Isotropic ARAP energy using Cauchy-Green invariants

HUANCHENG LIN and FLOYD M. CHITALU, TransGP, The University of Hong Kong
TAKU KOMURA, The University of Hong Kong

Fig. 1. *Our formulation is combined with a projected Newton solver to interpolate from a Vulcan salute (left) to an OK gesture (right). We can also compute effects like surface parameterizations, elastic deformations and more using our algorithm. These are achieved by deriving an novel expression of the ARAP/corotational energy, which is written in terms of invariants of the Cauchy-Green tensor and thus free of numerically factored rotations and their gradients.*

Isotropic As-Rigid-As-Possible (ARAP) energy has been popular for shape editing, mesh parametrisation and soft-body simulation for almost two decades. However, a formulation using Cauchy-Green (CG) invariants has always been unclear, due to a rotation-polluted trace term that cannot be directly expressed using these invariants. We show how this incongruent trace term can be understood via an implicit relationship to the CG invariants. Our analysis reveals this relationship to be a polynomial where the roots equate to the trace term, and where the derivatives also give rise to closed-form expressions of the Hessian to guarantee positive semi-definiteness for a fast and concise Newton-type implicit time integration. A consequence of this analysis is a novel analytical formulation to compute rotations and singular values of deformation-gradient tensors without explicit/numerical factorization which is significant, resulting in up-to 3.5× speedup and benefits energy function evaluation for reducing solver time. We validate our energy formulation by experiments and comparison, demonstrating that our resulting eigendecomposition using the CG invariants is equivalent to existing ARAP formulations. We thus reveal isotropic ARAP energy to be a member of the "Cauchy-Green club", meaning that it can indeed be defined using CG invariants and therefore that the closed-form expressions of the resulting Hessian are shared with other energies written in their terms.

CCS Concepts: • **Computing methodologies → Mesh geometry models**.

Authors' addresses: Huancheng Lin and Floyd M. Chitalu, lamws@connect.hku.hk, chitalu@hku.hk, TransGP, The University of Hong Kong; Taku Komura, taku@cs.hku.hk, The University of Hong Kong.

0730-0301/2022/12-ART275
https://doi.org/10.1145/3550454.3555507

Additional Key Words and Phrases: Second-order methods, geometry optimization, ARAP, Cauchy-Green invariants, corotational methods

ACM Reference Format:
Huancheng Lin and Floyd M. Chitalu and Taku Komura. 2022. Isotropic ARAP energy using Cauchy-Green invariants . *ACM Trans. Graph.* 41, 6, Article 275 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555507

1 INTRODUCTION

Isotropic ARAP, which is synonymous with the popular *corotational* model[1] [Chao et al. 2010; Hsieh 1977; McAdams et al. 2011; Rankin and Brogan 1986; Stomakhin et al. 2012], has been widely used in graphics literature for two decades. First appearing in the early 2000s by the name "stiffness warping" in [Müller et al. 2002] (see also [Etzmuß et al. 2003; Irving et al. 2004]), it has been one of the simplest yet key material models for soft-tissue deformation in character animation [Kugelstadt et al. 2018; McAdams et al. 2011], geometry processing [Sorkine and Alexa 2007] as well as computer vision [Myronenko and Song 2009; Sarabandi et al. 2020].

Despite the popularity and indispensable use of this model, challenges exists around some of its definition. The most common isotropic hyperelastic energies for solid mechanics simulation in computer graphics can be expressed with either tensors or Cauchy-Green (CG) invariants. However, the ARAP energy resists being written in terms of the latter. The energy is at most quadratic which is appealing for its simplicity but a rewriting using CG invariants is unknown due to a trace term containing a rotation from factorization. Moreover, this *numerical* factorization restricts proper

[1] We use the term "corotational" in reference to the class of elasticity models that involve explicit factoring out of rotations in some form.

analytic treatment of corotational energies and their derivatives as simple closed-form expressions since the factors–*e.g.* rotation(s) and stretch–are given by operations like polar decomposition (PD) and singular value decomposition (SVD). These decompositions have algebraic form (which is even imbued with intuitive geometric interpretation) but there exists no direct closed-form expressions to evaluate the constituting factors for non-symmetric inputs. Thus, iterative procedures are necessary to obtain these factors, which then restricts analytic derivations of energies like ARAP since derivatives (and integrals) of numerically obtained quantities do not have proper closed-form expressions. Thus, unlike the Dirichlet [Hélein and Wood 2008], Neo-Hookean [Bonet and Wood 2008] and St. Venant-Kirchhoff [Picinbono et al. 2000] energies, ARAP remains the only model within this group whose formulation has thus far seemed inexpressible in terms of CG invariants, which prohibits a generic (i.e. complete *rogues' gallery*) formulation of gradients and Hessians that is shared by these energies for Newton-type implicit time integration.

In this paper, we show that the isotropic ARAP model can be understood entirely in terms of CG invariants. The first step in such an analysis is usually to re-write the model as an isotropic, hyperelastic strain energy [Bonet and Wood 2008]. However, it has never been clear how to do this for the ARAP model. After first showing that simply rewriting the model as an isotropic, hyperelastic strain energy is impossible due to the unwieldy trace term, we demonstrate how this trace term can be both formulated and analysed using a polynomial with coefficients expressed in terms of the invariants.

Using this energy, we establish that our formulation resembles both Smith et al. [2018, 2019] (which has a similar structure) and the tensor-based ARAP energy. We show that the original problematic trace term can be understood as a root of a quartic equation in 3D (or quadratic in 2D, and in 1D) incorporating all three CG invariants to allow for a proper closed-form expression of the energy, its gradient, and Hessian that is free of rotation gradients obtained via numerical factorisation. Moreover, since this root is in closed-form the first derivative is an expression evaluating to the–previously explicitly factored–rotation which we can now compute *without* resort to any approximate/iterative procedures. Using our analysis, we also show how the singular values of deformation gradient tensors can be determined with relative ease using simple scalar expressions in terms of our roots, which is novel and significant.

Finally, we characterize the exact conditions under which this energy produces locally indefinite systems. Smith et al. [2019] characterise this indefiniteness by advocating for a new set of invariants, departing from the normative path of using CG invariants. Moreover, their first and third invariants are sign preserving, to indicate element inversion, which then permits a rewriting and analysis of energies like ARAP but with the expense of a–rotation-variant–PD [Sorkine-Hornung and Rabinovich 2017]. We revisit eigenanalysis to quantitatively establish the exact conditions under which indefiniteness occurs using well established derivations of the gradients and Hessians of the CG invariants [Bonet and Wood 2008]. Our closed-form, analytic expressions for the eigenvalues can also work in tandem with Smith et al. [2019]'s analysis to enable a fast, simple, and analytic method for projecting the Hessian back to positive semi-definiteness. The efficacy of our formulation is demonstrated in a variety of scenarios such as elastic deformation and surface parameterization (see Fig. 1, § 7 and supplementary material).

A summary of our contributions is as follows:

- An isotropic ARAP energy using Cauchy-Green invariants.
- A complete eigenanalysis of the energy.
- A simple closed-form description of singular values and rotation factors of the deformation gradient.
- An analysis showing that the energy is equivalent to existing formulations of isotropic ARAP energy.
- A fast, analytic positive semi-definiteness projection method for the Hessian.

2 RELATED WORK

The original corotational formulation [Rankin and Brogan 1986] (colloquially known as ARAP [Sorkine and Alexa 2007][2]) is widely used for many reasons, including its simplicity, and its popularization of arbitrarily large-rotation deformation formulations. In this work, we will focus on the constitutive model it uses for *stretching*.

Many related models have been used in the past, such as stiffness warping [Etzmuß et al. 2003], St. Venant Kirchhoff (St-VK) [Picinbono et al. 2000], and the corotational method [Chao et al. 2010; McAdams et al. 2011; Müller et al. 2002]. For isotropic materials it is possible and often expedient to formulate the constitutive equations using CG invariants, which contain all the information needed to evaluate the stored elastic energy function [Bonet and Wood 2008; Ogden 1997]. An overview, in the context of elastic deformation simulation, between FEM-based models is also available [Kim and Eberle 2020; Sifakis and Barbic 2012; Smith et al. 2018, 2019]. Of all these works, the eigenanalysis of Smith et al. [2019] using isotropic invariants is most similar to the results we present here, though our analysis maintains a CG invariant based description, where we now expand the space of expressible energies using these invariants to include inversion-safe ARAP [Sorkine and Alexa 2007] and thus corotational elasticity models like those by McAdams et al. [2011] and Stomakhin et al. [2012].

The isotropic ARAP model has also been commonly presented as either a tensor- or singular-value based formulation rather than as a CG invariant based model. The reason is that CG invariants cannot directly express linear distortion terms that appear in many popular models, like the corotational energy [McAdams et al. 2011]. This is the primary issue behind the inability to re-write ARAP using these invariants (see also Stomakhin et al. [2012]; Xu et al. [2015] for a further discussion). Smith et al. [2019] address this problem by introducing a new set of invariants that require a polar decomposition of the deformation gradient (see also [Kim and Eberle 2020]). We revisit this problem to certify ARAP as a model expressible in terms of the CG invariants and show that its isotropic formulation using these invariants for expressing linear distortion terms is possible.

The tensor based formulation too (*cf.* Eq. (1)) presents theoretical and practical challenges when deriving its Hessian, requiring rotational gradients from numerical factorisation. One solution has been to use auto-differentiation [Gao et al. 2009] but the resulting

[2]Note that Sorkine and Alexa [2007]'s approach uses a one-ring/edge-stencil description which shares a similar energy definition and optimization technique as other methods, but does not equate to an elastic membrane energy [Panetta 2019].

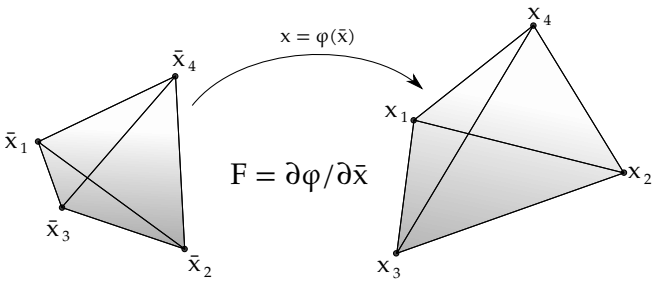


Fig. 2. *A concise description of the deformation that an elastic body has sustained. Here is show a single finite element with reference* $\bar{\mathbf{x}}$ *and spatial configurations* $\mathbf{x}$*, which are related via a deformation map* $\phi = \mathbf{F}\bar{\mathbf{x}} + \mathbf{t}$*, representing a local affine transformation. An important physical quantity derived directly from this map is the deformation gradient* $\mathbf{F}$ *from which one can measure stress resulting from the deformation.*

implementation is complex. Müller et al. [2002] and Irving et al. [2004] do not address this issue while others [Barbič 2012; Chao et al. 2010; McAdams et al. 2011; Wang et al. 2021] devise a variety of *specially-tailored* techniques of varying complexity for dealing with it. We achieve expressions for the isotropic ARAP energy's Hessian that surpass explicit need to compute rotation gradients as in previous methods.

Rotational alignment problems. Numerous other problems within computer graphics, vision, robotics and simulation involve determining the rotation aligning two objects. SVD is the standard solution, with rotation recovered as a product of numerically-determined singular vectors (see *e.g.* Irving et al. [2004]; Myronenko and Song [2009]; Twigg and Kacic-Alesic [2010] and Sorkine-Hornung and Rabinovich [2017]). Computation of only the rotation (*i.e.* skipping the diagonal matrix of singular values) is possible with parameterizations of the rotations using exponential maps, Caley maps, quaternions or the Rodrigues rotation formula with optimization. Examples of these numerical procedures include Byers and Xu [2008]; Kugelstadt et al. [2018]; Müller et al. [2016]; Panetta [2018, 2019]; Wu et al. [2018] and most recently Zhang et al. [2021], which are all iterative methods. We provide a closed-form expression for the sought rotation and thereby its derivatives.

3 THE COROTATIONAL ENERGIES

We will begin with a brief overview of the original (i.e. tensor-based) corotational formulation, following the notation of the original [Rankin and Brogan 1986] as closely as possible. Once the preliminaries are established, we will show how to cast these into a CG invariant based formulation.

