Accurate Dissipative Forces in Optimization Integrators

GEORGE E. BROWN, University of Minnesota
MATTHEW OVERBY, University of Minnesota
ZAHRA FOROOTANINIA, University of Minnesota
RAHUL NARAIN, Indian Institute of Technology

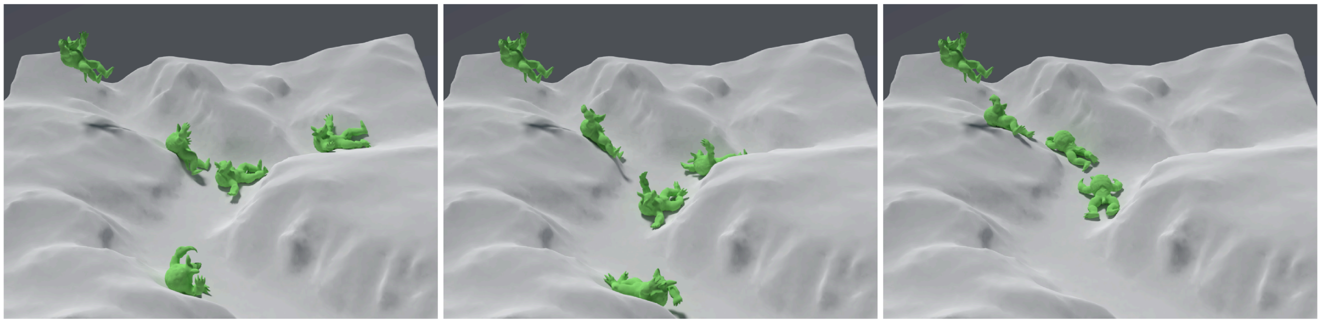

Fig. 1. A Neo-Hookean armadillo falls down a complex terrain with different friction coefficients, left: $\mu = 0$, center: $\mu = 0.15$, right: $\mu = 0.45$.

We propose a method for accurately simulating dissipative forces in deformable bodies when using optimization-based integrators. We represent such forces using *dissipation functions* which may be nonlinear in both positions and velocities, enabling us to model a range of dissipative effects including Coulomb friction, Rayleigh damping, and power-law dissipation. We propose a general method for incorporating dissipative forces into optimization-based time integration schemes, which hitherto have been applied almost exclusively to systems with only conservative forces. To improve accuracy and minimize artificial damping, we provide an optimization-based version of the second-order accurate TR-BDF2 integrator. Finally, we present a method for modifying arbitrary dissipation functions to conserve linear and angular momentum, allowing us to eliminate the artificial angular momentum loss caused by Rayleigh damping.

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

Additional Key Words and Phrases: Implicit integration, dissipation functions, nonlinear damping, friction, angular momentum

1 INTRODUCTION

Dissipative forces are ubiquitous in the natural world. Frictional contact in solids and granular materials, air resistance, viscosity, and internal damping in deformable bodies are just a few examples. In physics-based animation, a long-standing goal has been to produce visually plausible representations of these phenomena. Recently, researchers in graphics have begun exploring nonlinear damping [Xu and Barbič 2017], which has enabled the simulation of especially realistic damping effects.

In animation applications, there is also an ever-increasing need to extend the complexity of simulations that can be carried out in a fast and stable manner. However, achieving both speed and stability when modeling nonlinear phenomena is a challenging problem. Commonly used explicit integration methods are fast and simple but require small time steps for stability. Implicit methods such as backward Euler can be used instead, enabling larger time steps with much greater stability. Even so, most existing methods require solving a large system of equations every time step, whose computational expense increases steeply with the complexity of the simulation.

Optimization-based algorithms [Bouaziz et al. 2014; Liu et al. 2013, 2016; Overby et al. 2017] have been used to achieve interactive simulation rates for complex dynamical systems. However, most such methods are designed specifically for modeling conservative forces, with only some very simple dissipative forces added as an afterthought. Karamouzas et al. [2017] successfully modeled highly nonlinear, non-conservative interaction forces within their optimization-based framework. However, they based their method on the backward Euler scheme, which produces excessive artificial dissipation at large time steps that often overwhelms the desired dissipative behavior.

In this work, we present a novel approach for optimization-based simulation of systems with dissipative forces. Our contributions are summarized as follows:

- We provide an optimization-based formulation for a broad class of implicit time integrators, including the second-order TR-BDF2 scheme which has excellent momentum and energy conservation properties.

Authors' addresses: George E. Brown, University of Minnesota, brow2327@umn.edu; Matthew Overby, University of Minnesota, over0219@umn.edu; Zahra Forootaninia, University of Minnesota, foro0012@umn.edu; Rahul Narain, Indian Institute of Technology, narain@cse.iitd.ac.in.

0730-0301/2018/11-ART282 $15.00
https://doi.org/10.1145/3272127.3275011

- We introduce a simple approach for incorporating dissipative forces into such optimization integrators through the use of *dissipation functions*, a concept from classical mechanics that has only recently begun finding use in computer graphics [Karamouzas et al. 2017; Sánchez-Banderas and Otaduy 2017]. We show that the simple approach is first-order accurate, but can be elevated to second-order accuracy by the addition of a correction term.
- We provide dissipation functions for many dissipation models traditionally used in computer animation, and demonstrate some novel nonlinear dissipation models that enable artistic damping effects. Our fully coupled formulation enables implicit handling of frictional contact with kinematic obstacles in optimization-based integration.
- Finally, we propose a general method for modifying arbitrary dissipation models to conserve linear and angular momentum. In particular, our modification eliminates the artificial angular momentum loss caused by Rayleigh damping.

2 RELATED WORK

2.1 Time Integration

Implicit integration techniques are commonly used in computer graphics. Among these, a particularly prevalent choice is backward Euler, which is a first-order accurate backward differentiation formula with exceptional stability properties [Baraff and Witkin 1998; Terzopoulos et al. 1987]. However, backward Euler is also notorious for artificial damping, which can grow unacceptably large with large timesteps in stiff systems. Its second-order variant, the BDF2 method, has been used in some previous work [Choi and Ko 2002; English and Bridson 2008] to maintain stability while reducing the amount of numerical dissipation. Recently, Xu and Barbič [2017] introduced to graphics the TR-BDF2 method [Bank et al. 1985; Bathe 2007; Bathe and Baig 2005] which combines the trapezoidal rule and BDF2, resulting in a stable integrator with even lower artificial dissipation.

In problems with even moderate complexity, standard iterative schemes for carrying out implicit integration are either prohibitively expensive or take too long to converge for interactive applications. One widely used alternative is position-based dynamics [Bender et al. 2015; Müller et al. 2007], which deals only with positions. In this method, forces are treated as hard constraints and the state is updated with Gauss-Seidel iterations. This approach was generalized by XPBD [Macklin et al. 2016] to enable the use of finite stiffness constraints for better modeling of elastic forces.

Another strategy is to recast implicit integration as a numerical optimization problem [Gast et al. 2015; Kharevych et al. 2006; Martin et al. 2011]. Recent work has focused on local-global approaches for solving the resulting optimization problem, which offer computationally cheap descent iterations with rapid initial progress.

A block coordinate descent technique was used by Liu et al. [2013] for mass-spring systems. Bouaziz et al. [2014] generalized this approach to simulate a broader class of constraints and finite elements, and termed it *projective dynamics*. This approach has subsequently been applied to SPH fluids [Weiler et al. 2016] and character skinning [Komaritzan and Botsch 2018]. A number of acceleration techniques have been proposed to further speed up projective dynamics, including the Chebyshev semi-iterative method [Wang et al. 2015], parallel graph-colored Gauss-Seidel [Fratarcangeli et al. 2016], L-BFGS [Liu et al. 2016], and model reduction [Brandt et al. 2018]. In the closely related area of geometric optimization, many novel techniques have recently been introduced that dramatically accelerate the convergence of numerical optimization [Claici et al. 2017; Kovalsky et al. 2016; Peng et al. 2018; Rabinovich et al. 2017; Shtengel et al. 2017; Zhu et al. 2018]. As demonstrated by Peng et al. [2018], these methods can be useful for simulation problems as well.

Projective dynamics has been further generalized to nonlinear constitutive models by interpreting it as an instance of ADMM [Overby et al. 2017] or quasi-Newton optimization [Liu et al. 2016]. The ADMM formulation has been applied to skin slide simulation [Saito and Yuen 2017] and to character self-intersection removal for cloth simulation [Minor 2018].

Recently, Dinev et al. [2018a,b] have proposed two energy-conserving algorithms for deformable body simulation. Both approaches enable exact energy conservation for conservative systems, but require dissipative forces to be applied in a separate step decoupled from the conservative force integration. By contrast, we formulate a single optimization problem that combines the conservative and dissipative forces into a single objective.

2.2 Damping

Damping in graphics dates back to early work in deformable body simulation [Terzopoulos and Witkin 1988], where velocity-based damping was applied directly to each node. In finite element simulations, local damping forces are commonly computed using the strain-rate tensor [Debunne et al. 2001; O'Brien and Hodgins 1999; Platt and Barr 1988]. For cloth, Baraff and Witkin [1998] model damping using the time derivative of constraint functions.

Rayleigh damping, in which the damping forces are proportional to the product of the stiffness matrix and the (generalized) velocities, is another widely used model in computer graphics [Barbič and James 2005; Gast et al. 2015; Ren et al. 2013; Sifakis and Barbič 2012; Wang et al. 2015]. However, unlike the earlier strain-rate approaches, Rayleigh damping does not always conserve angular momentum. Some direct and iterative techniques to impose angular momentum conservation were proposed by Schmedding et al. [2009].

In the context of optimization-based integration, Kharevych et al. [2006] added stiffness-proportional damping forces by defining them in terms of the object's potential energy function. To obtain forces proportional to velocities, they defined the energy in terms of the strain measured *relative* to the current state. Gast et al. [2015] use a lagged version of Rayleigh damping to define a damping energy function for optimization-based integration. While the method of Kharevych et al. [2006] conserves angular momentum, the Rayleigh damping model used by Gast et al. [2015] does not.

Data-driven damping methods have also been used to obtain more realistic damping behavior. For cloth simulation, parameter optimization from data measurements was used for spring-based damping [Bhat et al. 2003] and for modeling internal friction [Miguel et al. 2013]. Recently, an example-based method [Xu and Barbič

2017] has been used which optimizes parameters for nonlinear damping.

Dissipation functions [Goldstein et al. 2002; Marsden and Ratiu 1999] are a technique for modeling non-conservative forces in Lagrangian mechanics. In graphics, Karamouzas et al. [2017] introduced a dissipation function to model anticipatory collision avoidance between pedestrians for implicit crowd simulation. Concurrently, Sánchez-Banderas and Otaduy [2017] modeled damping in yarn-level cloth simulation using dissipation functions, permitting fine-grained control over the damping of stretching, bending, and shear modes. As pointed out by these works, the damping models proposed earlier by Baraff and Witkin [1998] and Kharevych et al. [2006] are also closely related to dissipation functions.

2.3 Friction

There is a long history of frictional contact simulation in computer graphics, dating back to seminal work by Baraff [1991]. Usually, an isotropic Coulomb friction model is used, although models for anisotropic and asymmetric frictional forces have also been proposed [Pabst et al. 2009]. In this section, we focus primarily on frictional contact solvers applicable to deformable bodies, which is the setting of our work.

A classical approach is to apply contact forces in a separate collision resolution step, decoupled from the time integration of the internal dynamics [Bridson et al. 2002; Provot 1997]. However, as this is inadequate for systems with strong coupling between the elastic, normal, and frictional forces, there has been much work on *implicit* frictional contact algorithms which compute all these forces simultaneously. One approach is to approximate the equations of frictional contact as a linear complementarity problem [Duriez et al. 2006; Otaduy et al. 2009] or second-order cone complementarity problem [Daviet et al. 2011], and solve it using Gauss-Seidel iterations. Kaufman et al. [2008] proposed a staggered projection method in which normal and frictional forces are determined by two coupled minimizations computed alternately. Frictional dynamics in granular flows have also been modeled using staggered projections [Narain et al. 2010], Gauss-Seidel solvers [Daviet and Bertails-Descoubes 2016], or plastic flow models [Klár et al. 2016]. Recently, Jiang et al. [2017] applied a continuum plasticity model to simulate frictional contact in cloth using the material point method.

