Repulsive Shells

JOSUA SASSEN, University of Bonn, Germany and École normale supérieure Paris-Saclay, France
HENRIK SCHUMACHER, Chemnitz University of Technology, Germany and University of Georgia, USA
MARTIN RUMPF, University of Bonn, Germany
KEENAN CRANE, Carnegie Mellon University, USA

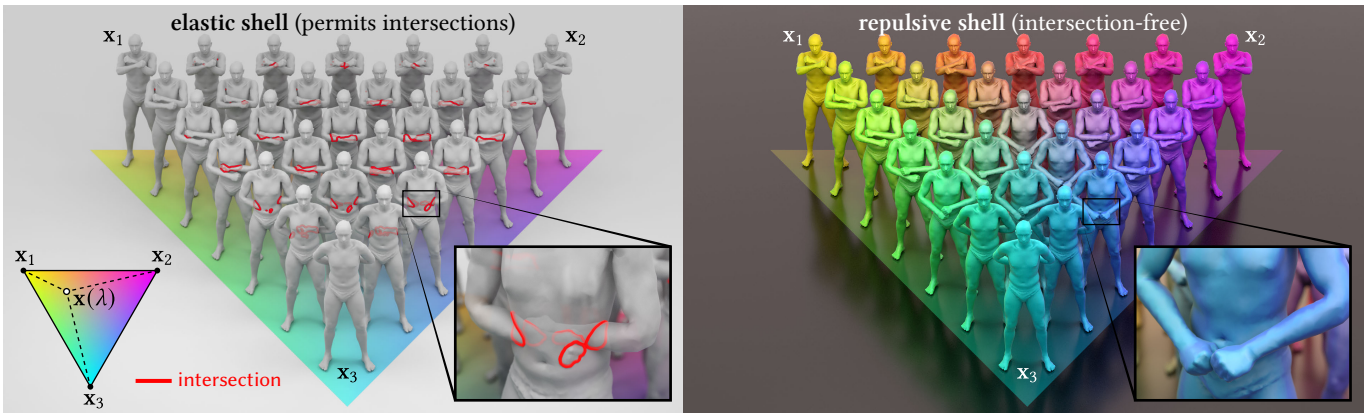


Fig. 1. *Left:* using an elasticity-based prior for shape interpolation provides natural deformation, but can also yield severe self-intersection. *Right:* we develop a shape space where tasks like interpolation, extrapolation, and averaging are naturally intersection-free. Here we show geodesic barycentric interpolation of poses $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$. Note that no rig nor additional information is needed—only the three poses, given as three sets of vertex coordinates on a common mesh.

This paper develops a shape space framework for collision-aware geometric modeling, where basic geometric operations automatically avoid interpenetration. Shape spaces are a powerful tool for surface modeling, shape analysis, nonrigid motion planning, and animation, but past formulations permit nonphysical intersections. Our framework augments an existing shape space using a repulsive energy such that collision avoidance becomes a first-class property, encoded in the Riemannian metric itself. In turn, tasks like intersection-free shape interpolation or motion extrapolation amount to simply computing geodesic paths via standard numerical algorithms. To make optimization practical, we develop an adaptive collision penalty that prevents mesh self-intersection, and converges to a meaningful limit energy under refinement. The final algorithms apply to any category of shape, and do not require a dataset of examples, training, rigging, nor any other prior information. For instance, to interpolate between two shapes we need only a single pair of meshes with the same connectivity. We evaluate our method on a variety of challenging examples from modeling and animation.

Authors' addresses: Josua Sassen, josua.sassen@uni-bonn.de, University of Bonn, Endenicher Allee 60, Bonn, Germany, 53115, École normale supérieure Paris-Saclay, 4 Avenue des Sciences, Gif-sur-Yvette, France, 91190; Henrik Schumacher, henrik.schumacher@mathematik.tu-chemnitz.de, Chemnitz University of Technology, Chemnitz, Germany, 09107, University of Georgia, 200 D. W. Brooks Drive, Athens, USA, GA, 30602; Martin Rumpf, martin.rumpf@uni-bonn.de, University of Bonn, Endenicher Allee 60, Bonn, Germany, 53115; Keenan Crane, kmcrane@cs.cmu.edu, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, USA, PA, 15213.

0730-0301/2024/7-ART140
https://doi.org/10.1145/3658174

CCS Concepts: • **Computing methodologies** → **Shape modeling**; • **Mathematics of computing** → **Mathematical optimization**; **Quadrature**.

Additional Key Words and Phrases: Shape spaces, nonrigid deformation

ACM Reference Format:
Josua Sassen, Henrik Schumacher, Martin Rumpf, and Keenan Crane. 2024. Repulsive Shells. *ACM Trans. Graph.* 43, 4, Article 140 (July 2024), 22 pages. https://doi.org/10.1145/3658174

1 INTRODUCTION

"An ounce of prevention is worth a pound of cure."
—Benjamin Franklin

The status quo in geometric computing is a bit odd: even though solid objects cannot pass through themselves, nor easily change global topology, decades of algorithms build on shape representations that permit self-intersection (*e.g.*, Bézier curves or triangle meshes) or freely allow topological changes (*e.g.*, level set [Osher and Fedkiw 2005] or density-based encodings [Mildenhall et al. 2021]). The rationale, of course, is that computation is much cheaper if one need not detect or resolve intersections. Yet using such representations for shape synthesis or analysis is bound to yield unsatisfactory results—and misleading conclusions. This paper provides a repulsion-based framework for ensuring that basic geometric operations are automatically intersection-free.

Collisions are of course well-studied in physical simulation, but these techniques address only one specific class of problems: detection and resolution of collisions in direct response to local contact. In broader geometric computing there are other problems where

one must actively avoid the influence of self-intersecting states. For instance, when studying statistics of nonrigid shapes (*e.g.*, in computational anatomy [Miller et al. 1997]), one would like a notion of "average" that yields valid, non-intersecting configurations (Figure 1). Likewise, when planning motion trajectories for deformable bodies (*e.g.*, in soft robotics [Fairchild et al. 2021]), one might aim to proactively deform a surface in anticipation of obstacles (Figure 4), rather than reactively slamming on the breaks at the moment of impact. The *shape space* perspective [Younes 2010], provides a principled starting point for these broader tasks.

In this paper we specifically develop a Riemannian shape space where each point corresponds to an intersection-free *embedded* surface. To our knowledge, all past work instead considers spaces of *immersions*, which permit intersections. The basic idea is to augment an existing shape space with a repulsive potential that penalizes configurations in near-contact. A critical observation is that we cannot merely *add* this repulsive penalty to the path energies used for optimization—which yields undesirable behavior (Figure 3). Instead, we form what we call a *graph manifold* over the original space (Figure 2), yielding more natural trajectories (Figure 15). To ensure good numerical behavior, we also develop spatially adaptive quadrature for our repulsive potential that both prevents collisions on coarse meshes, and converges to a meaningful continuous potential under spatial refinement (Section 5).

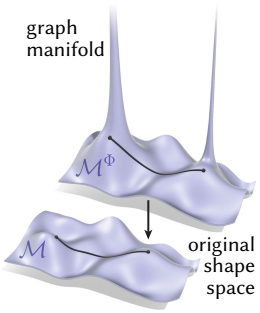


Fig. 2. Cartoon of our basic approach: by "graphing" a repulsive potential Φ over an existing shape space $\mathcal{M}$, we get a *graph manifold* $\mathcal{M}^\Phi$ where shape trajectories are automatically collision-free.

1.1 Outline

We begin in Section 2 with a simple "toy" model—a space of repulsive *points*—which helps illustrate our approach, unencumbered by details about the surface case. We then define surface energies in Section 3, and introduce the full-blown space of repulsive shells in Section 4. Section 5 develops our novel *adaptive TPE* potential, Section 6 describes numerical optimization, and Section 7 presents a variety of results. Finally, in Section 8, we discuss limitations of our approach and opportunities for future improvement.

1.2 Related Work

1.2.1 Nonrigid Registration, Motion Planning, and Modeling. From an application perspective our method connects to a large body of work from robotics, animation, and computational anatomy on, *e.g.*, nonrigid registration, interpolation, modeling, and motion planning—far too vast to cover in detail. In general, there are two basic strategies for incorporating collision avoidance into such tasks.

One is to apply a globally injective (or more properly, *diffeomorphic*) deformation to all of space, precluding new intersections between embedded submanifolds. This class includes methods that apply an elastic barrier to a bulk medium [Ball 1981], and those that directly discretize the space of diffeomorphisms [Beg et al. 2005]. A

Fig. 3. Simply penalizing the total repulsive potential over a time-varying trajectory yields undesirable deformation: between the fixed start/end configurations, the surface "explodes" away from itself to reduce potential.

corresponding volumetric discretization must have sufficient resolution to capture surface-surface interactions, which can come at the cost of expensive dynamic re-meshing [Müller et al. 2015; Jiang et al. 2017], or lead to corruption of embedded geometry (see Figure 19). Most importantly, although this approach formally precludes intersections (purely by virtue of preserving global topology), it provides no mechanism for proactively *avoiding* contact—nor using proximity to drive motion planning or shape modeling.

A second class of methods adapts numerical integrators for collision detection and response to broader planning and motion synthesis tasks. For instance, elastodynamic models have been used to guide classic (*e.g.*, tree-based) planning algorithms [Gayle et al. 2005b], to drive optimal control frameworks [Wojtan et al. 2006; Barbič et al. 2009; Coros et al. 2012; Du et al. 2021; Li et al. 2022], to apply collision detection and response during shape modeling [Harmon et al. 2011; Fang et al. 2021], and to find collision-free displacements along a predetermined trajectory [Gayle et al. 2005a; Rodriguez et al. 2006; Bergou et al. 2007; Moss et al. 2008]. Since these methods are rooted in dynamics and forward time integration, collision avoidance is largely reactive rather than proactive, *i.e.*, forces are localized in space and time around points of (near-)collision. Tasks like shape interpolation are difficult to handle in a symmetric way, due to the irreversible dynamics. More broadly, this class of methods does not address the richer tools provided by shape spaces, such as averaging multiple shapes, or performing deformation/motion transfer.

1.2.2 Shape Spaces. Our primary focus is on extending the shape space machinery to incorporate collision avoidance via a *graph manifold* construction (Figure 5). Surprisingly little work treats collision avoidance in the shape space context—especially for deformable geometry. Geometric and computational mechanics largely consider *contact*, modeled via *differential inclusions* [Blagodatskikh and Filippov 1985; Fetecau 2003] or *normal cones* [Moreau 1988; Kaufman et al. 2005]—neither of which provide a "nice" (*i.e.*, globally differentiable) Riemannian picture. Some work considers collision-avoiding points on finite-dimensional manifolds, akin to our didactic example in Section 2.1. For example, both [Assif et al. 2018, Section III] and [Klein et al. 2023] integrate a total repulsive energy over time—but as seen in Figures 3 and 7, this approach yields undesirable strong artificial expansion, motivating the graph manifold approach.

For triangle meshes, shape spaces were first studied by Kilian et al. [2007]; our work builds specifically on the space of *discrete shells*

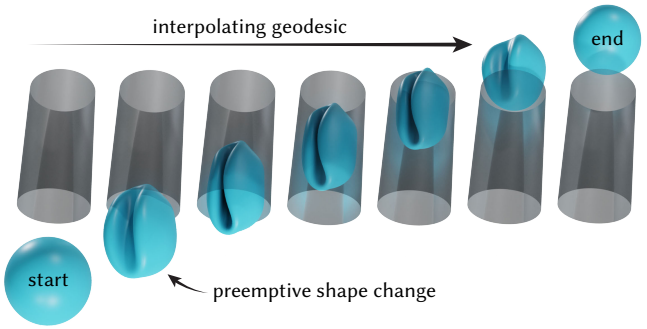


Fig. 4. In contrast to forward dynamical simulation, where shapes deform in response to impact, geodesics in our shape space preemptively deform geometry to avoid collisions. Here for instance, a spherical shell "folds up" in anticipation of squeezing through a narrow tube.

defined by Heeren et al. [2012, 2014]. Again, however, these spaces consider only immersed rather than embedded surfaces. Closer to our approach, Herzog and Loayza-Romero [2022] define a metric on the space of planar triangular meshes, but do not consider physics-based functionals suitable for shape/motion interpolation, and treat only intersections at the boundary of a planar mesh—not easily generalized to curved surfaces. Finally, Zhu et al. [2017] use a no-overlap constraint to achieve intersection-free interpolation of planar shapes, with similar limitations.

1.2.3 Repulsive Potentials. Though our *graph manifold* construction can adopt any repulsive potential, we found it essential to model long-range repulsion, rather than just local collision response (detailed below). Long-range forces help to globally steer geometry away from contact, rather than locally deforming it only near the place and time of collision. Numerically, we found it important to use a potential that both prevents intersections on coarse meshes (Figure 12), and converges to a well-defined potential under refinement (Figure 20). Convergence is especially helpful for ensuring that coarse solutions are predictive of fine results (Section 6.4).

Collision Potentials. Potentials designed for collision response have produced impressive results in recent years—especially in close-contact scenarios [Li et al. 2020, 2021; Lan et al. 2021]. Fundamentally, such potentials are a numerical device for enforcing inequality constraints that characterize non-intersecting states—akin to *log barrier methods* [Boyd and Vandenberghe 2004, Section 11.2]. In practice, unfortunately, potentials like *incremental potential contact (IPC)* are unsatisfactory for shape space tasks, as they do not provide sufficient long-range guidance (Section 7.2.2). Moreover, in the continuous setting, a log barrier integrated over surface patches can remain finite even as they are pushed toward one another.