3.1 The Original Formulation

The ARAP method uses the Frobenius norm $||\cdot||_{\mathrm{F}}^2$ to measure the local strain energy density per finite element of elastic material (*cf.* Fig. 2). Using PD, this strain energy quantifies stretching which is measured as the difference between a deformation gradient $\mathbf{F}$ ($\in \mathbb{R}^{3\times3}$ in 3D volume elements; $\in \mathbb{R}^{3\times2}$ in 2D membrane elements; and $\in \mathbb{R}^{3\times1}$ in 1D strand elements) and a rigid body rotation $\mathbf{R}$ ($\mathbf{F} = \mathbf{RS}$ via PD) to give

$$\Psi = \frac{\mu}{2}||\mathbf{F}-\mathbf{R}||_{\mathrm{F}}^2, \tag{1}$$

$$= \frac{\mu}{2}\left(||\mathbf{F}||_{\mathrm{F}}^2 + ||\mathbf{R}||_{\mathrm{F}}^2 - 2\mathrm{tr}(\mathbf{F}^\intercal\mathbf{R})\right), \tag{2}$$

$$\equiv \frac{\mu}{2}\left(||\mathbf{F}||_{\mathrm{F}}^2 + 3 - 2\mathrm{tr}(\mathbf{S})\right), \tag{3}$$

$$\equiv \frac{\mu}{2}\left(||\mathbf{S}||_{\mathrm{F}}^2 + 3 - 2\mathrm{tr}(\mathbf{S})\right), \tag{4}$$

which is a scalar function that is simplified by using the Frobenius identity

$$||\mathbf{A}-\mathbf{B}||_{\mathrm{F}}^2 = ||\mathbf{A}||_{\mathrm{F}}^2 + ||\mathbf{B}||_{\mathrm{F}}^2 - 2\mathrm{tr}(\mathbf{A}^\intercal\mathbf{B}), \tag{5}$$

and where Eq. (4) results from the rotation invariance of the norm $||\cdot||_{\mathrm{F}}^2$. The Lamé constant $\mu = E/2(1+\nu)$ is the shear modulus, which is a material parameter controlling resistance to stretch: the variable E is Young's modulus and ν is Poisson's ratio. The appealing nature of Eq. (1) comes from the fact that it is quadratic (*i.e.* in the entries of $\mathbf{F}$), which is appealingly less non-linear than *e.g.* the St-VK model. In essence, the energy is minimized by the finding rotation $\mathbf{R}$ maximising the trace (*cf.* Eq. (2)).

The 1st Piola-Kirchhoff stress (PK1) is then,

$$\begin{aligned}\mathbf{P}(\mathbf{F}) \equiv \frac{\partial\Psi}{\partial\mathbf{F}} &= \frac{\partial\left(\frac{\mu}{2}\left(||\mathbf{F}||_{\mathrm{F}}^2 + ||\mathbf{R}||_{\mathrm{F}}^2 - 2\mathrm{tr}(\mathbf{F}^\intercal\mathbf{R})\right)\right)}{\partial\mathbf{F}},\\ &= \frac{\mu}{2}\left(\frac{\partial||\mathbf{F}||_{\mathrm{F}}^2}{\partial\mathbf{F}} + \cancelto{0}{\frac{\partial||\mathbf{R}||_{\mathrm{F}}^2}{\partial\mathbf{F}}} - 2\frac{\partial\mathrm{tr}(\mathbf{F}^\intercal\mathbf{R})}{\partial\mathbf{F}}\right),\\ &= \frac{\mu}{2}\left(2\mathbf{F} - 2\mathbf{R}\right),\\ &= \mu(\mathbf{F}-\mathbf{R}),\end{aligned} \tag{6}$$

which is simplified based on the invariance of the Frobenius norm when applied to orthonormal matrices resulting in a constant. Nodal forces are determined from Eq. (6) by $\mathbf{f} = -v\frac{\partial\Psi}{\partial\mathbf{x}}$, where

$$\frac{\partial\Psi}{\partial\mathbf{x}} = \frac{\partial\mathbf{F}}{\partial\mathbf{x}} : \frac{\partial\Psi}{\partial\mathbf{F}} \tag{7}$$

$$\equiv \mathrm{vec}\left(\frac{\partial\mathbf{F}}{\partial\mathbf{x}}\right)^\intercal \mathrm{vec}\left(\frac{\partial\Psi}{\partial\mathbf{F}}\right). \tag{8}$$

These forces are scaled by the local element volume v, where $\mathbf{x}$ represents the stacked nodal positions of the element. The equivalent notation $\mathrm{vec}\,(\cdot)$ denotes *vectorization*, which is a stacking of matrix columns in the simplest case

$$\mathbf{A} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \Leftrightarrow \mathrm{vec}\,(\mathbf{A}) = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \mathbf{a}.$$

Moreover, the change of deformation gradient w.r.t nodal positions $\partial\mathbf{F}/\partial\mathbf{x}$ ($\in \mathbb{R}^{3\times3\times12}$ for a linear tetrahedral element) is a 3rd-order tensor, where respective vectorisation will produce a 2nd-order tensor ($\in \mathbb{R}^{9\times12}$) with each resulting column representing a stacking of the corresponding columns of a block-entry ($\in \mathbb{R}^{3\times3}$). This flattening convention of vectors (and higher-order tensors) has properties that

benefit our derivations. For brevity, we refer readers to Sec.3 in Smith *et al.* [2019] (see also Kim [2020] and Golub *et al.* [2012]) for a primer on the vector, matrix and general tensor notation that we adopt throughout.

The force gradient too (*i.e.* the Hessian, or tangent stiffness matrix) is required for implicit timestepping, which is defined by $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = -v\frac{\partial^2 \Psi}{\partial \mathbf{x}^2}$, where

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{F}^\intercal}{\partial \mathbf{x}} \frac{\partial^2 \Psi}{\partial \mathbf{F}^2} \frac{\partial \mathbf{F}}{\partial \mathbf{x}}, \tag{9}$$

$$\equiv \text{vec}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^\intercal \text{vec}\left(\frac{\partial \Psi^2}{\partial \mathbf{F}^2}\right) \text{vec}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right). \tag{10}$$

Although the term $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ may be constant (*e.g.* for a tetrahedral mesh-based discretization), Eq. (9) presents an immediate obstacle if one wishes to get an analytic expression for the ARAP energy's Hessian: The 4th-order tensor

$$\frac{\partial^2 \Psi}{\partial \mathbf{F}^2} = \frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}} = \mu\left(\frac{\partial \mathbf{F}}{\partial \mathbf{F}} - \frac{\partial \mathbf{R}}{\partial \mathbf{F}}\right),$$

contains a rotation gradient $\frac{\partial \mathbf{R}}{\partial \mathbf{F}}$ which presents a challenge when deriving a general analytic description because there are no explicit entries of $\mathbf{F}$ that appear in $\mathbf{R}$ (from numerical factorisation).

3.2 No *explicit* Cauchy-Green invariant formulation exists

Alternatively, it *should* be possible to define any isotropic hyperelastic model like ARAP by using the three CG invariants [Bonet and Wood 2008]

$$I_C = \text{tr}(\mathbf{C}) \qquad II_C = ||\mathbf{C}||_\text{F}^2 \qquad III_C = \det(\mathbf{C}), \tag{11}$$

where the variable $\mathbf{C} = \mathbf{F}^\intercal\mathbf{F}$ is the right Cauchy-Green tensor, which can be computed inexpensively[3]. Using the chain rule, an energy of the form $\Psi(I_C, II_C, III_C)$ will have a PK1 stress tensor as

$$\mathbf{P}(\mathbf{F}) = \frac{\partial \Psi\left(I_C, II_C, III_C\right)}{\partial \mathbf{F}}$$

$$= \frac{\partial \Psi}{\partial I_C}\frac{\partial I_C}{\partial \mathbf{F}} + \frac{\partial \Psi}{\partial II_C}\frac{\partial II_C}{\partial \mathbf{F}} + \frac{\partial \Psi}{\partial III_C}\frac{\partial III_C}{\partial \mathbf{F}}. \tag{12}$$

The corresponding Hessian (a 4th-order tensor *e.g.* $\in \mathbb{R}^{3\times3\times3\times3}$ in 3D) is given by

$$\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}} = \sum_i \frac{\partial \Psi}{\partial i}\mathbb{H}_i + \sum_i \sum_j \frac{\partial^2 \Psi}{\partial i \partial j}\mathbb{G}_j \otimes \mathbb{G}_i \tag{13}$$

where $\mathbb{G}_i$ and $\mathbb{H}_i$ are gradient and Hessian of the i-th invariant w.r.t $\mathbf{F}$, respectively, in tensor form with $i, j \in \{I_C, II_C, III_C\}$. The symbol $\otimes$ denotes the Kronecker product operator, which is given by

$$\mathbb{A} \otimes \mathbb{B} = \begin{bmatrix} [a_{11}\mathbb{B}] & [a_{12}\mathbb{B}] & \cdots \\ [a_{21}\mathbb{B}] & [a_{22}\mathbb{B}] & \\ \vdots & & \ddots \end{bmatrix},$$

with variable a_{ij} representing a scalar entry in the respective tensor $\mathbb{A}$. We also refer the reader to Appendix A for the actual definitions of tensor variables $\mathbb{G}_i$ and $\mathbb{H}_i$ that are derived from the deformation gradient $\mathbf{F}$ and used in Eq. (13).

Eq. (13)–like the PK1 in Eq. (12)–is characterised by an expedient generic structure to make for a practically convenient mix-and-match approach to implementing isotropic energies. However, while this analytic invariant-based description can be adopted easily for energies like St-VK and Neo-Hookean, it is unclear how this applies to ARAP for which a direct rewriting of its energy shown in Eq. (2) in terms of the CG invariants (in Eq. (11)) yields

$$\Psi = I_C - 2\text{tr}\left(\mathbf{F}^\intercal\mathbf{R}\right) + d, \tag{14}$$

where d is the spatial dimension of the finite element (*e.g.* 1, 2 or 3 dimensions). In particular, the trace term (with rotation $\mathbf{R}$ from numerical factorisation) precludes analytic rewriting–and thereby a derivation of the gradient and the Hessian–of the second term using invariants. The fundamental problem is that this term is a linear expression in the singular values $\mathbf{F}$ but the CG invariants denote non-linear expressions. Smith *et al.* [2018; 2019] address this issue by introducing a new set of invariants that is applicable to general Neo-Hookean elasticity and dependant on PD. We instead seek a solution that fits within and complements the generic structure of the Hessian defined using the CG invariants as in Eq. (13) while being completely free of explicit rotation-related terms in the resulting expressions.

3.3 An *implicit* formulation exists

Having described the challenge of formulating a rewriting of the ARAP energy in terms of CG invariants, we now describe a solution that will also bypass the explicit dependence on rotation gradients that are fundamentally dependent on numerical factorisation of the deformation gradient. Our method will additionally enable a complete analytic derivation of the Hessian and allow efficient projection to positive semi-definiteness for implicit timestepping.

Following from Eq. (11), Stomakhin et al. [2012] (see also Xu et al. [2015]) showed that the CG invariants may also be written using the singular values of the deformation gradient

$$I_C = \text{tr}\left(\Sigma^2\right) = \sum_{i=1}^{d} \sigma_i^2 \tag{15}$$

$$II_C = \text{tr}\left(\Sigma^4\right) = \sum_{i=1}^{d} \sigma_i^4 \tag{16}$$

$$III_C = \det\left(\Sigma^2\right) = \prod_{i=1}^{d} \sigma_i^2, \tag{17}$$

where σ_i denotes one such value that is obtained by computing SVD with $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^\intercal$.

From this perspective, it should be clear the trace term of Eq. (14) will reduce to

$$\text{tr}\left(\mathbf{F}^\intercal\mathbf{R}\right) = \text{tr}\left(\mathbf{S}\right) = \text{tr}\left(\Sigma\right) = \sum_{i=1}^{d} \sigma_i. \tag{18}$$

Raising Eq. (18) to the forth power and simplifying after substituting the three invariant definitions from Equations 15 to 17, we obtain a polynomial whose coefficients are the CG invariants and the trace term is a root. This polynomial will be a quartic in 3D

[3] Although it is possible to define an isotropic material by a relation between Ψ and Σ *e.g.* as described by Stomakhin *et al.* [2012] where $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^\intercal$ from SVD, this too may not necessarily the preferred approach, since the overhead of an SVD computation would be necessary.

$$\tilde{\mathcal{P}}(t) = t^4 - 2I_C t^2 - 8\sqrt{III_C}t + I_C^2 - 4II_C^*, \tag{19}$$

where $II_C^* = \frac{1}{2}\left(I_C^2 - II_C\right)$. Closed-form expressions for quartic roots exist, which can also be written in terms of the invariants[4].

We rewrite the ARAP energy in Eq. (14) by first defining a function $f\colon \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ which evaluates to the trace. Moreover, we define this f to be the expression of the quartic root corresponding to the trace, thereby making it a function of the invariants. With this formulation, the ARAP energy can be rewritten as

$$\tilde{\Psi}_{\text{impl}} = I_C - 2\left(\tilde{\mathcal{P}}(t)\Big|_{t=f} = 0\right) + d, \tag{20}$$

where the problematic trace term of Eq. (14) is now replaced with the differentiable f satisfying $\tilde{\mathcal{P}}(f) = 0$. Readers are referred to our supplementary material, which provides the closed-form expressions for this f.