Many optimization-based methods in graphics do not incorporate friction into the optimization, applying it instead as a separate post-optimization update to the velocities [Bouaziz et al. 2014; Gast et al. 2015; Müller et al. 2007]. Recently, Overby et al. [2017] simulated friction by varying the external force term as the optimization proceeds. In our work, we model friction fully implicitly as an additional dissipative term in the optimization, allowing us to accurately reproduce dynamic and static frictional phenomena.

3 BACKGROUND

Notation. For brevity, we introduce compact notation which is used in the rest of this paper unless otherwise noted. We use superscripts on functions to denote the same superscript on each of the function's arguments; e.g. $\mathbf{f}^n = \mathbf{f}(\mathbf{x}^n, \mathbf{v}^n)$. Subscripts on norms denote the norm weighted by a matrix; e.g, $\|\mathbf{x} - \tilde{\mathbf{x}}\|^2_{\mathbf{M}} = (\mathbf{x} - \tilde{\mathbf{x}})^T \mathbf{M}(\mathbf{x} - \tilde{\mathbf{x}}) = \|\mathbf{M}^{\frac{1}{2}}(\mathbf{x} - \tilde{\mathbf{x}})\|^2$.

For functions of multiple vector arguments, we use ∇_i to denote the gradient with respect to the ith argument; e.g. $\nabla_2 R(\mathbf{x}, \mathbf{v})$ is the vector $[\frac{\partial R}{\partial v_1}, \ldots, \frac{\partial R}{\partial v_n}]^T$. We adopt the convention that differentiating a scalar with respect to a (column) vector yields a column vector; e.g. $\frac{\mathrm{d}U(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \nabla U(\mathbf{x})$, and $\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}} R(\mathbf{x}, \mathbf{A}\mathbf{v}) = \mathbf{A}^T \nabla_2 R(\mathbf{x}, \mathbf{A}\mathbf{v})$.

3.1 Dynamical model

Time evolution in physical systems is governed by Newton's second law, which describes the relation between dynamics and kinematics. In simulations which utilize the finite element method, objects are represented by discrete mesh elements, and forces are applied on mesh vertices. In a system with n vertices, enforcing Newton's law is tantamount to solving a $6n$-dimensional ordinary differential equation,

$$\dot{\mathbf{x}} = \mathbf{v}, \tag{1a}$$

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}, \mathbf{v}), \tag{1b}$$

where $\mathbf{M} \in \mathbb{R}^{3n \times 3n}$ is the system mass matrix, and $\mathbf{x} \in \mathbb{R}^{3n}$ and $\mathbf{v} \in \mathbb{R}^{3n}$ are the system displacement and velocity vectors, respectively.

We assume that the total force vector, $\mathbf{f} \in \mathbb{R}^{3n}$, can be decomposed as the sum of three terms: external forces $\mathbf{f}_{\text{ext}}$ (e.g. gravity), conservative internal forces $\mathbf{f}_c(\mathbf{x})$, and dissipative forces $\mathbf{f}_d(\mathbf{x}, \mathbf{v})$ respectively. We take the external forces $\mathbf{f}_{\text{ext}}$ to be constant over a single time step. Conservative forces, such as internal elastic forces, are of the form

$$\mathbf{f}_c(\mathbf{x}) = -\nabla U(\mathbf{x}) \tag{2}$$

for some potential energy function U. For dissipative forces, we restrict our attention to those which can be expressed as the gradient of a dissipation function R via

$$\mathbf{f}_d(\mathbf{x}, \mathbf{v}) = -\nabla_2 R(\mathbf{x}, \mathbf{v}). \tag{3}$$

Thus, our force model takes the form

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = \mathbf{f}_{\text{ext}} - \nabla U(\mathbf{x}) - \nabla_2 R(\mathbf{x}, \mathbf{v}). \tag{4}$$

If the dissipation function is independent of $\mathbf{x}$, we may simply write it as $R(\mathbf{v})$, with the corresponding force being $\mathbf{f}_d(\mathbf{v}) = -\nabla R(\mathbf{v})$.

Assuming the existence of a dissipation function is not too severe a restriction, since many commonly used dissipative forces in mechanics can be expressed in this form, including Rayleigh damping and Coulomb friction. Furthermore, this subset of forces is naturally suited for inclusion in optimization-based integrators. In Section 4 we show that a first-order accurate method can be obtained by simply adding a scalar multiple of R to the objective function.

In case the potential functions U and R are nonsmooth, the dynamics can be expressed in terms of subdifferentials [Rockafellar 1970]. We assume that U is the sum of a convex nonsmooth term U^{ns} and a smooth term U^s. Then the conservative forces satisfy

$$\mathbf{f}_c(\mathbf{x}) \in -\nabla U^s(\mathbf{x}) - \partial U^{ns}(\mathbf{x}), \tag{5}$$

where $\partial U^{ns}(\mathbf{x})$ is the subdifferential of U^{ns} at $\mathbf{x}$, i.e. the set of all vectors $\mathbf{g}$ such that $U^{ns}(\mathbf{y}) - U^{ns}(\mathbf{x}) \geq \mathbf{g}^T(\mathbf{y} - \mathbf{x})$ for all $\mathbf{y}$. An analogous property holds for the dissipative forces, with the subdifferential taken with respect to $\mathbf{v}$.

Table 1. Some common time integration schemes expressed in the form (6).

Method	$\mathbf{y}^p$	α	$\mathbf{y}^q$
Forward Euler	$\mathbf{y}^n$	1	$\mathbf{y}^n$
Backward Euler	$\mathbf{y}^n$	1	$\mathbf{y}^{n+1}$
Implicit midpoint	$\mathbf{y}^n$	1	$\frac{1}{2}(\mathbf{y}^n + \mathbf{y}^{n+1})$
Trapezoidal	$\mathbf{y}^n + \frac{1}{2}\Delta t \boldsymbol{\phi}(\mathbf{y}^n)$	1/2	$\mathbf{y}^{n+1}$
BDF2	$\frac{4}{3}\mathbf{y}^n - \frac{1}{3}\mathbf{y}^{n-1}$	2/3	$\mathbf{y}^{n+1}$

3.2 Implicit integration

For simplicity, let us first consider a generic first-order ODE $\dot{\mathbf{y}} = \boldsymbol{\phi}(\mathbf{y})$, where $\mathbf{y} \in \mathbb{R}^d$. Many one-stage time integration schemes are of the form

$$\mathbf{y}^{n+1} = \mathbf{y}^p + \alpha \Delta t \boldsymbol{\phi}(\mathbf{y}^q) \tag{6}$$

where $\alpha \in \mathbb{R}$ is a constant, $\mathbf{y}^p \in \mathbb{R}^d$ is fixed at the beginning of the time step, and the quadrature point $\mathbf{y}^q \in \mathbb{R}^d$ depends linearly on $\mathbf{y}^{n+1}$. Backward Euler, for example, is obtained by setting $\alpha = 1$, $\mathbf{y}^p = \mathbf{y}^n$, and $\mathbf{y}^q = \mathbf{y}^{n+1}$; other examples are listed in Table 1. For simplicity, in this paper we will focus on the case $\mathbf{y}^q = \mathbf{y}^{n+1}$, which includes backward Euler, the trapezoidal scheme, and BDF2, but omits implicit midpoint.

Applied to the equations of motion (1), such a time integration scheme yields the equations

$$\mathbf{x}^{n+1} = \mathbf{x}^p + \alpha \Delta t \mathbf{v}^{n+1}, \tag{7a}$$

$$\mathbf{M}\mathbf{v}^{n+1} = \mathbf{M}\mathbf{v}^p + \alpha \Delta t \mathbf{f}(\mathbf{x}^{n+1}, \mathbf{v}^{n+1}). \tag{7b}$$

This method is *implicit*, since future state information appears on both sides of these equations. As previously discussed, such methods generally require solving large nonlinear systems of equations every timestep. However, when the forces are expressible as gradients of potentials, one can recast this system of equations as an optimization problem. For example, in the absence of dissipative forces, the backward Euler update equations are equivalent to

$$\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}} \left(\frac{1}{2\Delta t^2} \|\mathbf{x} - \tilde{\mathbf{x}}_{\mathrm{BE}}^{n+1}\|_{\mathbf{M}}^2 + U(\mathbf{x}) \right), \tag{8a}$$

$$\mathbf{v}^{n+1} = \frac{1}{\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^n), \tag{8b}$$

where $\tilde{\mathbf{x}}_{\mathrm{BE}}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^n + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}_{\mathrm{ext}}$ is the predicted future state without the contribution of $\mathbf{f}^{n+1}$ [Gast et al. 2015; Martin et al. 2011]. An analogous optimization formulation for the implicit midpoint scheme has recently been provided by Dinev et al. [2018a].

For the general family of integrators (7), it is useful to define the discretized time derivative operator

$$\delta(\mathbf{y}) = \frac{1}{\alpha \Delta t}(\mathbf{y} - \mathbf{y}^p), \tag{9}$$

so that the time integration equations are simply

$$\delta(\mathbf{x}^{n+1}) = \mathbf{v}^{n+1}, \tag{10a}$$

$$\mathbf{M}\delta(\mathbf{v}^{n+1}) = \mathbf{f}(\mathbf{x}^{n+1}, \mathbf{v}^{n+1}). \tag{10b}$$

If all the forces are nondissipative, i.e. $\mathbf{f}(\mathbf{x}, \mathbf{v}) = \mathbf{f}_{\mathrm{ext}} - \nabla U(\mathbf{x})$, we can eliminate $\mathbf{v}^{n+1}$ from the second equation to obtain

$$\frac{1}{\alpha^2 \Delta t^2} \mathbf{M}\big(\mathbf{x}^{n+1} - (\mathbf{x}^p + \alpha \Delta t \mathbf{v}^p)\big) = \mathbf{f}_{\mathrm{ext}} - \nabla U(\mathbf{x}^{n+1}). \tag{11}$$

This corresponds to the minimization

$$\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}} \Big(\frac{1}{2\alpha^2 \Delta t^2} \|\mathbf{x} - \tilde{\mathbf{x}}^{n+1}\|_{\mathbf{M}} + U(\mathbf{x}) \Big) \tag{12a}$$

$$\text{where} \quad \tilde{\mathbf{x}}^{n+1} = \mathbf{x}^p + \alpha \Delta t \mathbf{v}^p + \alpha^2 \Delta t^2 \mathbf{M}^{-1} \mathbf{f}_{\mathrm{ext}}. \tag{12b}$$

Similar to the backward Euler case, $\tilde{\mathbf{x}}^{n+1}$ is the predicted future state without $\mathbf{f}^{n+1}$. Therefore, time integration for a nondissipative system is exactly equivalent to solving the minimization (12a) followed by setting $\mathbf{v}^{n+1} = \delta(\mathbf{x}^{n+1})$. In Section 4, we extend this formulation to support dissipative forces.

3.3 ADMM for local-global optimization

The objective functions in (8a) and (12a) are optimized over the entire global displacement vector $\mathbf{x} \in \mathbb{R}^{3n}$. This global solve is expensive to compute. When applicable, using a local-global approach can greatly improve the performance. Overby et al. [2017] expressed the backward Euler optimization in a form compatible with the ADMM optimization algorithm [Boyd et al. 2011]. We briefly review their approach here. In Section 4 we will do this more generally for the higher-order TR-BDF2 integrator we use in this work.

First, we assume that the potentials U are the sum of several terms of the form

$$U(\mathbf{x}) = \sum_i U_i(\mathbf{D}_i \mathbf{x}) = U_*(\mathbf{D}\mathbf{x}), \tag{13}$$

where each U_i depends only on a low-dimensional vector of local coordinates $\underline{\mathbf{x}}_i = \mathbf{D}_i \mathbf{x}$. Here, each $\mathbf{D}_i$ is a matrix which encodes the linear mapping from global coordinates of the entire system to the local coordinates relevant for the particular energy term, such as the edge vector for a spring or the deformation gradient for a finite element. The aggregate transformation $\mathbf{D}$ is obtained by vertically concatenating the individual $\mathbf{D}_i$ matrices.

