A Simple Geometric Model for Elastic Deformations

Isaac Chao
Caltech

Ulrich Pinkall
TU Berlin

Patrick Sanan
Caltech

Peter Schröder
Caltech, TU München

Abstract

We advocate a simple geometric model for elasticity: *distance between the differential of a deformation and the rotation group*. It comes with rigorous differential geometric underpinnings, both smooth and discrete, and is computationally almost as simple and efficient as linear elasticity. Owing to its geometric non-linearity, though, it does not suffer from the usual linearization artifacts. A material model with standard elastic moduli (Lamé parameters) falls out naturally, and a minimizer for static problems is easily augmented to construct a fully variational 2nd order time integrator. It has excellent conservation properties even for very coarse simulations, making it very robust.

Our analysis was motivated by a number of heuristic, physics-like algorithms from geometry processing (editing, morphing, parameterization, and simulation). Starting with a continuous energy formulation and taking the underlying geometry into account, we simplify and accelerate these algorithms while avoiding common pitfalls. Through the connection with the Biot strain of mechanics, the intuition of previous work that these ideas are "like" elasticity is shown to be spot on.

Keywords: Digital Geometry Processing, Discrete Differential Geometry, elasticity, geometric modeling, shape space interpolation, morphing, parameterization.

1 The Elastic Energy

We jump right into the meat of things and defer discussion of relations to previous algorithms until we complete our setup.

Given a smooth map $f : M \to \tilde{M}$, $f(p) = q$ describing the deformation of a reference configuration $M \subset \mathbb{R}^n$ into a deformed configuration $\tilde{M} \subset \mathbb{R}^n$, we study the energy

$$E(f) = \tfrac{1}{2}\int_M \operatorname{dist}\left(df, \mathrm{SO}(n)\right)^2 = \tfrac{1}{2}\int_M \min_{R\in \mathrm{SO}(n)} |df - R|^2. \qquad (1)$$

At a point $p \in M$, the integrand measures the distance between the deformation differential, df, and the nearest rotation, thus characterizing how far f is from an isometry. (Our df corresponds to what is often called the deformation gradient F in mechanics.) In 2D this is the setting of planar morphing and parameterization while the 3D case principally covers solid mechanics.

A minimizer of $E(f)$ subject to boundary conditions is characterized by vanishing variations $\delta_g E(f) := \frac{d}{d\epsilon}|_{\epsilon=0} E(f + \epsilon g) = 0$. Let g be an arbitrary admissible variation. Then,

$$\delta_g E(f) = \int_M \langle dg - \delta_g R, df - R\rangle = \int_M \langle dg, df - R\rangle, \qquad (2)$$

where $\langle A, B\rangle = \operatorname{tr}(A^T B)$ denotes the standard inner product between linear maps and $R \in \mathrm{SO}(n)$ the minimizer of the squared distance (dropping explicit mention of its dependence on df). We have used $df - R \perp \delta_g R$, which follows from R being critical with respect to the squared distance; thus, no derivatives of R appear in the gradient of E.

Due to the dependence of R on df, the Euler-Lagrange equation is a non-linear Poisson problem

$$\Delta f = \operatorname{div} R.$$

The energy Hessian follows from taking a further variation h

$$\delta^2_{g,h} E(f) = \int_M \langle dg, dh\rangle - \int_M \langle dg, \delta_h R\rangle.$$

The first term is the standard Laplace-Beltrami operator which does not depend on f, while the second term varies with f through the dependence of R on df. Here variations of R do enter since they are in general not orthogonal to dg.

2 Discrete Setting

From now on let $M = (V, E, F)$ or (V, E, F, T) be a 2- or 3-manifold simplicial complex consisting of vertices, edges, triangular facets, and, possibly, tetrahedra, with associated initial vertex positions $p_i \in \mathbb{R}^2$ resp. $\mathbb{R}^3$. The deformation function f is given as a piecewise affine mapping from p_i to new positions q_i. The integral over M becomes a sum over triangles resp. tetrahedra. The corresponding discrete energy in 2D is the well known piecewise linear Dirichlet energy [Pinkall and Polthier 1993]

$$\tfrac{1}{2}\int_{p_{ijk}} |df - R|^2 = \tfrac{1}{4}\sum_{e_{ab}\in\{i,j,k\}} \cot\alpha^c_{ab}|q_{ab} - R^{ijk} p_{ab}|^2,$$

where α^c_{ab} denotes the angle at c opposite edge e_{ab}; $p_{ab} = p_b - p_a$, the directed edge from p_a to p_b; and q_{ab} its image under df. Note that R^{ijk} is constant per triangle since df is. The corresponding cotan formula for a tetrahedron [Meyer et al. 2002] is

$$\tfrac{1}{2}\int_{p_{ijkl}} |df - R|^2 = \tfrac{1}{12}\sum_{e_{ab}\in\{i,j,k,l\}} |p_{cd}| \cot\alpha^{cd}_{ab}|q_{ab} - R^{ijkl} p_{ab}|^2,$$

with α^{cd}_{ab} the dihedral angle opposite e_{ab}, *i.e.*, at edge e_{cd} and R^{ijkl} constant per tetrahedron.