Long-Range Potentials. Long-range repulsive potentials for embedded curves and surfaces have a long history in geometric topology, prompted by questions from knot theory [Fukuhara 1988; O'Hara 1991; Buck and Orloff 1995; Kusner and Sullivan 1998]. The basic idea is to mimic the Coulomb potential between electrostatically charged particles x and y, *i.e.*, $1/|x-y|^\alpha$ for some falloff parameter $\alpha > 0$. However, integrating this kernel over all pairs of points x, y on an n-manifold M does not yield a useful energy: the integral is not well-defined (infinite) for $\alpha \geq n$, and yet too weak to prevent intersections (*i.e.*, finite) for $\alpha \leq 2n$. Naïvely repelling all pairs of mesh vertices hence yields awful numerical behavior, as seen in [Yu et al. 2021a, Figure 2]. In response, several "desingularized" energies have been developed—for a brief history, see Yu et al. [2021b, Section 3.1] and Yu et al. [2021a, Section 1]. We specifically use the *tangent-point energy (TPE)* [Buck and Orloff 1995; Gonzalez and Maddocks 1999; Banavar et al. 2003; Strzelecki and von der Mosel 2013; Bartels et al. 2022], defined in Section 3.2.1 for surfaces.

Discretization. In the continuous setting, TPE provides an infinite barrier against self-intersection (for surfaces without boundary). However, this behavior does not automatically carry over to meshes, where a careless discretization can "miss" important singularities in the integrand. We hence develop a novel *adaptive TPE* scheme (Section 5) which closely approximates the continuous energy. To our knowledge, this scheme is the first to both (i) reliably prevent collisions on coarse meshes and (ii) converge to a meaningful energy under spatial refinement. For instance, the midpoint TPE scheme of Yu *et al.* can miss key singularities [Yu et al. 2021a, Section 10], whereas IPC does not correspond to a continuous potential. However, we stress emphatically that we do not offer any rigorous no-intersection guarantee in the discrete case—a question of detailed analysis left to future work.

Finally, a common misconception is that an all-pairs potential like TPE must be significantly slower to evaluate than a local potential like IPC. However, this turns out not to be true in practice. By applying the *fast multipole method* (Section 5), which itself incurs little overhead, the cost of using TPE is dominated by a small number of near-field interactions. For instance, in a typical scenario far-field interactions account for only about 5% of overall cost—and in many examples TPE is even *cheaper* to evaluate than IPC (Figure 16).

1.3 Notation

We use $\langle \cdot, \cdot \rangle$ for the standard inner product on $\mathbb{R}^3$, and use $\overline{\mathbb{R}} := \mathbb{R}_{\geq 0} \cup \{\infty\}$ for the (positively) extended real numbers. We use upright d to denote the differential of any function f, *i.e.*, $\mathrm{d}_x f$ is the differential of f at position x; in the finite-dimensional case, we sometimes represent the differential by the Jacobian matrix J_f. Throughout we use regular letters (*e.g.*, x) to indicate spatial coordinates in $\mathbb{R}^n$, and bold letters (*e.g.*, $\mathbf{x}$) to indicate points in a configuration space $\mathcal{M}$.

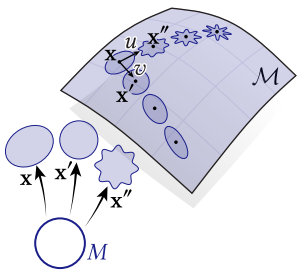


Fig. 5. For a fixed domain M such as the circle *(bottom left)*, each configuration $\mathbf{x}\colon M \to \mathbb{R}^n$ can be viewed as a point in a shape space $\mathcal{M}$ *(top right)*. A choice of metric g on $\mathcal{M}$ defines the cost of moving from one configuration to another. Here for example we might define g so that $\|u\|_g$ is larger than $\|v\|_g$.

2 SHAPE SPACES

In general, a *shape space* describes all the different ways a domain of fixed topology can be embedded in $\mathbb{R}^n$, as well as the cost of moving around in this space. For instance, motions that induce significant bending or stretching might be considered more costly than those that merely translate or rotate an object. More precisely, for a fixed domain M, a *configuration*

manifold $\mathcal{M}$ consists of all maps $\mathbf{x}\colon M \to \mathbb{R}^n$ of a certain class, *i.e.*, all ways of assigning a specific geometry to M. A tangent vector $u \in T_{\mathbf{x}}\mathcal{M}$ at a point $\mathbf{x}$ then describes an infinitesimal change to the embedded shape, *i.e.*, a velocity at each point of M (Figure 5).

A *shape space* $(\mathcal{M}, g)$ associates a carefully-designed Riemannian metric g to $\mathcal{M}$, which defines the cost of small, infinitesimal deformations. In particular, this metric is a smoothly-varying inner product $g_{\mathbf{x}}\colon T_{\mathbf{x}}\mathcal{M} \times T_{\mathbf{x}}\mathcal{M} \to \mathbb{R}$. The quantity $\|u\|_g \coloneqq \sqrt{g_{\mathbf{x}}(u,u)}$ hence assigns a notion of length to any vector $u \in T_{\mathbf{x}}\mathcal{M}$. One can use this metric to find the shortest interpolating trajectory between two shapes—or perform a number of other tasks (*e.g.*, extrapolation, averaging, or deformation transfer [Heeren et al. 2014]). For instance, the overall cost of a trajectory $\mathbf{x}\colon [0,1] \to \mathcal{M}$ can be measured via the *path energy*

$$\mathcal{E}(\mathbf{x}(t)) \coloneqq \int_0^1 g_{\mathbf{x}(t)}(\dot{\mathbf{x}}(t), \dot{\mathbf{x}}(t))\, \mathrm{d}t. \tag{1}$$

The path energy bounds the usual length $\mathcal{L}(\mathbf{x}(t)) \coloneqq \int_0^1 \|\dot{\mathbf{x}}(t)\|_g \, \mathrm{d}t$ from above, *i.e.*, $\mathcal{L}^2(\mathbf{x}(t)) \leq \mathcal{E}(\mathbf{x}(t))$ for all paths $\mathbf{x}(t)$. Critical points of path energy coincide with *geodesics* that not only locally minimize length, but also have constant speed $\|\dot{\mathbf{x}}(t)\|_g$ for all t. The *geodesic distance* $\mathrm{dist}_g(\mathbf{x}, \mathbf{y})$ gives the *globally* shortest length of any path between $\mathbf{x}$ and $\mathbf{y}$. More explicitly, if $\mathcal{C}$ is the set of continuously differentiable paths $\mathbf{x}\colon [0,1] \to \mathcal{M}$ with endpoints $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{x}(1) = \mathbf{x}_1$, then

$$\mathrm{dist}_g(\mathbf{x}_0, \mathbf{x}_1) \coloneqq \inf_{\mathbf{x} \in \mathcal{C}} \mathcal{L}(\mathbf{x}(t)).$$

Our Approach. Suppose we have an existing shape space, and want to incorporate collision avoidance. Our basic approach is to augment the original metric, using a repulsive potential to define a so-called *graph manifold*, as detailed in Section 2.3. Encoding collision avoidance in the metric itself provides a *repulsive shape space* on top of which we can build broader geometric computing tasks, using standard numerical methods from the shape space literature (*e.g.*, performing motion extrapolation by following the exponential map, or averaging shapes by computing a *Karcher mean*).

2.1 Warm-up: Repulsive Points

Consider a collection of m points in the 2D Euclidean plane. From the shape space perspective, our domain is then the set $M = \{1, \ldots, m\}$, and our configuration space is the manifold $\mathcal{M} = (\mathbb{R}^2)^m$. Each configuration is then a map $\mathbf{x}\colon M \to (\mathbb{R}^2)^m$ that assigns 2D coordinates to each of our elements in M. We can of course encode this map as a stacked vector of individual coordinates $\mathbf{x} = (x_1, \ldots, x_m)$. A trajectory of all points over time is then described by a curve $\mathbf{x}(t)\colon [0,1] \to \mathcal{M}$, and the velocities of each point at any time $t \in [0,1]$ are given by a tangent vector $\dot{\mathbf{x}}(t) \in T_{\mathbf{x}(t)}\mathcal{M}$ (again, just a column vector in $(\mathbb{R}^2)^m$).

Which trajectories should be considered "good?" If we want to expend the least effort getting from one configuration to another, then a natural choice is to penalize any motion whatsoever by letting our metric g be a simple dot product

$$g_{\mathbf{x}}(u, v) \coloneqq \sum_{k=1}^{m} u_k \cdot v_k. \tag{2}$$

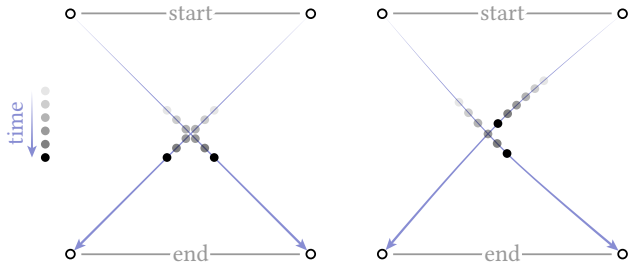


Fig. 6. For points in the plane, simply minimizing path length yields trajectories that collide *(left)*, whereas our graph manifold approach naturally avoids collision while remaining as straight as possible *(right)*.

For this choice of metric, minimizers of the path energy $\mathcal{E}$ (Equation 1) yield constant-speed straight-line trajectories for each point. However, these points can come arbitrarily close and might collide (Figure 6, *left*), which we must avoid somehow.

2.2 Penalty Formulation

A naïve first attempt at avoiding collision is to add a repulsive potential to the path energy $\mathcal{E}$, such as the *Coulomb potential*

$$\Phi(\mathbf{x}) \coloneqq \sum_{\substack{i=1 \\ i \neq j}}^{m} \sum_{j=1}^{m} \frac{1}{|x_i - x_j|}. \tag{3}$$

Doing so yields an augmented path energy

$$\widetilde{\mathcal{E}}^{\Phi}(\mathbf{x}) \coloneqq \int_0^1 g_{\mathbf{x}(t)}(\dot{\mathbf{x}}(t), \dot{\mathbf{x}}(t))\mathrm{d}t + \beta \int_0^1 \Phi(\mathbf{x}(t))\, \mathrm{d}t, \tag{4}$$

where $\beta > 0$ is a parameter controlling the strength of repulsion. However, minimizers of this energy fail to satisfy even some very basic properties: for instance, if the initial and final configurations are related by a constant offset $c \in \mathbb{R}^2$, the minimizer will not be a simple translation, but will rather strongly expand in the middle of the trajectory (Figure 7, *top*). This same phenomenon occurs with the repulsive shell space defined in Section 4 (see especially Figure 3) where, again, configurations are displaced far more than necessary in order to reduce the repulsive potential.

2.3 Graph Manifold

A natural alternative, and the key idea behind our formulation, is to ask that the *change* in repulsive potential is as small as possible throughout the trajectory: if points start out close together, they may remain close together—but should still be prohibited from colliding. In the Riemannian picture, we use a repulsive potential $\Phi\colon \mathcal{M} \to \overline{\mathbb{R}}$ to define an augmented shape space where configurations naturally avoid collision. In particular, we define what we call a *graph manifold* by appending the potential value $\Phi(\mathbf{x})$ to each point $\mathbf{x} \in \mathcal{M}$, effectively embedding our original shape space in a space one dimension higher:

$$\mathcal{M}^{\Phi} \coloneqq \{(\mathbf{x}, \Phi(\mathbf{x})) \mid \mathbf{x} \in \mathcal{M}\} \subset \mathcal{M} \times \overline{\mathbb{R}}.$$

Intuitively, configurations with collisions now correspond to "infinitely tall mountains" that we do not wish to climb (*e.g.*, Figure 8, *top*).

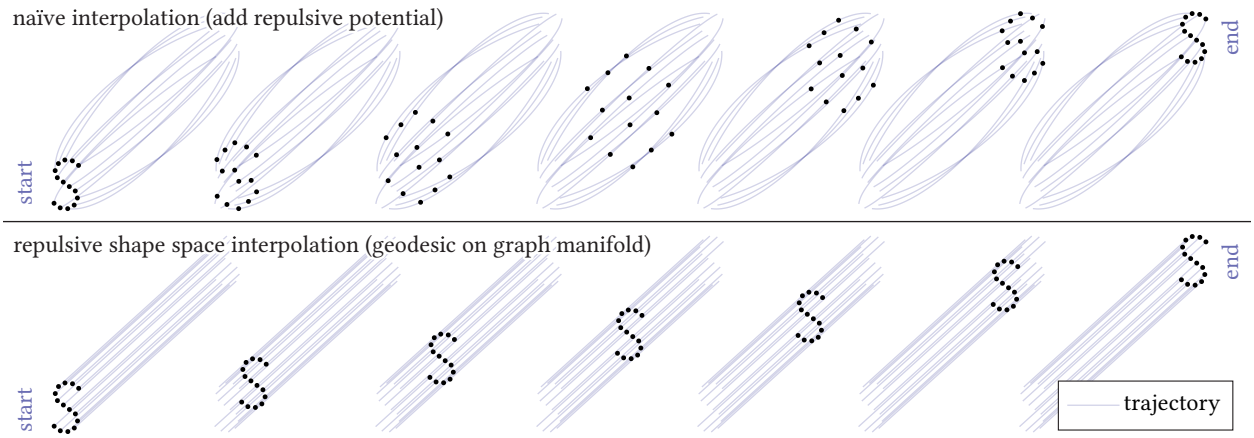


Fig. 7. Simply penalizing the total repulsive energy along a trajectory leads to undesirable behavior—here, the point configuration "explodes" in the middle *(top)*, rather than just translating *(bottom)*.