Equipped with this notation, we rewrite the optimization problem in (8a) as

$$\min_{\mathbf{x}} \left(\frac{1}{2\Delta t^2} \|\mathbf{x} - \tilde{\mathbf{x}}_{\mathrm{BE}}^{n+1}\|_{\mathbf{M}}^2 + U_*(\mathbf{D}\mathbf{x}) \right). \tag{14}$$

Treating $\mathbf{D}\mathbf{x}$ as an independent variable $\underline{\mathbf{x}}$, this can be expressed in the form

$$\begin{aligned} &\min_{\mathbf{x}, \underline{\mathbf{x}}} \ \frac{1}{2\Delta t^2} \|\mathbf{x} - \tilde{\mathbf{x}}_{\mathrm{BE}}^{n+1}\|_{\mathbf{M}}^2 + U_*(\underline{\mathbf{x}}) \\ &\text{s.t. } \mathbf{W}(\mathbf{D}\mathbf{x} - \underline{\mathbf{x}}) = \mathbf{0}, \end{aligned} \tag{15}$$

where $\mathbf{W}$ is an invertible matrix that acts as a weighting of different potential terms. This form permits the use of the ADMM algorithm [Boyd et al. 2011], which solves the optimization problem iteratively, alternating between $\mathbf{x}$ and $\underline{\mathbf{x}}$ updates until convergence. Good heuristics for choosing $\mathbf{W}$ are discussed by Overby et al. [2017]. In brief, we choose $\mathbf{W}$ to be a block diagonal matrix where each block specifies the relative weight of the corresponding local energy term, with stiffer energies receiving larger weights.

Finally, the ADMM update equations which solve (15) are

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \Big(\frac{1}{2\Delta t^2} \|\mathbf{x} - \tilde{\mathbf{x}}^{n+1}\|_{\mathbf{M}}^2 + \frac{1}{2} \|\mathbf{W}(\mathbf{D}\mathbf{x} - \underline{\mathbf{x}}^{(k)} + \bar{\mathbf{u}}^{(k)})\|^2 \Big), \tag{16a}$$

$$\underline{\mathbf{x}}^{(k+1)} = \arg\min_{\underline{\mathbf{x}}} \Big(U_*(\underline{\mathbf{x}}) + \frac{1}{2} \|\mathbf{W}(\mathbf{D}\mathbf{x}^{(k+1)} - \underline{\mathbf{x}} + \bar{\mathbf{u}}^{(k)})\|^2 \Big), \tag{16b}$$

$$\bar{\mathbf{u}}^{(k+1)} = \bar{\mathbf{u}}^{(k)} + \mathbf{D}\mathbf{x}^{(k+1)} - \underline{\mathbf{x}}^{(k+1)}, \tag{16c}$$

where k denotes the iteration number, $\bar{\mathbf{u}}$ is a new variable introduced for ADMM, and the initial guesses are $\mathbf{x}^{(0)} = \tilde{\mathbf{x}}^{n+1}$, $\underline{\mathbf{x}}^{(0)} = \mathbf{D}\tilde{\mathbf{x}}^{n+1}$, and $\bar{\mathbf{u}}^{(0)} = 0$. The benefit of ADMM is that each of these steps can be carried out efficiently. The objective in (16a) is a quadratic form with a constant Hessian matrix, which be minimized using a prefactored linear solve. The minimization in (16b) splits into separate low-dimensional minimizations

$$\underline{\mathbf{x}}_i^{(k+1)} = \arg\min_{\underline{\mathbf{x}}_i} \Big(U_i(\underline{\mathbf{x}}_i) + \frac{1}{2} \|\mathbf{W}(\mathbf{D}_i\mathbf{x}^{(k+1)} - \underline{\mathbf{x}}_i + \bar{\mathbf{u}}_i^{(k)})\|^2 \Big), \tag{17}$$

which can be performed in parallel, and for which closed-form solutions may be available.

4 METHOD

In this section, we first set up the *first-order* generalized optimization-based integration problem with the dissipation function R added. We show that the method is first-order accurate. In the next section, we derive a correction term that can be added to raise the accuracy to second order. Finally, in the following section we present the full second-order TR-BDF2 scheme.

4.1 Optimization integration with dissipative forces

In the presence of a nontrivial dissipation function R, the equation for $\mathbf{x}^{n+1}$ analogous to (11) is

$$\begin{aligned} \frac{1}{\alpha^2\Delta t^2} \mathbf{M}\big(\mathbf{x}^{n+1} - (\mathbf{x}^p + \alpha\Delta t \mathbf{v}^p)\big) &= \mathbf{f}(\mathbf{x}^{n+1}, \delta(\mathbf{x}^{n+1})) \\ &= \mathbf{f}_{\text{ext}} - \nabla U(\mathbf{x}^{n+1}) - \nabla_2 R(\mathbf{x}^{n+1}, \delta(\mathbf{x}^{n+1})). \end{aligned} \tag{18}$$

To lighten the notation, we will often abbreviate $\mathbf{f}(\mathbf{x}^{n+1}, \delta(\mathbf{x}^{n+1}))$, $U(\mathbf{x}^{n+1})$, and $R(\mathbf{x}^{n+1}, \delta(\mathbf{x}^{n+1}))$ as $\mathbf{f}^{n+1}$, U^{n+1}, and R^{n+1} respectively.

Let us consider the optimization-based scheme

$$\tilde{\mathbf{x}}^{n+1} = \mathbf{x}^p + \alpha\Delta t \mathbf{v}^p + \Delta t^2 \alpha^2 \mathbf{M}^{-1} \mathbf{f}_{\text{ext}}, \tag{19a}$$

$$\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}} \Big(\frac{1}{2\alpha^2\Delta t^2} \|\mathbf{x} - \tilde{\mathbf{x}}^{n+1}\|_{\mathbf{M}}^2 + U(\mathbf{x}) + \alpha\Delta t R(\mathbf{x}, \delta(\mathbf{x})) \Big), \tag{19b}$$

$$\mathbf{v}^{n+1} = \delta(\mathbf{x}^{n+1}). \tag{19c}$$

The optimality condition for (19b) is

$$\frac{1}{\alpha^2\Delta t^2} \mathbf{M}(\mathbf{x}^{n+1} - \tilde{\mathbf{x}}^{n+1}) + \nabla U^{n+1} + \alpha\Delta t \nabla_1 R^{n+1} + \nabla_2 R^{n+1} = 0, \tag{20}$$

or equivalently

$$\mathbf{M}\delta(\mathbf{v}^{n+1}) = \mathbf{f}^{n+1} - \alpha\Delta t \nabla_1 R^{n+1}. \tag{21}$$

The velocity update obtained from this integrator is therefore

$$\mathbf{v}^{n+1} = \mathbf{v}^p + \alpha\Delta t \mathbf{M}^{-1} \mathbf{f}^{n+1} - \Delta t^2 \alpha^2 \mathbf{M}^{-1} \nabla_1 R^{n+1}, \tag{22}$$

in which the final term is an $O(\Delta t^2)$ error proportional to $\nabla_1 R$.

From this analysis, it is clear that if the dissipative term R is independent of its first argument $\mathbf{x}$, this optimization problem yields the *exact* solution to (18). Even when this is not the case, for most dissipative forces the dependence of R on positions is much weaker than that on velocities, so in practice the associated error usually is dominated by the error due to the time integration itself. Therefore, we believe that this first-order scheme for dissipative forces is sufficiently accurate for the large time steps used in graphics.

It is true in principle that the $O(\Delta t^2)$ error incurred by this method prevents it from being second-order accurate even if the underlying integrator is. We describe in the following section a modified scheme that provides theoretical second-order accuracy, at the cost of an additional objective term.

4.2 Attaining second-order accuracy

The additional error in the previous method arises because the term $\alpha\Delta t R$ in the objective function gives rise to forces

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} \alpha\Delta t R(\mathbf{x}, \delta(\mathbf{x})) &= \alpha\Delta t \nabla_1 R + \alpha\Delta t \nabla_2 R \frac{\mathrm{d}\delta(\mathbf{x})}{\mathrm{d}\mathbf{x}} &(23) \\ &= \alpha\Delta t \nabla_1 R + \nabla_2 R, &(24) \end{aligned}$$

despite the actual dissipative force in the physical model being $\nabla_2 R$ alone. If we wish to elevate the method to second-order accuracy, we should replace the term $\alpha\Delta t R$ in the objective with a different term S such that

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} S(\mathbf{x}, \delta(\mathbf{x})) = \nabla_2 R(\mathbf{x}, \delta(\mathbf{x})) + O(\Delta t^2). \tag{25}$$

In particular, we choose

$$S(\mathbf{x}, \mathbf{v}) = \alpha\Delta t \big(R(\mathbf{x}, \mathbf{v}) - R(\mathbf{x}, \mathbf{v}^n) \big), \tag{26}$$

which yields

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} S(\mathbf{x}, \delta(\mathbf{x})) &= \nabla_2 R(\mathbf{x}, \delta(\mathbf{x})) + \alpha\Delta t (\nabla_1 R(\mathbf{x}, \delta(\mathbf{x})) - \nabla_1 R(\mathbf{x}, \mathbf{v}^n)) \\ &= \nabla_2 R(\mathbf{x}, \delta(\mathbf{x})) + \alpha\Delta t \, O(\delta(\mathbf{x}) - \mathbf{v}^n) \\ &= \nabla_2 R(\mathbf{x}, \delta(\mathbf{x})) + O(\Delta t^2). \end{aligned} \tag{27}$$

This reduces the error term on the velocity update in (22) to $O(\Delta t^3)$.

This strategy improves the theoretical order of accuracy of the method, but it introduces higher-order errors arising from the $R(\mathbf{x}, \mathbf{v}^n)$ term. As we show in Section 7.1, this can be a net negative in cases where the original $\alpha\Delta t \nabla_1 R$ error is not significant. We have presented it here primarily for theoretical completeness, and as a starting point for future work.

We now summarize our optimization-based time stepping scheme. For convenience, in the following scheme we combine the conservative and dissipative potentials to define an "effective potential" which is either

$$\Phi(\mathbf{x}) = U(\mathbf{x}) + \alpha\Delta t R(\mathbf{x}, \delta(\mathbf{x})) \tag{28}$$

Fig. 2. A sheet of cloth, modeled with strain-limited linear elasticity and quadratic bending elements [Bergou et al. 2006], swings under the influence of gravity. Energy and angular momentum are quickly damped using Backward Euler (left). Both BDF2 (middle) and TR-BDF2 (right) result in a minimally damped swing, but the TR-BDF2 result exhibits visibly more wrinkles. For comparable computational cost, we have run TR-BDF2 with twice the time step size (40 ms) as compared to backward Euler and BDF2.

for the first-order accurate method, or

$$\Phi(\mathbf{x}) = U(\mathbf{x}) + S(\mathbf{x}, \delta(\mathbf{x})) \tag{29}$$

$$= U(\mathbf{x}) + \alpha\Delta t\big(R(\mathbf{x}, \delta(\mathbf{x})) - R(\mathbf{x}, \mathbf{v}^n)\big) \tag{30}$$

for second order. Then, our generic time-stepping scheme is

$$\tilde{\mathbf{x}}^{n+1} = \mathbf{x}^p + \alpha\Delta t\mathbf{v}^p + \Delta t^2\alpha^2\mathbf{M}^{-1}\mathbf{f}_{\text{ext}}, \tag{31a}$$

$$\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}} \left(\frac{1}{2\alpha^2\Delta t^2}\|\mathbf{x} - \tilde{\mathbf{x}}^{n+1}\|_{\mathbf{M}}^2 + \Phi(\mathbf{x})\right), \tag{31b}$$

$$\mathbf{v}^{n+1} = \delta(\mathbf{x}^{n+1}) = \frac{1}{\alpha\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^p). \tag{31c}$$

4.3 Optimization-based TR-BDF2

TR-BDF2 is a composite integrator which combines the trapezoidal method (TR) and the second-order backward difference formula (BDF2). This combination yields an L-stable time integration scheme which shows excellent conservation of energy and angular momentum, as illustrated in Figure 2.