The second term of the energy $\tilde{\Psi}_{\text{impl}}$ in Eq. (20) is now stated *implicitly* in one of the roots of the polynomial $\tilde{\mathcal{P}}(t)$. This new form is expressed with the CG invariants which represent the terms in f, while the former energy in Eq. (14) is based solely on the singular values of $\mathbf{F}$. Closed-form derivatives can therefore be obtained easily, which can then be used to construct the analytic PK1 and Hessian. Eq. (19) unfortunately has a key drawback that we must address.

3.4 Inversion awareness

While Eq. (19) possesses the desired qualities to account for stretch, compression and even volume, it lacks *inversion awareness* which is crucial for robust simulation. In computer graphics (*e.g.* [Irving et al. 2004; Teran et al. 2005]) it is well known that finite elements might invert for a multitude of reasons, including numerical error. This predicament relates to $\tilde{\mathcal{P}}(t)$ as well since (as we will show) it is practically convenient that information indicating inversion is captured in the expression. Otherwise, the forces (*cf.* § 4) that arise from the derivatives of $\tilde{\mathcal{P}}(t) = 0$ are only valid for solutions $t \neq \mathrm{tr}\,(\mathbf{F}^\intercal\mathbf{R})$ when the element is inverted. Moreover, Eq. (18) is a solution only when the element has positive volume. This lack of inversion awareness is a known side effect of the third invariant III_C [Smith et al. 2019]. One solution is to flip the sign of the third term

$$t^4 - 2I_C t^2 + 8\sqrt{III_C}t + I_C^2 - 4II_C^*,$$

to then use this new expression in the specific case of inversion. We avoid this approach, because the ensuing analytic derivatives of Eq. (22) must be evaluated as separate routines for the normal and inverted case per element. Moreover, this increases the complexity of booking keeping during implementation and should therefore be done judiciously.

Intuitively, the problem with Eq. (19) is that the relationship of Eq. (18) to the change in element volume is expressed with the third invariant $III_C = J^2$, which discards the sign information about the volume ratio $J = \det(\mathbf{F})$. Consequently, we ask whether we can evaluate the *exact* ratio so that when the element is inverted, this information is captured in the sign of J. To answer this question, we replace $\sqrt{III_C}$ with J:

$$\mathcal{P}(t) = t^4 - 2I_C t^2 - 8Jt + I_C^2 - 4II_C^*, \tag{21}$$

with which we arrive at our implicit ARAP energy

$$\boxed{\Psi_{i\text{ARAP}} = I_C - 2\left(\underbrace{\mathcal{P}(t)|_{t=f} = 0}_{\mathcal{P}-\text{Energy}}\right) + d.} \tag{22}$$

This is strikingly similar to the Smith et al. [2019]'s ARAP formulation except that their 1st invariant $I_1 \equiv \mathrm{tr}(\Sigma)$ is now replaced with the expression for the solution $\mathcal{P}(t)|_{t=f} = 0$. The polynomial $\mathcal{P}(t)$ and its derivatives are also independent of rotations, which is implied by Eq. (18) since the problematic trace term of Eq. (14) is uniquely defined by the singular values of $\mathbf{F}$. This property follows from rotation invariance of Eq. (1)

$$\|\mathbf{F} - \mathbf{R}\|_{\mathrm{F}}^2 = \|\mathbf{R}^\intercal\mathbf{F} - \mathbf{R}^\intercal\mathbf{R}\|_{\mathrm{F}}^2 = \|\mathbf{S} - \mathbf{I}\|_{\mathrm{F}}^2,$$

resulting in the Biot strain tensor (RHS) from mechanics literature [Biot 1938] (see also Chao et al. [2010]). In essence, we have performed a rotation removal on $\mathrm{tr}\,(\mathbf{F}^\intercal\mathbf{R})$ but without PD nor SVD, which will result in analytic expressions of the PK1 and the Hessian of Eq. (22) that are free of their (numerical) rotations.

4 DERIVATIVES OF ARBITRARY $\mathcal{P}$-ENERGIES

The implicit potential $\Psi_{i\text{ARAP}}$ reveals that the variables $\partial\Psi/\partial i$ and $\partial^2\Psi/\partial i j$ in Eq. (13) (and Eq. (12)) are dependent on the 1st- and 2nd-order partial derivatives of our root function f w.r.t the invariants, which are needed to construct the PK1 and Hessian. In this light, we now show that the analytic gradient and Hessian of any energy expressed solely in terms of this f can be written down in proper closed-form.

We can solve for the partial derivatives of the root function $\partial f/\partial i$ from Eq. (21) by first noting that $\mathcal{P}(t) = 0$ holds regardless of the values of the CG invariants. Therefore, by applying the partial operator $\partial/\partial i$, we obtain

$$\frac{\partial\mathcal{P}(t)}{\partial i} = 0, \tag{23}$$

with which we analytically solve for $\partial f/\partial i$ by rearranging the terms in the left-hand-side expression:

$$\frac{\partial f}{\partial I_C} = \frac{2f^2 + 2I_C}{\alpha}, \quad \frac{\partial f}{\partial II_C} = \frac{-2}{\alpha}, \quad \frac{\partial f}{\partial J} = \frac{8f}{\alpha}, \tag{24}$$

where $\alpha = 4f^3 - 4I_C f - 8J$ and with the volume ratio J is heretofore used in-place of the third invariant III_C[5]. The corresponding 2nd-order partial derivatives are of the form

$$\frac{\partial^2 f}{\partial i \partial j} = \frac{p_{ij}}{\alpha}, \tag{25}$$

where the numerator p_{ij} represents scalar expressions that we provide in Appendix B.

[4] Refer to our supplementary material for the full derivation of Eq. (19), including the versions for one- and two-dimensional finite elements.

[5] We emphasise that our motivation for using this J is merely a practical one. It is in fact possible to use III_C but this requires a case-by-case (*i.e.* inverted *vs.* non-inverted) handling of derivatives of f, which can be avoided.

From Eq. (24) and Eq. (25), the 1st- and 2nd-order partial derivatives of the energy in Eq. (22) and w.r.t the invariants are

$$\frac{\partial \Psi}{\partial I_C} = 1 - 2\frac{\partial f}{\partial I_C}, \quad \frac{\partial \Psi}{\partial II_C} = -2\frac{\partial f}{\partial II_C}, \quad \frac{\partial \Psi}{\partial J} = -2\frac{\partial f}{\partial J}, \tag{26}$$

$$\frac{\partial^2 \Psi}{\partial ij} = -2\frac{\partial^2 f}{\partial ij}, \tag{27}$$

which we use to construct the (previously unknown) PK1 and Hessian of the problematic trace term in Eq. (14) by following Eq. (12) and Eq. (13), respectively.

The expressions in Eq. (26) and Eq. (27) are evaluated by setting t equal to the trace in Eq. (18). Moreover, the singular values of the deformation gradient $\mathbf{F}$ can be determined via simple scalar expressions of the polynomial roots, as shown in our supplementary material: For example, in 3D with $\mathbf{F} \in \mathbb{R}^{3\times3}$ we have

$$\begin{aligned} \sigma_1 &= (x_1 + x_4)/2 \\ \sigma_2 &= (x_2 + x_4)/2 \\ \sigma_3 &= (x_3 + x_4)/2, \end{aligned} \tag{28}$$

as the closed-form expressions of the singular values, where x_i is the ith root of the polynomial $\mathcal{P}(t)$ in Eq. (21). From these singular values it is also possible to directly determine the eigenvalues of the local Hessian $\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}}$ (*cf.* Eq. (32)) as shown in § 5. This novel approach to compute the singular values thus works in tandem with (and to the benefit of) existing methods.

5 EIGENANALYSIS OF THE ENERGY

Numerical minimization of $\Psi_{i\text{ARAP}}$ using 2nd-order Newton methods can be a challenge due to stalls and even divergence when the energy Hessian is indefinite. Here, we revisit the problem of ensuring positive semi-definiteness by projection, but without any iterative procedures such as SVD or PD, thanks to the new energy formulation that provides a closed-form solution for the trace-term.

5.1 Ensuring Positive Semi-Definiteness of the Hessian

To ensure positive semi-definiteness of the Hessian in Eq. (22), the eigenvalues and eigenvectors can be stated in closed-form [Kim and Eberle 2020]. Considering only the 3D case for simplicity, this analytic eigenstructure is revealed by a vectorization of each block entry of the Hessian into a single column

$$\text{vec}\left(\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}_{ij}} \in \mathbb{R}^{3\times3}\right) \Rightarrow \mathbb{R}^{9\times1}. \tag{29}$$

Thus, applying this rule for all blocks $\partial/\partial \mathbf{F}_{ij}$, one arrives at the reduced form

$$\text{vec}\left(\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}}\right) \in \mathbb{R}^{9\times9}. \tag{30}$$

Eq. (30) is evaluated by vectorizing tensor terms in Eq. (13) to give

$$\text{vec}\left(\frac{\partial \mathbf{P}(\mathbf{F})}{\partial \mathbf{F}}\right) = \sum_i \frac{\partial \Psi}{\partial i}\mathbf{H}_i + \sum_i \sum_j \frac{\partial^2 \Psi}{\partial i \partial j}\mathbf{g}_j \mathbf{g}_i^{\mathsf{T}} \tag{31}$$

where $\mathbf{g}_i = \text{vec}\left(\mathbb{G}_i\right), \mathbf{H}_i = \text{vec}\left(\mathbb{H}_i\right)$, and the overloaded symbols $i, j \in \{I_C, II_C, J\}$ refer to the invariants here.

The eigenpairs of our vectorised Hessian are determined by solving the analytic eigen problem: Eigenvalues are determined by evaluating roots of the characteristic polynomial $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$, where $\mathbf{A} \in \mathbb{R}^{9\times9}$ is our vectorised Hessian and $\mathbf{I} \in \mathbb{R}^{9\times9}$ is the identity matrix to give

$$\lambda_0 = 1 - \frac{2}{\sigma_1 + \sigma_2} \quad \lambda_1 = 1 - \frac{2}{\sigma_2 + \sigma_3} \quad \lambda_2 = 1 - \frac{2}{\sigma_1 + \sigma_3}$$
$$\lambda_{3\ldots8} = 1,$$

as our eigenvalues in 3D where $\mathbf{F} \in \mathbb{R}^{3\times3}$. We also have

$$\lambda_0 = 1 - \frac{1}{\sigma_1} \quad \lambda_1 = 1 - \frac{1}{\sigma_2} \quad \lambda_2 = 1 - \frac{2}{\sigma_1 + \sigma_2}$$
$$\lambda_{3,4,5} = 1$$

in 2D where $\mathbf{F} \in \mathbb{R}^{3\times2}$, and σ_i are the singular values of the deformation gradient $\mathbf{F}$ which can be computed directly from the roots of $\mathcal{P}(t)$ using Eq. (28). In 1D with $\mathbf{F} \in \mathbb{R}^{3\times1}$, we also have

$$\lambda_0 = 1 \quad \lambda_{1,2} = 1 - \frac{1}{\sigma_1},$$

where σ_1 corresponds to the current length of 1D strand element.

A corresponding vector $\mathbf{q}_i$ ($\in \mathbb{R}^9$ for a tetrahedral element; $\in \mathbb{R}^6$ for a triangular element; and $\in \mathbb{R}^3$ for a strand-like/2-point edge element) of each eigenvalue λ_i is obtained simply by solving $(\mathbf{A} - \lambda_i\mathbf{I})\,\mathbf{q}_i = 0$. The resemblance of our eigensystem to Smith et al. [2019] both validates our scheme and means that their corresponding eigenvector expressions (and the 2D forms by Panetta [2019]) can also be used in place of the explicit solve, which is practically expedient. We provide the analytic eigenvectors for a 1D strand energy in our supplementary material.

Constructing a Hessian that is positive semi-definite is thus reduced to applying

$$\boxed{\text{vec}\left(\frac{\partial \mathbf{P}}{\partial \mathbf{F}}\right) = \sum_{i=1}^{N} \langle \lambda_i \rangle \mathbf{q}_i \mathbf{q}_i^{\mathsf{T}},} \tag{32}$$

where $N \in \mathbb{R}^9$ in 3D ($\in \mathbb{R}^6$ in 2D; and $\in \mathbb{R}^3$ in 1D) and the operator $\langle\cdot\rangle$ denotes the Macauley brackets

$$\langle x \rangle = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}.$$

In summary, by rewriting the ARAP potential as in Eq. (22) and obtaining the closed-form expressions of any energy written solely in terms of the root $\mathcal{P}(t) = 0$, we can evaluate the PK1 and exact positive semi-definite Hessian of the ARAP energy written in terms of the CG invariants and bypass the need to evaluate rotation gradients.

