oded Stein, Jiajin Li, and Justin Solomon. 2022. A Splitting Scheme for Flip-Free Distortion Energies. *SIAM Journal on Imaging Sciences* 15, 2 (2022), 925–959.
- Ari Stern and Eitan Grinspun. 2009. Implicit-Explicit Variational Integration of Highly Oscillatory Problems. *Multiscale Modeling & Simulation* 7, 4 (2009), 1779–1794. doi:10.1137/080732936 arXiv:<https://doi.org/10.1137/080732936>
- Matthew West. 2004. *Variational integrators*. Ph.D. Dissertation. <https://www.proquest.com/dissertations-theses/variational-integrators/docview/305200634/se-2> Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-03-02.
- Yahoo!JAPAN. 2013. Skull. <https://www.thingiverse.com/thing:182365> Asset licensed under CC BY-NC 4.0.

Variational Elastodynamic Simulation Supplemental Material

Algorithm 1 TIME STEPPING

```

1: function ADMM( $q^k, U^k, P^k$ )
2:   Initialize  $q, U, P, Y \leftarrow q^k, U^k, P^k, 0$ 
3:   while PrimalResidual  $> \delta_p$  and PrimalDual  $> \delta_d$  do
4:      $q_i \leftarrow$  solution of (9)
5:      $P_i \leftarrow$  solution of (11)
6:      $U_i \leftarrow$  solution of (10)
7:     update  $Y_i$  via (13) and (optionally)  $\rho_i$  though (35)
8:   end while
9:   return  $q, U, P$ 
10: end function

```

A Proof of Theorem 4.2

In this section, we provide a proof of the equivalency of our optimization problem (7) with the minimization of the constrained action functional (5). Our strategy is to show that the two problems have identical first-order optimality constraints. We start by deriving these constraints for the original problem (5). The Lagrangian of the constrained minimization of the action functional is given by

$$\sum_k h \left[K(v^{k+1}) - E(P^{k+1}) + \sum_i \text{Tr}((R_i^{k+1})^\top (D_i q_{\text{mid}}^{k+1} - U_i^{k+1} P_i^{k+1})) \right], \quad (18)$$

where the R_i 's are the dual variables for the constraint. We omit the conditions that each P_i is a PSD matrix and each U_i is an orthogonal matrix as these can be enforced quite easily.

Taking the derivative of (18) with respect to q^k , P_i^k , and R_i^k , we obtain the following pair of conditions for each k :

$$0 = \frac{1}{h} M(2q^k - q^{k-1} - q^{k+1}) + \frac{1}{2} \sum_i D_i^\top (R_i^{k+1} + R_i^k) \quad (19)$$

$$0 = -h w_i \nabla f(P_i^k) - (U_i^k)^\top R_i^k \quad (20)$$

$$0 = D_i q_{\text{mid}}^{k+1} - U_i^{k+1} P_i^{k+1} \quad (21)$$

From (20) together with the orthogonality of U_i^k , we conclude that for each k , we have $R_i^k = -h w_i U_i^k \nabla f(P_i^k)$. Substituting this into (19) and rearranging yields

$$M v^{k+1} + \frac{h}{2} \sum_i w_i D_i^\top U_i^{k+1} \nabla f(P_i^{k+1}) = M v^k - \frac{h}{2} \sum_i w_i D_i^\top U_i^k \nabla f(P_i^k) \quad (22)$$

The left-hand side of this expression is entirely at time $k+1$ on the staggered time grid, whereas the right-hand side is entirely at time k . But also notice that these two sides do not exactly advance in parallel, as they differ by a sign.