To minimize this energy with standard Newton trust region solvers [Benson et al. 2007] we need its gradient and Hessian.

2.1 The First Discrete Variation

This is now just a matter of differentiating the above expressions with respect to the q_i. For a single triangle $\{i, j, k\}$ we obtain

$$\delta_i E(q) = \tfrac{1}{2}(\cot\alpha^k_{ij} q_{ji} + \cot\alpha^j_{ik} q_{ki}) - R^{ijk}\tfrac{1}{2}(\cot\alpha^k_{ij} p_{ji} + \cot\alpha^j_{ik} p_{ki})$$

The first term represents a summand in the Laplace-Beltrami operator applied to the q variables, while the second term is the area gradient of the original triangle p_{ijk} with respect to vertex i rotated by R^{ijk}. As in the continuous setting (Eq. (2)), criticality ensures that no derivatives of R^{ijk} appear in the discrete setting. These expressions are summed for all triangles incident to vertex i. The sum of the rotated area gradients is the divergence of the rotations incident on vertex i.

The expression for a tetrahedron is entirely analogous with terms evaluating the Laplacian of q and the *volume* gradient of the tetrahedron p_{ijkl}, with respect to vertex i, rotated by R^{ijkl}. Fully detailed expressions are contained in the attached C++ code.

2.2 The Closest R

For a df with positive determinant, R is most efficiently found by computing the right polar decomposition $df = RY$ with Newton's algorithm [Higham 1986, p. 1168] (with warm start). In case of inverted elements ($\det df < 0$) the polar decomposition returns an *orientation reversing* orthogonal transformation instead of the closest rotation. In these cases, as well as for singular df, the closest rotation (or *a* closest rotation for rank 0 or 1 in 3D resp. 0 in 2D) is given as a simple function of the SVD [Myronenko and Song 2009, Lemma 1]. The more costly SVD procedure is invoked only for df with non-positive determinant. (In practice we use $\det df < \epsilon$ which also covers all rank deficient cases.)

Comment on inverted elements The distance $|df - R|^2$ is always continuous, *even* as the element inverts. (Additionally it is smooth if $\det df > 0$ or $\det df <= 0$ and the smallest singular value is unique.) The energy only increases for an inverted element since the nearest rotation is further away. Compare this to fourth order measures such as $|df^T df - I|^2$. In that case a mirrored tetrahedron $df = -I$ has zero strain. This has led to the development of, at times elaborate, measures to deal with inverted elements in elasticity (see, *e.g.*, [Irving et al. 2004; Schmedding and Teschner 2008]). In our setting we merely need to ensure that the nearest rotation and not by accident the nearest orientation reversing orthogonal transform is computed.

Comment on performance One could use an SVD in all cases but we do not recommend this for performance reasons. Over a broad set of simulations we found that 10% of the entire runtime is spent computing R. Within the R routine only about 1-3% of all calls invoke the SVD, but those few calls consume 10-20% of the time in the R routine. Because the polar decomposition (with warm starts) is so much cheaper than the SVD, using the SVD every time would increase overall runtime by 50% or more.

The resulting algorithm is quite robust as demonstrated by a "torture test:" the tetrahedralized Dragon model, scaled by $-3/4$, is used as an initial configuration. In this configuration every tetrahedron is inverted and squished, yet the original shape is quickly recovered without any adverse effects (see the attached movie).

2.3 The Second Discrete Variation

It produces the standard cotan-Laplace operator—independent of f—and an additional term which depends on f (through Y and R) and is constant per triangle resp. tetrahedron

$$\langle dg, \delta_h R \rangle = 4 \left\langle \mathrm{X}(R^T dg), \mathrm{X}(R^T dh) \right\rangle_W , \qquad (3)$$

a weighted (by W) scalar product of the anti-symmetric part ($\mathrm{X}(.)$ denotes extraction of the anti-symmetric part of a matrix) of the backward rotated differentials of the variations g and h. In practice this expression amounts to a linear combination of precomputed outer product matrices multiplied by R^T and the weight matrix, which is a simple linear function of Y (see App. B). Fully detailed expressions can be found in the attached C++ code.

Implementation notes Ignoring file I/O and display routines, our overall code is quite compact and consists of four main components: energy, gradient and Hessian callbacks, as well as the polar decomposition resp. SVD routine to find the nearest rotation. The callbacks iterate over the mesh data structure, evaluating energy, gradient, and Hessian expressions on a per element basis (as given above and documented in the attached code), accumulating the results as they go along. Owing to the use of standard black box Newton trust region solver [Benson et al. 2007]—to which the callbacks are passed—little else is needed to produce all results shown in this paper.