5.2 Relationship to Previous ARAP Eigensystems

Using a recent analysis by Wang et al. [2021], we discuss our analytic eigensystem for the ARAP stretch term, and the relation to force filtering by Teran et al. [2005] and the analytic eigensystems approach of Smith et al. [2019] when applied to Eq. (9).

To compute the problematic rotation gradient of Eq. (9), one can start from PD $\mathbf{F} = \mathbf{RS}$ and its derivative $\frac{\partial \mathbf{F}}{\partial \mathbf{F}_{ij}} = \frac{\partial \mathbf{R}}{\partial \mathbf{F}_{ij}}\mathbf{S} + \mathbf{R}\frac{\partial \mathbf{S}}{\partial \mathbf{F}_{ij}}$ that

gives

$$\frac{\partial \mathbf{R}}{\partial \mathbf{F}_{ij}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{F}_{ij}} - \mathbf{R} \frac{\partial \mathbf{S}}{\partial \mathbf{F}_{ij}} \right) \mathbf{S}^{-1}. \tag{33}$$

Using Sylvester's equation (*cf.* §3.7 in [Wang et al. 2021]), the unknown term $\frac{\partial \mathbf{S}}{\partial \mathbf{F}_{ij}}$ in Eq. (33) may be determined by

$$\operatorname{vec}\left(\frac{\partial \mathbf{S}}{\partial \mathbf{F}_{ij}} \right) = (\mathbf{S} \oplus \mathbf{S})^{-1} \operatorname{vec}\left(\frac{\partial \mathbf{F}^\intercal \mathbf{F}}{\partial \mathbf{F}_{ij}} \right), \tag{34}$$

where $\oplus$ denotes Kronecker sum.

Thus, by using Eq. (33), one can evaluate the tensor-based Hessian in Eq. (9) and solve a similar eigen problem as Teran et al. [2005]. We found that these steps lead to the exact same eigenpairs as our invariant-based method (and of course Smith et al. [2019]). Notably however, the tensor-based formulation is limited to cases where $\mathbf{S} \oplus \mathbf{S}$ is invertible (see Eq. (34)): The solution to Sylvester's equation is unique exactly when an element is non-degenerate (*e.g.* not collapsed to a plane, line or point) since $\det(\mathbf{F}) = \det(\mathbf{S}) \neq 0$ is assumed. Conversely, our model is robust under extreme, inverted configurations as our results show in § 7.

In comparison to Smith et al. [2019], they require computing the rotation matrix via iterative procedures *à la* SVD or PD for evaluating the energy and computing the eigenvectors $\mathbf{q}_i$ of the Hessian, which can be avoided in our method. Our energy evaluation can bypass these procedures to improve solver performance, which is significant because the energy is evaluated multiple times per Newton iteration e.g. during the line-search (see Algo. (1), Line (10)).

In fact, our formulation also provides a fast and exact method to compute the analytic rotation matrix, which could also be helpful for Smith et al. [2019] and others. The closed-form root function $f \equiv \operatorname{tr}(\mathbf{F}^\intercal \mathbf{R})$ is a *re*-presentation of the trace, leading to a concise expression for the–previously numerically factored–rotation of $\mathbf{F}$. From Petersen and Pedersen [2012] and by the chain-rule, we have

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{F}} &\equiv \frac{\partial \operatorname{tr}(\mathbf{F}^\intercal \mathbf{R})}{\partial \mathbf{F}} \triangleq \mathbf{R} \\ &= \frac{\partial f}{\partial I_C} \frac{\partial I_C}{\partial \mathbf{F}} + \frac{\partial f}{\partial II_C} \frac{\partial II_C}{\partial \mathbf{F}} + \frac{\partial f}{\partial J} \frac{\partial J}{\partial \mathbf{F}}. \end{aligned} \tag{35}$$

As we have closed-forms for $\frac{\partial f}{\partial i}$ (see Eq. (24)) and $\frac{\partial i}{\partial \mathbf{F}}$ (see Appendix A), this gives a closed-form for the rotation $\mathbf{R} \Leftrightarrow \mathbf{S}^{-1}\mathbf{F}$. Higher order derivatives follow naturally; and conventional rotation factors of $\mathbf{F}$ that have previously been computed with SVD can be obtain using the diagonalization of $\mathbf{S} = \mathbf{R}^\intercal \mathbf{F} = \mathbf{V}\Sigma\mathbf{V}^\intercal$, from which we get $\mathbf{V}$ and then $\mathbf{U} = \mathbf{R}\mathbf{V}$ [6].

6 FINAL ENERGY AND IMPLEMENTATION

Having discussed some practical advantages of our method in § 5.2, we now describe the final energy that we use in this section, and summarise implementation details necessary to simulate the results that we show in § 7.

[6] See *e.g.* Eberly [2020a,b] for the reference implementation, which we use for our 2×2 and 3×3 symmetric matrix diagonalization.

(a) *No volume preservation* **(b)** *With volume preservation*

Fig. 3. *Comparison between stretching a cube with and without volume preservation.*

6.1 Final energy

Isotropic ARAP is by definition a measure of the deviation of the edge lengths (in principle-stretch space) from unit length, which implies that volume preservation is unaccounted for. Thus, an additional term is included into our energy, which attempts to maintain the original volume [Ogden 1997]. Our final energy, which accounts for this change in volume, is given by

$$\boxed{\Psi_{\text{final}} = \frac{\mu}{2} \Psi_{i\text{ARAP}} + \frac{\lambda}{2}(J-1)^2,} \tag{36}$$

which is rest stable [Smith et al. 2018] where the new Lamé constant $\lambda = Ev/(1+v)(1-2v)$ controls the material's tendency for volume preservation (*cf.* Fig. 3) The (vectorised) gradient of Eq. (36) is given by

$$\frac{\partial \Psi_{\text{final}}}{\partial \mathbf{F}} = \mu \left(\sum_i \Psi_i \mathbf{g}_i \right) + \lambda (J-1) \mathbf{g}_J, \tag{37}$$

and the Hessian by

$$\begin{aligned} \operatorname{vec}\left(\frac{\partial^2 \Psi_{\text{final}}}{\partial \mathbf{F}^2} \right) = \mu \left(\sum_i \Psi_i \mathbf{H}_i + \sum_i \sum_j \Psi_{ij} \mathbf{g}_j \mathbf{g}_i^\intercal \right) \\ + \lambda \left(\mathbf{g}_J \mathbf{g}_J^\intercal + (J-1)\mathbf{H}_J \right), \end{aligned} \tag{38}$$

where the first term is projected to positive semi-definite state using Eq. (32), with the variables $\mathbf{g}_J$ and $\mathbf{H}_J$ representing the vectorized gradient and Hessian of the volume ratio J, respectively (see also Smith et al. [2019], §4.2 for the corresponding analytic eigensystem).

6.2 Implementation

To perform *dynamic* simulation, we follow the standard approach for computing the new vertex positions $\mathbf{x}^{t+1} \in \mathbb{R}^{3n}$ by minimising of an incremental potential (IP) [Kane et al. 2000], which we summarise here: Given n vertex positions $\mathbf{x}_t \in \mathbb{R}^{3n}$ and velocities $\mathbf{v}_t \in \mathbb{R}^{3n}$ at time $t \in \mathbb{R}$, this IP (with Rayleigh damping) is defined as

$$\begin{aligned} E(\mathbf{x}, \mathbf{x}^t, \mathbf{v}^t) = \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^\intercal \mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}) + \Delta t^2 \Psi_{\text{final}} \\ + k \frac{1}{2\Delta t^2}(\mathbf{x} - \mathbf{x}^t)^\intercal \mathbf{M}(\mathbf{x} - \mathbf{x}^t) - \Delta t^2 \mathbf{x}^\intercal (\mathbf{f}_d + \mathbf{f}_e), \end{aligned} \tag{39}$$

where Δt is the time step size, $\mathbf{M} \in \mathbb{R}^{3n \times 3n}$ is the diagonal mass matrix and $\hat{\mathbf{x}} = \mathbf{x}^t + \Delta t \mathbf{v}^t + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}_e$. k is the damping coefficient. The variables $\mathbf{f}_d$ and $\mathbf{f}_e$ denote friction forces (if collisions are handled) and forces that are external to the body, respectively.

In practice, we compute the new positions $\mathbf{x}^{t+1}$ by iteratively minimising Eq. (39) using a Newton-type solver. At each iteration,

Algorithm 1 Projected Newton Solver

procedure MINIMISEINCREMENTALPOTENTIAL($\mathbf{x}^t$)
  $\mathbf{x} \leftarrow \mathbf{x}^t$
  $\mathbf{x}_{\text{prev}} \leftarrow \mathbf{x}$
  $E_{\text{prev}} \leftarrow E(\mathbf{x})$
  repeat
    $\mathbf{g} \leftarrow \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}}$
    $\mathbf{H} \leftarrow \text{projectedHessian}\left(\frac{\partial^2 E(\mathbf{x})}{\partial \mathbf{x}^2}\right)$ ▷ Section § 5
    $\mathbf{p} \leftarrow -\mathbf{H}^{-1}\mathbf{g}$ ▷ Solve Eq. (40)
    $\alpha \leftarrow 1$
    do ▷ Line search
      $\mathbf{x} \leftarrow \mathbf{x}_{\text{prev}} + \alpha\mathbf{p}$
      $\alpha \leftarrow \frac{\alpha}{2}$
    while $E(\mathbf{x}) > E_{\text{prev}}$
    $\mathbf{x}_{\text{prev}} \leftarrow \mathbf{x}$
    $E_{\text{prev}} \leftarrow E(\mathbf{x})$
  until $\|\mathbf{g}\|_\infty < \epsilon$
  return $\mathbf{x}$ ▷ New positions $\mathbf{x}^{t+1}$
end procedure

we minimise the energy by evaluating the following linear system at the *current iterate* $\mathbf{x}$ (see also Line (8) in Algo. (1))

$$\left[\mathbf{M} + \Delta t^2 \frac{\partial^2 \Psi_{\text{final}}}{\partial \mathbf{x}^2} + k\frac{1}{2\Delta t^2}\mathbf{M}\right]\mathbf{p} = -\left(\mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}) + \Delta t^2 \frac{\partial \Psi_{\text{final}}}{\partial \mathbf{x}} + k\frac{1}{2\Delta t^2}\mathbf{M}(\mathbf{x} - \mathbf{x}^t) - \Delta t^2(\mathbf{f}_d + \mathbf{f}_e)\right), \tag{40}$$

using preconditioned conjugate gradients (PCG) [Shewchuk 1994] where $\mathbf{p}$ is the vector of displacements, the variables $\frac{\partial \Psi_{\text{final}}}{\partial \mathbf{x}}$ and $\frac{\partial^2 \Psi_{\text{final}}}{\partial \mathbf{x}^2}$ are computed from $\frac{\partial \Psi_{\text{final}}}{\partial \mathbf{F}}$ (Eq. (37)) and $\frac{\partial \mathbf{P}(F)}{\partial \mathbf{F}}$ (Eq. (38)) using Eq. (7) and Eq. (9), respectively. A simplified example (without collisions) is outlined in Algo. (1).

7 RESULTS AND DISCUSSION

We present our results in this section, which are produced on an Ubuntu system with a 16-core 3.8GHZ Intel Xeon Platinum CPU and 32GB of RAM. Examples are based on a tetrahedral, triangle or strand mesh based discretization. Simulations are solved with a standard Newton solver with a line search, and our linear systems are solved using the Eigen implementation of the PCG method [Guennebaud et al. 2021]; the only exception is our cloth simulation, which is solved with Sparsesuite (cholmod) [Davis and Hu 2011] due to the number of (triangle) elements (32k). All Newton solves were run until the absolute infinity-norm of the force residual was less than $10^{-4}\Delta t$ as Smith et al. [2019] with all other PCG settings set to their defaults.