Applied to the generic ODE $\dot{\mathbf{y}} = \boldsymbol{\phi}(\mathbf{y})$, TR-BDF2 proceeds by first using the trapezoidal method to obtain a mid-step prediction for the state $\mathbf{y}^{n+\gamma}$ at time $t + \gamma\Delta t$, where $\gamma \in (0, 1)$ is a parameter:

$$\mathbf{y}^{n+\gamma} = \mathbf{y}^n + \frac{1}{2}\gamma\Delta t\boldsymbol{\phi}(\mathbf{y}^n) + \frac{1}{2}\gamma\Delta t\boldsymbol{\phi}(\mathbf{y}^{n+1}). \tag{32}$$

Then a three-point backward difference formula is applied at t, $t+\gamma\Delta t$, and $t+\Delta t$ to determine the final state at $t+\Delta t$. The backward difference formula takes the form

$$\delta_{\text{BDF2}}(\mathbf{y}^{n+1}) = \frac{1}{\Delta t}(c_1\mathbf{y}^n + c_2\mathbf{y}^{n+\gamma} + c_3\mathbf{y}^{n+1}) \tag{33}$$

where $c_1 = \frac{1-\gamma}{\gamma}$, $c_2 = -\frac{1}{\gamma(1-\gamma)}$, and $c_3 = \frac{2-\gamma}{1-\gamma}$, yielding the time integration equation

$$\mathbf{y}^{n+1} = -\frac{c_2}{c_3}\mathbf{y}^{n+\gamma} - \frac{c_1}{c_3}\mathbf{y}^n + \frac{1}{c_3}\Delta t\boldsymbol{\phi}(\mathbf{y}^{n+1})$$

$$= \frac{1}{\gamma(2-\gamma)}\mathbf{y}^{n+\gamma} - \frac{(1-\gamma^2)}{\gamma(2-\gamma)}\mathbf{y}^n + \frac{1-\gamma}{2-\gamma}\Delta t\boldsymbol{\phi}(\mathbf{y}^{n+1}). \tag{34}$$

Since both the TR and BDF2 stages are of the form (6), it is straightforward to derive optimization formulations for them using (31). For the TR stage, we have

$$\alpha = \frac{\gamma}{2}, \tag{35a}$$

$$\mathbf{x}^p = \mathbf{x}^n + \frac{1}{2}\gamma\Delta t\mathbf{v}^n, \tag{35b}$$

$$\mathbf{v}^p = \mathbf{v}^n + \frac{1}{2}\gamma\Delta t\mathbf{M}^{-1}\mathbf{f}^n, \tag{35c}$$

and for BDF2,

$$\alpha = \frac{1-\gamma}{2-\gamma}, \tag{36a}$$

$$\mathbf{x}^p = \frac{1}{\gamma(2-\gamma)}\mathbf{x}^{n+\gamma} - \frac{(1-\gamma)^2}{\gamma(2-\gamma)}\mathbf{x}^n, \tag{36b}$$

$$\mathbf{v}^p = \frac{1}{\gamma(2-\gamma)}\mathbf{v}^{n+\gamma} - \frac{(1-\gamma)^2}{\gamma(2-\gamma)}\mathbf{v}^n. \tag{36c}$$

While some previous work has used $\gamma = \frac{1}{2}$ [Bathe 2007; Bathe and Baig 2005; Xu and Barbič 2017], we follow the original formulation of TR-BDF2 [Bank et al. 1985] in using $\gamma = 2 - \sqrt{2}$, which is the only choice for which TR-BDF2 is L-stable [Hosea and Shampine 1996]. This choice also yields the same value of α for both stages, permitting the global matrix in ADMM to be reused.

Finally, the TR stage requires the total force vector $\mathbf{f}^n$ at the beginning of the time step. Computing this explicitly is unreliable in the presence of nonsmooth potentials. We do so only for the first time step, assuming that the system is initialized away from nonsmooth states. Subsequently, we compute the force at the end of each BDF2 stage using the impulse-momentum theorem,

$$\mathbf{f}^{n+1} = \mathbf{M}\delta_{\text{BDF2}}(\mathbf{v}^{n+1}), \tag{37}$$

and store it for use in the TR stage of the next time step.

4.4 Local-global optimization

In section 3.3, we took advantage of the separability of the conservative potential U to permit a local-global optimization for backward Euler. The same strategy can be applied in our setting, this time using effective potentials Φ instead of U. The effective potential Φ is again assumed to be a sum of local potentials Φ_i via

$$\Phi(\mathbf{x}) = \sum_i \Phi_i(\mathbf{D}_i\mathbf{x}) = \Phi_*(\mathbf{D}\mathbf{x}), \tag{38}$$

$$\text{where} \quad \Phi_i(\underline{\mathbf{x}}_i) = U_i(\underline{\mathbf{x}}_i) + S_i(\underline{\mathbf{x}}_i, \delta(\underline{\mathbf{x}}_i)), \tag{39}$$

defined in terms of local conservative and dissipative potentials U_i and S_i respectively. Here $\delta(\underline{\mathbf{x}}_i)$ denotes the time-discretized local velocity, equal to $\mathbf{D}_i\delta(\mathbf{x}_i) = \frac{1}{\alpha\Delta t}(\underline{\mathbf{x}}_i - \mathbf{D}_i\mathbf{x}^p)$. With this effective potential, the ADMM iterations still have the same structure as (16), but with local updates

$$\underline{\mathbf{x}}_i^{(k+1)} = \arg\min_{\underline{\mathbf{x}}_i}\left(\Phi_i(\underline{\mathbf{x}}_i) + \frac{1}{2}\|\mathbf{W}(\mathbf{D}_i\mathbf{x}^{(k+1)} - \underline{\mathbf{x}}_i + \bar{\mathbf{u}}_i^{(k)})\|^2\right). \tag{40}$$

There is some freedom in the choice of the terms Φ_i. For a damped elastic object, we may create a single local potential Φ_i for each element that includes both the elastic energy and the dissipation function, or two independent local potentials, one equalling U_i and the other equalling S_i. The choice is determined largely by performance considerations: using separate terms may take more iterations to converge, but the tradeoff is often worthwhile if each local optimization (40) can be solved more efficiently.

5 FORCES

In the following subsections, we provide local dissipation functions $R_i(\underline{\mathbf{x}}_i, \underline{\mathbf{v}}_i)$ for a variety of dissipative forces. From these functions one can use (26) to compute S_i if needed for second-order accuracy. Whenever possible, we also discuss closed-form minimizers or fast iterative methods for (40) with these dissipation functions.

To lighten the notational burden, we will omit the subscript i on local quantities when it does not hurt clarity.

5.1 Viscous damping

First, we discuss damping models for finite-element elastic objects. In this setting, the local coordinates $\underline{\mathbf{x}}$ and $\underline{\mathbf{v}}$ contain the entries of the deformation gradient $\mathbf{F}$ and its time derivative $\dot{\mathbf{F}}$ in vectorized form, and gradients of local potentials represent Piola-Kirchhoff stresses, e.g. $\mathbf{P} = \frac{\mathrm{d}U}{\mathrm{d}\mathbf{F}}$. For clarity, we will express the dissipation function as $R(\mathbf{F}, \dot{\mathbf{F}})$ instead of $R(\underline{\mathbf{x}}, \underline{\mathbf{v}})$ when convenient.

5.1.1 Strain-rate proportional damping. In an elastic material with deformation gradient $\mathbf{F}$, the Green strain tensor $\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})$ is a rotation-invariant measure of the instantaneous elastic deformation. Its time derivative, the strain rate tensor

$$\dot{\mathbf{E}} = \frac{1}{2}(\dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\dot{\mathbf{F}}), \tag{41}$$

is invariant to rigid motions and measures the rate at which the elastic deformation is changing. A simple model for viscous damping can be obtained by assuming that the damping forces depends linearly and isotropically on $\dot{\mathbf{E}}$ [O'Brien and Hodgins 1999]. In this case, the viscous Piola-Kirchhoff stress is of the form

$$\mathbf{P} = \mathbf{F}(2\psi\dot{\mathbf{E}} + \phi\,\mathrm{tr}(\dot{\mathbf{E}})\mathbf{I}), \tag{42}$$

where ψ and ϕ are material parameters controlling shear and bulk damping. This damping model corresponds to a dissipation function

$$R(\mathbf{F}, \dot{\mathbf{F}}) = \psi\|\dot{\mathbf{E}}(\mathbf{F}, \dot{\mathbf{F}})\|^2 + \frac{\phi}{2}\mathrm{tr}^2\dot{\mathbf{E}}(\mathbf{F}, \dot{\mathbf{F}}). \tag{43}$$

We have $\nabla_2 R(\mathbf{F}, \dot{\mathbf{F}}) = \mathbf{P}$ as desired, while

$$\nabla_1 R(\mathbf{F}, \dot{\mathbf{F}}) = \dot{\mathbf{F}}(2\psi\dot{\mathbf{E}} + \phi\,\mathrm{tr}(\dot{\mathbf{E}})\mathbf{I}). \tag{44}$$

On the face of it, computing the local step with this model requires a 9-dimensional optimization, since $\mathbf{F}$ is a 3×3 matrix. In Appendix A, we show how the rotation invariance of $\mathbf{E}$ allows us to simplify this to a 6-dimensional minimization coupled with a polar decomposition.

5.1.2 Rayleigh damping. An alternative to strain-rate proportional damping is Rayleigh damping, in which the damping forces due to each element are

$$\mathbf{f}(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = \beta\mathbf{K}(\underline{\mathbf{x}})\underline{\mathbf{v}}, \tag{45}$$

where β is a scaling parameter and $\mathbf{K}(\underline{\mathbf{x}})$ is a matrix characterizing the element's stiffness in local coordinates, usually taken to be the tangent stiffness matrix $-\partial^2 U/\partial\underline{\mathbf{x}}^2$.

For fully implicit Rayleigh damping, we define the dissipation function

$$R(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = \frac{\beta}{2}\underline{\mathbf{v}}^T\mathbf{K}(\underline{\mathbf{x}})\underline{\mathbf{v}}. \tag{46}$$

Its gradients are

$$\nabla_2 R(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = \beta\mathbf{K}(\underline{\mathbf{x}})\underline{\mathbf{v}}, \tag{47}$$

$$\nabla_1 R(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = \frac{\beta}{2}\left[\underline{\mathbf{v}}^T\frac{\partial\mathbf{K}(\underline{\mathbf{x}})}{\partial\underline{x}_i}\underline{\mathbf{v}}\right]_{i=1,2,3}. \tag{48}$$

Differentiating $\mathbf{K}$ with respect to $\mathbf{x}$ may be difficult to compute in general. However, the gradient in (48) only requires the product of the derivative with $\mathbf{v}$, which can in fact be computed using the time-derivative of $\mathbf{K}$:

$$\underline{\mathbf{v}}^T\frac{\partial\mathbf{K}}{\partial\underline{x}_i}\underline{\mathbf{v}} = -\sum_{j,k} v_j\left(\frac{\partial}{\partial x_i}\frac{\partial^2 U}{\partial x_j\partial x_k}\right)v_k \tag{49}$$

$$= -\sum_j v_j\sum_k\left(\frac{\partial}{\partial x_k}\frac{\partial^2 U}{\partial x_i\partial x_j}\right)v_k = \underline{\mathbf{v}}^T\dot{\mathbf{K}}. \tag{50}$$

We further approximate $\dot{\mathbf{K}}$ by employing the same scheme used in the time integrator,

$$\dot{\mathbf{K}} \approx \delta(\mathbf{K}) = \frac{1}{\alpha\Delta t}(\mathbf{K}(\underline{\mathbf{x}}) - \mathbf{K}(\mathbf{x}^p)). \tag{51}$$

One little-discussed drawback of Rayleigh damping with the tangent stiffness matrix is that the resulting forces may not conserve angular momentum. In Section 6 we discuss this phenomenon and propose a simple correction scheme that ensures that the forces are angular momentum conserving.

5.1.3 Damping of principal strains. In isotropic hyperelastic models, the potential U_i can be viewed as a function only of the principal strains, i.e. the 3 singular values of $\mathbf{F}$, rather than on the full 3×3 matrix $\mathbf{F}$. Then it is possible to reduce the local step to a 3-dimensional optimization, leading to a significant speedup [Overby et al. 2017].

To achieve the same speedup in dissipative models, we propose a simplified damping model that acts directly on the rate of change of principal strains. Let $\boldsymbol{\sigma}$ be the vector of principal strains, obtained from $\mathbf{F}$ via the singular value decomposition. Applying the discretized time derivative directly to $\boldsymbol{\sigma}$, we can approximate

$$\dot{\boldsymbol{\sigma}} \approx \delta(\boldsymbol{\sigma}) := \frac{1}{\alpha\Delta t}(\boldsymbol{\sigma} - \boldsymbol{\sigma}^p), \tag{52}$$

where $\boldsymbol{\sigma}^p$ contains the singular values of the matrix corresponding to $\underline{\mathbf{x}}^p$.