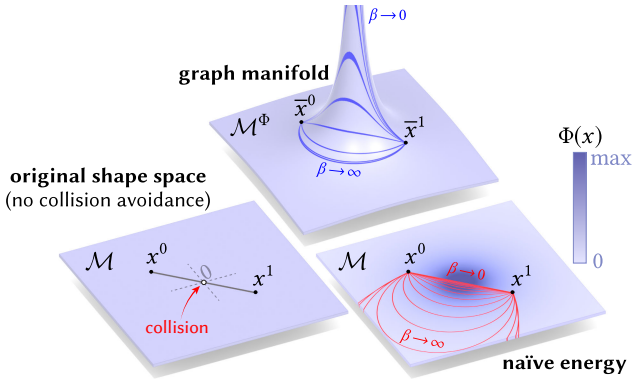


Fig. 8. The shape space for two repulsive points x, y where $x = (x^0, x^1)$, and y is fixed to the origin $(0, 0)$. *Bottom left:* if we penalize only local motions, collisions can occur. *Bottom right:* adding a repulsive potential to our cost function avoids collisions, but can take us far from the ideal trajectory. *Top center:* we instead find geodesics on the graph of the repulsive potential, which yields collision-avoiding motion close to the ideal trajectory. In both cases we plot trajectories for several values of the strength parameter β.

Velocities are in turn vectors tangent to this mountain range—in particular, the tangent spaces of the graph manifold are given by

$$T_{(\mathbf{x},\Phi(\mathbf{x}))}\mathcal{M}^\Phi = \{(u, \mathrm{d}_\mathbf{x}\Phi(u)) \mid u \in T_\mathbf{x}\mathcal{M}\},$$

where at each point $\mathbf{x} \in \mathcal{M}$ the directional derivative $\mathrm{d}_\mathbf{x}\Phi(u)$ describes the change in potential Φ due to a infinitesimal motion u along the original shape space. We define the metric on $\mathcal{M}^\Phi$ as

$$\overline{g}^\Phi_{(\mathbf{x},\Phi(\mathbf{x}))}\left((u,s),(v,t)\right) := g_\mathbf{x}(u,v) + \beta\, s\, t,$$

for $(u, s), (v, t) \in T_{(\mathbf{x},\Phi(\mathbf{x}))}\mathcal{M}^\Phi$ where β controls the strength of repulsion (as in Equation 4). We can also express this metric with respect to the original configuration space $\mathcal{M}$, namely

$$g^\Phi_\mathbf{x}(u,v) := g_\mathbf{x}(u,v) + \beta\, \mathrm{d}_\mathbf{x}\Phi(u)\, \mathrm{d}_\mathbf{x}\Phi(v). \tag{5}$$

Importantly, by expressing the metric on $\mathcal{M}$, we avoid working with a submanifold $\mathcal{M}^\Phi$ of $\mathcal{M} \times \mathbb{R}$, and need only solve *unconstrained* optimization problems.

The final metric g^Φ captures the same cost as in the original shape space, *plus* the cost of increasing—or decreasing—the repulsive potential Φ. Geodesic paths in $\mathcal{M}^\Phi$ hence try to maintain a roughly constant repulsive potential, while still keeping as straight as possible with respect to the original metric g. Importantly, g^Φ (Equation 5) is well-defined only in regions of $\mathcal{M}$ where Φ is finite—but as shown below, these are the only regions we ever need to consider.

The path energy on our graph manifold is now just the usual path energy $\mathcal{E}$ (Equation 1), but with the original metric g replaced by the graph metric g^Φ. For our toy example, minimizers of path energy now exhibit the expected behavior, like translational invariance (Figure 7, *bottom*). The parameter β controls the relative importance of collision avoidance (Figure 8, *top center*). Most importantly, the fact that a potential Φ tends toward infinity for self-intersecting configurations carries over to our distance function dist_{g^Φ}: To move any point $\mathbf{x} \in \mathcal{M}$ of finite potential $\Phi(\mathbf{x}) < \infty$ to a point $\mathbf{y}^*$ on the boundary of the feasible region (*i.e.*, to a point where $\Phi = \infty$), one must travel an infinite distance, *i.e.*, $\mathrm{dist}_{g^\Phi}(\mathbf{x}, \mathbf{y}) \to \infty$ as $\mathbf{y} \to \mathbf{y}^*$.

To see this, denote by $\mathcal{C}$ the set of continuously differentiable curves $c\colon [0,1] \to \mathcal{M}$ satisfying $c(0) = x$ and $c(1) = y$. We use the definition of the Riemannian distance and see

$$\begin{aligned}
\mathrm{dist}_\Phi(x,y) &:= \inf_{c\in\mathcal{C}} \int_0^1 \sqrt{g(\dot c(t),\dot c(t)) + (\mathrm{d}_{c(t)}\Phi\,\dot c(t))^2}\,\mathrm{d}t \\
&\geq \inf_{c\in\mathcal{C}} \int_0^1 \sqrt{(\mathrm{d}_{c(t)}\Phi\,\dot c(t))^2}\,\mathrm{d}t \\
&= \inf_{c\in\mathcal{C}} \int_0^1 |\mathrm{d}_{c(t)}\Phi\,\dot c(t)|\,\mathrm{d}t \geq |\Phi(y) - \Phi(x)|.
\end{aligned}$$

From the last estimate, it immediately follows that $\mathrm{dist}_{g^\Phi}(x,y) = \infty$ if $\Phi(y) = \infty$. In other words, beyond merely ensuring that invalid states $\mathbf{x}$ have infinite energy, we have made certain that geodesic paths in our repulsive shape space cannot reach invalid states in

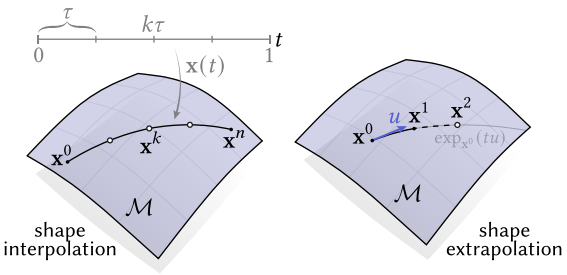


Fig. 9. The change in shape over time is discretized by a sequence of points $\mathbf{x}^0, \ldots, \mathbf{x}^n \in \mathcal{M}$. Interpolation is achieved by holding the endpoints fixed and minimizing path energy *(left)*, whereas extrapolation is achieved by solving for the next point that continues a geodesic *(right)*.

finite time. Since the proof does not depend on the particular potential Φ, it will remain true for the repulsive shell space defined in Section 4.

2.4 Discrete Geodesics

We can approximate a path in shape space by a sequence of points in the configuration space $\mathcal{M}$. For instance, in the case of repulsive points, our path is encoded by configurations $\mathbf{x}^0, \ldots, \mathbf{x}^n \in (\mathbb{R}^2)^m$. Here and throughout superscripts index time. Our basic computational task is then to find paths that approximate geodesics with respect to g^Φ. In particular, we can either *interpolate* between two given poses (Section 2.4.2), or *extrapolate* an initial velocity to get a longer motion (Section 2.4.3). For operations beyond basic geodesic interpolation and extrapolation (such as shape averaging or spline interpolation), our repulsive shape space is compatible with the computational machinery found in past work on shape spaces [Heeren et al. 2014, 2016, 2018; von Tycowicz et al. 2015; Sassen et al. 2020b].

2.4.1 Discrete Path Energy. To perform either task, we must first discretize the path energy from Equation 1. Here we follow the general approach from Rumpf and Wirth [2015], and, for any sequence of points $\mathbf{x}^0, \ldots, \mathbf{x}^n$, consider a piecewise geodesic $\mathbf{x}\colon [0,1] \to \mathcal{M}$ interpolating the points $\mathbf{x}^k$ at regular intervals, *i.e.*, $\mathbf{x}(k\tau) = \mathbf{x}^k$, where $\tau := 1/n$. Each segment then has constant absolute speed $c_k := \operatorname{dist}(\mathbf{x}^{k-1}, \mathbf{x}^k)/\tau$, and the overall path energy is

$$\int_0^1 \|\dot{\mathbf{x}}(t)\|^2_{g^\Phi}\, dt = \sum_{k=1}^n \tau |c_k|^2 = \sum_{k=1}^n \operatorname{dist}^2_\Phi(\mathbf{x}^{k-1}, \mathbf{x}^k)/\tau.$$

Since $\tau = 1/n$, we hence define the *discrete path energy* as

$$\widehat{\mathcal{E}}(\mathbf{x}) := n \sum_{k=1}^n \operatorname{dist}^2_\Phi(\mathbf{x}^{k-1}, \mathbf{x}^k). \tag{6}$$

In the case of repulsive points, the geodesic distance is easy to approximate: the usual Euclidean distance provides the exact distance along the original configuration space $\mathcal{M}$, and a simple squared difference of potential values (resulting from the combination of midpoint quadrature and difference quotients) provides a lower bound on the additional "vertical" distance along the graph manifold. Overall, then, we approximate $\operatorname{dist}^2_\Phi(\mathbf{x}, \mathbf{y})$ by

$$D^2_\Phi(\mathbf{x}, \mathbf{y}) := |\mathbf{x} - \mathbf{y}|^2 + (\Phi(\mathbf{x}) - \Phi(\mathbf{y}))^2. \tag{7}$$

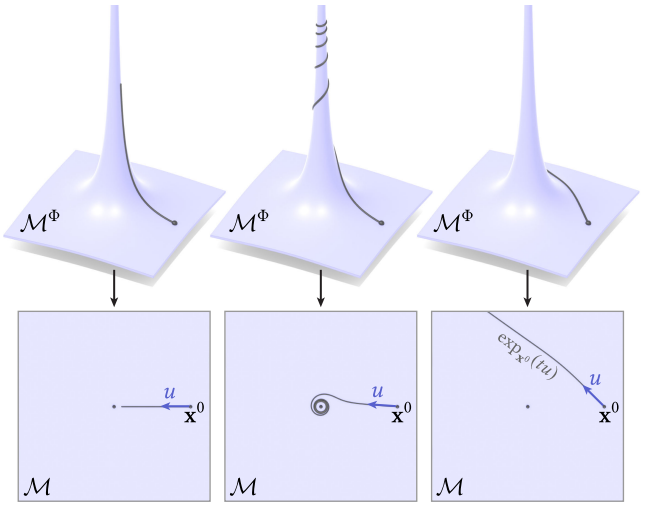


Fig. 10. The exponential map in our repulsive shape space is obtained by following a constant-speed geodesic on the graph manifold $\mathcal{M}^\Phi$, then projecting this geodesic back to the original configuration space $\mathcal{M}$. Here we show trajectories for a pair of repulsive points, where one point is fixed to the origin. Notice in particular that it takes infinite time to reach a collision *(bottom left)*, which is equivalent to climbing an infinitely tall peak *(top left)*.

For shells (Section 3), though we can no longer explicitly evaluate configuration space distance, it remains easy to approximate—see Section 4.2.1.

Note that consistency of the discrete and continuous path energies follows from the general theory of time-discrete geodesic calculus developed by Rumpf and Wirth [2015]. In fact, under mild conditions on the metric g fulfilled for the models in this paper, minimizers of the discrete energy are guaranteed to converge to continuous geodesics. Here, the crucial step is to show that the Hessian $\mathrm{d}^2_{\mathbf{y}} D^2_\Phi(\mathbf{x}, \mathbf{y})|_{\mathbf{y}=\mathbf{x}}$ coincides with the metric $g^\Phi_{\mathbf{x}}$. This amounts to a calculation very similar to the computation of the path energy's Hessian in Section 6.1.

2.4.2 Geodesic Interpolation. To interpolate between two shape configurations, we can minimize the discrete path energy while holding the endpoints $\mathbf{x}^0$ and $\mathbf{x}^n$ fixed. Examples for repulsive points are shown in Figure 6, *right* and Figure 7, *bottom*. In contrast to the naïve penalty formulation from Section 2.2, geodesics on the graph manifold roughly preserve the closeness exhibited at the endpoints, typically leading to more natural interpolation (compare also Figures 3 and 15).

2.4.3 Geodesic Extrapolation. Suppose that instead we are given an initial configuration $\mathbf{x}^0 \in \mathcal{M}$ and velocity $u \in T_{\mathbf{x}^0}\mathcal{M}$, and wish to extrapolate this motion forward in time. In the discrete case, the first step can be approximated by just taking a small step of size $\tau > 0$, yielding a configuration $\mathbf{x}^1 := \mathbf{x}^0 + \tau u$. To get the next configuration, consider any three consecutive points $\mathbf{x}^{k-1}, \mathbf{x}^k, \mathbf{x}^{k+1}$ along a discrete geodesic. Then, the middle point $\mathbf{x}^k$ minimizes the discrete path energy $2(\operatorname{dist}^2_\Phi(\mathbf{x}^{k-1}, \mathbf{x}^k) + \operatorname{dist}^2_\Phi(\mathbf{x}^k, \mathbf{x}^{k+1}))$ (Equation 6). Hence, $\mathbf{x}^k$ satisfies the optimality condition

$$\mathrm{d}_{\mathbf{x}^k}(\operatorname{dist}^2_\Phi(\mathbf{x}^{k-1}, \mathbf{x}^k) + \operatorname{dist}^2_\Phi(\mathbf{x}^k, \mathbf{x}^{k+1})) = 0. \tag{8}$$

Now, given $\mathbf{x}^{k-1}$ and $\mathbf{x}^k$, we can then solve Equation 8 for the next, unknown configuration $\mathbf{x}^{k+1}$. Iterating this process defines a *discrete exponential map* $\exp_{\mathbf{x}^0}(tu)$ stepwise building up a discrete geodesic. Following Rumpf and Wirth [2015, Theorem 5.9], this discrete exponential map converges to the continuous one on $(\mathcal{M}, g^\Phi)$.

In the specific case of repulsive points, Equation 8 becomes

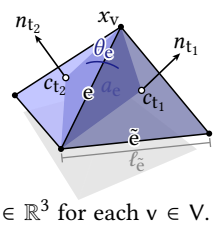
$$(\mathbf{x}^{k-1} - 2\mathbf{x}^k + \mathbf{x}^{k+1}) + (\Phi(\mathbf{x}^{k-1}) - 2\Phi(\mathbf{x}^k) + \Phi(\mathbf{x}^{k+1}))\, \mathrm{d}_{\mathbf{x}^k}\Phi = 0.$$

This equation is nonlinear in the unknown point $\mathbf{x}^{k+1}$, and can be solved using a root-finding method such as Newton's method. Figure 10 shows several examples for a pair of points; see Figure 27 for an example using elastic shells. Unlike interpolating geodesics, which try to roughly preserve the value of the repulsive potential Φ, the exponential map will increase, decrease, or maintain Φ along the trajectory $\exp_{\mathbf{x}^0}(tu)$, depending on the initial change in Φ.