Compression and stretch tests. We setup a cylinder to be compressed (2/3× of original length) and stretched (3× original length). The model is composed of 142,720 tetrahedra where we translate hardconstrained vertices on two opposing faces. Three models are compared with: corotational energy [McAdams et al. 2011]; Stable Neo-Hookean energy (SNE) [Smith et al. 2018]; and force-filtering [Teran et al. 2005] with the rotation gradients computed with the

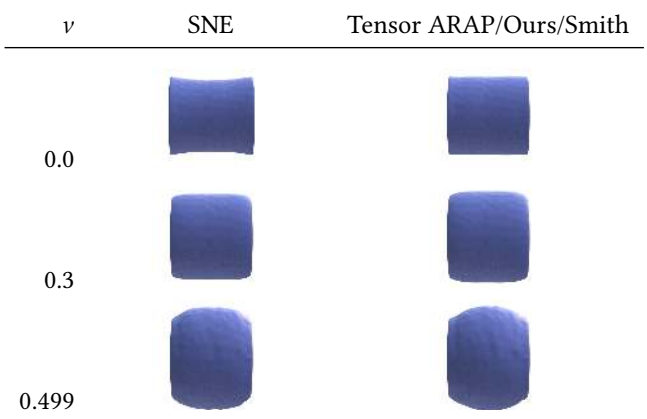


Fig. 4. *Beam compression test with three settings for Poisson's Ratio ν. We compare our method with Stable Neo-Hookean (SNE) [Smith et al. 2018]; the tensor based ARAP formulation [Teran et al. 2005] (cf. Eq. (9)) where rotation gradients are computed with the Sylvester equation in Eq. (33); and Smith et al. [2019]'s distortion energy (Smith)*

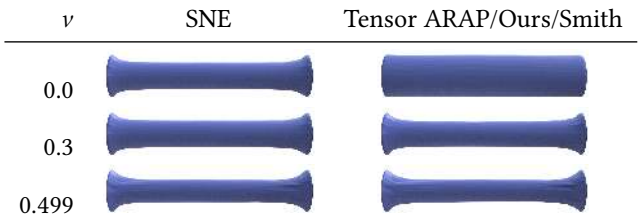
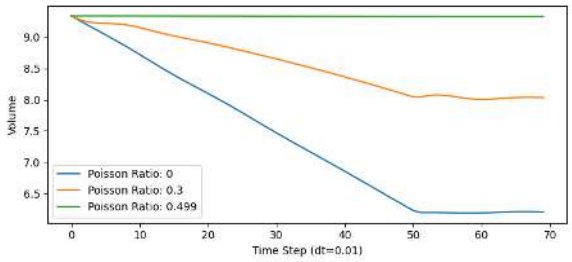


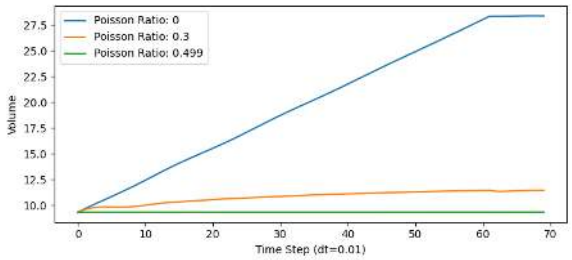
Fig. 5. *Beam stretch test. See also Fig. 4*

Sylvester equation (§ 5.2). We show results for Poisson's ratio set to $\nu = 0.1$, 0.3 and 0.49 in Fig. 4 and Fig. 5. Our method is able to preserve volume in all cases according the setting of ν. We successively penalize volume-loss (or volume-gain) in the limit of $\nu \approx 0.499$ (the beam loses 0.12% volume when compressing and gains 0.14% when stretching) and similarly eliminate any tendency for penalization when $\nu \approx 0$. Our results are also shared with the force filtering method. Conversely, SNE works well under stretching but shows artefacts when $\nu \approx 0$ due to Smith et al. [2018]'s reparameterization of the Lamé constants μ and λ for their model to reproduce the PK1 of linear elasticity (see §3.4 in [Smith et al. 2018]). Further results are shown in Fig. 6 which plots our methods rate of volume loss for the three setting of ν. We preserve volume best under stretching, we are gaining approximately 0.05 m^3 volume and losing 0.02 m^3 per time step on average when $\nu = 0.3$. (see orange line with $\nu = 0.3$ in Fig. 6a and Fig. 6b). Our method is overall robust and captures bulging deformations well at $\nu \approx 0.499$ and remains able to preserve edge length when $\nu \approx 0.0$ without requirement for reparameterization.

Dynamics. In Fig. 7, we show our ARAP model undergoing dynamic, extreme deformations using the Stanford Bunny model. We fix the base and then pull on the ears as well as recovering from flat planar state. In both instances, the bunny recovers as soon as



(a) *Compression*



(b) *Stretching*

Fig. 6. *A plot of total volume loss with three setting of Possion's Ratio ν. The Young's Modulus is set to 5000.*

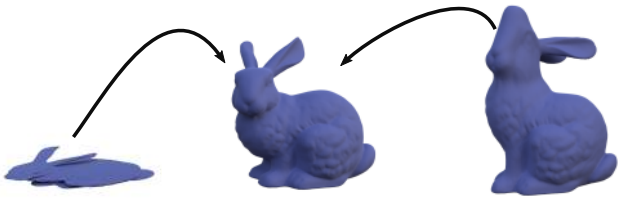


Fig. 7. *Recovery from severe compression and stretch.*

the external force is removed demonstrating that our formulation reproduces the expected deformation behaviour.

Scramble test. In Fig. 8, we perform a 'scramble' test similar to Smith et al. [2018], Stomakhin et al. [2012] and Teran et al. [2005]. The vertices of a unit cube are randomly placed within a cube of twice the rest volume, with vertices of the corner maintained in their proper positions to constrain rigid body modes. The results demonstrate that our method is robust to inversions and other degenerate configurations (*e.g.* element flattening) to gracefully recover the original rest configuration.

Cloth. Our energy is also applicable to triangle meshes, where deformation is fundamentally expressed in the respective 2D plane. A notable aspect is that these meshes yield a non-square deformation gradient $\mathbf{F} \in \mathbb{R}^{3\times2}$, which poses a subtle nuisance since polar decomposition is assumed to work only on square matrices. Working in 2D is one solution but this requires explicit projections of vertices, casting the deformation problem to the plane. Sumner and Popović [2004]'s auxiliary vertex scheme is a solution but increases the number of degrees of freedom. Instead, by using a non-square factorisation, which we describe in Appendix C, we can maintain the same energy formulation and without augmenting with extra degrees of freedom (DOF).

To simulate cloth, our energy (*cf.* Eq. (22)) acts as the stretching term (replacement for Eq. (11) in [Kim 2020]), where we add the Baraff-Witkin style shearing term from Kim [2020] and a Discrete Shells bending energy from Grinspun et al. [2003]. Using a stretching test applied to a piece of cloth, we have found that the resulting deformations are qualitatively similar to Kim's Baraff-Witkin model (see Fig. 10) and exactly identical to his ARAP shell energy.

We also evaluate our cloth model using a *Virtual Try-On* scenario (see Fig. 9), dressing up an animated model whose motion deforms the outfit due to contact and frictional forces. Our formulation remains robust and provides an adequate choice for modelling complex cloth with contacts, where the conditions of indefiniteness are also known. Collisions between the cloth (dress) and body (including self-collisions) are handled using the Incremental Potential Contact (IPC) method of Li et al. [2020].

Performance. We evaluate performance by comparing the average time and number of conjugate gradient iterations performed by the Newton solver at each time step, which is shown in Tab. 1. Our method exhibits faster performance than related methods, including the state-of-the-art [Smith et al. 2019], and with notable capabilities for volume preservation which is commonly associated with Neo-Hookean models.

Performance differences with SNE [Smith et al. 2018] are attributed to the differences in the non-linearity of the energy terms relating to volume preservation, which tend to affect the convergence rate of our conjugate gradient solves and line-search. These differences may be further exacerbated by element inversion, which Smith et al. [2018] also penalize with a regularized origin barrier energy term but we treat with a standard quadratic term in J. The fact that our model is faster when $\nu = 0$ may be evidence for this case, and perhaps suggests a source of potential future improvement in performance by combining our implicit ARAP stretch term with their origin barrier.

3D Shape Interpolation. Given an initial shape and a final shape, our formulation can also be used to interpolate between two deformations as in Smith et al. [2019] and Chao et al. [2010]. We use a deformation energy Ψ_{src} which is measured relative to the initial shape, and Ψ_{tgt} which is measured relative to the final shape. The combined (interpolation) energy is then $\Psi = (1-t)\Psi_{\text{src}} + t\Psi_{\text{tgt}}$, $t \in [0, 1]$, which we minimise given t. Both Ψ_{src} and Ψ_{tgt} are based on our ARAP energy in Eq. (36). To evaluate our method we first interpolated between two poses of a hand model with bending deformations Fig. 1 which our method handles robustly. We use models from Yeh et al. [2010]: the *source* shape is on the left, interpolated shapes are in the middle, and the *target* shape is on the right (t = 0, 0.25, 0.5, 0.75, 1.0). The hand model had 25k elements and running at average 5.2s per iteration with 5.3 iterations per time step. We also evaluate our method using two poses of a twisted bar (Fig. 11) to validate our method with more complex geometry exhibiting rotational distortions. In this case, the bar model had 51k elements and took average 2.3s per iteration with 4.6 iterations per time step.

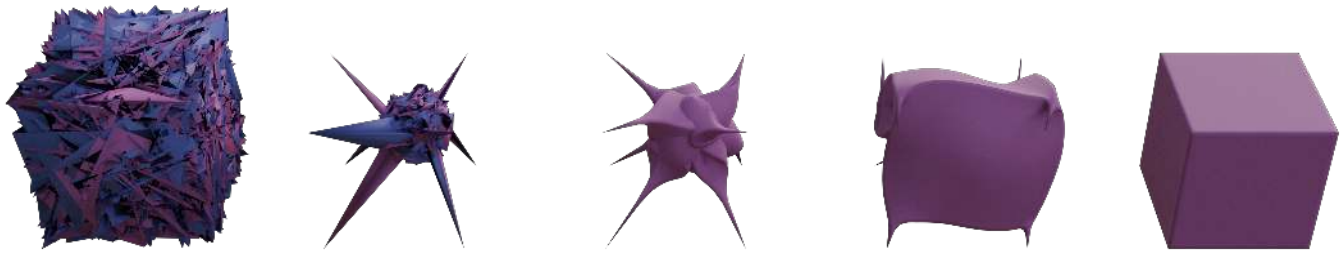


Fig. 8. *A classic benchmark where we randomly set the vertices of a cube object in a space twice its volume to then successfully recover the shape.*

Table 1. *Performance summary of elastic object simulation, with comparison against Stable Neo-Hookean elasticity (SNE) [Smith et al. 2018], force filtering (FF) [Teran et al. 2005] with Sylvestor's equation for computing rotation gradients, and Smith et al. [2019]'s distortion energy (Smith). The material parameters are as follows: Young's Modulus (E); Poisson's Ratio ν. In all case we use a timestep of 0.01s, where the number of iterations and time is presented as an average.*

				Ours		Smith		FF		SNE	
Scenario	#Elems	E	ν	Iters.	Time	Iters.	Time	Iters.	Time	Iters.	Time
Beam Stretch	142720	5000	0.499	12.2	116.60	12.2	117.15	12.2	117.90	11.5	104.24
Beam Compress	142720	5000	0.499	7.2	72.18	7.2	72.39	7.2	72.54	6.0	52.89
Beam Stretch	142720	5000	0.3	5.1	10.33	5.1	10.98	5.1	11.04	3.8	8.12
Beam Compress	142720	5000	0.3	3.8	8.07	3.8	8.27	3.8	8.66	3.3	6.79
Beam Stretch	142720	5000	0.0	3.1	5.06	3.1	5.26	3.1	5.91	3.0	5.71
Beam Compress	142720	5000	0.0	3.1	5.29	3.1	5.44	3.1	5.75	13.2	17.14
Bunny Stretch	637584	5000	0.3	21.2	437.00	21.2	444.16	21.2	454.06	14.6	238.22
Bunny Compress	637584	1500	0.3	14.5	191.50	14.5	197.72	14.5	203.72	12.0	177.37

Fig. 9. *Virtual Try-on: we can also simulate cloth deformation using our formulation with (self) collision contact and friction.*

Our method is also able to resolve interpolations between twisted geometries seamlessly.