By analogy with Rayleigh damping, we would like to define dissipative forces of strength $\beta\mathbf{K}_{\boldsymbol{\sigma}}(\boldsymbol{\sigma})\dot{\boldsymbol{\sigma}}$ along the principal strain directions, where $\mathbf{K}_{\boldsymbol{\sigma}} = \partial^2 U_i/\partial\boldsymbol{\sigma}^2$ is the Hessian of the element's elastic energy with respect to principal strains. One way to do this is to define a dissipation function $R(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}) = \frac{\beta}{2}\dot{\boldsymbol{\sigma}}^T\mathbf{K}_{\boldsymbol{\sigma}}\dot{\boldsymbol{\sigma}}$, although this would then require a second-order correction. Instead, inspired by Gast et al. [2015], we define the effective potential

$$\Phi(\boldsymbol{\sigma}) = \beta\left(\delta(\boldsymbol{\sigma})^T\nabla U_i(\boldsymbol{\sigma}) - \frac{1}{\alpha\Delta t}U_i(\boldsymbol{\sigma})\right), \tag{53}$$

which has exactly the correct gradient with respect to $\boldsymbol{\sigma}$. Since this depends only on the singular values of $\mathbf{F}$, we can treat this the same

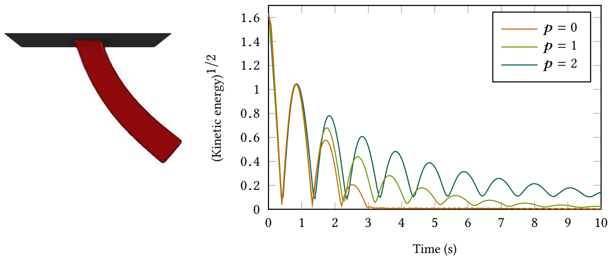

Fig. 3. Energy behavior of power-law damping on an elastic pendulum. The strength of damping is chosen so that each model damps roughly the same amount of kinetic energy in the first swing. To make the behavior more clearly visible, we plot the square root of kinetic energy, which is proportional to the RMS velocity.

way as an isotropic hyperelastic energy. In particular, in the local step (40), the singular vectors of $\mathbf{F}$ must be the same as those of the matrix corresponding to $\mathbf{D}_i\mathbf{x}^{(k+1)} + \bar{\mathbf{u}}_i^{(k)}$, so the objective can be minimized over $\boldsymbol{\sigma}$ alone.

Note that this model achieves its simplicity by ignoring changes in $\mathbf{V}$, so motions in which the principal strain axes rotate within the material will not be damped by this model. However, since it also ignores changes in $\mathbf{U}$, it conserves angular momentum and does not damp out rigid motions unlike Rayleigh damping. In practice we find this to be a worthwhile tradeoff.

5.2 Nonlinear damping

While the viscous damping forces described above are linear in the velocities $\underline{\mathbf{v}}$, many other dissipative forces in nature have a nonlinear relationship with velocity. In this section, we discuss some nonlinear models which enable more interesting damping effects than are possible with viscous damping.

5.2.1 Power-law damping. Consider a dissipation function $R(\underline{\mathbf{x}}, \underline{\mathbf{v}})$ which is quadratic in $\underline{\mathbf{v}}$ and so gives rise to linear damping forces; each of the models in Section 5.1 falls into this category. We define its power-law variant by raising it to an exponent $q = (p+1)/2$,

$$R'(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = A^{1-q} R(\underline{\mathbf{x}}, \underline{\mathbf{v}})^q, \tag{54}$$

where A is a proportionality constant with the same units as R. As an illustrative example, if we take $R = \|\underline{\mathbf{v}}\|^2$ and $A = 1$, the dissipation function and its associated forces are simply

$$R'(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = \|\underline{\mathbf{v}}\|^{p+1} \tag{55}$$

$$\implies \nabla_2 R'(\underline{\mathbf{x}}, \underline{\mathbf{v}}) = (p+1)\|\underline{\mathbf{v}}\|^p \frac{\underline{\mathbf{v}}}{\|\underline{\mathbf{v}}\|}, \tag{56}$$

showing the desired power-law behavior of the force magnitude. The power-law model will inherit properties such as isotropy, rotation invariance, and angular momentum conservation from the underlying linear model R.

The local minimization for power-law damping can be numerically challenging, especially for the $p = 0$ case in which $R'(\underline{\mathbf{x}}, \underline{\mathbf{v}})$ is nonsmooth at $\underline{\mathbf{v}} = \mathbf{0}$. In Appendix B, we show that for a certain choice of underlying linear model, a fast update can be performed using only a one-dimensional root finding step.

The power-law damping model enables a range of qualitatively distinct energy beahviors, as demonstrated in Figure 3. In the three cases $p < 1$, $p = 1$, and $p > 1$, the kinetic energy of the system decays superexponentially, exponentially, and subexponentially respectively. Thus, using the parameter p the user can control whether the system should come to rest in finite time, show lively long-lived oscillations, or anything in between.

5.2.2 Spline-based damping. Inspired by recent work on artistic design of elasticity [Xu et al. 2015] and damping [Xu and Barbič 2017], we propose a simple spline-based method to give the user direct control over nonlinear damping properties. The user provides a spline function $s : \mathbb{R} \to \mathbb{R}$ specifying a damping multiplier as a function of principal strain. For each element, we apply s to the three singular values and use the resulting multipliers to modulate an existing dissipation function R. This is similar to the example-based damping design approach of Xu and Barbič [2017], but applied to per-element deformations rather than modal bases.

One simple method that works for a generic R is to combine them into a single multiplier:

$$R'(\mathbf{F}, \dot{\mathbf{F}}) = (\max_{j=1,2,3} s(\sigma_j)) R(\mathbf{F}, \dot{\mathbf{F}}). \tag{57}$$

Many alternative approaches are possible. In the context of principal strain damping, one could scale the damping of each principal strain value separately using $R' = \frac{\beta}{2}\dot{\boldsymbol{\sigma}}^T \mathbf{S}^{1/2}\mathbf{K}_{\boldsymbol{\sigma}}\mathbf{S}^{1/2}\dot{\boldsymbol{\sigma}}$, where $\mathbf{S}$ is the diagonal matrix with entries $s(\sigma_j)$. One could also apply the spline function to the principal strain rates $\dot{\boldsymbol{\sigma}}$ instead of to the strains $\boldsymbol{\sigma}$ themselves, causing the damping to be nonlinear in velocity.

5.3 Coulomb Friction

Friction is a dissipative force which acts to reduce the tangential velocity of objects in contact. According to the dry Coulomb model, the force of friction $\mathbf{f}_f$ between two contacting objects lies in the tangent plane at the point of contact and satisfies

$$\begin{cases} \|\mathbf{f}_f\| \le \mu\|\mathbf{f}_n\| & \text{if } \mathbf{v}_t = \mathbf{0}, \\ \mathbf{f}_f = -\mu\|\mathbf{f}_n\|\frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} & \text{otherwise.} \end{cases} \tag{58}$$

Here, μ is the coefficient of friction, $\mathbf{f}_n$ is the normal force, and $\mathbf{v}_t$ is the tangential relative velocity. As is typical in graphics, we have assumed that μ is the same for both static and kinetic friction.

To model friction in our optimization framework, we create for each contact pair a dissipative term such that $\mathbf{D}_i\mathbf{v}$ gives the relative velocity at the point of contact. The dissipation function for Coulomb friction is

$$R(\underline{\mathbf{v}}) = \mu\|\mathbf{f}_n\|\|\mathbf{T}\underline{\mathbf{v}}\|, \tag{59}$$

where the matrix $\mathbf{T}$ performs orthogonal projection to the contact tangent plane. The subdifferential of R is

$$\partial R(\underline{\mathbf{v}}) = \begin{cases} \{\mathbf{T}\mathbf{f} : \|\mathbf{f}\| \le \mu\|\mathbf{f}_n\|\} & \text{if } \mathbf{T}\underline{\mathbf{v}} = \mathbf{0}, \\ \{\mu\|\mathbf{f}_n\|\frac{\mathbf{T}\underline{\mathbf{v}}}{\|\mathbf{T}\underline{\mathbf{v}}\|}\} & \text{otherwise,} \end{cases} \tag{60}$$

so $\mathbf{f}_f \in -\partial R(\underline{\mathbf{v}})$ is equivalent to the desired Coulomb law. Furthermore, the nonsmoothness of the dissipation function (59) does not pose a problem for us, since the local optimization (40) has a simple closed form in this case.

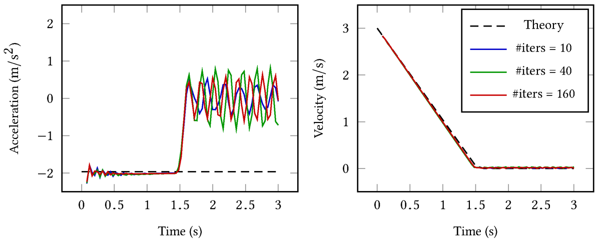

Fig. 4. Dynamic friction validation. An undamped elastic box slides along a horizontal surface and decelerates due to friction ($\mu = 0.2$). The rate of deceleration matches the physical prediction for different numbers of ADMM iterations.

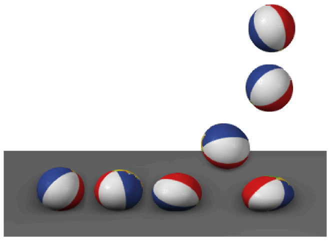

Fig. 5. A spinning elastic ball is dropped onto a flat surface. Due to frictional contact with the surface the ball picks up net horizontal velocity and subsequently bounces and rolls off to the side.

The definition of the frictional dissipation function requires knowledge of the normal forces $\mathbf{f}_n$. We assume these can be obtained from the collision solver as in Overby et al. [2017]. In our implementation the normal forces are recomputed after each global step, and the updated values are used for all friction computations in the subsequent local step. Technically, under this scheme, we are no longer solving an optimization problem, since the objective term (59) changes at each iteration. However, in practice we find the method still converges to physically valid behavior in all tested cases.

In Figure 4 we demonstrate the convergence of the method relative to an analytical prediction. Another example is shown in Figure 5, where a spinning ball dropped onto a flat surface picks up net motion in the direction tangent to the contact normal.

6 CONSERVATION OF ANGULAR MOMENTUM

It is a well-known consequence of Noether's theorem that conservative forces preserve linear and angular momentum if and only if the associated potential is invariant to translation and rotation. The analogous condition does not hold for dissipative forces. In particular, the dissipation function for Rayleigh damping is invariant to translation and rotation, but Rayleigh damping fails to conserve angular momentum even in the continuous-time case. In this section, we derive the theoretical requirements for a dissipation function to conserve angular momentum, and propose techniques to modify any dissipation function to satisfy these requirements.

6.1 Theory

Consider an arbitrary dissipation function R, giving rise to forces $\mathbf{f} = \nabla_2 R$. For linear and angular momentum to be conserved, the net force and net torque must be zero, that is,

$$\sum_i \mathbf{f}_i = \mathbf{0}, \tag{61a}$$

$$\sum_i \mathbf{x}_i \times \mathbf{f}_i = \mathbf{0}. \tag{61b}$$

This is true if and only if $\mathbf{f}$ is orthogonal to all *rigid* velocities $\mathbf{v}^r$, that is, all velocities of the form

$$\mathbf{v}^r = \begin{bmatrix} \mathbf{t} + \boldsymbol{\omega} \times \mathbf{x}_1 \\ \vdots \\ \mathbf{t} + \boldsymbol{\omega} \times \mathbf{x}_n \end{bmatrix} \tag{62}$$

determined by a linear and an angular velocity $\mathbf{t}, \boldsymbol{\omega} \in \mathbb{R}^3$. Since $\mathbf{f} = \nabla_2 R$, this implies that R must be invariant to addition of rigid velocities: $R(\mathbf{x}, \mathbf{v}) = R(\mathbf{x}, \mathbf{v} + \mathbf{v}^r)$ for any rigid $\mathbf{v}^r$. Conversely, if R has this invariance property, then $\nabla_2 R$ automatically satisfies the conservation equations (61).