3 SURFACE ENERGIES

We next define the elastic energy (Section 3.1), and the repulsive energy (Section 3.2) that will be used to construct our shape space of repulsive shells in Section 4. To develop algorithms well-behaved with respect to tessellation and mesh refinement, we start with a smooth formulation, then develop a principled discretization. Although we make here a specific choice of elastic energy and repulsive potential, our basic framework of the *graph manifold* is largely agnostic to these choices—see Section 7.2.2 for further discussion.

Notation. In the smooth case, we consider a surface M, whose geometry is given by an embedding $\mathbf{x}\colon M \to \mathbb{R}^3$. We use $\mathbf{n}$ to denote the corresponding unit normals, and dA for the area element of the embedded surface. In the discrete case, we have a triangle mesh $\mathsf{M} = (\mathsf{V}, \mathsf{E}, \mathsf{T})$ with manifold connectivity. Its geometry is given by vertex coordinates $x_\mathsf{v} \in \mathbb{R}^3$ for each $\mathsf{v} \in \mathsf{V}$. For each triangle $\mathsf{t} \in \mathsf{T}$ we use a_t, n_t, and c_t to denote the area, unit normal, and barycenter respectively. For each edge $\mathsf{e} \in \mathsf{E}$ we use ℓ_e for the edge length. For each interior edge $\mathsf{e} \in \mathsf{E}_{\text{int}} \subseteq \mathsf{E}$, with adjacent triangles $\mathsf{t}_1, \mathsf{t}_2$, we let θ_e be the *dihedral angle*, *i.e.*, the angle between triangle normals n_{t_1} and n_{t_2}, and define an associated *edge area* $a_\mathsf{e} := (a_{\mathsf{t}_1} + a_{\mathsf{t}_2})/3$.



3.1 Shell Energy

We begin with introducing the elastic shell energy—in principle following Heeren et al. [2012]. A *thin shell* is, roughly speaking, a solid object with very small thickness $\delta > 0$, dominated by *elastic* behavior: it always tries to restore a deformed shape to some fixed reference configuration. Our embedding $\mathbf{x}$ describes the reference configuration of the midsurface and another embedding $\tilde{\mathbf{x}}$ describes the deformed configuration (on a mesh, just a second set of vertex coordinates $\tilde{x}_\mathsf{v} \in \mathbb{R}^3$). The deviation of the deformed configuration from the reference is then quantified by the elastic potential energy

$$\mathcal{W}(\mathbf{x}, \tilde{\mathbf{x}}) := \mathcal{W}_{\text{membrane}}(\mathbf{x}, \tilde{\mathbf{x}}) + \mathcal{W}_{\text{bending}}(\mathbf{x}, \tilde{\mathbf{x}}),$$

given as a sum of membrane (Section 3.1.1) and bending terms (Section 3.1.2). We use $\widehat{\mathcal{W}}$ for the corresponding discrete energy.

3.1.1 Membrane Energy.

Smooth. A *membrane energy* accounts for tangential stretching and shearing of the surface, and in our case is given by

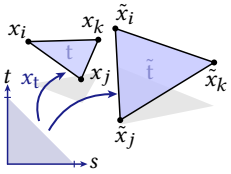
$$\mathcal{W}_{\text{membrane}}(\mathbf{x}, \tilde{\mathbf{x}}) := \delta \int_M W(\mathrm{I}^{-1}\tilde{\mathrm{I}})\, dA, \tag{9}$$

where $\mathrm{I} := \mathrm{J}_x^\mathsf{T}\mathrm{J}_x$ is the metric or *first fundamental form* induced by $\mathbf{x}$ (and likewise for $\tilde{\mathrm{I}}$). The quantity $\mathrm{I}^{-1}\tilde{\mathrm{I}}$ is called the *Cauchy–Green strain tensor*, and encodes the change of metric due to the deformation. The function W describes the stress-strain response of the material—we use the *neo-Hookean energy density*

$$W(A) := \tfrac{\mu}{2}\operatorname{tr} A + \tfrac{\lambda}{4}\det A - \left(\tfrac{\mu}{2} + \tfrac{\lambda}{4}\right)\log(\det A) - \mu - \tfrac{\lambda}{4}. \tag{10}$$

Here λ and μ are positive material constants (called the *first and second Lamé parameters, resp.*).

Discrete. A triangle $\mathsf{t} \in \mathsf{T}$ with vertices $i, j, k \in \mathsf{V}$ is embedded in $\mathbb{R}^3$ via the map $x_\mathsf{t}(s,t) := x_i + s(x_j - x_i) + t(x_k - x_i)$ on the standard triangle $\{(s,t) \in \mathbb{R}^2_{\geq 0} | s + t \leq 1\}$ (see inset). Hence, the metric is constant in each triangle, given by $\mathrm{I}_\mathsf{t} := \mathrm{J}_\mathsf{t}^\mathsf{T}\mathrm{J}_\mathsf{t}$, with



$$\mathrm{J}_\mathsf{t} := \begin{bmatrix} x_j - x_i & x_k - x_i \end{bmatrix} \in \mathbb{R}^{3\times 2},$$

and likewise for $\tilde{\mathrm{J}}_\mathsf{t}$ (where each $x_i \in \mathbb{R}^3$ is viewed as a column vector). The energy in Equation 9 can then be discretized via the sum

$$\widehat{\mathcal{W}}_{\text{membrane}}(\mathbf{x}, \tilde{\mathbf{x}}) = \delta \sum_{\mathsf{t}\in\mathsf{T}} a_\mathsf{t} W(\mathrm{I}_\mathsf{t}^{-1}\tilde{\mathrm{I}}_\mathsf{t}). \tag{11}$$

The logarithmic term in W (Equation 10) ensures that each triangle remains locally embedded, *i.e.*, the three vertices can never become collinear. However, it does not guarantee the mesh will remain globally embedded—a global repulsive energy (*à la* Section 3.2) is needed to keep non-adjacent triangles away from intersecting configurations. In practice, this energy can be implemented using simple expressions in terms of just the areas and edge lengths—see, for example, [Sassen et al. 2020a, Appendix B].

3.1.2 Bending Energy.

Smooth. A *bending energy* accounts for bending of the surface in the normal direction. We use the energy

$$\mathcal{W}_{\text{bending}}(\mathbf{x}, \tilde{\mathbf{x}}) := \delta^3 \int_M \|\widetilde{\mathrm{II}} - \mathrm{II}\|_F^2\, dA, \tag{12}$$

where $\|\cdot\|_F$ is the Frobenius norm, and II is the *second fundamental form*, given by $\mathrm{II}(X, Y) := \langle \mathrm{d}\mathbf{x}(X), \mathrm{d}\mathbf{n}(Y)\rangle$ for all tangent vectors X, Y (and likewise for $\widetilde{\mathrm{II}}$). Intuitively, this energy penalizes a change in curvature along any direction.

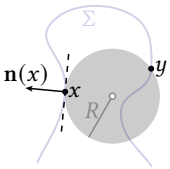
Discrete. For a triangle mesh, we use the discrete bending energy

$$\widehat{\mathcal{W}}_{\text{bending}}(\mathbf{x}, \tilde{\mathbf{x}}) = \delta^3 \sum_{\mathsf{e}\in\mathsf{E}_{\text{int}}} (2\tan(\tilde{\theta}_\mathsf{e}/2) - 2\tan(\theta_\mathsf{e}/2))^2 \ell_\mathsf{e}^2/a_\mathsf{e},$$

where we have applied a standard discretization of normal curvature [Crane and Wardetzky 2017]. Similar to the logarithm in the membrane energy, the use of the tan function prevents foldovers (*i.e.*, $\theta_\mathsf{e} = \pi$). In practice, most dihedral angles are small, and we find it works just as well to use a simpler penalty $(\tilde{\theta}_\mathsf{e} - \theta_\mathsf{e})^2$, corresponding to the discrete bending energy from Grinspun et al. [2003].

3.2 Repulsive Energy

At the most basic level, a *repulsive energy* is an energy that is finite only if a surface is non-intersecting (*i.e.*, *embedded*). Below we recall the smooth definition of the *tangent-point energy* (Section 3.2.1) from Strzelecki and von der Mosel [2013] and recap a simple midpoint discretization (Section 3.2.2); a more reliable adaptive discretization is introduced in Section 5.



3.2.1 Tangent-Point Energy (Smooth). Let $\Sigma := \mathbf{x}(M) \subset \mathbb{R}^3$ denote the embedded surface. For any two points $x, y \in \Sigma$ the *tangent-point radius* $R(x, y)$ is defined as the radius of the smallest sphere through x and y that is tangent to Σ at x. This radius can be expressed as

$$R(x, y) := \frac{|x - y|^2}{2\,|\langle \mathbf{n}(x), x - y\rangle|},$$

where $\mathbf{n}(x)$ is the unit normal at x. The *tangent-point energy* is then

$$\Phi(\mathbf{x}) := \int_\Sigma \int_\Sigma \frac{|\langle \mathbf{n}(x), x - y\rangle|^\alpha}{|x - y|^{2\alpha}} \mathrm{d}x\mathrm{d}y, \tag{13}$$

where $\alpha \geq 1$ is a parameter that controls the strength of repulsion.

For surfaces without boundary, this energy is finite if and only if (i) the surface has no self-intersections, and (ii) $\mathbf{x}$ is sufficiently regular [Strzelecki and von der Mosel 2013; Blatt 2013]. In particular, polyhedral surfaces (which are only C^0) will have infinite energy: normals do not become parallel as points approach opposite sides of an edge. Hence, the numerator does not approach zero, as it would for a smoother surface. We must hence be careful when defining our adaptive quadrature scheme in Section 5; we first revisit a simpler quadrature rule that provides the basis for this scheme.

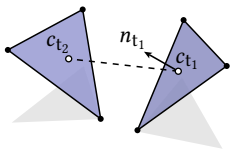
Although we define the tangent-point energy for surfaces in $\mathbb{R}^3$, it can be more generally defined for any sufficiently regular, closed, n-dimensional, embedded submanifold $M \subset \mathbb{R}^m$—and in fact, for even more general sets. For further discussion and precise regularity conditions, see [Strzelecki and von der Mosel 2013] and [Blatt 2013].

3.2.2 Tangent-Point Energy (Discrete). Yu et al. [2021a] discretize tangent-point energy by applying midpoint quadrature to Equation 13 for each triangle pair, yielding the expression

$$\widehat{\Phi}(\mathbf{x}) := \sum_{\substack{\mathsf{t}_1 \in \mathsf{T} \\ \mathsf{t}_1 \neq \mathsf{t}_2}} \sum_{\mathsf{t}_2 \in \mathsf{T}} a_{\mathsf{t}_1} a_{\mathsf{t}_2} K(c_{\mathsf{t}_1}, c_{\mathsf{t}_2}, n_{\mathsf{t}_1}), \tag{14}$$

where k is the discrete tangent-point kernel

$$K(x, y, n) := \frac{|\langle n, x - y\rangle|^\alpha}{|x - y|^{2\alpha}}. \tag{15}$$

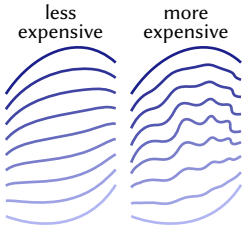


Omitting the summand for $\mathsf{t}_1 = \mathsf{t}_2$ (which would be undefined) is consistent with the fact that every planar piece of a surface has zero tangent-point energy. Moreover, since we evaluate the integrand only at midpoints—and not near edges—the fact that the surface is only C^0 does not cause the energy to blow up in this case. Consistency of this discretization has been verified experimentally by Yu et al. [2021a, Section 8.1]. However, several challenges remain, as will be discussed in Section 5.

4 REPULSIVE SHELLS

With the basic pieces in place, we can finally define our collision-aware shape space for surfaces. At a high level, we apply the same graph manifold construction as in Section 2.3, but adopt the shell and repulsion energies from Section 3. For configurations far from self-intersection, our shape space resembles the well-studied space of elastic shells (Section 4.1). However, when approaching contact, our metric is dominated by the repulsive term (Section 4.2), preventing self-intersection. As in the case of repulsive points, merely *adding* repulsion to the path energy yields undesirable results (Figure 3). Our graph manifold construction instead remains faithful to the appearance of the original shape (Figure 15, *bottom*).

4.1 Shape Space of Elastic Shells



The elastic energy $\mathcal{W}$ assigns a cost to a single displacement, but not a continuous trajectory. To penalize motions that unnaturally stretch or bend the surface, we adopt the *viscous dissipation* model previously used to develop a shape space for elastic shells [Wirth et al. 2011; Heeren et al. 2012, 2014]. The basic idea of viscosity is that material dissipates energy due to internal friction when deformed. In contrast to elastic energy, which depends purely on strain (*i.e.*, displacement), viscous dissipation is determined at each point of a trajectory by the *strain rate*, *i.e.*, the change in displacement per unit time. Hence, unlike elastic energy, total dissipation is path-dependent: sudden stretching and bending costs more than gradual deformation, even if we end up at the same final configuration (see inset).

From a geometric perspective, strain rates correspond to tangent vectors $u \in T_{\mathbf{x}}\mathcal{M}$. To penalize dissipation we hence define the elastic metric g as

$$g_{\mathbf{x}}(u, v) := \tfrac{1}{2}\, \mathrm{d}^2_{\mathbf{y}} \mathcal{W}(\mathbf{x}, \mathbf{y})\Big|_{\mathbf{y}=\mathbf{x}} (u, v), \tag{16}$$

since the Hessian $\mathrm{d}^2_y \mathcal{W}$ provides a local quadratic model of the energy $\mathcal{W}$. Moreover, since $y = x$ is always a minimizer of this energy, the Hessian at this point is positive semidefinite. Indeed, because $\mathcal{W}$ is rigid body motion invariant its Hessian is positive definite up to infinitesimal rigid motions and thus g is a valid Riemannian metric when modding out rigid motions. Geodesics with respect to this metric are obtained exactly as described in Section 2.4; a variety of examples (without repulsion) are shown in Heeren et al. [2012].