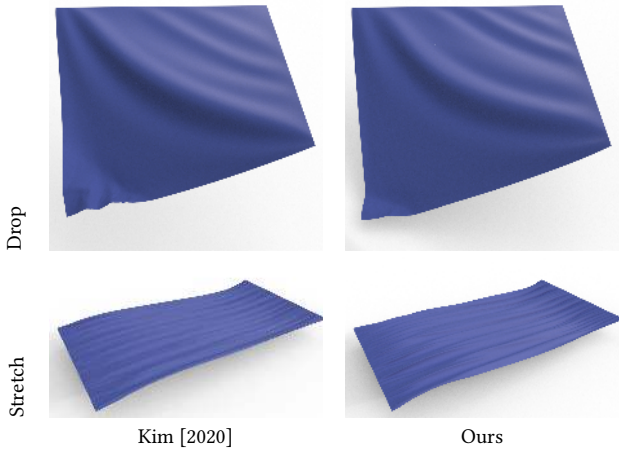


Fig. 10. *Left: Stretch and drape test using Kim [2020]'s FEM force formulation of Baraff-Witkin cloth. Right: Stretch and drape test, using our isotropic ARAP FEM formulation.*

Surface Parameterization. We test our approach on parametrization, minimizing the distortion energy expressed in the 2D plane with results shown in Fig. 12. We use the Tutte Embedding [Floater 1997] as the initial parameterization in our setup and proceed to minimize the energy using our projected Newton solver. Notably,

Table 2. *Summary of cloth simulation performance. The stretching and shearing stiffness parameters are both set to 10000.*

Scenario	#Elems	Δt	Bending Stiffness	Iters.	Time
Cloth Stretch	32768	0.01	0.2	16.8	25.67
Cloth Stretch	32768	0.01	0	14.5	22.78
Cloth Drop	32768	0.01	0.2	5.6	12.51
Cloth Drop	32768	0.01	0	4.9	8.87
Virtual Try-on	8280	0.016	0	6.7	8.63

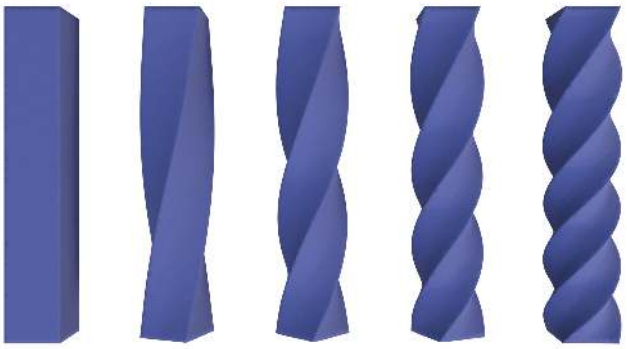


Fig. 11. *Bar twist shape interpolation.*

performing parameterization by minimizing the distortion energy in Eq. (36) will work in most instances but is generally insufficient to guarantee bijectivity. The forces arising from the quadratic volume preservation term are insufficient to prevent inversions. An example is also shown in Fig. 13 with non-bijective parameterization of the camel mesh, where areas with flipped triangles are highlighted.

Inspired by barrier formulations (*e.g.* Smith and Schaefer [2015] and Li et al. [2020]) which strive to prevent such inversion, we augment our energy with an in-plane barrier as follows

$$\Psi_{\text{FP}} = \sum_{i=1}^{3} \begin{cases} -\left(\hat{h} - h_i\right)^2 \log\left(\frac{h_i}{\hat{h}}\right), & h_i \leq \hat{h}, \\ 0, & h_i > \hat{h}. \end{cases}$$

Where h_i denotes the distance between the i-th vertex in the triangle and its opposite edge and $\hat{h}$ is a threshold (*e.g.* average triangle height) determining when the barrier force will be activated to prevent flipping. The final parameterization energy we use is given by

$$\Psi_{\text{param}} = \mu\Psi_{i\text{ARAP}} + \lambda\Psi_{\text{FP}}, \tag{41}$$

where μ and λ are set to 10 and 1, respectively throughout our experiments which are shown in Fig. 12. Our method successfully parameterizes complex models, ensuring bijectivity and with minimal distortion in the bulging extremities of surfaces.

Multiple Materials. To demonstrate the easiness of using our CG invariant based energy, we show a case of dropping three armadillo models with three different energies (defined using the CG invariants) to the ground in one scenario. These are our final elastic energy Eq. (36), the Stable Neo-Hookean elasticity [Smith et al. 2018], and

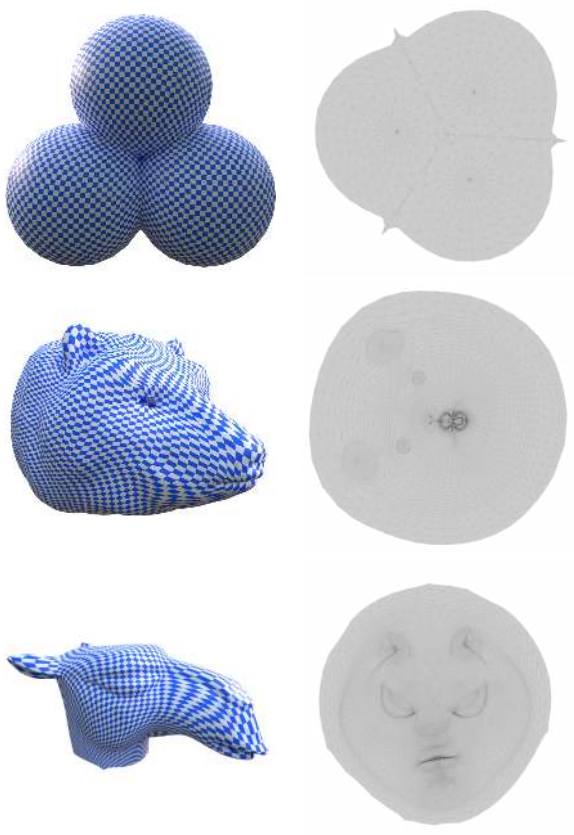
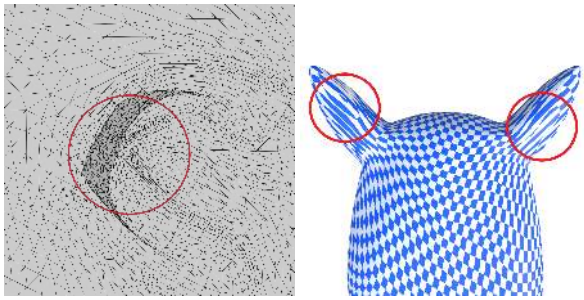


Fig. 12. *Surface parameterizations computed with our formulation (*cf. *Eq. (41))*



(a) *Flipped Local Triangles* **(b)** *After Texturing*

Fig. 13. *Non-bijective parameterization causes bad looking result*

the Saint Venant–Kirchhoff model [Sin et al. 2013], respectively. The Young's Modulus is set to 1e4 and the Poisson's Ratio is set to 0.4. Fig. 14 shows our results. Fig. 15 also demonstrates the capability of our formulation for generalising to elastic strands. The generality of our formulation additionally facilitates coupling between elements of different dimensions *e.g.* volume- and elastic strand elements.

Fig. 14. *Dropping three armadillo simultaneously with different materials. Energy from left (yellow) to right (green): SNE [Smith et al. 2018], ours, St-Vk [Sin et al. 2013]*

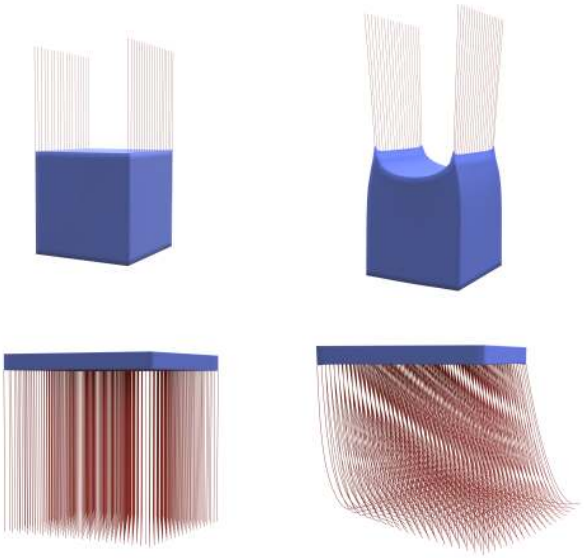


Fig. 15. *Simulating elastic strands/rods using 1D finite-elements with our implicit ARAP energy.*

Energy evaluation. We evaluate the performance of our root finding during energy evaluation by comparing with two existing SVD procedures. We measure the average time to evaluate the energy, comparing with the Eigen [Guennebaud et al. 2021] and McAdams et al. [2011]. Our results are shown in Fig. 16, which reveals that our approach (without hardware-specific optimizations) is approximately 2× faster than McAdams et al. [2011] and upto 3.7× faster than Eigen.

Rotational alignment. Tab. 3 summarises the performance of our method to extract rotations by comparing with existing methods (same setup as in Fig. 16). Comparisons are based on the source-code provided by [Zhang et al. 2021], which is in the public domain. Our method is on average 3.2× faster than the four methods we have compared with. We achieve at-least 2× speedup over the state-of-the-art [Zhang et al. 2021], and up-to 4.66× speedup over the SVD implementation provided in the Eigen library [Guennebaud et al. 2021]. Our approach is not only faster but also inherently simpler than those requiring SVD of 3×3 matrices, eigenvectors of a 4×4 matrices, or Caley- and exponential maps with optimization.

Table 3. *Rotational alignment speedup: A summary of the performance speedup of our method (evaluating Eq. (35)) versus existing approaches to compute the rotation* **R** *from* **F** *(value in parenthesis represents number of tetrahedra/deformation gradients.).*

	Bunny (700k)	Cube (200k)	Cylinder (150k)
Eigen SVD [Guennebaud et al. 2021]	3.77×	3.77×	3.76×
Horn [1987]	4.65×	4.66×	4.63×
Müller et al. [2016]	2.36×	2.36×	2.36×
Cayley Gershigorin [Zhang et al. 2021]	1.99×	2.02×	2.31×

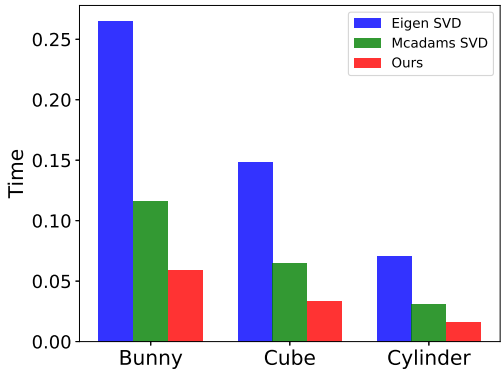


Fig. 16. *Time (seconds) taken to evaluate the energy function on three scenarios (#elements): bunny (700k), cube (200k), cylinder(150k). Each scenario is defined by a tetrahedralized model, where 'deformed' vertices* **x** *are scrambled within a volume twice the default bounding volume extents for generating deformation gradients with random stretch and rotation factors.*

8 DISCUSSION AND CONCLUSION

A Cauchy-Green invariant formulation of the isotropic ARAP energy has been presented. We introduce expressions that prescribe an implicit relationship to the linear distortion terms of ARAP. The gradient and Hessian of the energy are then evaluated with relative ease, and impart little-to-no additional cost in computation time since our new terms boil down to reducible scalars from the invariants. We have also described an approach to guarantee positive semi-definiteness of the finite element Hessian, enabling fast and concise constructions for Newton-type solvers. The energy can be applied to a wide variety of simulation-problem domains, including cloth, parametrisation, thin-strand modelling and it is robust to extreme deformations and element inversions. In sum, we have demonstrated that a rewriting of linear distortion terms using the Cauchy-Green invariants is possible, allowing for unified treatment of all respective isotropic materials which now includes corotational energies. Moreover, the complete *rogues' gallery* of $\mathbf{g}_*$ and $\mathbf{H}_*$ can be therefore used to mix-and-match a range of isotropic energies,

which now include ARAP. Our method is also useful for domain-specific programming languages targeting physics based animation like Taichi [Hu et al. 2019], Ebb [Bernstein et al. 2016] and Simit [Kjolstad et al. 2016] with an expedient approach to implementing isotropic energies written using the Cauchy-Green invariants.

Capturing the space of all isotropic materials. Our method overcomes an inherently mathematical restriction which–until now–has prevented the expression of linear corotational energies using CG invariants. Moreover, isotropic materials that are expressible using these invariants have been generally understood to form only a (proper) subset of all isotropic energies because the invariants are functions of the *squares* of the singular values σ_i^2. This understanding in-part follows from the intuition provided by the Taylor expansion of a general isotropic material $\Psi_{\text{gen}}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ *e.g.* around $\sigma_i = 1, \forall i$, which will have terms that contain both odd *and* even powers of these singular values. We have demonstrated how this mathematical restriction–as previously understood–is a loose imposition, which we now overcome.

Relations to the state-of-the-art. Comparing our formulation with the **S**-centric analysis of Smith et al. [2019], perhaps the most distinct invariant relative to our CG counterparts is their first $I_1 = \text{tr}(\mathbf{S})$, which we actually reveal to be computable without SVD to give merit to our formulation. Their second invariant $I_2 = I_C$ is the first CG invariant, and their third invariant $I_3 = J$ can be determined directly from the primary variable **F** without special treatment.

Our final energy in Eq. (36) also exhibits equivalent tendency for volume preservation as the Stable Neo-Hookean model of Smith et al. [2018] but without reparameterization, offering a CG invariant based formulation for modelling materials with high resistance to volume loss. The alternative Neo-Hookean model of Bonet and Wood [2008] requires a log-term for volume preservation, which is undefined in the inverted configuration unlike ours.