3 Elasticity Simulation

To get a material model with Lamé parameters we note that the difference $df - R = R(Y - I)$ can be further decomposed orthogonally into a trace-free part and a multiple of the identity

$$Y - I = \left(Y - \tfrac{\mathrm{tr}(Y)}{n} I\right) + \left(\tfrac{\mathrm{tr}(Y)}{n} - 1\right) I =: \bar{Y} + yI$$

with $n = 2, 3$ the dimension of M. Each can now be weighted independently with $\alpha \geq 0$ (bulk deformation) and $\beta \geq 0$ (shear deformation)

$$E_{\alpha,\beta}(f) = \tfrac{1}{2} \int_M |\beta\bar{Y} + \alpha y I|^2 = \tfrac{1}{2} \int_M \beta^2 |\bar{Y}|^2 + \tfrac{\alpha^2}{n} (\mathrm{tr}(Y) - n)^2 .$$

The energy density for an isotropic Hookean material, $\langle \sigma, \epsilon \rangle$ with $\sigma = 2\mu\epsilon + \lambda\,\mathrm{tr}(\epsilon) I$ and Lamé parameters μ and λ maps to $E_{\alpha,\beta}(f)$ for $\epsilon = Y - I$, $2\mu = \beta^2$, and $\lambda = (\alpha^2 - \beta^2)/n$. With $E_{\alpha,\beta}(f)$ we have a geometrically non-linear elastic potential energy with standard material parameters and a minimization solver can find static equilibrium configurations of elastic bodies. (The corresponding gradient and Hessian expressions, as functions of α and β, are included in the attached code samples for the first and second variation.) Figure 1 demonstrates deformations and the influence of the α and β parameters.

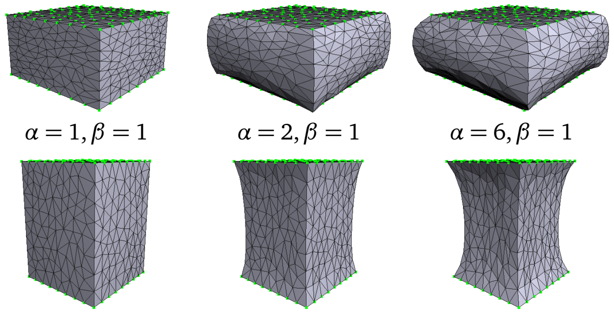
$\alpha = 1, \beta = 1$ $\alpha = 2, \beta = 1$ $\alpha = 6, \beta = 1$

Figure 1: *Examples of tetrahedral mesh compression (down to 53% height) and expansion (up 135% height) for different α parameters.*

Dynamics simulations are a straightforward extension of the static elasticity case and many avenues are possible. For our experiments we implemented the *fully variational integrator* (FVI) of [Kharevych et al. 2006], based on the stationary Hamilton-Pontryagin principle. We chose it because it exhibits—like all variational integrators [Marsden and West 2001]—excellent long term energy stability and exact preservation of momenta, even for large time steps. From a practical point of view an FVI has the significant benefit that it converts the usual non-linear root finding problem in the time stepper into an energy minimization

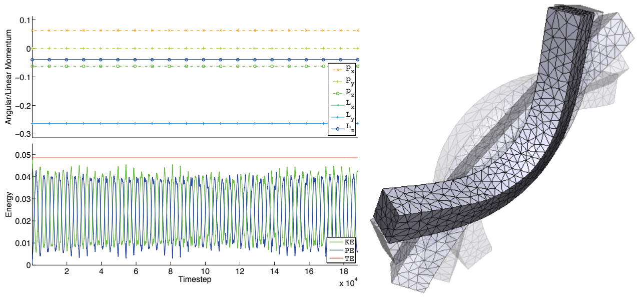

Figure 2: *Tracking the momentum and energy of an elastic body with some initial motion and bending deformation. Momentum and energy are tracked precisely even for very long simulation runs.*

problem. *And* this is achieved with only a minor modification to the energy minimization code we already have!

Assume per vertex momentum ($\mathfrak{p}^k$), position ($\mathfrak{q}^k$), and velocity variables ($\mathfrak{v}^k$) indexed by discrete time $k \geq 0$. Define an ancillary energy

$$\varepsilon(\mathfrak{q}) := \tfrac{2}{h^2}(\mathfrak{q}-\mathfrak{q}^k)^T M(\mathfrak{q}-\mathfrak{q}^k) + W(\mathfrak{q}) - \tfrac{2}{h}(\mathfrak{p}^k)^T(\mathfrak{q}-\mathfrak{q}^k),$$

with h denoting the time step size, M the mass matrix, and $W(.) = E_{\alpha,\beta}(.)$ the elastic potential of the body. Define $\mathfrak{q}^* = \operatorname{argmin}_{\mathfrak{q}} \varepsilon(\mathfrak{q})$, then the transition $k \to k+1$ proceeds as

$$\begin{aligned}
\mathfrak{v}^{k+1} &= \tfrac{2}{h}(\mathfrak{q}^* - \mathfrak{q}^k) \\
\mathfrak{q}^{k+1} &= \mathfrak{q}^k + h\mathfrak{v}^{k+1} \\
\mathfrak{p}^{k+1} &= M\mathfrak{v}^{k+1} - \tfrac{h}{2}\operatorname{grad} W(\mathfrak{q}^k + \tfrac{1}{2}h\mathfrak{v}^{k+1})
\end{aligned}$$

Minimization of $\varepsilon(\mathfrak{q})$ requires its gradient and Hessian. Both are simple modifications of the corresponding expressions for $W(\mathfrak{q}) = E_{\alpha,\beta}(\mathfrak{q})$

$$\begin{aligned}
\operatorname{grad}\varepsilon(\mathfrak{q}) &= \operatorname{grad} W(\mathfrak{q}) + \tfrac{4}{h^2}M(\mathfrak{q}-\mathfrak{q}^k) - \tfrac{2}{h}\mathfrak{p}^k \\
\operatorname{Hess}\varepsilon(\mathfrak{q}) &= \operatorname{Hess} W(\mathfrak{q}) + \tfrac{4}{h^2}M.
\end{aligned}$$

In this manner the existing minimization code needs only a small modification to serve as a time stepper for dynamics simulations. To complete the system one would likely want to add external forces and damping and we refer the interested reader to [Kharevych et al. 2006] for the details. There one also finds the derivation of the energy minimization time stepper.