4.2 Shape Space of Repulsive Shells

As with repulsive points, merely *adding* tangent-point energy Φ to the integrand of the elastic path energy $\mathcal{E}$ will yield undesirable behavior. Instead, we define our repulsive shape space as the manifold $(\mathcal{M}, g^\Phi)$ where

- the configuration space $\mathcal{M}$ is the collection of C^2 embeddings $\mathbf{x}\colon M \to \mathcal{M}$ of a surface M,
- the metric g on $\mathcal{M}$ is the shell metric from Equation 16, and
- the potential Φ used to construct the graph manifold is the tangent-point energy given in Equation 13.

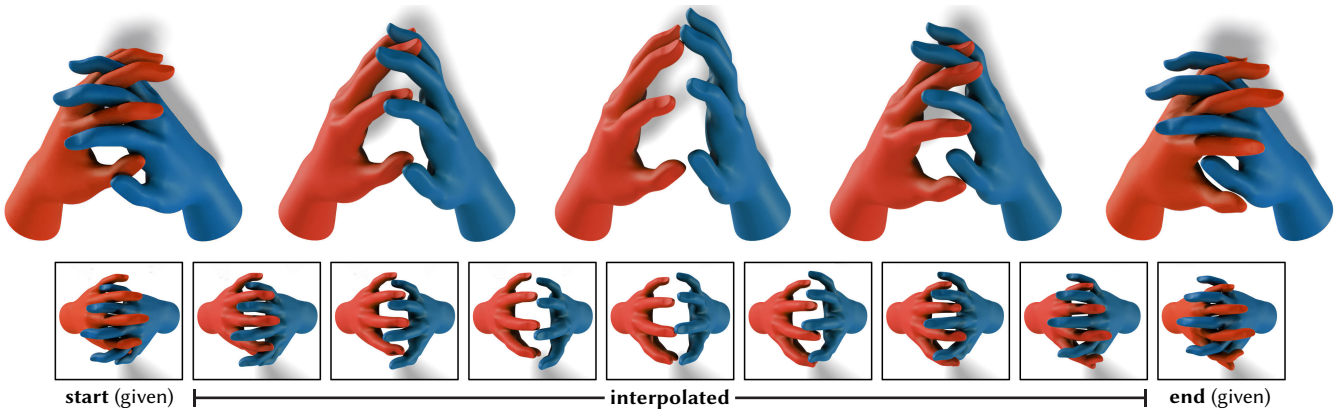


Fig. 11. Geodesics in the space of repulsive shells correspond to intersection-free trajectories that exhibit natural deformation behavior. Here, for instance, we change the way the fingers of two hands are interleaved by interpolating between two given poses *(far left and right)*, avoiding any intersection even at moments of near-contact. Note also the natural bending of the fingers and wrist to avoid collision, despite the lack of any skeletal rig.

We otherwise follow the exact same construction as in Section 2.3. Note that since both the elastic energy $\mathcal{W}$ and the tangent-point energy Φ are rigid-motion invariant, the path energy on this manifold will ignore any rigid translation or rotation along the trajectory. We discuss how to account for rigid motion in Section 7.4.3.

4.2.1 Discrete Path Energy. To discretize the path energy, we follow the treatment from Section 2.4, with only one modification. Namely, we take advantage of analysis done by Rumpf and Wirth [2015], who show that the elastic energy itself approximates the geodesic distance in shell space up to third order. Hence, we can approximate the squared geodesic distance on $(\mathcal{M}, g^{\Phi})$ in time and space by

$$D^2_{\Phi}(\mathbf{x}, \mathbf{y}) := \widehat{\mathcal{W}}(\mathbf{x}, \mathbf{y}) + (\widehat{\Phi}(\mathbf{x}) - \widehat{\Phi}(\mathbf{y}))^2, \tag{17}$$

where $\widehat{\mathcal{W}}$ is the discrete shell energy from Section 3.1, and $\widehat{\Phi}$ is a discretization of tangent-point energy (ultimately, the adaptive one defined in Section 5). Notice that we do *not* directly discretize the metric from Equation 16 by evaluating the Hessian of the discrete elastic energy—in which case we would need 3rd-order derivatives to minimize path energy.

5 ADAPTIVE QUADRATURE

In the continuous setting, tangent-point energy is guaranteed to prevent self-intersections for surfaces without boundary [Strzelecki and von der Mosel 2013]. However, the discrete TPE from Section 3.2.2 still allows self-intersections, since only face centers repel each other (Figure 12). Uniformly refining the mesh everywhere is not practical: even with very large meshes, one may still not capture important singularities in the integrand (Figure 17). Our *adaptive TPE* scheme instead distributes quadrature points exactly (and only) where needed to resolve repulsive forces.

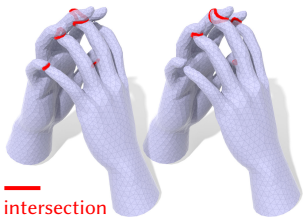


Fig. 12. Without adaptive quadrature, surfaces can exhibit self-intersection. Here: interpolated configurations near start/end pose from Figure 11.

Our basic idea is to apply a hierarchical *fast multipole method* in a lazy fashion to an (implicit) infinite spatial subdivision of the mesh. Unlike Yu et al. [2021a], who use multipole to coarsen interactions between distant elements, we also use it to *refine* interactions between nearby elements. The energy value obtained in this fashion is equal to the continuous tangent-point energy of the (continuous) piecewise linear surface, up to an additive user-specified error tolerance. Hence, if the continuous TPE approaches an infinite barrier, so too must the multipole approximation.

A subtlety is that TPE is *always* infinite for surfaces that are only C^0. To keep it finite we omit neighboring element pairs from the adaptive scheme, relying instead on the discrete elastic energy from Section 3.1 to prevent local intersections. However, since the tangent-point kernel is, by construction, nonsingular within small geodesic neighborhoods, these pairs make a vanishingly small contribution to the overall energy (Yu et al. [2021b, Figure 9] provide a nice illustration). In turn, as the mesh is refined towards a sufficiently regular continuous surface, the numerical approximation converges to the continuous energy—as observed experimentally by Yu et al. [2021a, Section 8.1]. We again stress that we provide no formal guarantee that intersections absolutely cannot occur with adaptive TPE, but make a fairly compelling argument in Section 5.3.1, and observe zero failure cases in practice.

The use of ordinary multipole also reduces the overall cost of energy and derivative evaluation from $O(|\mathsf{T}|^2)$ to $O(|\mathsf{T}|\log(|\mathsf{T}|))$. The final adaptive scheme amounts to a simple subroutine executed only for a small number of interacting elements, usually incurring very little computational overhead (Figure 17).

5.1 Acceptance Criterion

In any n-body problem, collections of distant particles are well-approximated by a single, more massive particle located at their centroid (Beatson and Greengard [1997] provide a nice tutorial). Likewise, the basic strategy for accelerating tangent-point interactions is to approximate the energetic interactions of geometric patches via hierarchical clustering [Yu et al. 2021a,b]. A key question

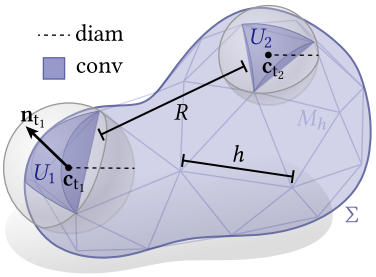
is how to decide when elements should be replaced by a proxy, and how much error this approximation incurs. Here we relate geometric approximation error to error in approximation of the repulsive energy—this analysis guides our adaptive scheme in Section 5.3.

MAC for TPE. Suppose our smooth surface $\Sigma \subset \mathbb{R}^3$ is twice-differentiable (C^2). Our *multipole acceptance criterion (MAC)* for TPE considers whether two surface patches $U_1, U_2 \subset \Sigma$ are small relative to the distance between them. In particular, if $\operatorname{diam}(U)$ is the greatest distance between any two points in U and $\operatorname{conv}(U)$ is its convex hull, then our MAC is

$$\max\{\operatorname{diam}(U_1), \operatorname{diam}(U_2)\} \leq \theta \operatorname{dist}(\operatorname{conv}(U_1), \operatorname{conv}(U_2)), \quad (18)$$

where $\operatorname{dist}(A, B)$ is the smallest distance between any two points $a \in A, b \in B$. The parameter $\theta > 0$ controls how aggressively we replace exact patch interactions with a simpler proxy—and hence the accuracy of our approximation. In particular, if we replace each patch U with its center of mass $c(U)$ and average normal $\overline{n}(U)$, then we get a multipole approximation of the TPE restricted to these surfaces patches $\Phi(U_1, U_2)$ with error of order $O(\theta^2)$:

$$\Phi(U_1, U_2) = \operatorname{area}(U_1)\operatorname{area}(U_2)\frac{|\langle \overline{n}(U_1), c(U_1) - c(U_2)\rangle|^\alpha}{|c(U_1) - c(U_2)|^{2\alpha}} + O(\theta^2). \quad (19)$$



In practice, we cannot evaluate the multipole approximation directly, because we have only a triangulation and not the smooth surface Σ. Suppose, however, that Σ_h is a *closely inscribed* triangle mesh (in the sense of Morvan and Thibert [2002]) with maximum edge length h. Consider then two triangles t_1, t_2 in Σ_h, and the two corresponding surface patches $U_1, U_2 \subset \mathbb{R}^3$ obtained by closest-point projection onto Σ (see inset). Since the triangle normal n_{t_1} approximates $\overline{n}(U_1)$ up to $O(h)$ [Morvan and Thibert 2002], we get

$$\Phi(U_1, U_2) = a_{t_1} a_{t_2} \frac{|\langle n_{t_1}, c_{t_1} - c_{t_2}\rangle|^\alpha}{|c_{t_1} - c_{t_2}|^{2\alpha}} + O(h + \theta^2). \quad (20)$$

Hence, as long as we pick $\theta^2 \in O(h)$ and satisfy the MAC for this θ value, our discrete energy $\widehat{\Phi}(\Sigma_h)$ will approximate the continuous energy $\Phi(\Sigma)$ up to an additive error of size at most linear in h. This calculation agrees with experiments by Yu et al. [2021a, Section 8.1].

5.2 Multipole Approximation

We apply the MAC from Section 5.1 to accelerate evaluation of the tangent-point energy and its derivatives. Like Yu et al. [2021a], this approximation dramatically speeds up evaluation of distant interactions (with very little approximation error), but unlike Yu *et al.* we use a 0th-order fast multipole scheme, rather than Barnes-Hut. In doing so, we get about an order of magnitude speedup over Yu et al. [2021a]—*e.g.*, for the four examples in Figure 16, our scheme is roughly 7, 6, 9, and 10–20 times faster, respectively.

5.2.1 *Energy Approximation.*

Bounding volume hierarchy. We start by building a standard *bounding volume hierarchy (BVH)* for the mesh, *i.e.*, a binary tree where triangles of Σ_h are hierarchically clustered into *axis-aligned bounding boxes (AABBs)* [Yu et al. 2021a, Section 4]. We will use U_1, U_2 to denote the two children of any BVH node U. Each node also stores the total area a_U, center of mass c_U, and area-weighted average normal n_U of all the triangles it contains. In particular, for each node U we compute

$$a_U = a_{U_1} + a_{U_2}, \quad c_U = \frac{a_{U_1} c_{U_1} + a_{U_2} c_{U_2}}{a_U}, \quad n_U = \frac{a_{U_1} n_{U_1} + a_{U_2} n_{U_2}}{a_U}. \quad (21)$$

For efficiency reasons, we store $k > 1$ triangles in each leaf node (in practice, $k = 2$), taking totals/averages of these quantities.

Block cluster tree. The BVH in turn defines a *block cluster tree (BCT)* that can be used to approximate the energy and its differential. In particular, each node of the BCT is a pair (U, V) of nodes from the BVH, with children $(U_1, V_1), (U_1, V_2), (U_2, V_1), (U_2, V_2)$, starting with the root node (Σ_h, Σ_h).

Energy evaluation. To evaluate the energy of any node, we check if the AABBs of U and V satisfy the MAC (Equation 18). If so, we evaluate the multipole approximation (Equation 19), using the aggregate area/center/normal data and the minimum box-box distance. Otherwise, we sum up the contributions of all four children. If these children are individual triangles, we apply the adaptive scheme from Section 5.3. Applying this algorithm to the root node of the BCT yields the overall energy approximation. Recall that for two triangles sharing a vertex or edge, we do not apply adaptive multipole, but instead just use the standard midpoint scheme to avoid infinite energy contribution.

Approximation error. The error of energy approximation is of order $O(\theta^2)$. Cost of course increases as $\theta \to 0$; in our experiments $\theta = \frac{1}{4}$ was already sufficient to approximate the energy with a relative error around 0.1%.

Derivative Approximation. Note that since the set of admissible leaves in the BCT can change as the surface deforms, the multipole approximation of $\widehat{\Phi}$ is not itself differentiable. Instead, as in Yu et al. [2021b, Appendix B.1], we approximate the derivative of the discrete energy, rather than try to take the derivative of the approximation. In particular, for any admissible pair (U, V) used in the energy approximation we compute the differential of $\widehat{\Phi}(U, V)$ with respect to the cluster centers c_U, c_V, areas a_U, a_V, and normals n_U, n_V, and propagate these derivatives down the BVH toward the leaves (via the chain rule), ultimately yielding derivatives of the discrete energy with respect to the vertex locations. In particular, at interior nodes, we differentiate the relationships given in Equation 21.