Alternative method to find singular values σ_i. From the Cayley-Hamilton theorem, the CG invariants (as elementary symmetric polynomials) can also be understood as the coefficients of the characteristic polynomial

$$\det(\mathbf{C} - \lambda \mathbf{I}),$$

where **I** is the identity tensor and λ is a scalar variable. Moreover, this polynomial (cubic) has roots that are available as analytical expressions and evaluate to the eigenvalues σ_i^2 of the right Cauchy-Green tensor **C**. Thus, as an alternative to our approach (*cf.* § 3.3 and § 3.4) one may recover σ_i via square roots with an additional sign-copy from the volume ratio J to the smallest the singular value (see also Franca [1989] and Sarabandi et al. [2020] for further discussion). Our advantage is the provision of direct closed-form expressions for the singular values without aforementioned sign-copies, simplifying derivations of energy gradients (yielding rotations) and Hessians for a more prudent implementation with less book-keeping. Readers are also referred to Marsden and Hughes [1994] for further detail about isotropic elasticity and invariants (see *e.g.* Propositions 5.8 to 5.10 of Chapter 3 on page 220).

Limitations and Future work. Root-finding is generally known to be potentially unstable for certain values of the coefficients, which we did not address in this paper. Our procedures are based on Ferrari's method with complex numbers using the real part of the answer as our final result and is sufficient for our purposes. However, further investigation is needed as several stages of solving the quartic involve square roots, which leads to branching out to special cases and awkward treatments of discriminants.

Our method has advantage over state-of-the-art analysis [Smith et al. 2019] but we still require rotation factors **U** and **V** to reuse their 'closed-form' eigenvectors of the local Hessian. However, no proof exists to show that expressions defined in terms of these factors represent the simplest/most-efficient analytic form. Our provision of a proper analytic rotation **R** suggests otherwise, thus leaving an open problem to prove this hypothesis which may potentially lead to eigenvector expressions that depend directly on this **R**.

ACKNOWLEDGMENTS

This work was partially funded by the Research Grant Council of Hong Kong (GRF 17210222). This work was also supported by the Innovation and Technology Commission of the HKSAR Government under the InnoHK initiative, and the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust. Finally, we thank the reviewers for their detailed and insightful feedback.

REFERENCES

Jernej Barbič. 2012. *Exact Corotational Linear FEM Stiffness Matrix.* Technical Report. Technical Report, Computer Science Department, USC.

Jan Bender and Crispin Deul. 2013. Technical Section: Adaptive Cloth Simulation Using Corotational Finite Elements. *Comput. Graph.* 37, 7 (nov 2013), 820–829.

Gilbert Louis Bernstein, Chinmayee Shah, Crystal Lemire, Zachary Devito, Matthew Fisher, Philip Levis, and Pat Hanrahan. 2016. Ebb: A DSL for Physical Simulation on CPUs and GPUs. *ACM Trans. Graph.* 35, 2, Article 21 (may 2016), 12 pages.

Maurice A. Biot. 1938. Theory of elasticity with large displacements and rotations.

Javier Bonet and Richard D. Wood. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis* (2 ed.). Cambridge University Press.

Ralph Byers and Hongguo Xu. 2008. A New Scaling for Newton's Iteration for the Polar Decomposition and its Backward Stability. *SIAM J. Matrix Anal. Appl.* 30, 2 (2008), 822–843.

Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. 2010. A Simple Geometric Model for Elastic Deformations. *ACM Trans. Graph.* 29, 4, Article 38 (jul 2010), 6 pages.

Timothy A. Davis and Yifan Hu. 2011. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Softw.* 38, 1, Article 1 (dec 2011), 25 pages.

David Eberly. 2020a. *A Robust Eigensolver for* 2×2 *Symmetric Matrices.* White Paper. Geometric Tools, PO Box 1866, Mountain View, CA 94042, USA. 5 pages. https://www.geometrictools.com/Documentation/RobustEigenSymmetric2x2.pdf

David Eberly. 2020b. *A Robust Eigensolver for* 3×3 *Symmetric Matrices.* White Paper. Geometric Tools, PO Box 1866, Mountain View, CA 94042, USA. 18 pages. https://www.geometrictools.com/Documentation/RobustEigenSymmetric3x3.pdf

Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. 2003. A Fast Finite Element Solution for Cloth Modelling. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG '03).* IEEE Computer Society, USA, 244.

Michael S Floater. 1997. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design* 14, 3 (1997), 231–250.

L.P. Franca. 1989. An algorithm to compute the square root of a 3 × 3 positive definite matrix. *Computers Mathematics with Applications* 18, 5 (1989), 459–466.

Zhan Gao, Theodore Kim, Doug L. James, and Jaydev P. Desai. 2009. Semi-Automated Soft-Tissue Acquisition and Modeling Forsurgical Simulation *(CASE'09).* IEEE Press, 268–273.

Gene H Golub and Charles F Van Loan. 2012. *Matrix computations.* Vol. 3. JHU Press.

Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete Shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03).* Eurographics Association, Goslar, DEU, 62–67.

Gaël Guennebaud, Benoît Jacob, et al. 2021. Eigen v3.8.0. http://eigen.tuxfamily.org.

Berthold K. P. Horn. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A* 4, 4 (Apr 1987), 629–642. https://doi.org/10.1364/

JOSAA.4.000629
B. J. Hsieh. 1977. Nonlinear formulation of finite element by corotational coordinates. (1 1977).
Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. 2019. Taichi: A Language for High-Performance Computation on Spatially Sparse Data Structures. *ACM Trans. Graph.* 38, 6, Article 201 (nov 2019), 16 pages.
Frédéric Hélein and John C. Wood. 2008. Harmonic maps: Dedicated to the memory of James Eells. In *Handbook of Global Analysis*, Demeter Krupka and David Saunders (Eds.). Elsevier, Amsterdam, 417–491.
G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Grenoble, France) *(SCA '04)*. Eurographics Association, Goslar, DEU, 131–140.
Couro Kane, Jerrold E Marsden, Michael Ortiz, and Matthew West. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *International Journal for numerical methods in engineering* 49, 10 (2000), 1295–1325.
Theodore Kim. 2020. *A Finite Element Formulation of Baraff-Witkin Cloth.* Eurographics Association, Goslar, DEU.
Theodore Kim and David Eberle. 2020. Dynamic Deformables: Implementation and Production Practicalities. In *ACM SIGGRAPH 2020 Courses* (Virtual Event, USA) *(SIGGRAPH '20)*. Association for Computing Machinery, New York, NY, USA, Article 23, 182 pages.
Fredrik Kjolstad, Shoaib Kamil, Jonathan Ragan-Kelley, David I. W. Levin, Shinjiro Sueda, Desai Chen, Etienne Vouga, Danny M. Kaufman, Gurtej Kanwar, Wojciech Matusik, and Saman Amarasinghe. 2016. Simit: A Language for Physical Simulation. *ACM Trans. Graph.* 35, 2, Article 20 (May 2016), 21 pages.
Tassilo Kugelstadt, Dan Koschier, and Jan Bender. 2018. Fast Corotated FEM using Operator Splitting. *Computer Graphics Forum (SCA)* 37, 8 (2018).
Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Trans. Graph. (SIGGRAPH)* 39, 4, Article 49 (2020).
Jerrold E. Marsden and Thomas JR. Hughes. 1994. *Chapter 3.* Dover.
Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 37, 12 pages.
Matthias Müller, Jan Bender, Nuttapong Chentanez, and Miles Macklin. 2016. A Robust Method to Extract the Rotational Part of Deformations. In *Proceedings of the 9th International Conference on Motion in Games* (Burlingame, California) *(MIG '16)*. Association for Computing Machinery, New York, NY, USA, 55–60.
Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. 2002. Stable Real-Time Deformations *(SCA '02)*. Association for Computing Machinery, New York, NY, USA, 49–54.
Andriy Myronenko and Xubo Song. 2009. On the closed-form solution of the rotation matrix arising in computer vision problems. https://doi.org/10.48550/ARXIV.0904.1613
R. W. Ogden. 1997. *Non-linear Elastic Deformations.* Dover.
Julian Panetta. 2018. *Optimizing over SO(3).* Technical Report. University of California, Davis.
Julian Panetta. 2019. *Analytic Eigensystems for Isotropic Membrane Energies.* Technical Report. University of California, Davis.
K. B. Petersen and M. S. Pedersen. 2012. The Matrix Cookbook. Version 20121115.
Guillaume Picinbono, Herve Delingette, and Nicholas Ayache. 2000. Real-Time Large Displacement Elasticity for Surgery Simulation: Non-Linear Tensor-Mass Model. In *Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '00)*. Springer-Verlag, Berlin, Heidelberg, 643–652.
C. C. Rankin and F. A. Brogan. 1986. An Element Independent Corotational Procedure for the Treatment of Large Rotations. *Journal of Pressure Vessel Technology* 108, 2 (05 1986), 165–174.
Soheil Sarabandi, Arya Shabani, Josep M. Porta, and Federico Thomas. 2020. On Closed-Form Formulas for the 3-D Nearest Rotation Matrix Problem. *IEEE Transactions on Robotics* 36, 4 (2020), 1333–1339. https://doi.org/10.1109/TRO.2020.2973072
Jonathan R Shewchuk. 1994. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.* Technical Report. USA.
Eftychios Sifakis and Jernej Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) *(SIGGRAPH '12)*. Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages.
Fun Shing Sin, Daniel Schroeder, and Jernej Barbič. 2013. Vega: non-linear FEM deformable object simulator. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 36–48.
Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (mar 2018), 15 pages.
Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (feb 2019), 15 pages.
Jason Smith and Scott Schaefer. 2015. Bijective Parameterization with Free Boundaries. *ACM Trans. Graph.* 34, 4, Article 70 (jul 2015), 9 pages.
Olga Sorkine and Marc Alexa. 2007. As-Rigid-as-Possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Barcelona, Spain) *(SGP '07)*. Eurographics Association, Goslar, DEU, 109–116.
Olga Sorkine-Hornung and Michael Rabinovich. 2017. *Least-Squares Rigid Motion Using SVD.* Technical Report. Technical Note, Department of Computer Science, ETH Zurich.
Alexey Stomakhin, Russell Howes, Craig Schroeder, and Joseph M. Teran. 2012. Energetically Consistent Invertible Elasticity *(EUROSCA'12)*. Eurographics Association, 25–32.
Robert W. Sumner and Jovan Popović. 2004. Deformation Transfer for Triangle Meshes. *ACM Trans. Graph.* 23, 3 (aug 2004), 399–405.
Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation *(SCA '05)*. Association for Computing Machinery, New York, NY, USA, 181–190.
Christopher D. Twigg and Zoran Kacic-Alesic. 2010. Point Cloud Glue: Constraining Simulations Using the Procrustes Transform. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, Zoran Popovic and Miguel Otaduy (Eds.). The Eurographics Association.
Bohan Wang, George Matcuk, and Jernej Barbič. 2021. Modeling of Personalized Anatomy using Plastic Strains. *ACM Trans. on Graphics (TOG)* 40, 2 (2021).
Jin Wu, Ming Liu, Zebo Zhou, and Rui Li. 2018. Simple Fast Vectorial Solution to The Rigid 3D Registration Problem. *CoRR* abs/1806.00627 (2018).
Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbič. 2015. Nonlinear Material Design Using Principal Stretches. *ACM Trans. Graph.* 34, 4, Article 75 (jul 2015), 11 pages.
I-Cheng Yeh, Chao-Hung Lin, Olga Sorkine, and Tong-Yee Lee. 2010. Template-based 3D Model Fitting Using Dual-domain Relaxation. *IEEE Transactions on Visualization and Computer Graphics* 99, RapidPosts (2010).
Jiayi Eris Zhang, Alec Jacobson, and Marc Alexa. 2021. Fast Updates for Least-Squares Rotational Alignment. *Computer Graphics Forum* (2021).

A INVARIANT DERIVATIVES

In this section, we summarise the gradients and Hessian of the Cauchy-Green invariants for self containment, which are also found in [Kim and Eberle 2020].