Figure 2 demonstrates the exact momentum conservation and long term stable energy behavior which distinguish these integrators. A more complex example is given in Figure 3.

Insight The simple energy of Eq. (1), measuring the distance between the differential of the deformation and the rotation group, provides a full-fledged model for isotropic elasticity, in particular when this difference is further decomposed into the orthogonal subspaces of trace-free matrices and multiples of the identity.

In the mechanics literature a linearized version of $Y-I$ is known as the Biot strain [Biot 1938]. It is the first order entry in a ladder of generalized Lagrange strains with the standard Green-Lagrange strain $df^T df - I = Y^2 - I$ as the second order member. Because $Y-I$ is lower order in the state variables, it is computationally far more attractive. Compare, *e.g.*, the efforts to “reign in” the computational complexity of the standard Green-Lagrange strain undertaken in [Kikuuwe et al. 2009].

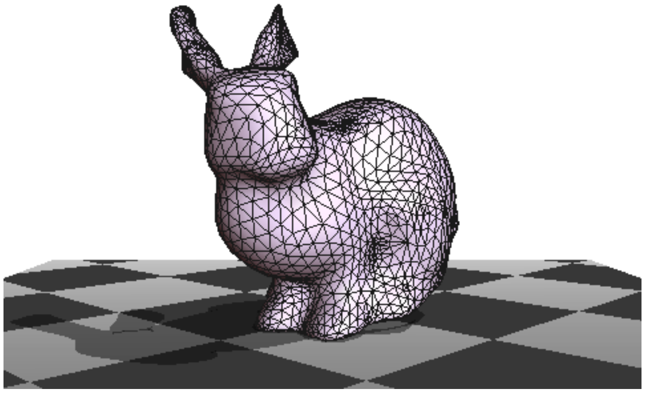

Figure 3: *A Bunny being dropped and colliding with the ground plane with no damping whatsoever. The energy is perfectly preserved. (See also the attached movie.)*

Shape matching approaches to qualitative physics simulations [Müller et al. 2005] use reasoning reminiscent of our energy. Considering groups of nearby points, they use a shape matching metaphor to determine the best fit rigid transform of the group of points from initial to deformed configuration. The resulting rotation $\tilde{R}$ arises from the polar decomposition of $\tilde{R}\tilde{Y} = df\, I_m$ where I_m is the inertia tensor of the point set (up to a scale factor). This additional factor of I_m makes the resulting $\tilde{R}$ dependent on the shape of the group of points. As a consequence the material behavior may become anisotropic if the groups of points are not isotropic in shape. No continuous energy is given.

3.1 Relationship to Co-Rotational Methods

Per element rotations applied to df appear also in co-rotational methods [Rankin and Brogan 1986]. These have become popular in computer graphics since [Müller et al. 2002; Etzmuß et al. 2003] as a post facto fix for the disturbing artifacts due to the linearized Green-Lagrange strain $(df^T df - I) \simeq (df^T + df)/2 - I$ when large deformations or rotations occur. We emphasize that this is quite different from our non-linear setting which uses R to measure the distance from df to an isometry. The following experiment demonstrates that this difference is not mere semantics.

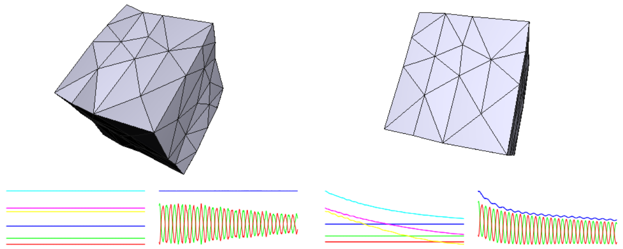

Figure 4: *Comparison of our non-linear energy (left) with a co-rotational method (right). Linear and angular momenta (6 lines) and energy (kinetic, potential, total) are shown in the plots. For our non-linear energy, momenta are exactly preserved and total energy stays within a small constant band. Fixing rotations at the beginning of each time step (co-rotational) destroys these system invariants. See also the attached movie.*

We will fix the rotations at the beginning of each time step. Now the backward rotated $R^T df$ at each element gives the strain used in a co-rotational method. The time stepper energy minimization is quadratic and a single solve moves from $k \to k+1$ (the integration is still a 2nd order accurate implicit method). Figure 4 shows what happens in a comparison with the non-linear energy. An elastic body in 3D is subject to some initial momentum and a twist/compression without any external forces. Time step size was identical in both cases. At the bottom of each image are graphs of momenta (left) and energy (right). For this simulation they should stay constant and they do when using the non-linear energy in contrast to the co-rotational method.