5.3 Adaptive Subdivision

Next, we detail our adaptive strategy to alleviate the issue of the midpoint discretization "missing" singularities in Φ and thus not preventing intersections. To this end, we consider a lazy evaluation of the multipole method on an infinite subdivision of the mesh.

Consider in particular a triangle pair (t_1, t_2), which we can view as the root node of a block cluster tree. The children of any node

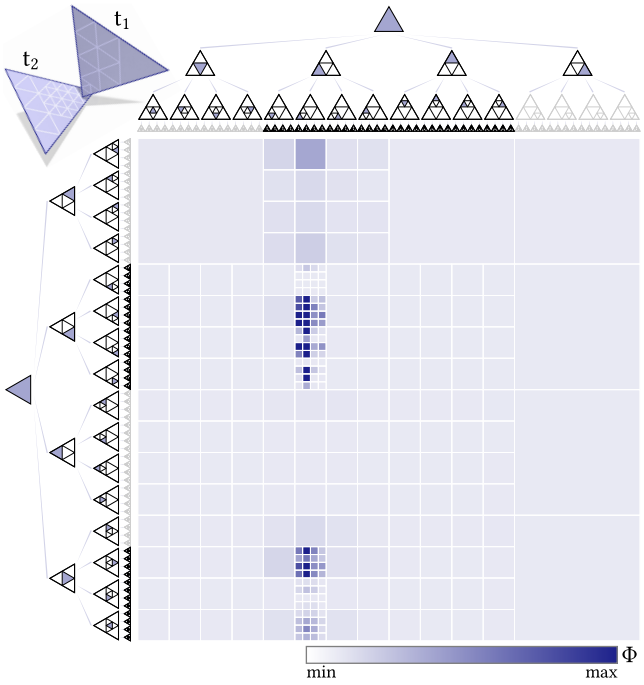


Fig. 13. When two triangles t_1, t_2 are in near-contact, it may be insufficient to approximate the repulsive energy using just their centers. Instead, we pretend that each triangle has been subdivided hierarchically into infinitely many pieces, and continue running the usual multipole approximation algorithm. Here we show the sub-trees implicitly used for approximation; each block corresponds to an energy term in our sum. Note however that we do not explicitly build these trees.

(t_1', t_2') in this tree (including the root) are defined as the sixteen triangle pairs obtained by splitting both t_1' and t_2' into four along edge midpoints (Figure 13). Note, then, that the tree has no leaves: implicitly, we have an infinite BCT defined by repeated subdivision.

Even though we do not have an explicit representation of the tree, we can still apply a multipole algorithm via lazy evaluation. In particular, we push the root node onto a stack. Until the stack is empty, we check if the top element satisfies the MAC. If so, we add the multipole approximation to our total energy; if not we split each triangle into four and push the sixteen corresponding triangle pairs back onto the stack. To evaluate the MAC for a triangle pair we no longer use the AABB approximation, which is unnecessarily conservative. Instead, we use the exact triangle diameter (equal to the longest edge length) and the exact triangle-triangle distance. Algorithm 3 (found in the supplement) provides pseudocode for a simple stack-based implementation. In practice we use a much faster *stack-free* implementation that implicitly encodes the vertex coordinates, rather than allocating them in memory.

Derivatives. We also adaptively approximate the energy differential. For optimization, we need only derivatives with respect to original mesh vertices. Hence, rather than propagate information "down" from coarse nodes to vertices in leaves of the tree, we propagate information "up" from subdivisions of a triangle pair toward the original triangles. This computation is made easy by the fact that (i) all subdivided triangles share normals with their coarse ancestor, (ii) the subdivided area at depth k is simply $(1/4)^k$ times the original area k, and (iii) the center of a subdivided triangle is a barycentric linear combination of the three original vertices. This procedure is made explicit in Algorithm 3 (found in the supplement).

5.3.1 No-intersection Property. Though we do not provide a rigorous proof, we can at least sketch a fairly compelling argument for why our adaptive scheme should approach infinite energy for surface trajectories that approach self-contact. There are two basic cases to consider: adjacent and non-adjacent elements. Intersections between adjacent elements, which are not included in the discrete adaptive tangent-point energy, are prevented by the infinite barriers in the discrete membrane and bending energy (due to the logarithm and tangent functions—see Section 3.1). Other schemes such as IPC use elastic penalties in the same way: to prevent local collisions and fold-overs not accounted for by repulsion [Li et al. 2020, Sections 2 and 4.4]. For non-adjacent elements, suppose we do not truncate the tree to any fixed depth. The resulting approximation error then depends purely on the MAC parameter θ (Equation 18). Recall in particular that multipole makes an *additive* error of size $O(h + \theta^2)$ (Equation 20). Hence, no matter how large the (unknown) constant C in this approximation error, there will always be a separation distance R between the two closest points of contact such that the approximation error is below any tolerance ε. Hence, one could ensure that no intersections occur by, *e.g.*, applying backtracking line search to the total energy (elastic plus adaptive TPE).

In practice, since we aim primarily to obtain descent directions for optimization, we do not need a particularly accurate approximation of TPE. We hence use a larger θ parameter for adaptive subdivision than for the global BCT ($\theta = 10$ in all our experiments). Adaptive refinement is of course more expensive than the midpoint scheme, but in most scenarios it is active for only a small number of triangle pairs in near-contact (or not at all). See further experiments and discussion in Sections 7.2.1 and Section 8.

6 NUMERICAL OPTIMIZATION

Computing both interpolating and extrapolating geodesics leads to spacetime optimization problems. Here we describe how to solve these problems efficiently.

6.1 Interpolation

To find a geodesic $\mathbf{x}^0, \ldots, \mathbf{x}^n \in (\mathbb{R}^3)^{|V|}$ interpolating given configurations $\mathbf{x}^0, \mathbf{x}^n$ we use a *trust-region* method to minimize the discrete path energy Equation 6, following [Nocedal and Wright 2006, Alg. 4.1]. Each step of this method minimizes a local quadratic model of the energy over a limited-size ball around the current guess. In our case, the "current guess" means the current trajectory, viewed as a point in $(\mathbb{R}^{3|V|})^{n-1}$ corresponding to the degrees of freedom $\mathbf{x}^1, \ldots, \mathbf{x}^{n-1}$. In the case of repulsive shells, our discrete path energy $\widehat{\mathcal{E}}(\mathbf{x})$ is the sum of an elastic term

$$\widehat{\mathcal{E}}_{\mathcal{W}}(\mathbf{x}) \coloneqq n \sum_{k=1}^{n} \widehat{\mathcal{W}}(\mathbf{x}^{k-1}, \mathbf{x}^k)$$

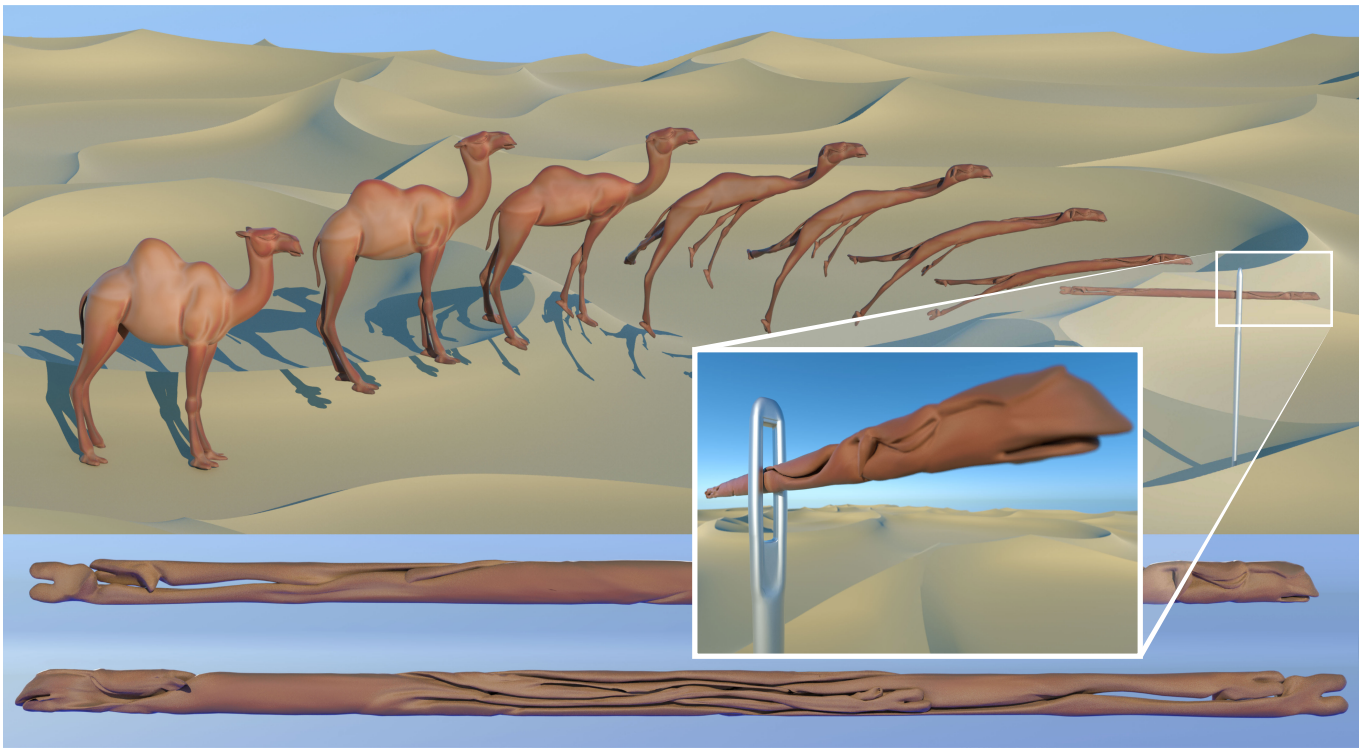


Fig. 14. Passing a camel through the eye of a needle. The initial mesh *(far left)* is first progressively compressed into a cylinder *(far right)*; we then find a geodesic between initial and compressed states *(left to right)*. *Bottom:* close-up of compressed geometry from left/right sides; inset shows occluded piece.

plus a repulsive term

$$\widehat{\mathcal{E}}_\Phi(\mathbf{x}) \coloneqq n \sum_{k=1}^{n} (\widehat{\Phi}(\mathbf{x}^{k-1}) - \widehat{\Phi}(\mathbf{x}^k))^2,$$

where the latter can be viewed as a nonlinear least squares energy. Hence, to define our local quadratic model we use the Hessian of $\widehat{\mathcal{E}}_\mathcal{W}(\mathbf{x})$, plus the Gauß-Newton approximation of the Hessian of $\widehat{\mathcal{E}}_\Phi(\mathbf{x})$.

For this approximation, we consider the vector

$$\mathcal{V}(\mathbf{x}) = \left(\widehat{\Phi}(\mathbf{x}^0) - \widehat{\Phi}(\mathbf{x}^1), \ldots, \widehat{\Phi}(\mathbf{x}^{n-1}) - \widehat{\Phi}(\mathbf{x}^n)\right) \in \mathbb{R}^n$$

and express the repulsive term as $\widehat{\mathcal{E}}_\Phi(\mathbf{x}) = n\mathcal{V}(\mathbf{x})^T\mathcal{V}(\mathbf{x})$. Its first and second derivatives are then

$$\begin{aligned} \mathrm{d}_\mathbf{x}\widehat{\mathcal{E}}_\Phi &= 2n\,(\mathrm{d}_\mathbf{x}\mathcal{V})^T\mathcal{V}(\mathbf{x}), \\ \mathrm{d}^2_\mathbf{x}\widehat{\mathcal{E}}_\Phi &= 2n\,(\mathrm{d}_\mathbf{x}\mathcal{V})^T\mathrm{d}_\mathbf{x}\mathcal{V} + 2n\,(\mathrm{d}^2_\mathbf{x}\mathcal{V}(\mathbf{x}))^T\mathcal{V}(\mathbf{x}), \end{aligned}$$

respectively. Near a minimizer $\mathcal{V}(\mathbf{x})$ is small, and hence the Hessian is well-captured by the Gauß-Newton approximation

$$H_\mathrm{GN} \coloneqq 2n\,\mathrm{d}_\mathbf{x}\mathcal{V}^T\mathrm{d}_\mathbf{x}\mathcal{V} \in \mathbb{R}^{3(n-1)|\mathsf{V}|\times 3(n-1)|\mathsf{V}|}.$$

This matrix is symmetric and block-tridiagonal, with layout

$$H_\mathrm{GN} = \begin{bmatrix} A_1 & B_1 & & \\ B_1^T & \ddots & \ddots & \\ & \ddots & \ddots & B_{n-2} \\ & & B_{n-2}^T & A_{n-1} \end{bmatrix}, \quad \begin{aligned} A_k &\coloneqq 4n\,\mathrm{d}_{\mathbf{x}^k}\widehat{\Phi}\,\mathrm{d}_{\mathbf{x}^k}\widehat{\Phi}^T, \\ B_k &\coloneqq -2n\,\mathrm{d}_{\mathbf{x}^k}\widehat{\Phi}\,\mathrm{d}_{\mathbf{x}^{k+1}}\widehat{\Phi}^T, \end{aligned}$$

where all blocks have size $3|V| \times 3|V|$. Overall, our quadratic model has the matrix representation $\frac{1}{2}\mathbf{x}^T B\mathbf{x} + c^T\mathbf{x}$, where

$$B \coloneqq \mathrm{d}^2_\mathbf{x}\widehat{\mathcal{E}}_\mathcal{W} + H_\mathrm{GN} \in \mathbb{R}^{3(n-1)|\mathsf{V}|\times 3(n-1)|\mathsf{V}|}$$

and $c \in \mathbb{R}^{3(n-1)|\mathsf{V}|\times 3(n-1)|\mathsf{V}|}$ is the differential of the path energy $\widehat{\mathcal{E}}(\mathbf{x})$ with respect to $\mathbf{x}$, computed *à la* Section 5 (see Supplement, Appendix B for details).