For I_C, we have

$$\frac{\partial I_C}{\partial \mathbf{F}} = 2\mathbf{F} \quad \frac{\partial^2 I_C}{\partial \mathbf{F}^2} = 2\frac{\partial \mathbf{F}}{\partial \mathbf{F}}$$
$$\mathbf{g}_1 = 2\mathrm{vec}\,(\mathbf{F}) \quad \mathbf{H}_1 = 2\mathbf{I}_{9\times 9}.$$

For II_C, we have

$$\frac{\partial II_C}{\partial \mathbf{F}} = 4\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{F} \quad \frac{\partial^2 II_C}{\partial \mathbf{F}^2} = 4\left(\frac{\partial \mathbf{F}}{\partial \mathbf{F}}\mathbf{F}^\mathsf{T}\mathbf{F} + \mathbf{F}\frac{\partial \mathbf{F}^\mathsf{T}}{\partial \mathbf{F}}\mathbf{F} + \mathbf{F}\mathbf{F}^\mathsf{T}\frac{\partial \mathbf{F}}{\partial \mathbf{F}}\right)$$
$$\mathbf{g}_2 = 4\mathrm{vec}\,(\mathbf{F}\mathbf{F}^\mathsf{T}\mathbf{F}) \quad \mathbf{H}_2 = 4\left(\mathbf{I}_{3\times 3} \otimes \mathbf{F}\mathbf{F}^\mathsf{T} + \mathbf{F}^\mathsf{T}\mathbf{F} \otimes \mathbf{I}_{3\times 3} + \mathbf{D}\right),$$

where $A \otimes B$ denotes the Kronecker product and the matrix $\mathbf{D}$ is given by

$$\mathbf{D} = \begin{bmatrix} \mathbf{f}_0\mathbf{f}_0^\mathsf{T} & \mathbf{f}_1\mathbf{f}_0^\mathsf{T} & \mathbf{f}_2\mathbf{f}_0^\mathsf{T} \\ \mathbf{f}_0\mathbf{f}_1^\mathsf{T} & \mathbf{f}_1\mathbf{f}_1^\mathsf{T} & \mathbf{f}_2\mathbf{f}_1^\mathsf{T} \\ \mathbf{f}_0\mathbf{f}_2^\mathsf{T} & \mathbf{f}_1\mathbf{f}_2^\mathsf{T} & \mathbf{f}_2\mathbf{f}_2^\mathsf{T} \end{bmatrix}.$$

here $\mathbf{f}_i$ denotes the ith column of $\mathbf{F}$.

Finally, for J, we have

$$\mathbf{g}_3 = 2J\mathbf{g}_J \quad \mathbf{H}_3 = 2\mathbf{g}_J\mathbf{g}_J^\mathsf{T} + 2J\mathbf{H}_J,$$

where $J = \det(\mathbf{F})$ and

$$\mathbf{g}_J = \mathrm{vec}\left(\begin{bmatrix} \mathbf{f}_1 \times \mathbf{f}_2 & \mathbf{f}_2 \times \mathbf{f}_0 & \mathbf{f}_0 \times \mathbf{f}_1 \end{bmatrix}\right)$$

$$\mathbf{H}_J = \begin{bmatrix} \mathbf{0} & -\hat{\mathbf{f}}_2 & \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 & \mathbf{0} & -\hat{\mathbf{f}}_0 \\ -\hat{\mathbf{f}}_1 & \hat{\mathbf{f}}_0 & \mathbf{0} \end{bmatrix},$$

where $\hat{\mathbf{x}}$ is the matrix cross-product operator

$$\hat{\mathbf{x}} = \begin{bmatrix} 0 & -\mathbf{x}_2 & \mathbf{x}_1 \\ \mathbf{x}_2 & 0 & -\mathbf{x}_0 \\ -\mathbf{x}_1 & \mathbf{x}_0 & 0 \end{bmatrix}.$$

Corresponding derivatives of III_C (*i.e.* $\mathbf{g}_{III_C} = \mathrm{vec}\left(\partial III_C/\partial \mathbf{F}\right)$ and $\mathbf{H}_{III_C} = \mathrm{vec}\left(\partial^2 III_C/\partial \mathbf{F}^2\right)$) are evaluated similarly but using $\mathbf{C}$ in place of $\mathbf{F}$.

B TRACE POLYNOMIAL DERIVATIVES

Here we summarise the numerator expressions p_{ij}

$$i, j \in \{1, 2, 3\} \iff \{I_C, II_C, J\},$$

that are required to evaluate Eq. (25). These can be written in several forms and are computed easily since they are only comprised of reducible scalar terms

$$\begin{aligned}
p_{11} &= 4f\frac{\partial f}{\partial I_C} + 2 - \left(12f^2\frac{\partial f}{\partial I_C} - 4f - 4I_C\frac{\partial f}{\partial I_C}\right)\frac{\partial f}{\partial I_C} \\
p_{12} &= 4f\frac{\partial f}{\partial II_C} - \left(12f^2\frac{\partial f}{\partial II_C} - 4I_C\frac{\partial f}{\partial II_C}\right)\frac{\partial f}{\partial I_C} \\
p_{13} &= 4f\frac{\partial f}{\partial J} - \left(12f^2\frac{\partial f}{\partial J} - 4I_C\frac{\partial f}{\partial J} - 8\right)\frac{\partial f}{\partial I_C} \\
p_{21} &= -\left(12f^2\frac{\partial f}{\partial I_C} - 4f - 4I_C\frac{\partial f}{\partial I_C}\right)\frac{\partial f}{\partial II_C} \\
p_{22} &= -\left(12f^2\frac{\partial f}{\partial II_C} - 4I_C\frac{\partial f}{\partial II_C}\right)\frac{\partial f}{\partial II_C} \\
p_{23} &= -\left(12f^2\frac{\partial f}{\partial J} - 4I_C\frac{\partial f}{\partial J} - 8\right)\frac{\partial f}{\partial II_C} \\
p_{31} &= 8\frac{\partial f}{\partial I_C} - \left(12f^2\frac{\partial f}{\partial I_C} - 4f - 4I_C\frac{\partial f}{\partial I_C}\right)\frac{\partial f}{\partial J} \\
p_{32} &= 8\frac{\partial f}{\partial II_C} - \left(12f^2\frac{\partial f}{\partial II_C} - 4I_C\frac{\partial f}{\partial II_C}\right)\frac{\partial f}{\partial J} \\
p_{33} &= 8\frac{\partial f}{\partial J} - \left(12f^2\frac{\partial f}{\partial J} - 4I_C\frac{\partial f}{\partial J} - 8\right)\frac{\partial f}{\partial J}
\end{aligned}$$

C ROTATIONS FOR LOWER-DIMENSIONAL ELEMENTS

The example case of 3D cloth simulation [Kim 2020] presents a scenario in which (triangular) elements are furnished with a non-square (*e.g.* 3×2) deformation gradient that would be used to evaluate Eq. (33). However, standard convention stipulates that the (in-plane) rotation $\mathbf{R}$ for evaluating gradients is computed from a square $\mathbf{F}$. In keeping with this dictum, methods (*e.g.* Bender and Deul [2013]; Etzmuß et al. [2003] and Panetta [2019]) will augment the element with an auxiliary vertex [Sumner and Popović 2004] or normal [Panetta 2019] so as to ensure a square $\mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ deformation gradient, which increases the number of degrees of freedom in $\mathbf{F}$. We revisit this problem and advocate a general polar factorization of a (potentially) non-square deformation gradient to produce a (semi-) orthogonal rotation $\mathbf{R} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ and a symmetric matrix $\mathbf{S} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ that is positive semi-definite. We further prove that minimizing the ARAP energy with the resulting rotation is equivalent to the case where $\mathbf{F}$ is a square matrix of dimension $\mathbb{R}^{\mathrm{n}\times\mathrm{n}}$.

Non-square polar factorization. One can extract a rotation $\mathbf{R}$ from a non-square $\mathbf{F} = \mathbf{U}\boldsymbol{\Sigma}^\star\mathbf{V}^\mathsf{T} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ $(m > n)$ with a standard SVD routine, where $\mathbf{U} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$ is orthogonal and $\boldsymbol{\Sigma}^\star \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ is a rectangular diagonal matrix

$$\boldsymbol{\Sigma}^\star = \left[\begin{array}{ccc} & \boldsymbol{\Sigma} & \\ \hline 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{array}\right], \tag{42}$$

The sub-matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ has non-negative diagonal entries, and $\mathbf{V} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ is also orthogonal.

The rotation and scaling factors from this 'semi' polar decomposition (semi-PD) are then

$$\mathbf{R} = \mathbf{U}\left[\begin{array}{ccc} & \mathbf{V}^\mathsf{T} & \\ \hline 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{array}\right] \tag{43}$$

$$\mathbf{S} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T}. \tag{44}$$

The (semi-) orthogonal rotation $\mathbf{R}$ is now a rank n matrix, which can be demonstrated as follows:

$$\begin{aligned}
\mathbf{R}^\mathsf{T}\mathbf{R} &= \left[\begin{array}{c|ccc} & 0 & \cdots & 0 \\ \mathbf{V} & \vdots & \ddots & \vdots \\ & 0 & \cdots & 0 \end{array}\right]\mathbf{U}^\mathsf{T}\mathbf{U}\left[\begin{array}{ccc} & \mathbf{V}^\mathsf{T} & \\ \hline 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{array}\right] \\
&= \mathbf{I}_{n\times n}.
\end{aligned}$$

This rank n rotation $(n \leq m)$ nonetheless serves as an adequate term with which to measure and minimise deviation from the corresponding deformation gradient.

Proof. Eq. (1) is in fact the energy of an n-dimensional element embedded in m-dimensional space with deformation gradient $\mathbf{F} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$.

We use Chao et al. [2010]'s model to show this

$$\Psi_{\mathrm{Chao}} = \frac{\mu}{2}\int_{\Omega_e} \min_{\mathbf{R}^\dagger \in SO(3)} \|\mathbf{F} - \mathbf{R}^\dagger\|_{\mathrm{F}}^2, \tag{45}$$

representing an energy density function over the domain of an element Ω_e which has a solution $\mathbf{R}^\dagger$ for a *fixed* $\mathbf{F}$ to minimise $||\mathbf{F} - \mathbf{R}^\dagger||_{\mathrm{F}}^2$. This minimisation of Eq. (45) is also equivalent to maximising the trace $\mathrm{tr}(\mathbf{F}^\mathsf{T}\mathbf{R}^\dagger)$, which is revealed by applying a similar expansion

as Eq. (2) using Eq. (5). This property will be fundamental to our proof.

By noting that $\mathbf{F} = \mathbf{U}(\Sigma^\star)\mathbf{V}^\intercal$ and $\mathbf{F}^\intercal = \mathbf{V}(\Sigma^\star)^\intercal\,\mathbf{U}^\intercal$, we demonstrate that the (potentially non-square and) orthogonal matrix $\mathbf{R}$ from Eq. (43) *is* the solution $\mathbf{R}^\dagger$ for minimizing Eq. (45). Let $\mathbf{A} = \sqrt{\Sigma}$, where Σ is the diagonal matrix of singular values of $\mathbf{F}$, and rewrite the trace by

$$\begin{aligned}\mathrm{tr}(\mathbf{F}^\intercal\mathbf{R}^\dagger) &= \mathrm{tr}\left(\mathbf{V}(\Sigma^\star)^\intercal\mathbf{U}^\intercal\mathbf{R}^\dagger\right)\\ &= \mathrm{tr}\left(\mathbf{V}\mathbf{A}_L\mathbf{A}_R\mathbf{U}^\intercal\mathbf{R}^\dagger\right)\\ &= \langle \mathbf{V}\mathbf{A}_L, \mathbf{A}_R\mathbf{U}^\intercal\mathbf{R}^\dagger\rangle,\end{aligned}$$

where

$$\mathbf{A}_L = \left[\begin{array}{c|c} \mathbf{A} & \begin{matrix}0\\0\end{matrix} \end{array}\right], \quad \mathbf{A}_R = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{\mathbf{A}} & \begin{matrix}0\\0\end{matrix} \\ \hline 0 & 0 & 0 \end{array}\right]$$

and $\langle\cdot,\cdot\rangle$ denotes matrix inner product. From the Cauchy–Schwarz inequality, we have

$$\mathrm{tr}\left(\mathbf{F}^\intercal\mathbf{R}^\dagger\right) \le \|\mathbf{V}\mathbf{A}_L\|_\mathrm{F}^2 \cdot \|\mathbf{A}_R\mathbf{U}^\intercal\mathbf{R}^\dagger\|_\mathrm{F}^2,$$

with which we can show that the rotation $\mathbf{R}$ from our semi-PD in Eq. (43) will minimise Eq. (45) by

$$\begin{aligned}\|\mathbf{V}\mathbf{A}_L\|_\mathrm{F}^2 \cdot \|\mathbf{A}_R\mathbf{U}^\intercal\mathbf{R}^\dagger\|_\mathrm{F}^2 &= \|\mathbf{A}_L\|_\mathrm{F}^2 \cdot \|\mathbf{A}_R\|_\mathrm{F}^2\\ &= \|\mathbf{A}\|_\mathrm{F}^2 \cdot \|\mathbf{A}\|_\mathrm{F}^2\\ &= \mathrm{tr}(\mathbf{A}^2)\\ &= \mathrm{tr}(\Sigma)\\ &= \mathrm{tr}(\mathbf{S})\\ &= \mathrm{tr}(\mathbf{F}^\intercal\mathbf{R}). \qquad (46)\end{aligned}$$

We arrive at Eq. (46) using the isotropy property of orthogonal matrices (invariance of Frobenius norm under rotations). Thus, the energy for an n-dimensional element (*e.g.* triangle) in m-dimensional space is also Eq. (1), but the rotation matrix now comes from our semi-PD in Eq. (43) and Eq. (44). □