Numerical vs. variational damping The numerical damping in the co-rotational method causes not only loss of energy and momentum invariants, but also depends on time step size. This is illustrated in an attached movie showing side by side runs with stepsizes of 0.001 and 0.0002 (the latter subsampled by 5 for display). With the rotation fixed at the beginning of the time step, the state determined for the end of the time step uses the wrong (old) rotation, assigning potential energy to the incremental rotation during the time step. When the rotation is then reset at the beginning of the next time step, this energy is lost. Numerical experiments easily verify this lossage mechanism. As illustrated in the video smaller time steps can lessen, but never fix this phenomenon. There is always some potential energy, corresponding to a small but finite rotation, which is discarded.

In application practice one generally wishes to have damping, but it should not depend on the time step size. This is easily achieved with variational damping (for details see [Kharevych et al. 2006]) as demonstrated by the side by side comparison in the attached movie. Again 0.001 and 0.0002 time step sizes are compared. Linear and angular momentum are conserved, but energy (due to losses in the potential energy) decays at the selected rate even though the time step sizes are different by a factor of 5.

Analysis The increased robustness due to conservation of system invariants in our method comes at a cost: additional linear system solves invoked by the Newton solver and computation of the additional terms in the Hessian (Eq. (3)). The latter adds only 3.5% total runtime overhead and we consider it negligible. To compare the methods we count the total number of iterations in the linear algebra solver over the entire run: 3113 for the co-rotational and 7959 for the non-linear energy. One could therefore compare with a co-rotational method run at one third the step size. As demonstrated above (for a reduction in stepsize by a factor of 5), this does not fix the problem of the co-rotational formulation, it only lessens the rate at which system invariants are lost. The inclusion of the additional term in the Hessian (Eq. (3)) is essential.

4 Other Applications

4.1 Parameterization

The 2D case of Eq. (1) was studied by Liu *et al.* [2008] for purposes of surface parameterization. They treated the energy as a functional depending on both df and R and optimized each in an alternating manner, fixing R to find the best df, updating R to be optimal, and repeating until convergence. They also considered distance to the closest similarity and describe a mixed functional (their Eq. 5) which trades off similarity against isometry. However, in that functional the isometry constraint enters to fourth order (their $(a^2+b^2-1)^2$), while we can control the trade-off with only second order. For $(\alpha=0, \beta=1)$ we get as-similar-as-possible (ASAP) solutions and for $(\alpha=\beta=1)$ as-rigid-as-possible (ARAP) (see Fig. 5 and compare with Fig. 7 of Liu *et al.*), with $\alpha\in[0,1]$ controlling the trade-off.

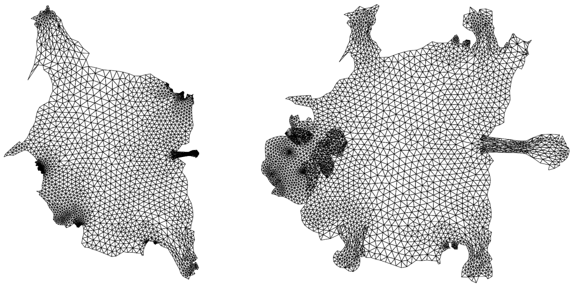

Figure 5: *Our method applied in the 2D setting of parameterization. For $\alpha=0$, $\beta=1$ ASAP,* i.e.*, discrete conformal maps result, and for $\alpha=\beta=1$ ARAP maps similar to Liu* et al. *result.*

Comparison The alternating minimization approach of Liu *et al.* has the advantage of only using a constant system matrix, the Laplace-Beltrami operator, which may therefore be pre-factored. In our Newton solver the Hessian changes and thus cannot be prefactored. But the number of iterations are far fewer for us than in their alternating solver because of the quadratic convergence of Newton versus their linear convergence (see Fig. 6 for data from the 3D case; the results are similar in 2D). We also require no specific isometry constraints (of higher order) to manage the trade-off between similarities and isometries. Instead the trade-off is a natural consequence of the split of Y into trace-free and multiple-of-the-identity parts.

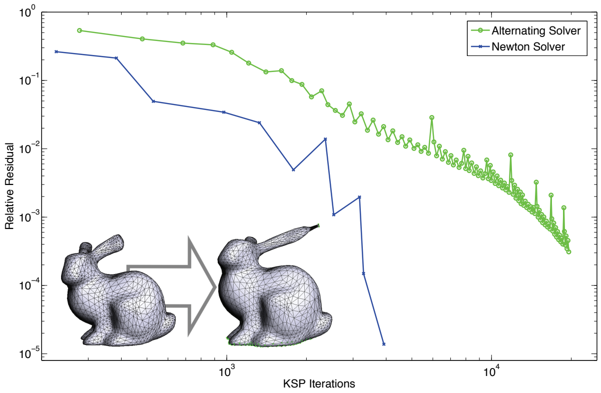

Figure 6: *Convergence of the alternating solver resp. Newton solver. The abscissa shows number of linear system iterations as a common measure of time.*

4.2 Surface Modeling

For the case of triangle meshes in $\mathbb{R}^3$ Sorkine and Alexa [2007] model a discrete deformation energy which, as in the case of Liu *et al.*, is a function of df and R, using the same alternating optimization approach. Here the rotations are chosen at vertices to be nearest to the cross covariance of original and deformed edges summed over the incident edges with 2D cotan weights. (Our R results if the sum is taken over edges incident to a triangle.) As a consequence they incorporate the extrinsic curvature of the surface and their discrete energy is a mix of membrane and bending terms. No continuous energy is given and it is unclear how the bending contribution depends on the geometry or how its relative weighting with respect to the membrane energy is determined.