To avoid assembling the dense blocks of the Gauß-Newton approximation, we use an iterative solver for the trust region subproblem. In particular, we use *Steihaug's conjugate gradient (CG) method* [Nocedal and Wright 2006, Alg. 7.2], which extends conventional CG to indefinite problems with constraints on the norm of the solution.

As with any CG method, we must apply a preconditioner to achieve fast convergence. We use the block-diagonal matrix $P \in \mathbb{R}^{3(n-1)|\mathsf{V}|\times 3(n-1)|\mathsf{V}|}$ with blocks $P_k = \mathrm{d}^2_\mathbf{y}\widehat{\mathcal{W}}(\mathbf{x}_k, \mathbf{y})|_{\mathbf{y}=\mathbf{x}_k}$ for each time step $k = 1, \ldots, n-1$. In other words, we take the Hessian of the elastic energy with respect to the second configuration $\mathbf{y}$, assuming the first configuration is fixed at $\mathbf{x}_k$, then evaluate this Hessian at $\mathbf{y} = \mathbf{x}_k$. This preconditioner behaves like a Newton preconditioner for the elastic energy in space, and like the identity

(or L^2) preconditioner in time. Since the Euler-Lagrange equations for the path energy look like a 1D Laplace equation in time, it might also be helpful to use an H^1 preconditioner in time (*i.e.*, take the 2nd-order centered difference of consecutive configurations). However, the L^2-in-time scheme is easier to assemble and cheaper to invert, as we can easily exploit its block structure.

6.1.1 Initialization. In order to optimize geodesics, we must begin in a feasible state. This means that each of the individual time steps $\mathbf{x}_0, \ldots, \mathbf{x}_n$ must be intersection free—however, we are not required to provide a *time-continuous* initial trajectory that is intersection-free. This setup gives us a fair bit of flexibility in initializing the solver. In many situations (*e.g.*, Figure 4), it is sufficient to start with a piecewise continuous trajectory where $\mathbf{x}_k$ is initialized to either $\mathbf{x}_0$ or $\mathbf{x}_n$ depending on whether k is greater or less than $n/2$.

For more difficult examples, one may need to be more intelligent about initialization. For instance, in Figure 11, we use a middle configuration $\mathbf{x}_{n/2}$ where the hands have been placed far apart—but still use a piecewise constant initialization (now with three pieces). Other examples likewise use a single additional middle configuration, computed in a problem-specific way—see Section 7 for more details. In general this approach provides one reasonable initialization strategy: a user need only specify a very small number of "hints" about where a collision-free trajectory might need to travel, and optimization takes care of the rest. One could also supply a continuous initial trajectory and "tighten" it to a geodesic, but we did not find it necessary to provide nearly this much information for any of our examples.

6.2 Extrapolation

To extrapolate a pair of given surface configurations $\mathbf{x}_{k-1}$, $\mathbf{x}_k$ to the next configuration $\mathbf{x}_{k+1}$, we solve Equation 8, but now use Equation 17 to approximate dist^2_Φ (Figure 27 shows an example). In particular, if we let $f\colon \mathbb{R}^{3|\mathsf{V}|} \to \mathbb{R}^{3|\mathsf{V}|}$ be the function mapping $\mathbf{x}_{k+1}$ to the left-hand side of Equation 8, then we seek to solve the nonlinear equation $f(\mathbf{x}_{k+1}) = 0$. Since this equation is nonlinear, we use Newton's method with step size τ chosen via Armijo line search. In particular, each step of Newton's method computes $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_{k+1} - \tau\,(\mathrm{d}_{\mathbf{x}_{k+1}} f)^{-1} f(\mathbf{x}_{k+1})$. Hence, we have to invert the Jacobian $\mathrm{d}_{\mathbf{x}_{k+1}} f$ in each iteration. The Jacobian itself is given by

$$\mathrm{d}_{\mathbf{x}_{k+1}} f = H_M(\mathbf{x}_k, \mathbf{x}_{k+1}) - 2\,\mathrm{d}_{\mathbf{x}_k}\widehat{\Phi} \otimes \mathrm{d}_{\mathbf{x}_{k+1}}\widehat{\Phi}, \tag{22}$$

where $H_M \coloneqq \mathrm{d}^2_{\mathbf{x}_k,\mathbf{x}_{k+1}} \widehat{\mathcal{W}} \in \mathbb{R}^{3|\mathsf{V}|\times 3|\mathsf{V}|}$ is the *mixed Hessian* of the elastic energy, *i.e.*, the partial derivative of each of the $3|\mathsf{V}|$ equations in the system $\mathrm{d}_{\mathbf{x}_k}\widehat{\mathcal{W}}(\mathbf{x}_k, \mathbf{x}_{k+1}) = 0$, with respect to each of the $3|\mathsf{V}|$ coordinates of $\mathbf{x}_{k+1}$. Since the remaining term is just a rank-one update of this mixed Hessian, we can apply the Sherman–Morrison formula to get

$$(\mathrm{d}_{\mathbf{x}_{k+1}} f)^{-1} = H_M^{-1} - \frac{2\,H_M^{-1}(\mathrm{d}_{\mathbf{x}_k}\widehat{\Phi})\,(\mathrm{d}_{\mathbf{x}_{k+1}}\widehat{\Phi})^T H_M^{-1}}{1 + 2\,(\mathrm{d}_{\mathbf{x}_{k+1}}\widehat{\Phi})^T\,H_M^{-1}\,(\mathrm{d}_{\mathbf{x}_k}\widehat{\Phi})}. \tag{23}$$

To efficiently invert the Jacobian, we can hence prefactor $H_M(\mathbf{x}_k, \mathbf{x}_{k+1})$ and perform backsubstitutions. Our initial guess for $\mathbf{x}_{k+1}$ is always just the previous step $\mathbf{x}_k$. Though it is tempting to, *e.g.*, perform linear extrapolation, using the previous step ensures that we do not introduce self-intersections—moreover, as in the elastic case, a linear extrapolation can in some cases lead to a much worse initial guess (*e.g.*, in situations of tight pinching).

6.3 Weighted Averages

How do you average a given set of data points? In a vector space (*e.g.*, given points in the plane), one can simply take the arithmetic mean. But this elementary definition no longer applies to data that lives on a manifold, such as rotations, which live in SO(3)—or configurations of an elastic shell, which belong to a shape space $(\mathcal{M}, g)$. Here, a variational definition is more natural: given data points $\mathbf{x}_1, \ldots, \mathbf{x}_k$ and fixed weights $\lambda_1, \ldots, \lambda_k \in [0, 1]$ that sum to one, find a configuration

$$\mathbf{x}(\lambda) \coloneqq \arg\min_{\overline{\mathbf{x}}} \sum_{i=1}^{k} \lambda_i\,\mathrm{dist}_g(\overline{\mathbf{x}}, \mathbf{x}_i)^p. \tag{24}$$

For $p = 2$, any such minimizer is called a (weighted) *Karcher mean* [Karcher 2014], and coincides with the usual arithmetic mean for $\mathcal{M} = \mathbb{R}^n$ and $\lambda_i \equiv 1/k$.

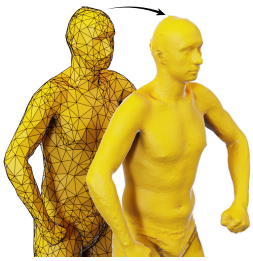
For repulsive shells, we can approximate Equation 24 via the distance formula given in Equation 17, yielding a natural notion of intersection-free (weighted) averages between shapes in very different configurations. Figure 1 shows one example, where the repulsive component of our metric is crucial for avoiding large intersections that otherwise occur in the elastic shape space. More accurate averages might be achieved by, *e.g.*, replacing the geodesic distance with the path energy and simultaneously optimizing k geodesics—though we do not pursue that approach here.

6.4 Upsampling

A nice benefit of using an adaptive discretization (Section 5), as well as a consistent discretization in space (Section 3) and time (Section 2.4) is that, in our experiments, coarse trajectories are usually predictive of fine scale behavior. We can hence upsample in space or time to get higher-fidelity results with less computational cost.

6.4.1 Temporal Upsampling. For all of our interpolation examples, we use a simple upsampling scheme to achieve temporal resolution suitable for animation (60fps). Starting with a coarse solution, we insert a new configuration $\mathbf{x}_{k+1/2} = \mathbf{x}_k$ between any two consecutive configurations $\mathbf{x}_k, \mathbf{x}_{k+1}$, then minimize the path energy of this finer trajectory. Similar to the initialization scheme from Section 6.1.1, this strategy gives a piecewise constant trajectory that is guaranteed to be free of intersections. In practice we typically start with only eight samples in time, subdividing to as many as 512 samples for final animations. Figure 30 shows one example indicating that the coarse preview is highly predictive of the final result—making this approach quite attractive for iterating on a design.

6.4.2 Spatial Upsampling. We also explored a simple strategy for spatial upsampling. In addition to accelerating computation, this strategy enables one to de-couple the simulation mesh from the mesh used for final output or visualization—preserving the original



connectivity if needed. As a precomputation, we compute correspondence between a fine mesh M_{fine} used for visualization and a coarse mesh $\mathsf{M}_{\text{coarse}}$ used for computation. (We generated coarse meshes via remeshing and smoothing from *OpenFlipper* [Möbius and Kobbelt 2012].) More explicitly, for each vertex i of M_{fine} we find the closest point on $\mathsf{M}_{\text{coarse}}$. If this point is on a triangle, we store its barycentric coordinates and the (signed) normal offset. If the closest point is instead on a vertex i^* of $\mathsf{M}_{\text{coarse}}$, we express the offset as a linear combination of the normals $n_1, \ldots, n_d$ around coarse vertex i^*, where d is the degree of i^*. In particular, we use QR decomposition to find the smallest-norm solution of the linear system $[n_1 \cdots n_d]c = x_i^{\text{fine}} - x_{i*}^{\text{coarse}}$, where $c \in \mathbb{R}^d$. For each configuration $\mathbf{x}_k$, we can then trivially compute offsets from the coarse mesh to recover a fine upsampling. This simple scheme could of course be improved in a number of ways—for instance, we do not perform any further optimization after spatial upsampling, which can leave small intersections in regions of very close contact (*e.g.*, Figure 21), though we did not find it affects overall qualitative appearance (and it has no impact on optimization). Intersection-free upsampling is an interesting question for future work.

7 EVALUATION AND EXAMPLES

We evaluated our method on a variety of challenging examples. To gauge the effectiveness of the elastic + TPE energy, we first consider basic minimization tasks (Section 7.3), subject to various constraints. We then explore how this energy facilitates collision-avoiding interpolation and extrapolation (Section 7.4). We also provide numerical evaluation and comparison with several alternatives (Section 7.2.2).

Note that although we consider here only interpolation, extrapolation, and averaging, the metric (and optimization framework) we have developed fits squarely into the standard shape space framework—and could hence be more broadly applied to task such as parallel transport for detail transfer [Heeren et al. 2014] or nonlinear statistics of shapes [Fletcher et al. 2004; Heeren et al. 2018].

7.1 Implementation

We implemented our method in C++ using the *OpenMesh* [Botsch et al. 2002] library for mesh data structures, *Eigen* [Guennebaud et al. 2010] for numerical linear algebra, *SuiteSparse* [Davis 2006] for direct linear solvers, the *GOAST* [Heeren and Sassen 2020] for optimization and variational problems (including the discrete elastic energy), and the *Repulsor* [Schumacher 2023] library for the tangent-point energy. Most experiments were run on a Intel i7 1260p laptop with 4 performance cores and 32GB RAM; to generate final 60fps frame rate animation we used a single workstation with two 32-core AMD EPYC 7601 processors with 1TB RAM. Evaluation of the surface energies and their derivatives were trivially parallelized over time steps (using OpenMP); further parallelization could easily be achieved in a variety of ways (*e.g.*, by solving linear systems in parallel, or porting to the GPU). We found in practice the most expensive step is most often computing the initial coarse trajectory. Subsequent refinement in time is progressively cheaper, since here we already have a good initialization, and need to perform relatively few optimization steps. For instance, Figures 11 and 15 each took first 10 minutes to find a coarse trajectory, then 5 and 2-3 minutes, respectively, to refine. Some of the more involved examples require a stepwise minimization with changing parameters to obtain the final results (see *e.g.*, Section 7.4.2). The runtime for such examples is typically much higher, *i.e.*, they can take multiple hours for detailed meshes. Our simulation meshes in Figures 1, 4, 11, 14, 18, 21, 22, 23, 26, 27, 28, 29 have 3995, 5120, 6766, 11534, 4996, 5996, 9992, 5400, 8152/8344/6204, 5966, 10192, 5498 triangles, respectively.

7.2 Numerical Tests

7.2.1 Adaptive Quadrature. The cost of adaptive subdivision (Section 5.3) depends on the number of triangle pairs that violate the MAC, and hence on the dimension of the contact region. In the worst case, this contact will occur over dense two-dimensional regions—for instance, in the synthetic case of two parallel plates separated by a tiny distance $\delta > 0$ we would have to refine everywhere. To examine behavior on a more representative example, we consider a pair of hand models meeting at fingertips (0-dimensional contact), along fingers (1-dimensional contact), and across palms (2-dimensional contact), as shown in Figure 17. Here we plot both the evaluation cost and our approximation of the tangent-point energy Φ, as a function of the separation distance $\delta > 0$ and the mesh resolution. Examining the energy plots, we observe that non-adaptive quadrature significantly under-estimates the energy in regions of close contact (even at very high mesh resolutions), whereas adaptive quadrature captures a contribution with the expected asymptotic growth of $O(1/\delta^\alpha)$, where recall that α is the integrability parameter of Φ from Section 3.2. Simultaneously, the growth in evaluation cost is quite small—not even quite as large as the linear and quadratic growth one might expect from 1- and 2-dimensional contact (*resp.*), and also quite small in absolute terms. We have observed this behavior consistently across all examples: unless nearby surfaces have identical curvature, there will invariably be point-like peaks which minimize the surface-surface distance (like knuckles on the palms), and hence dominate the energy (growing as $1/\delta^\alpha$ with distance).