Next, we obtain the first-order optimality constraints of the problem in (7). Its Lagrangian is given by

$$K\left(\frac{q - z^k}{h}\right) + E(P) + \sum_i \text{Tr}(Y_i^\top (D_i q_{\text{mid}} - U_i P_i)), \quad (23)$$

where the Y_i 's are the dual variables for the constraint. Taking the derivatives of (23) with respect to q^k , P_i^k , and Y_i , we obtain the

following constraints for each k :

$$0 = \frac{M}{h^2} (q - z^k) + \sum_i D_i^\top \frac{Y_i}{2} \quad (24)$$

$$0 = w_i \nabla f(P_i) - U_i^\top Y_i \quad (25)$$

$$0 = D_i q_{\text{mid}} - U_i P_i \quad (26)$$

We combine the orthogonality of U_i with (25) and substitute the result into (24) to obtain

$$\frac{M}{h^2} (q - z^k) + \frac{1}{2} \sum_i w_i D_i^\top U_i \nabla f(P_i) = 0 \quad (27)$$

One now readily verifies that, to match the conditions (24)–(27) with the conditions obtained in (19)–(22), it suffices to make the following choices:

$$q = q^{k+1}$$

$$P_i = P_i^{k+1}$$

$$U_i = U_i^{k+1}$$

$$Y_i = w_i U_i^{k+1} \nabla f(P_i^{k+1})$$

$$z^k = (2q^k - q^{k-1}) - M^{-1} \frac{h^2}{2} \sum_i w_i D_i^\top U_i^k \nabla f(P_i^k)$$

Thus, we have shown that minimizing the action functional in (5) is equivalent to the minimization problem in (7). \square

B Adding External Forces

In this section, we complement our proof in §A and derivation of the corresponding algorithm in §5.2 by discussing what changes must be made to handle the case in which external conservative forces, such as gravity or penalty-based collision terms, and linear constraints are added to our model.

B.1 Analogue of Theorem 4.2

Suppose our external forces correspond to a potential energy g , and suppose we seek to impose a linear constraint on q of the form $Xq = Z$. For example, g could represent gravitational potential energy, modeled as a weighted sum over all vertices of the height of the vertex times the acceleration due to gravity; and $Xq = Z$ could represent an anchor constraint, where a certain subset of vertices is anchored by taking X to be the diagonal matrix whose diagonal entries are 1 if the corresponding vertex is to be anchored and zero otherwise, and Z encodes the anchor values. In this general setting, the constrained action takes the following shape, obtained from (5) by simply adding the potential energy function g corresponding to the forces into the action and enforcing the linear constraint:

$$S^d = \sum_k h \left[K(v^{k+1}) - g(q_{\text{mid}}^{k+1}) - E(P^{k+1}) \right], \quad (28)$$

subject to $D_i q_{\text{mid}}^{k+1} - U_i^{k+1} P_i^{k+1} = 0 \forall i$, and

$$X q_{\text{mid}}^{k+1} = Z.$$

Just as we did in §A, we can derive an optimization problem whose first-order optimality constraints match those of the problem of minimizing the constrained action (28). The derivation is completely analogous, the only difference being that the potential energy g and

the constraint $Xq_{\text{mid}}^{k+1} = Z$ must be incorporated into the augmented Lagrangian; for the sake of brevity, we choose not to repeat the derivation. The following result formulates the desired equivalent optimization problem:

THEOREM B.1. *Let z^k be defined by*

$$z^k = (2q^k - q^{k-1}) - M^{-1} \frac{h^2}{2} \left[2\nabla g(q_{\text{mid}}^k) + \sum_i w_i D_i^\top U_i^k \nabla f(P_i^k) \right]. \quad (29)$$

Then, critical points of the constrained action functional in (28) are solutions to the following minimization problem:

$$\begin{aligned} \arg \min_{q, P \in \mathcal{S}^3, U \in \text{SO}(3)} & K \left(\frac{q - z^k}{h} \right) + g(q_{\text{mid}}^{k+1}) + E(P), \\ \text{subject to} & D_i q_{\text{mid}} - U_i P_i = 0 \quad \forall i, \text{ and} \\ & X q_{\text{mid}} = Z. \end{aligned} \quad (30)$$

Comparing Theorems 4.2 and B.1, we see that the inclusion of external conservative forces changes the optimization problem in two ways: the associated potential energy function g is added to the objective, and a term involving the gradient of g is added to the quantity z^k . The imposition of the linear constraint on the action minimization problem simply propagates the exact same constraint into the optimization problem; in particular, it does not affect the definition of the quantity z^k .