If we modify their energy slightly by summing over all edges incident to the triangles incident to a given vertex, *i.e.*, we also add the “rim edges” to their “spoke edges” sum, we arrive at an energy which can be analyzed. In that case the rotation R^i at vertex i is closest to df integrated over the entire 1-ring of triangles of a given vertex. Using a Taylor series argument one can then show that the resulting continuous energy differs from ours by a bending term weighted by r^2 where r is the radius of integration over which R was chosen to be optimal

$$\tilde{E} = \int_M |df - R| + r^2 \langle dR, dRY \rangle.$$

As a consequence the bending contribution is dependent on the mesh element size (r) and goes to zero as the mesh is refined. (The vanishing of the bending term is not just a concern in the refinement limit, but is easily noticeable for meshes with strongly differing discretization rates in different parts of the surface.) Renormalizing this bending contribution and controlling it with a user selectable parameter independent of the discretization rate is an intriguing direction for future research.

Rigid frames in other work Botsch and co-workers [2006] introduced a surface modeling approach in which frames also play a central role, albeit in a way which is very different from ours. They represent the surface with a dual, SE(3) valued mesh. Each dual vertex carries a frame and each dual edge has associated with it an elastic energy between frames. The energy they formulate unfortunately has no continuum limit. Coming up with a continuous picture that is akin to what PriMo does would be quite interesting.

4.3 Geodesic Interpolation in Shape Space

A metric on the space of shapes may be defined with the aid of the elastic deformation energy. Given two shapes M_0 and M_1, the distance between them is the length of a geodesic

$$d^2(M_0, M_1) = \min_{\gamma} \int_0^1 \left(\int_{M_{\gamma(t)}} |\dot{Y}|^2 \right) dt$$

with γ ranging over all paths connecting M_0 and M_1. This point of view was pursued by Kilian *et al.* [2007] for triangle meshes in $\mathbb{R}^3$. Their model does not derive from an elastic energy, but can be understood as the squared residuals of a point mass and spring model with unit coefficients on all edges independent of how stretched they might be.

We can use our elastic energy to approximate such geodesic paths as follows. Recall that a “point” M_t on a geodesic path from M_0 to M_1 is a critical point of the functional

$$M_t = \text{argmin}_{M_x} (1-t) d^2(M_0, M_x) + t d^2(M_x, M_1).$$

As t ranges over $[0,1]$, M_t moves along the geodesic from M_0 to M_1. We now exploit the fact that the elastic energy agrees with the geodesic distance to second order since $|df_t - R_t|^2 = |\dot{Y}_0|^2 t^2 + \text{h.o.t}$ (for $f_t : M_0 \to M_t$), to replace the actual geodesic distance with our energy as a distance approximation

$$\tilde{M}_t := \text{argmin}_{M_x} (1-t) E(M_0 \to M_x) + t E(M_1 \to M_x).$$

(Here $E(M_a \to M_b)$ denotes the energy of the map from a reference shape M_a to a deformed shape M_b.) This generates intermediate shapes $\tilde{M}_{t\in[0,1]}$, which are static equilibria (for fixed t) of two weighted non-linear “springs” pulling on the intermediate shape from the beginning resp. end configuration. The corresponding energy minimization problem is no harder than what we already have. Setting up the gradient and Hessian does require twice as many evaluations since they are a convex combination ($(1-t)$ resp. t) of the gradients and Hessians of the individual energies. Figure 7 shows snapshots of the resulting trajectory between a straight and twisted/bent beam. One may now approximate a locally optimal geodesic trajectory arbitrarily closely by minimizing over more and more intermediate shapes (as was done by Kilian *et al.*).

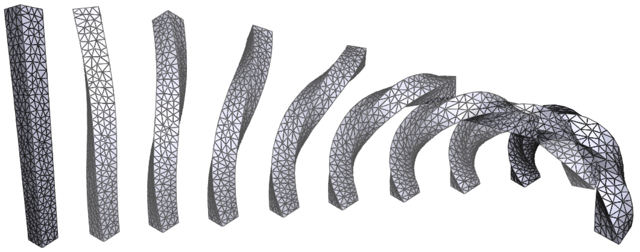

Figure 7: *Near geodesic interpolation between two shapes using the elastic energy as an approximation of geodesic distance. See also the attached movie.*

5 Summary

Our paper was motivated by the breadth of successful applications of Laplacian-like models for many algorithms which appeal to a physical modeling metaphor. Starting with a smooth model (Eq. (1)), which captures the core idea, a careful analysis of the geometric structures underlying it pays off in a number of ways. The biggest practical payoff comes from the additional term in the Hessian due to the rotations. It enables a far more efficient solver (Newton) and is the key to long time stability of our time integrator. The explicit role of rotations as a footpoint for a distance computation also illuminates the issue of inverted elements: one must not accidentally use an orientation reversing orthogonal transformation. (This is of course also true for inverted elements in standard elasticity, though it may not always be as apparent.) Our analysis also enabled us to make the connection with the Biot strain, validating intuitions expressed in a variety of previous work that these models are elasticity-like. In fact, when implemented correctly, we have a first class elasticity model, with textbook parameters, for the large displacement/rotation, small strain regime.