One natural way to construct such a dissipation function R is as follows. Suppose $\mathbf{s}(\mathbf{x})$ is a translation- and rotation-invariant function of the positions $\mathbf{x}$. One can show that its time derivative $\dot{\mathbf{s}}(\mathbf{x}, \mathbf{v}) = \frac{\mathrm{d}\mathbf{s}(\mathbf{x})}{\mathrm{d}\mathbf{x}} \mathbf{v}$ is invariant to addition of rigid velocities. By our result above, any dissipation function R that depends solely on $\dot{\mathbf{s}}$ conserves linear and angular momentum. For example, strain-rate proportional damping is of this form with $\mathbf{s} = \mathbf{E}$, as is principal strain damping with $\mathbf{s} = \Sigma$; this verifies again the fact that these models conserve linear and angular momentum.

If we are given a user-specified dissipation function R that is not of this form, how can we modify it to guarantee angular momentum conservation? It suffices to choose a decomposition of velocities into rigid and nonrigid components, $\mathbf{v} = \mathbf{v}^r + \mathbf{v}^{nr}$, and to define the corrected dissipation function

$$\bar{R}(\mathbf{x}, \mathbf{v}) = R(\mathbf{x}, \mathbf{v}^{nr}). \tag{63}$$

As long as the nonrigid velocity $\mathbf{v}^{nr}$ yielded by the decomposition is unchanged upon adding any rigid velocity to $\mathbf{v}$, our corrected function $\bar{R}$ will conserve linear and angular momentum. Below, we present one method to do so in general, and another method specialized for volumetric finite element models.

6.2 General method

Define the matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_3 & -[\mathbf{x}_1]_\times \\ \vdots & \vdots \\ \mathbf{I}_3 & -[\mathbf{x}_n]_\times \end{bmatrix}, \tag{64}$$

so that rigid velocities $\mathbf{v}^r = \mathbf{C} \begin{bmatrix} \mathbf{t} \\ \boldsymbol{\omega} \end{bmatrix}$ form the column space of $\mathbf{C}$. Given a velocity $\mathbf{v}$, a natural choice of decomposition is via the orthogonal projection to this space,

$$\mathbf{v}^r = \mathbf{P}^r \mathbf{v} := \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \mathbf{v}, \tag{65}$$

$$\mathbf{v}^{nr} = \mathbf{P}^{nr} \mathbf{v} := (\mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T)\mathbf{v}. \tag{66}$$

If $\mathbf{C}^T\mathbf{C}$ is not invertible due to the points $\mathbf{x}_i$ not being in general position, we retain only the linearly independent column(s) of $\mathbf{C}$ and solve the problem in the reduced space.

Since most forces of interest act only on a small subset of vertices involved in the local coordinates $\underline{\mathbf{x}}_i = \mathbf{D}_i\mathbf{x}$ and $\underline{\mathbf{v}}_i = \mathbf{D}_i\mathbf{v}$, we rewrite these conditions in terms of the local coordinates. The local rigid velocity, $\underline{\mathbf{v}}_i^r = \mathbf{D}_i\mathbf{v}^r$, lies in the column space of $\mathbf{C}_i = \mathbf{D}_i\mathbf{C}$. The local coordinates are typically translation-invariant, in which case the first three columns of $\mathbf{D}_i\mathbf{C}$ become zero. Therefore, we retain only the remaining nonzero columns in $\mathbf{C}_i$. We then define the nonrigid projection in local space $\mathbf{P}_i^{nr}$ analogously to (66), and obtain the corrected dissipation function $\bar{R}_i(\underline{\mathbf{x}}_i, \underline{\mathbf{v}}_i) = R_i(\underline{\mathbf{x}}_i, \mathbf{P}_i^{nr}\underline{\mathbf{v}}_i)$.

6.3 Velocity gradient approach for volumetric elements

When the deformation gradient is a square matrix, as is the case when simulating volumetric objects, a simpler angular momentum conservation strategy may be employed. In this case, we can define the velocity gradient $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$, an Eulerian quantity which measures how velocity varies with position. Its symmetric part, $\mathbf{D} = \frac{1}{2}(\mathbf{L}+\mathbf{L}^T)$, is the rate of deformation tensor, and entirely encodes the nonrigid component of the motion. Since this is precisely the component we seek to isolate for angular momentum conservation, we propose substituting $\mathbf{D}$ in place of $\mathbf{L}$ in the equation $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$. Doing so yields the nonrigid component of $\dot{\mathbf{F}}$ as

$$\dot{\mathbf{F}}^{nr} = \mathbf{D}\mathbf{F} = \frac{1}{2}(\dot{\mathbf{F}} + \mathbf{F}^{-T}\dot{\mathbf{F}}^T\mathbf{F}). \tag{67}$$

This method can also be derived in a different way, revealing a connection to strain-rate proportional damping. We wish to construct a mapping from $\dot{\mathbf{F}}$ to some velocity $\dot{\mathbf{F}}^{nr}$ that is invariant to rigid velocities. We may proceed by first computing an intermediate quantity that exhibits this invariance, then deriving $\dot{\mathbf{F}}^{nr}$ from it. Choosing the intermediate quantity to be $\dot{\mathbf{E}} = \frac{1}{2}(\dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\dot{\mathbf{F}})$, and setting $\dot{\mathbf{F}}^{nr} = \mathbf{F}^{-T}\dot{\mathbf{E}}$ yields exactly the above definition. Thus, this approach can be viewed as importing rigid invariance from strain-rate proportional damping to an arbitrary dissipation function.

Having defined the nonrigid velocity $\dot{\mathbf{F}}^{nr}$, our corrected dissipation function is

$$\bar{R}(\mathbf{F}, \dot{\mathbf{F}}) = R(\mathbf{F}, \dot{\mathbf{F}}^{nr}), \tag{68}$$

where $\dot{\mathbf{F}}^{nr}$ depends on both $\mathbf{F}$ and $\dot{\mathbf{F}}$. To perform optimization, we will need the gradient of $\bar{R}$ with respect to both arguments. This is most easily expressed in terms of matrix differentials:

$$\delta\bar{R} = \nabla_1 R : \delta\mathbf{F} + \nabla_2 R : \delta\dot{\mathbf{F}}^{nr}, \tag{69}$$

$$\begin{aligned}\delta\dot{\mathbf{F}}^{nr} = \frac{1}{2}(&\mathbf{F}^{-T}\dot{\mathbf{F}}^T\delta\mathbf{F} - \mathbf{F}^{-T}\delta\mathbf{F}^{-T}\mathbf{F}^{-T}\dot{\mathbf{F}}^T\mathbf{F} \\ &+ \delta\dot{\mathbf{F}} + \mathbf{F}^{-T}\delta\dot{\mathbf{F}}^T\mathbf{F}).\end{aligned} \tag{70}$$

7 RESULTS

We implemented our method on top of an ADMM solver for hyperelastic materials [Overby et al. 2017]. To speed up the computation, we computed the primal and dual residuals [Boyd et al. 2011] at each ADMM iteration and terminated early if they were below a chosen threshold. In this section, we demonstrate our method on a variety of validation tests and complex examples. We encourage the reader to view the animated results in the supplementary video, since the effects of damping are difficult to convey in static images.

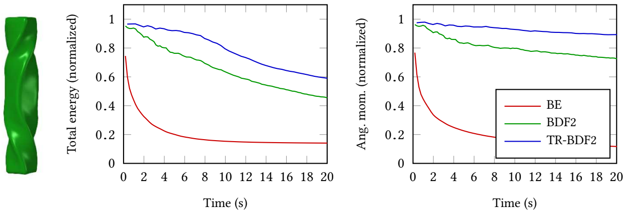

Fig. 6. An undamped elastic beam undergoing rotation and stretching, simulated with various methods. TR-BDF2 exhibits the best energy and angular momentum conservation for comparable computational cost.

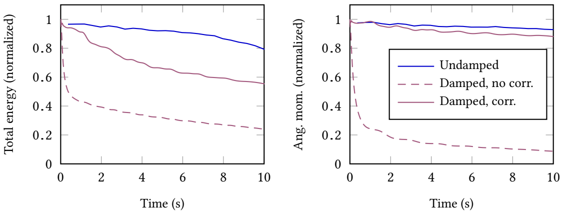

Fig. 7. A damped elastic beam simulated with TR-BDF2, using Rayleigh damping with and without our angular momentum correction. With correction, the remaining angular momentum loss is primarily that caused by the integrator itself.

7.1 Validation tests

We use the example shown in Figure 6 for several of our validation tests. This example consists of an elastic beam given initial velocities that induce rapid spinning and elongation. In the undamped case, the total energy and angular momentum of the system should remain conserved, but in practice most implicit integrators fail to do so exactly. As shown in the figure, backward Euler rapidly loses both energy and angular momentum. BDF2 performs significantly better, while TR-BDF2 does best of all. For a fair comparison, we ran backward Euler and BDF2 with half the time step used for TR-BDF2, since the latter must solve two optimization problems per time step. From here on, all further examples are simulated only with TR-BDF2.

We evaluate the same example with damping forces added in Figure 7. As shown, Rayleigh damping quickly slows down the rotation of the bar in a non-physical manner. Applying our angular momentum correction dramatically improves its behavior.

Finally, we validate the second-order accuracy of our method in Figure 8. We performed two tests using strain-rate proportional damping and Rayleigh damping, and compared our results against a reference solution computed using the 4th-order Runge-Kutta scheme with $\Delta t = 10^{-6}$ s. TR-BDF2 with our correction term described in Section 4.2 achieves second-order convergence in both tests, while the unmodified method is only first-order accurate in the dissipative forces. However, if the error term $\nabla_1 R$ is negligible, as in the case of strain-rate proportional damping, the method still

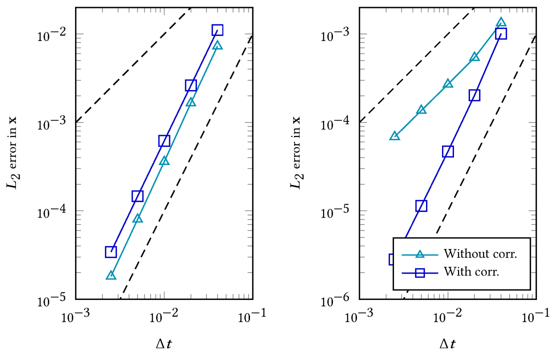

Fig. 8. Validation of the second-order accuracy of our method on (left) a stretching beam with strain-rate damping, and (right) an oscillating tetrahedron with Rayleigh damping. In both cases, the final state is compared to a reference solution obtained using RK4 with $\Delta t = 10^{-6}$ s. Dashed lines indicate first-order and second-order slopes.

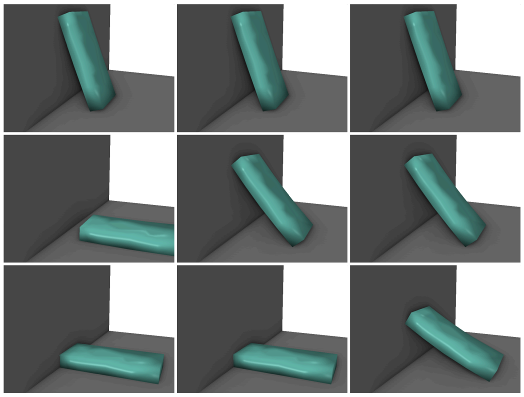

Fig. 9. An elastic block leaning against a wall due to static friction. Initial angles: 30°, 50°, 70° (bottom, middle, top rows). Friction coefficients: 0.2, 0.4, 0.6 (left, middle, right columns). Theory predicts the block to stay up only for the cases in the diagonal and upper triangle of this table.

exhibits second-order convergence without the correction term. In this case, the error is in fact lower than with the correction.

In the supplementary video, we show a comparison with a traditional implicit solver applied to TR-BDF2, which directly solves the discretized equations of motion (7) using Newton's method. The difference in error between the original integrator and our optimization-based formulation is visually unnoticeable.

We validate our Coulomb friction model using two tests for which analytical predictions are available. First, we tested dynamic friction by simulating an elastic block sliding across a flat plane with (Fig. 4). The expected horizontal deceleration is $-\mu g$, where $g = 9.81$ m/s^2 is the acceleration due to gravity, and μ was chosen to be 0.2. The resulting motion of the block closely matches the theoretical prediction, apart from some elastic oscillations once it comes to rest.

We exercise static friction more heavily in the following example. A stiff elastic block with friction coefficient μ is initialized leaning at an angle θ to the vertical against a static wall. Theoretically, a rigid rod in such a configuration will remain at rest when $\tan\theta < (1-\mu^2)/(2\mu)$, and will slide down otherwise. As shown in Figure 9, the results of our method agree with the theoretical predictions.