B.2 Modified ADMM Algorithm with Linear Constraints

The augmented Lagrangian in (8) already incorporates the potential energy term g . We now explain how the incorporation of linear constraints impacts the steps of our ADMM algorithm. Because we model these constraints as functions of position, the q update is affected, but the other updates remain unchanged.

The constraint $Xq_{\text{mid}} = Z$ is incorporated by adding on the following pair of terms:

$$\frac{\mu'}{2} \|Xq_{\text{mid}} - Z\|_F^2 + \text{Tr}(Y'^\top (Xq_{\text{mid}} - Z)).$$

Having modified the augmented Lagrangian in this way, we can differentiate it with respect to q to obtain the q update. Doing, so we see that the q update amounts to solving a system of the form $Aq = B$, where A and B are now given as follows:

$$\begin{aligned} A &= \frac{1}{h^2} M + \sum_i \frac{\rho_i}{4} D_i^\top D_i + \frac{1}{4} \mu' X^\top X \\ \text{and } B &= -\frac{1}{h^2} M z^k - \frac{1}{2} \sum_i D_i^\top Y_i + \sum_i \frac{\rho_i}{2} D_i^\top U_i P_i - \frac{\rho_i}{4} D_i^\top D_i q^k \\ &\quad + \frac{1}{2} \mu' X^\top Z - \frac{1}{2} X^\top Y' - \frac{\mu'}{4} X^\top X q^k. \end{aligned} \quad (31)$$

As before, the matrix A can be precomputed and factorized using Cholesky decomposition at the beginning of the algorithm.

B.3 Closed Form for q' Update

We conclude this section by deriving an explicit closed form for the q' update in the case where g corresponds to a quadratic penalty

potential. Because our penalty potential breaks down in per vertex fashion, for this purpose it suffices to consider the case of a single vertex with position q . Then we take

$$g(q) = \frac{1}{2} \kappa |\max\{0, -z(q)\}|^2,$$

where $z(q)$ denotes the z -coordinate of q . Let q^* denote the solution to the minimization problem (12). Differentiating the objective with respect to q' and setting the result equal to zero, we find the following piecewise equality:

$$z(q') = \begin{cases} z(q + W/\mu), & \text{if } z(q^*) \geq 0, \\ z\left(\frac{\kappa h v/4 + \mu q + W}{\kappa/2 + \mu}\right), & \text{if } z(q^*) < 0, \end{cases} \quad (32)$$

with the x and y components of q' remaining unchanged, as the penalty potential is a function purely of the z coordinate.

C Stopping Criterion and Rescaling

We terminate our ADMM algorithm if the following conditions on the primal and dual residuals are satisfied:

$$\begin{aligned} \epsilon_p &:= \sqrt{\sum_i \|J_i - U_i P_i\|_F^2} < \delta_p, \\ \epsilon_d &:= \sqrt{\sum_i \rho_i \|P_i - P_i^k\|_F^2} < \delta_d, \end{aligned} \quad (33)$$

where, following [Boyd et al. 2011; Stein et al. 2022], we define the primal and dual tolerances to be

$$\begin{aligned} \delta_p &:= 10^{-6} \sqrt{3m} + 10^{-5} \max \left(\sum_i \|J_i\|_F, \sum_i \|P_i\|_F \right), \\ \delta_d &:= 10^{-6} \sqrt{3m} + 10^{-5} \sum_i \|G_i^\top \rho_i^{-1} Y_i\|_F. \end{aligned} \quad (34)$$

Optional. A standard modification of the ADMM algorithm is to rescale the dual variable ρ_i at each iteration, adjusting Y_i accordingly, with the goal of improving convergence. A typical rule for rescaling (see §3.4.1 in [Boyd et al. 2011]) is as follows

$$\rho_i^{k+1} = \begin{cases} 1/2 \rho_i^k, & \text{if } \epsilon_{p_i} > 10 \epsilon_{d_i} \\ 2 \rho_i^k, & \text{if } \epsilon_{d_i} > 10 \epsilon_{p_i}. \end{cases} \quad (35)$$