As far as the co-rotational method is concerned our observations can be interpreted as a prescription for improving existing co-rotational simulation systems: add the missing Hessian term and invoke a Newton solver at each time step! Little else would need changing in existing codes to gain the benefit of increased stability. (Or, if one wishes, the ability to take much larger time steps without the usual dissipative effects of doing so.)

In future work the most interesting question for us is that of triangle meshes in $\mathbb{R}^3$ and how to incorporate a well controlled notion of thickness, with its attendant bending contribution, without having to resort to higher order models.

Acknowledgment This work was supported in part by NSF (CCF-0635112), Caltech’s Center for the Mathematics of Information, and the DFG Research Center MATHEON. Additional support was provided by the Rose Hill Foundation and the IAS at TU München.

References

BENSON, S., MCINNES, L. C., MORÉ, J., MUNSON, T., AND SARICH, J. 2007. TAO User Manual. Tech. Rep. ANL/MCS-TM-242 (Revision 1.9), Argonne National Laboratory.

BIOT, M. A. 1938. Theory of Elasticity with Large Displacements and Rotations. In *Proc. Fifth Int. Cong. Appl. Math.*, John Wiley & Sons, 117–122.

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. PriMo: Coupled Prisms for Intuitive Surface Modeling. In *Proc. Symp. Geom. Proc.*, 11–20.

ETZMUSS, O., KECKEISEN, M., AND STRASSER, W. 2003. A Fast Finite Element Solution for Cloth Modeling. In *Proc. Pac. Graph.*, 244–251.

HIGHAM, N. J. 1986. Computing the Polar Decomposition—With Applications. *SIAM J. Sci. Stat. Comp. 7*, 4, 1160–1174.

IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible Finite Elements for Robust Simulation of Large Deformations. In *Proc. Symp. Comp. Anim.*, 131–140.

KHAREVYCH, L., WEIWEI, TONG, Y., KANSO, E., MARSDEN, J., SCHRÖDER, P., AND DESBRUN, M. 2006. Geometric, Variational Integrators for Computer Animation. In *Proc. Symp. Comp. Anim.*, 43–52.

KIKUUWE, R., TABUCHI, H., AND YAMAMOTO, M. 2009. An Edge-based Computationally Efficient Formulation of Saint Venant-Kirchhoff Tetrahedral Finite Elements. *ACM Trans. Graph. 28*, 1, 1–13.

KILIAN, M., MITRA, N. J., AND POTTMANN, H. 2007. Geometric Modeling in Shape Space. *ACM Trans. Graph. 26*, 3, #64, 1–8.

LIU, L., ZHANG, L., XU, Y., GOTSMAN, C., AND GORTLER, S. J. 2008. A Local/Global Approach to Mesh Parameterization. *Comp. Graph. Forum 27*, 5, 1495–1504.

MARSDEN, J. E., AND WEST, M. 2001. Discrete Mechanics and Variational Integrators. *Acta Numerica*, 10, 357–514.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. 2002. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Vis. Math. III*. 35–57.

MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable Real-Time Deformations. In *Proc. Symp. Comp. Anim.*, 49–54.

MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless Deformations Based on Shape Matching. *ACM Trans. Graph. 24*, 3, 471–478.

MYRONENKO, A., AND SONG, X. 2009. On the Closed-Form Solution of the Rotation Matrix Arising in Computer Vision Problems. http://arxiv.org/abs/0904.1613v1, 4.

PINKALL, U., AND POLTHIER, K. 1993. Computing Discrete Minimal Surfaces and Their Conjugates. *Experiment. Math. 2*, 1, 15–36.

RANKIN, C. C., AND BROGAN, F. A. 1986. An Element Independent Corotational Procedure for the Treatment of Large Rotations. *ASME J. Press. Valve Techn. 108*, 2, 165–174.

SCHMEDDING, R., AND TESCHNER, M. 2008. Inversion Handling for Stable Deformable Modeling. *Vis. Comp. 24*, 7-9 (CGI 2008 Special Issue), 625–633.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible Surface Modeling. In *Proc. Symp. Geom. Proc.*, 109–116.

A Matrix Facts

Inner products of 2D and 3D matrices are given by $\langle A,B\rangle = \mathrm{tr}(A^T B)$ with the L_2-norm $|A|^2 = \langle A,A\rangle$ used to measure distances between matrices $\mathrm{dist}(A,B) = |A-B|$. All matrices with positive determinant possess a unique (right) polar decomposition $A = RY$ into a rotation R and a symmetric $Y = (A^T A)^{1/2}$. This R is closest to A in the L_2-norm. This follows from $|RY-R|^2 = |Y-I|^2$ and the L_2 orthogonality of symmetric and anti-symmetric matrices, the latter forming the tangent space to $SO(n)$ at the identity.