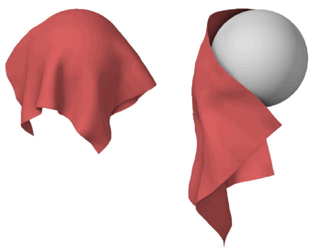

Fig. 10. A sheet of cloth draped on a sphere, showing frictional contact. The cloth stays on the sphere due to friction (left), but falls without it (right).

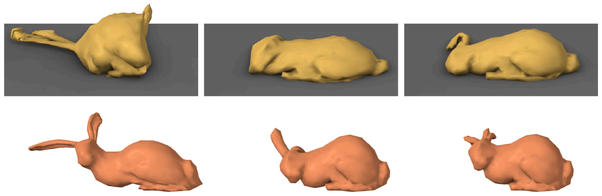

Fig. 11. An elastic object with power-law damping. Top row: Soft material model with (from left to right) no damping, $p = 0$ nonlinear damping, and $p = 1$ linear damping. Bottom row: Stiffer material with (from left to right) no damping, $p = 2$ nonlinear damping, and $p = 1$ linear damping.

7.2 Other examples

In Figure 1, we showed an example of a neo-Hookean elastic object sliding down a complex terrain, with strain-rate proportional damping and Coulomb friction for contact with the terrain. More examples of frictional behavior are shown in Figures 5 and 10.

In Figure 5 we show a spinning elastic ball being dropped onto a flat surface. Due to friction, the ball picks up horizontal velocity while in contact with the surface and subsequently rolls off to the side.

In Figure 10, a sheet of cloth with strain-limited linear elasticity for in-plane forces and the quadratic bending model [Bergou et al. 2006] for bending forces is dropped on a static sphere. In the absence of friction, the cloth easily slides off the sphere, but with friction, it stays on and creates realistic wrinkles as it comes to rest. Since the collision tests in our implementation are volumetric, we do not yet handle cloth self-collisions.

The qualitative effects of power-law damping are demonstrated in Figure 11. First, a highly compliant bunny is dropped onto a flat surface. The $p = 0$ case yields behavior similar to soft clay, permitting rapid deformation but ultimately bringing the object completely to rest in a deformed state. Linear damping with $p = 1$ cannot reproduce this unique behavior, and the motion decays only gradually as the object settles. Second, an elastic bunny is shaken violently via kinematic pins on its bottom surface. Power-law damping with $p = 2$ prevents excessive shearing and stretching of the object due to these rapid motions, while offering little resistance

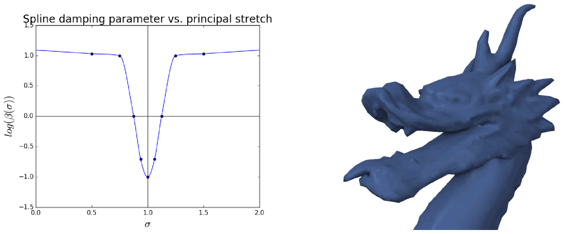

Fig. 12. Artistic control of dissipation with spline damping. A spline is designed to strongly damp velocities in the deformed state, and only minimally damp when near the rest configuration. This is used to simulate nonlinear effects in an elastic dragon model.

to the swinging of the ears and other small oscillations persisting afterwards. In the linear case, the parameter β must be chosen very large in order to suppress the deformations to a similar extent, causing the rest of the oscillations to also be heavily damped.

Finally, in Figure 12 we show an example inspired by [Xu and Barbič 2017], in which nonlinear damping is applied to a dragon model to better match the desired relaxation behavior. While they apply a spline damping curve to modal coefficients of a reduced model, we apply a spline directly to the principal strains of each element to control the element's damping forces, as described in Section 5.2.2. Spline-based damping allows the user to design arbitrary nonlinear damping behaviors that cannot be achieved simply by changing the Rayleigh damping coefficient.

Performance. Performance information for our examples is presented in Table 2. This data was collected using an Intel i9-7980XE CPU. In our software we took advantage of parallel structure to accelerate the local step of the algorithm. However, as a whole our implementation has not yet been well optimized. We believe that a more efficient implementation could make the performance of our method competitive with other approaches.

8 LIMITATIONS AND FUTURE WORK

Although TR-BDF2 conserves energy and angular momentum far better than backward Euler, it has a smaller region of stability. As a consequence, in some cases when especially stiff hyperelastic materials are used, it may be necessary to either perform more ADMM iterations or lower the time step. As is the case with other local-global optimization-based methods [Overby et al. 2017], it can be challenging to determine the suitable values for the weights. If weights are chosen poorly convergence may be greatly worsened. This issue is exacerbated in our context, since weights have to be chosen carefully for both conservative and dissipative forces. In the future, we would like to determine a more robust method for estimating good weight values. With a better heuristic, our method would require less tuning and convergence could be improved.

Our correction term cancels out the second-order error term caused by the dependence of R on $\mathbf{x}$, but it does introduce some error of its own. In practice, if the dissipative forces are only weakly dependent on $\mathbf{x}$, the net effect of the correction term can be harmful to the total error for large time steps. In the future, we would like to remove this drawback, as it would render the second-order correction a more practically useful technique.

In our method, we represent collisions as inequality constraints, modeled via infinite barrier potentials in the optimization. The drawback of this model is that we have little control over the restitution behavior of the collisions. While backward Euler yields purely inelastic collisions, we have observed that TR-BDF2 causes collisions to bounce, even when undesired like in a cloth-solid collision.

Our implementation currently only handles frictional contact with static obstacles, but does not handle frictional self-contact and multi-body contact. We believe that these can be included via non-penetration penalties using a Gauss-Seidel solver [Fratarcangeli et al. 2016; Overby et al. 2017]. However, this requires additional tuning of the penalty strength and associated ADMM weights, and we were not able to obtain successful results by the time of publication. Once these issues are resolved, it remains to be seen how our optimization-based formulation for implicit frictional contact compares in practice to existing approaches for this problem [Daviet et al. 2011; Kaufman et al. 2008; Otaduy et al. 2009]. It would be valuable to study the relative performance of these methods on challenging multi-body deformable systems, especially with multiple contacts and constraints. We hope to investigate this question in future work.

In principle, there is no theoretical requirement in our method to apply spline damping solely to individual elements. If we define the mapping $\mathbf{D}_i$ to correspond to a deformation mode in an arbitrary modal basis, the local coordinates $\underline{\mathbf{x}}_i$ and $\underline{\mathbf{v}}_i$ yield the magnitude and velocity of that mode. In this way, we could apply spline damping directly on any global deformation mode as advocated by Xu and Barbič [2017], on top of an existing full-resolution simulation. We have not tested this possibility, and we leave its investigation to future work.

There are many other damping phenomena which could be investigated in future work. Our power-law damping model with $p = 1$ yields behaviors reminiscent of the plastic deformation of soft clay, and we would like to explore extensions to general elastoplastic materials in the future. We would also like to apply ADMM to fluid simulations, inspired by projective fluids [Weiler et al. 2016]. In this context, nonlinear dissipation could model Coulomb friction in granular materials, and shear-dependent viscosity in non-Newtonian fluids like paints, gels, and foams.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments. R. Narain was supported by a Pankaj Gupta Young Faculty Fellowship. Z. Forootaninia was supported by a GAANN fellowship.

REFERENCES

R. E. Bank, W. M. Coughran, W. Fichtner, E. H. Grosse, D. J. Rose, and R. K. Smith. 1985. Transient Simulation of Silicon Devices and Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 4, 4 (October 1985), 436–451. https://doi.org/10.1109/TCAD.1985.1270142

David Baraff. 1991. Coping with Friction for Non-penetrating Rigid Body Simulation. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. ACM, New York, NY, USA, 31–41. https://doi.org/10.1145/122718.122722

Table 2. Performance data for the examples in Section 7.2. A timestep of $\Delta t = 40ms$ was used for all examples.

Example	#nodes	#energy terms	ADMM iters	Avg. global step (ms)	Avg. local step (ms)	Time per frame (ms)
Terrain (Fig. 1)	2278	16574	72	4.2	7.3	3050
Drape (Fig. 10)	1522	11657	33	13.1	1.0	1032
Ball (Fig. 5)	642	2423	35	1.4	0.6	164
Power-law (Fig. 11), $p = 0$	2435	20654	52	3.0	1.0	491
$p = 2$	2435	18926	5	133.7	9.9	1727
Dragon (Fig. 12)	5388	35878	11	13.8	77.0	2622

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 43–54. https://doi.org/10.1145/280814.280821

Jernej Barbič and Doug L. James. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Trans. Graph.* 24, 3 (July 2005), 982–990. https://doi.org/10.1145/1073204.1073300

Klaus-Jürgen Bathe. 2007. Conserving Energy and Momentum in Nonlinear Dynamics: A Simple Implicit Time Integration Scheme. *Comput. Struct.* 85, 7-8 (April 2007), 437–445. https://doi.org/10.1016/j.compstruc.2006.09.004

Klaus-Jürgen Bathe and Mirza M. Irfan Baig. 2005. On a Composite Implicit Time Integration Procedure for Nonlinear Dynamics. *Comput. Struct.* 83, 31-32 (Dec. 2005), 2513–2524. https://doi.org/10.1016/j.compstruc.2005.08.001

Jan Bender, Matthias Müller, and Miles Macklin. 2015. Position-Based Simulation Methods in Computer Graphics. In *EUROGRAPHICS 2015 Tutorials*. Eurographics Association.

Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP '06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 227–230. http://dl.acm.org/citation.cfm?id=1281957.1281987

Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. 2003. Estimating Cloth Simulation Parameters from Video. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 37–51. http://dl.acm.org/citation.cfm?id=846276.846282

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. https://doi.org/10.1145/2601097.2601116

Stephen 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, 1 (Jan. 2011), 1–122.

Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. 2018. Hyper-Reduced Projective Dynamics. 37, 4 (2018). To appear.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph.* 21, 3 (July 2002), 594–603. https://doi.org/10.1145/566654.566623

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph.* 21, 3 (July 2002), 604–611. https://doi.org/10.1145/566654.566624

S. Claici, M. Bessmeltsev, S. Schaefer, and J. Solomon. 2017. Isometry-Aware Preconditioning for Mesh Parameterization. *Comput. Graph. Forum* 36, 5 (Aug. 2017), 37–47. https://doi.org/10.1111/cgf.13243

Gilles Daviet and Florence Bertails-Descoubes. 2016. A Semi-implicit Material Point Method for the Continuum Simulation of Granular Materials. *ACM Trans. Graph.* 35, 4, Article 102 (July 2016), 13 pages. https://doi.org/10.1145/2897824.2925877

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. In *Proceedings of the 2011 SIGGRAPH Asia Conference (SA '11)*. ACM, New York, NY, USA, Article 139, 12 pages. https://doi.org/10.1145/2024156.2024173

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 (SIGGRAPH '01)*. ACM, New York, NY, USA, 31–36. https://doi.org/10.1145/383259.383262

Dimitar Dinev, Tiantian Liu, and Ladislav Kavan. 2018a. Stabilizing Integrators for Real-Time Physics. *ACM Transactions on Graphics* 37, 1 (2018), to appear.

Dimitar Dinev, Tiantian Liu, Jing Li, Bernhard Thomaszewski, and Ladislav Kavan. 2018b. FEPR: Fast Energy Projection for Real-Time Simulation of Deformable Objects. 37, 4 (2018). To appear.

Christian Duriez, Frederic Dubois, Abderrahmane Kheddar, and Claude Andriot. 2006. Realistic Haptic Rendering of Interacting Deformable Objects in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 12, 1 (Jan. 2006), 36–47. https://doi.org/10.1109/TVCG.2006.13

Elliot English and Robert Bridson. 2008. Animating Developable Surfaces Using Nonconforming Elements. *ACM Trans. Graph.* 27, 3, Article 66 (Aug. 2008), 5 pages. https://doi.org/10.1145/1360612.1360665

Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-seidel Method for Stable Soft Body Dynamics. *ACM Trans. Graph.* 35, 6, Article 214 (Nov. 2016), 9 pages. https://doi.org/10.1145/2980179.2982437

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 (Oct. 2015), 1103–1115. https://doi.org/10.1109/TVCG.2015.2459687

Herbert Goldstein, Charles P. Poole, and John L. Safko. 2002. *Classical Mechanics* (3rd ed.). Addison-Wesley.