D Assumptions for Convergence Analysis

Drawing from Stein et al. [2022], we impose the following assumptions for convergence guarantee:

- (A1) *Bound.* The gradient of the potential f evaluated at the sequence of iterates $P^{(j)}$ is bounded; i.e., for each i , there exists a constant $B_i > 0$ such that $\|\nabla f(P_i^{(j)})\| \leq B_i$.
- (A2) *Bound.* The change across ADMM iterations in dual variable Y corresponding to the polar decomposition constraint is bounded as follows:

$$\begin{aligned} \|Y_i^{(j+1)} - Y_i^{(j)}\|_F^2 &\leq \\ \gamma \|Y_i^{(j+1)} - Y_i^{(j)} + U_i^{(j+1)} Y_i^{(j+1)\top} U_i^{(j+1)} - U_i^{(j)} Y_i^{(j)\top} U_i^{(j)}\|_F^2 &\leq \end{aligned}$$

for some constant $\gamma > 0$.

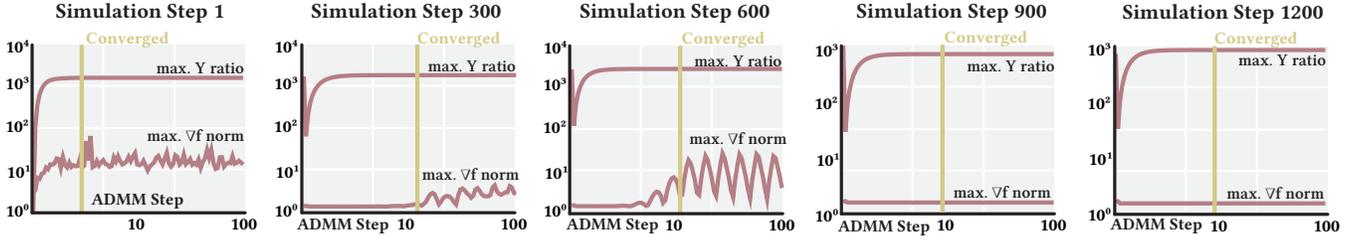


Fig. 1. The dual ratio and the norm of the potential gradient are assumed to be bounded as we detailed in §5.1 and §D. Here, we plot the maximum value for each versus ADMM iterations for a number of time steps over our simulation in Fig. 10. Indeed, we observe that (A1) and (A3) hold in practice since these quantities appear bounded.

(A3) *Lipschitz*. The gradient of the potential f is Lipschitz continuous along the sequence of iterates $P^{(j)}$; i.e., for each i we have

$$\|\nabla f(P_i^{(j+1)}) - \nabla f(P_i^{(j)})\| \leq F_i \|P_i^{(j+1)} - P_i^{(j)}\|$$

for some constant F_i .

We reiterate that the validity of Assumptions (A1) and (A2) can be verified in practice (see Figure 1), whereas Assumption (A3) can be checked analytically, as is done for symmetric gradient and symmetric Dirichlet energies in [Stein et al. 2022, Lemma 5.3].

E Closed form solutions

In this section, we present the close form equations to solve for P using each of the choices of elastic energy presented in §5.2.

We solve for P via finding the critical point of (11), which admits a unique minimizer. As recorded by Stein et al. [2022], this amounts to solving the following for each i :

$$w_i \nabla f(P_i) + \rho_i P_i = \rho_i Q_i \quad (36)$$

where $Q_i = \frac{1}{2}((U_i^\top (D_i q_{\text{mid}} + \rho_i^{-1} Y_i))^\top + U_i^\top (D_i q_{\text{mid}} + \rho_i^{-1} Y_i))$.