Inner products involving anti-symmetric matrices can be greatly simplified. Consider $\langle B,A_X\rangle$ for arbitrary B and anti-symmetric A_X. The result depends only on the anti-symmetric part of B, $B_X = \frac{1}{2}(B-B^T)$ due to the orthogonality of symmetric and anti-symmetric matrices. In the 2D case, anti-symmetric matrices A_X are multiples of J, the rotation by $\pi/2$, $A_X = aJ$. We will denote the extraction of the representative a as $\mathrm{X}(A_X) = a$ and more generally use $\mathrm{X}(B) = b$ for extracting the representative of the anti-symmetric part of an arbitrary B. Hence $\langle B,A_X\rangle = 2\,\mathrm{X}(B)\mathrm{X}(A_X)$. In 3D anti-symmetric matrices can be written as cross products with suitable vectors a, $A_X = a\times$. Again we use the notation $\mathrm{X}(A_X) = a$ for "pulling out" the representative, this time a 3-vector. Consequently $\langle B,A_X\rangle = 2\langle \mathrm{X}(B),\mathrm{X}(A_X)\rangle$. We summarize this by writing $\langle B,A_X\rangle = 2\langle \mathrm{X}(B),\mathrm{X}(A_X)\rangle$ in both the 2D and 3D cases.

The representative of the anti-symmetric part of a map could be computed by first finding the corresponding matrix representation and then "extracting" the representative as above. Alternatively, and more simply, it can be computed directly from the p (domain) and q (range) data as the discrete curl of q. For a given triangle f_{ijk} we get

$$c_{ijk} = \langle \tfrac{1}{2}(q_i+q_j), p_{ij}\rangle + \langle \tfrac{1}{2}(q_j+q_k), p_{jk}\rangle + \langle \tfrac{1}{2}(q_k+q_i), p_{ki}\rangle$$

as the boundary integral (using Stokes' theorem) of the corresponding piecewise linear interpolant. For maps over 2D triangles we then get

$$\mathrm{X}(A) = \frac{c_{ijk}}{2a_{ijk}},$$

i.e., the discrete curl is normalized by the area a_{ijk} of the triangle p_{ijk} and $\mathrm{X}(A)$ becomes its density. For maps over tetrahedra the 3-vector representative of the anti-symmetric part of the map is also given as a density, this time normalized by the tetrahedron volume v_{ijkl}

$$\mathrm{X}(A) = \frac{1}{6v_{ijkl}}(c_{kli}p_{ij} + c_{lji}p_{ik} + c_{ijk}p_{il}).$$

This expression uses the fact that discrete curls are divergence free, *i.e.*, they must sum to zero over the boundary of a tetrahedron. Hence four discrete curl values on the faces of a tetrahedron together with the divergence free constraint uniquely fix the discrete curl 3-vector $\mathrm{X}(A)$ associated to the tetrahedron as a whole.

B Second Variation

To compute the second variation of Eq. (1) we need a formula for $\langle dg, \delta_h R\rangle = \langle R^T dg, R^T \delta_h R\rangle$. Noting that $R^T\delta_h R$ is anti-symmetric this reduces to $\langle \mathrm{X}(R^T dg), \mathrm{X}(R^T\delta_h R)\rangle$. $\mathrm{X}(R^T dg)$ can be computed directly from knowledge of the back rotated differential dg of the variation g (see above). All that is needed is an expression for $\mathrm{X}(R^T\delta_h R)$.

With $df = RY$ we have $dh = \delta_h df = \delta_h RY + R\delta_h Y$ and thus $R^T dh = R^T\delta_h RY + \delta_h Y$. In order to get a useful expression for $R^T\delta_h R$ we consider the representative of the anti-symmetric part of $R^T dh$

$$\mathrm{X}(R^T dh) = \mathrm{X}(R^T\delta_h RY) = \tfrac{1}{2}\mathrm{X}(\{R^T\delta_h R, Y\}),$$

where we used the symbol $\{.,Y\}$ for the anti-commutator $\{A,Y\} = AY + YA$, dropped $\delta_h Y$ since it is symmetric, and exploited the anti-symmetry of $R^T\delta_h R$ for the second equality.

Hence, if we can invert the anti-commutator on its first argument we get an expression for $R^T\delta_h R$. Indeed, the anti-commutator $\{.,Y\}$ has an inverse on the anti-symmetric matrices if Y is positive semi-definite and no worse than rank deficient by one. Denote this inverse by $W := \{.,Y\}^{-1}$. Treating it as a map between *representatives* of anti-symmetric matrices we get

$$\mathrm{X}(R^T\delta_h R) = 2W(\mathrm{X}(R^T dh))$$

In 2D, $W = \mathrm{tr}(Y)^{-1}$ and in 3D, $W = (\mathrm{tr}(Y)I - Y)^{-1}$. The former follows from simply writing out the expressions while the latter is easily checked for diagonal Y with at least two positive diagonal entries. Validity for general symmetric Y then follows after multiplying the left and right sides with the appropriate coordinate transformations.

This gives the final representation

$$\langle dg, \delta_h R\rangle = 4\left\langle \mathrm{X}(R^T dg), \mathrm{X}(R^T dh)\right\rangle_W,$$

where we treat the intervening W as a scalar product weighting.