M.E. Hosea and L.F. Shampine. 1996. Analysis and implementation of TR-BDF2. *Applied Numerical Mathematics* 20, 1 (1996), 21 – 37. https://doi.org/10.1016/0168-9274(95)00115-8 Method of Lines for Time-Dependent Problems.

Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. Graph.* 36, 4, Article 152 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073623

Ioannis Karamouzas, Nick Sohre, Rahul Narain, and Stephen J. Guy. 2017. Implicit Crowds: Optimization Integrator for Robust Crowd Simulation. *ACM Trans. Graph.* 36, 4, Article 136 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073705

Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Trans. Graph.* 27, 5, Article 164 (Dec. 2008), 11 pages. https://doi.org/10.1145/1409060.1409117

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 (SCA '06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 43–51. http://dl.acm.org/citation.cfm?id=1218064.1218071

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-prager Elastoplasticity for Sand Animation. *ACM Trans. Graph.* 35, 4, Article 103 (July 2016), 12 pages. https://doi.org/10.1145/2897824.2925906

Martin Komaritzan and Mario Botsch. 2018. Projective Skinning. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games.*

Shahar Z. Kovalsky, Meirav Galun, and Yaron Lipman. 2016. Accelerated Quadratic Proxy for Geometric Optimization. *ACM Trans. Graph.* 35, 4, Article 134 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925920

Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-spring Systems. *ACM Trans. Graph.* 32, 6, Article 214 (Nov. 2013), 7 pages.

Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2016. Towards Real-time Simulation of Hyperelastic Materials. *arXiv preprint arXiv:1604.07378* (2016).

Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: Position-based Simulation of Compliant Constrained Dynamics. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. ACM, New York, NY, USA, 49–54. https://doi.org/10.1145/2994258.2994272

Jerrold E. Marsden and Tudor Ratiu. 1999. *Introduction to Mechanics and Symmetry.* Springer.

Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based Elastic Materials. *ACM Trans. Graph.* 30, 4, Article 72 (July 2011), 8 pages.

Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C. Schvartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A. Otaduy. 2013. Modeling and Estimation of Internal Friction in Cloth. *ACM Trans. Graph.* 32, 6, Article 212 (Nov. 2013), 10 pages. https://doi.org/10.1145/2508363.

2508389

David Minor. 2018. Making Space for Cloth Simulations Using Energy Minimization. In *ACM SIGGRAPH 2018 Talks (SIGGRAPH '18)*. ACM, New York, NY, USA, Article 41, 2 pages. https://doi.org/10.1145/3214745.3214764

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position Based Dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (April 2007), 109–118. https://doi.org/10.1016/j.jvcir.2007.01.005

Rahul Narain, Abhinav Golas, and Ming C. Lin. 2010. Free-flowing Granular Materials with Two-way Solid Coupling. *ACM Trans. Graph.* 29, 6, Article 173 (Dec. 2010), 10 pages. https://doi.org/10.1145/1882261.1866195

James F. O'Brien and Jessica K. Hodgins. 1999. Graphical Modeling and Animation of Brittle Fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 137–146. https://doi.org/10.1145/311535.311550

Miguel A. Otaduy, Takeo Igarashi, and Joseph J. LaViola, Jr. 2009. Interaction: Interfaces, Algorithms, and Applications. In *ACM SIGGRAPH 2009 Courses (SIGGRAPH '09)*. ACM, New York, NY, USA, Article 14, 66 pages. https://doi.org/10.1145/1667239.1667253

Matthew Overby, George E. Brown, Jie Li, and Rahul Narain. 2017. ADMM ⊇ Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Trans. Vis. Comput. Graph.* 23, 10 (2017), 2222–2234.

Simon Pabst, Bernhard Thomaszewski, and Wolfgang Straßer. 2009. Anisotropic Friction for Deformable Surfaces and Solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09)*. ACM, New York, NY, USA, 149–154. https://doi.org/10.1145/1599470.1599490

Yue Peng, Bailin Deng, Juyong Zhang, Fanyu Geng, Wenjie Qin, and Ligang Liu. 2018. Anderson Acceleration for Geometry Optimization and Physics Simulation. *ACM Trans. Graph.* 37, 4, Article 42 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201290

John C. Platt and Alan H. Barr. 1988. Constraints Methods for Flexible Models. *SIGGRAPH Comput. Graph.* 22, 4 (June 1988), 279–288. https://doi.org/10.1145/378456.378524

Xavier Provot. 1997. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation (Eurographics 1997)*. Springer, Vienna, 177–189.

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 2, Article 37a (April 2017). https://doi.org/10.1145/2983621

Zhimin Ren, Hengchin Yeh, and Ming C. Lin. 2013. Example-guided Physically Based Modal Sound Synthesis. *ACM Trans. Graph.* 32, 1, Article 1 (Feb. 2013), 16 pages. https://doi.org/10.1145/2421636.2421637

R. Tyrrell Rockafellar. 1970. *Convex Analysis*. Princeton University Press.

Jun Saito and Simon Yuen. 2017. Efficient and Robust Skin Slide Simulation. In *Proceedings of the ACM SIGGRAPH Digital Production Symposium (DigiPro '17)*. ACM, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/3105692.3105701

Rosa M. Sánchez-Banderas and Miguel A. Otaduy. 2017. Dissipation Potentials for Yarn-Level Cloth. http://gmrv.es/Publications/2017/SO17

Ruediger Schmedding, Marc Gissler, and Matthias Teschner. 2009. Optimized Damping for Dynamic Simulations. In *Proceedings of the 25th Spring Conference on Computer Graphics (SCCG '09)*. ACM, New York, NY, USA, 189–196. https://doi.org/10.1145/1980462.1980499

Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Geometric Optimization via Composite Majorization. *ACM Trans. Graph.* 36, 4, Article 38 (July 2017), 11 pages. https://doi.org/10.1145/3072959.3073618

Eftychios Sifakis and Jernej Barbič. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses (SIGGRAPH '12)*. ACM, New York, NY, USA, Article 20, 50 pages. https://doi.org/10.1145/2343483.2343501

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. ACM, New York, NY, USA, 205–214. https://doi.org/10.1145/37401.37427

Demetri Terzopoulos and Andrew Witkin. 1988. Physically Based Models with Rigid and Deformable Components. *IEEE Comput. Graph. Appl.* 8, 6 (Nov. 1988), 41–51. https://doi.org/10.1109/38.20317

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation Capture and Modeling of Soft Objects. *ACM Trans. Graph.* 34, 4, Article 94 (July 2015), 12 pages. https://doi.org/10.1145/2766911

Marcel Weiler, Dan Koschier, and Jan Bender. 2016. Projective Fluids. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. ACM, New York, NY, USA, 79–84. https://doi.org/10.1145/2994258.2994282

Hongyi Xu and Jernej Barbič. 2017. Example-based Damping Design. *ACM Trans. Graph.* 36, 4, Article 53 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073631

Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbič. 2015. Nonlinear Material Design Using Principal Stretches. *ACM Trans. Graph.* 34, 4, Article 75 (July 2015), 11 pages. https://doi.org/10.1145/2766917

Yufeng Zhu, Robert Bridson, and Danny M. Kaufman. 2018. Blended Cured Quasi-newton for Distortion Optimization. *ACM Trans. Graph.* 37, 4, Article 40 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201359

A LOW-DIMENSIONAL OPTIMIZATION FOR STRAIN-RATE PROPORTIONAL DAMPING

Let the polar decomposition of the deformation gradient be $\mathbf{F} = \mathbf{RS}$, where $\mathbf{R}$ is a matrix with orthonormal columns, and $\mathbf{S}$ is a symmetric matrix called the stretch tensor. We begin by noting that the strain rate $\dot{\mathbf{E}}$ depends only on $\mathbf{S}$ and its time derivative $\dot{\mathbf{S}}$, since

$$\mathbf{E} = \frac{1}{2}(\mathbf{S}^2 - \mathbf{I}), \tag{71}$$

$$\dot{\mathbf{E}} = \frac{1}{2}(\mathbf{S}\dot{\mathbf{S}} + \dot{\mathbf{S}}\mathbf{S}). \tag{72}$$

We apply the discretized time derivative operator to $\mathbf{S}$, defining

$$\dot{\mathbf{S}} \approx \delta(\mathbf{S}) := \frac{1}{\alpha \Delta t}(\mathbf{S} - \mathbf{S}_p) \tag{73}$$

where $\mathbf{S}_p$ is obtained from the polar decomposition of $\mathbf{F}^p$ arising from the time integration rule (6). Then, the strain-rate proportional damping term can be viewed as a function $\Phi(\mathbf{S})$ independent of $\mathbf{R}$.

Plugging this into the local optimization step (40), and substituting $\mathbf{F} = \mathbf{RS}$, we obtain the problem

$$\underset{\substack{\mathbf{R}\text{ ortho.,}\\ \mathbf{S}\text{ sym.}}}{\arg\min}\, \Phi(\mathbf{S}) + \frac{w^2}{2}\|\mathbf{RS} - \tilde{\mathbf{F}}\|^2, \tag{74}$$

where $\tilde{\mathbf{F}}$ is the matrix whose vectorization is $\mathbf{D}_i\mathbf{x}^{(k+1)} + \bar{\mathbf{u}}^{(k)}$. We solve this by alternately minimizing over $\mathbf{R}$ and $\mathbf{S}$. Minimization over $\mathbf{R}$ is an orthogonal Procrustes problem, $\arg\min_{\mathbf{R}} \|\mathbf{RS} - \tilde{\mathbf{F}}\|^2$, for which the minimizer is the rotational part in the polar decomposition of $\tilde{\mathbf{F}}\mathbf{S}$. Minimization over $\mathbf{S}$ is a 6-dimensional optimization which we perform using L-BFGS. Typically, few iterations of this alternating are necessary, and the total cost is cheaper than a full 9-dimensional optimization over $\mathbf{F}$.

B FAST UPDATES FOR POWER-LAW DAMPING

We take a simple power-law damping model,

$$R(\dot{\mathbf{S}}) = \beta\|\dot{\mathbf{S}}\|^{p+1} = \beta(\|\dot{\mathbf{S}}\|^2)^{(p+1)/2} \tag{75}$$

for some exponent $p \geq 1$. Note that this dissipation function is rigid-invariant and so conserves linear and angular momentum for any p. The local optimization problem is then

$$\underset{\substack{\mathbf{R}\text{ ortho.}\\ \mathbf{S}\text{ sym.}}}{\arg\min}\, \beta(\|\tfrac{1}{\alpha\Delta t}(\mathbf{S} - \mathbf{S}_p)\|^2)^{(p+1)/2} + \frac{w^2}{2}\|\mathbf{RS} - \tilde{\mathbf{F}}\|^2. \tag{76}$$

Taking the differential with respect to $\mathbf{S}$ and simplifying, we obtain

$$\left(\underbrace{\frac{\beta(p+1)}{(\alpha\Delta t)^{p+1}}\|\mathbf{S} - \mathbf{S}^p\|^{p-1}}_{c(\mathbf{S})}(\mathbf{S} - \mathbf{S}^p) + w^2(\mathbf{S} - \mathbf{R}^T\tilde{\mathbf{F}})\right) : \delta\mathbf{S} = 0. \tag{77}$$

Since $\delta\mathbf{S}$ is an arbitrary symmetric matrix, the symmetric part of $c(\mathbf{S})(\mathbf{S} - \mathbf{S}^p) + w^2(\mathbf{S} - \mathbf{R}^T\tilde{\mathbf{F}})$ must be zero, i.e.

$$c(\mathbf{S})(\mathbf{S} - \mathbf{S}^p) + w^2(\mathbf{S} - \operatorname{sym}(\mathbf{R}^T\tilde{\mathbf{F}})) = 0 \tag{78}$$

where $\operatorname{sym} \mathbf{A} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$. Denote $\hat{\mathbf{S}} = \operatorname{sym}(\mathbf{R}^T \tilde{\mathbf{F}})$. Equation (78) implies that $\mathbf{S} - \mathbf{S}^p$ and $\mathbf{S} - \hat{\mathbf{S}}$ are linearly dependent, so $\mathbf{S}$ must lie on the line segment between $\mathbf{S}^p$ and $\hat{\mathbf{S}}$, and can be found via bisection or Newton's method in one dimension.