ARAP. The potential gradient for ARAP is given by $\nabla f(P) = P$ and we can solve for P via

$$P = \frac{1}{(w_i/\rho_i) + 1} \left(\frac{w_i}{\mu_i} \mathbf{I}_3 + Q_i \right). \quad (37)$$

Symmetric Gradient. In this case, we have $\nabla f(P) = P - P^{-1}$ and (36) can be written as a quadratic equation for each i , whose solution we obtain via the closed form:

$$P_i = \frac{1}{2(w_i + \rho_i)} \left(\rho_i Q_i + \sqrt{\rho_i^2 Q_i^2 + 4w_i(w_i + \rho_i) \mathbf{I}_3} \right). \quad (38)$$

Symmetric Dirichlet. For this energy we have $\nabla f(P) = P - P^{-3}$. Hence 36 becomes the following:

$$(w_i + \rho_i) P_i^4 - \rho_i Q_i P_i^3 - w_i \mathbf{I}_3 = 0. \quad (39)$$

While this is a quartic equation, in principle, it does have a closed form solution. In practice, one might prefer to use a standard quartic solver.

F Proximal Operator for neo-Hookean energy

In this section, we describe how to modify (11) to use the neo-Hookean potential energy in our framework, with the caveat that our convergence theory does not apply. Despite not reflecting the

conditions of our convergence analysis, we show that in practice this modification successfully obtains convergence for a number of λ, κ values (see Fig. 17, 14, 16 and 15 for preliminary experiments).

In the large- λ regime, the optimization landscape of neo-Hookean energies becomes increasingly non-convex, posing additional numerical challenges. These difficulties have motivated a number of approaches to improve stability and convergence of the simulation of neo-Hookean materials, we refer the reader to the works by Smith et al. [2018] and Chen et al. [2024a,b].

Proximal Operator. Define $P_{0i} := U_i^\top (J_i + \rho_i^{-1} Y_i)$. With this notation, we obtain the following first-order optimality condition:

$$w_i \kappa (P_i - P_i^{-\top}) + w_i \lambda (\log \det P_i) P_i^{-\top} + \rho_i (P_i - P_{0i}) = 0. \quad (40)$$

This implies that each P_i and P_{0i} commute and hence can be simultaneously diagonalized,

$$P_{0i} = V \text{diag}(\sigma_{0ik}) V^\top, \quad P_i = V \text{diag}(\sigma_{ik}) V^\top, \quad (41)$$

which allows us to compute the proximal operator on singular value space, similar to [Chen et al. 2023]. Let $t := \sigma_{i0} \sigma_{i1} \sigma_{i2}$ denote the determinant of P_i . Then, for each singular value σ_{ik} , we obtain the scalar condition

$$\sigma_{ik}^2 - b \sigma_{0ik} \sigma_{ik} + \ell \log t - r = 0. \quad (42)$$

where we define $b := \frac{\rho_i}{w_i \kappa + \rho_i}$, $\ell := \frac{w_i \lambda}{w_i \kappa + \rho_i}$, and $r := \frac{w_i \kappa}{w_i \kappa + \rho_i}$.

Solving the resulting quadratic gives two possible roots per singular value:

$$\sigma_{ik} = \frac{1}{2} b \sigma_{0ik} \pm \frac{1}{2} \sqrt{b^2 \sigma_{0ik}^2 - 4(\ell \log t - r)}. \quad (43)$$

We seek a value of t such that all σ_{ik} are real and nonnegative, and satisfy $t = \sigma_{i0} \sigma_{i1} \sigma_{i2}$, which defines an implicit fixed-point problem over t . To ensure real-valued solutions, we enforce

$$t \leq \min_k \exp \left[\frac{4r + b^2 \sigma_{0ik}^2}{4\ell} \right]. \quad (44)$$

We also require that all $\sigma_{ik} \geq 0$ to preserve positive-definiteness, which can restrict the choice of signs in the quadratic roots. These observations allow us to bracket the feasible range for t and perform a bisection search to recover a consistent triplet $(\sigma_{i0}, \sigma_{i1}, \sigma_{i2})$ satisfying $t = \sigma_{i0} \sigma_{i1} \sigma_{i2}$ and $\sigma_{ik} \geq 0$. While this approach does not yield a closed-form solution, it results in a one dimensional root-finding problem that we solve numerically.