7.2.2 Alternative Formulations. As noted in Section 3, nothing about our graph manifold construction depends on the specific form of the tangent-point energy Φ—one could try replacing it with any other repulsive potential. To this end, we did several experiments using the *incremental potential contact (IPC)* energy in place of TPE, using the authors' own reference implementation for energy and derivative evaluation [Li et al. 2020].

Despite the all-pairs nature of TPE, we found that in practice the hierarchical evaluation provided by the multipole scheme actually tends to yield a *lower* evaluation cost than the IPC penalty (Figure 16), except in regions of very tight near-contact, such as the scrunched-up camel (Figure 16, *right*). The fact that the IPC penalty and TPE have a similar cost should come as no surprise: both methods use a spatial acceleration data structure to prune down all $O(n^2)$ element pairs into a dramatically smaller evaluation set. In particular, the IPC implementation uses a spatial partitioning strategy, via a regular grid with $O(1/d^3)$ elements; TPE uses an

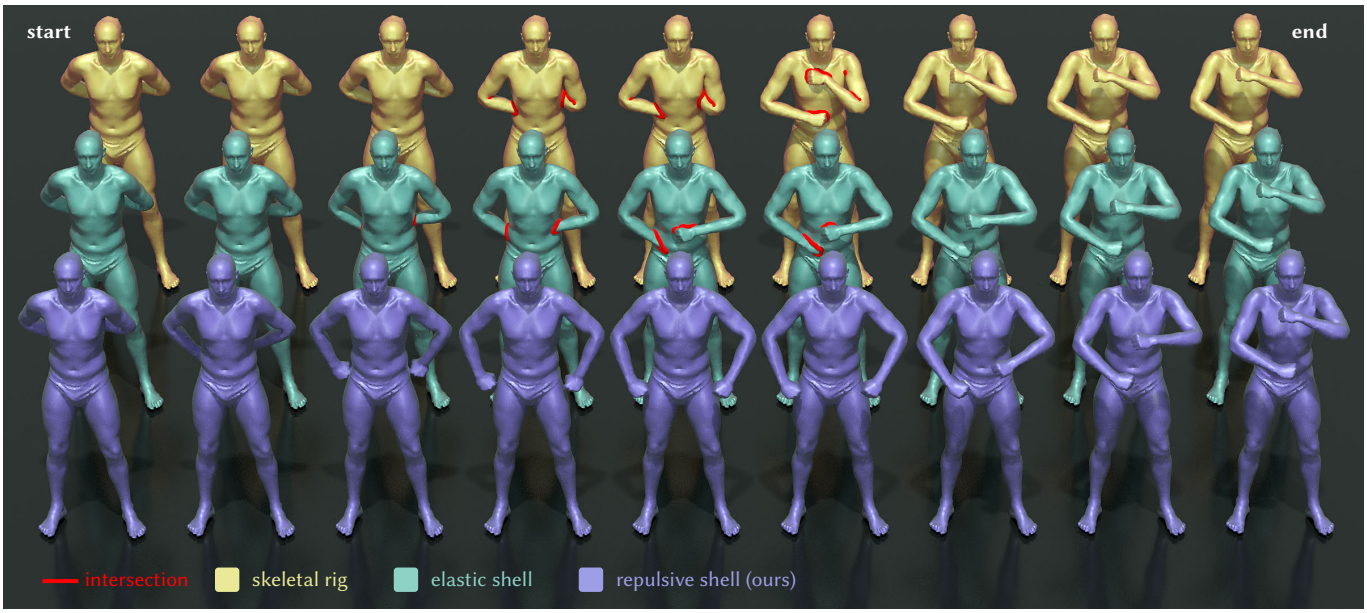


Fig. 15. Interpolation between far-left and far-right poses, using a skeletal rig *(top)*, a geodesic in the space of elastic shells *(center)*, and a geodesic in our repulsive shell space *(bottom)*. Notice that the repulsive metric does not merely resolve local intersections near moments of contact—rather, it alters the overall motion plan, yielding different global poses that proactively avoid intersection.

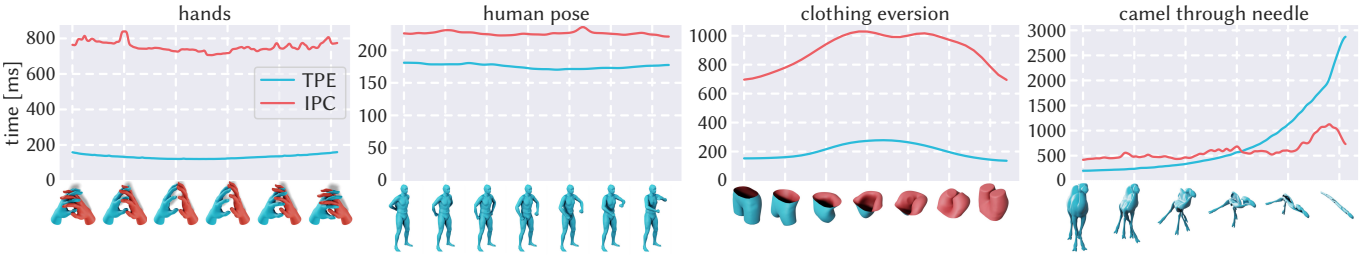


Fig. 16. Timings of energy evaluations. We show the time it takes to evaluate TPE and the IPC barrier energy without parallelization on an AMD EPYC 7601 along various geodesics. These are, from left to right, the geodesics from Figures 11, 15, 20 (initialization), and 14.

object partitioning strategy, via a bounding volume hierarchy (BVH) with $O(n \log n)$ elements. Moreover, since our TPE scheme uses a fast multipole scheme, we get accurate energy evaluation *without* omitting distant interactions (which account for only about 5% of total evaluation cost for the examples in Figure 16).

We also found that a collision potential like the IPC penalty yields less than satisfactory behavior for shape space optimization, often yielding sudden "jumps" in the configuration (Figure 20)—even with extensive tuning of the IPC target distance $\widehat{d}$ and the repulsion strength β. We speculate that the compact support of the IPC penalty means that it does not provide useful guiding forces far from contact, with more localized peaks in the energy landscape than for TPE. This behavior is reflected in Figure 18, where we compare TPE- and IPC penalty-based formulations in our trust-region framework *without* applying continuous collision detection (CCD) to prevent self-intersection—instead, we rely solely on the strength of the actual forces (*i.e.*, energy gradients) provided by the TPE and IPC potentials. CCD could perhaps be helpful in our numerical framework as well, but these experiments suggest it would likely be invoked less often (incurring lower cost). On the whole, we found that TPE works best for our shape space problems, though further exploration (*e.g.*, using a convergent variant of IPC [Li et al. 2023] or an integrated barrier approach using a short-range repulsive energy) may yield interesting alternatives.

A completely different alternative is to use a method that tries to diffeomorphically deform the ambient space. For instance, Eisenberger and Cremers [2020] restrict deformation velocities to a fixed basis of low-frequency divergence-free fields. In case of detailed surfaces, the deformations arising from the low-frequency field are not capable to resolve these details leading to significant artifacts in regions of close local contact (Figure 19). The use of a Lagrangian surface mesh, combined with our adaptive quadrature scheme, ensures that we can resolve both geometric and energetic features at the appropriate level of detail.

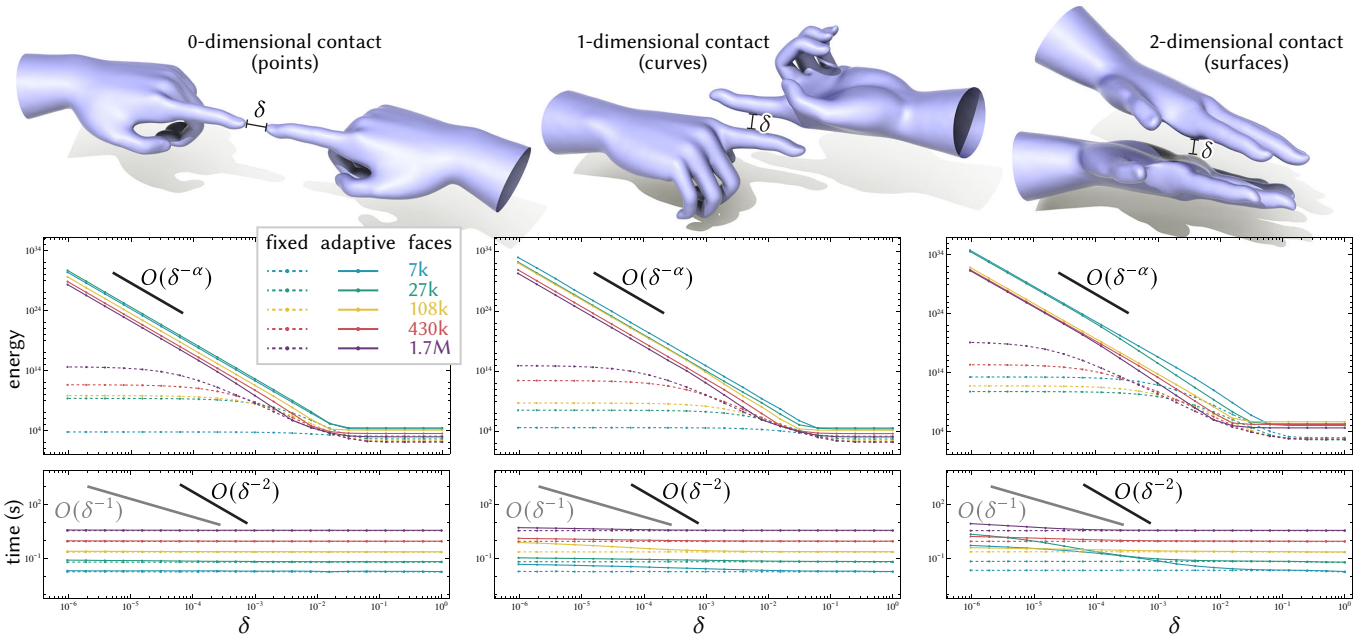


Fig. 17. To gauge the cost and accuracy of our adaptive energy evaluation strategy, we consider situations with regions of 0-, 1-, and 2-dimensional near-contact, separated by a decreasing distance $\delta > 0$ *(top row)*. Our adaptive strategy captures the correct rate of energy growth, whereas a fixed midpoint scheme (dashed lines) fails to approach infinity even under substantial refinement *(middle row)*. Meanwhile, the adaptive scheme is not significantly more expensive, even for higher-dimensional near contact *(bottom row)*.

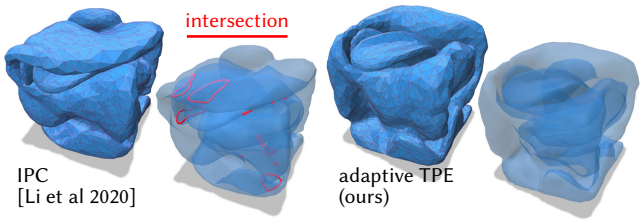


Fig. 18. IPC provides a logarithmic collision barrier and relies on continuous collision detection (CCD) as a failsafe due to the only weak repulsive forces. *Left:* without CCD, using the IPC penalty as Φ in Equation 25 can yield large self-intersections. *Right:* here our adaptive tangent-point energy still nicely prevents collision, even without expensive checks.

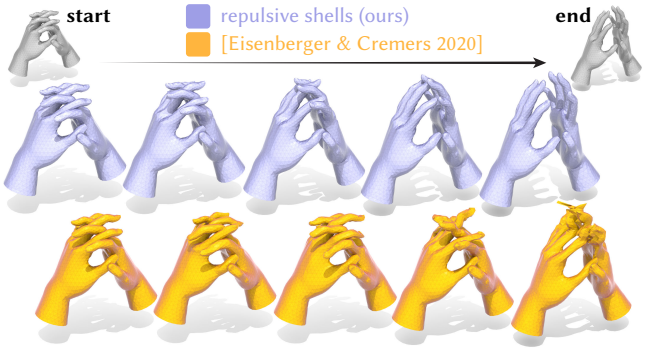


Fig. 19. Methods based on injectively deforming the space around an object can struggle with shapes in near-contact due to insufficient spatial resolution. Here we compare shape interpolation via our method *(top)* versus a method based on divergence-free volumetric flows *(bottom)*.

7.3 Energy Minimization

The combination of elastic and tangent-point energy is useful for applications beyond the shape space framework—in principle, one could use the repulsive energy as a regularizer in any scenario where an elastic energy is already used [Bartels et al. 2022]. Here, we will specifically consider the variational problem

$$\min_{\mathbf{x}} \ \mathcal{W}(\mathbf{x}_0, \mathbf{x}) + \beta \Phi(\mathbf{x}), \tag{25}$$

where $\mathbf{x}_0$ is a given reference configuration as a basic minimization task before moving on to interpolation and extrapolation. We will add further terms to Equation 25 for each application.

7.3.1 Nonrigid Packing. To pack nonrigid objects into tight spaces (as might be useful for, *e.g.*, 3D printing), we can add a barrier term to Equation 25 that forces the surface to stay inside a given domain $\Omega \subset \mathbb{R}^3$. In general, if $\phi(x)$ is the signed distance to the domain boundary $\partial\Omega$, we use a barrier $1/\phi^2(x)$. For instance, in Figure 21 we use a barrier

$$\mathcal{E}_{\text{box}}(\mathbf{x}) := \sum_{v \in V} \sum_{i=1}^{3} (l_i - \mathbf{x}_{v,i})^{-2} + (\mathbf{x}_{v,i} - u_i)^{-2}$$

representing a rectangular box, where $l, u \in \mathbb{R}^3$ are the lower and upper bounds of the box, *resp.* In Figure 22 we instead use a spherical

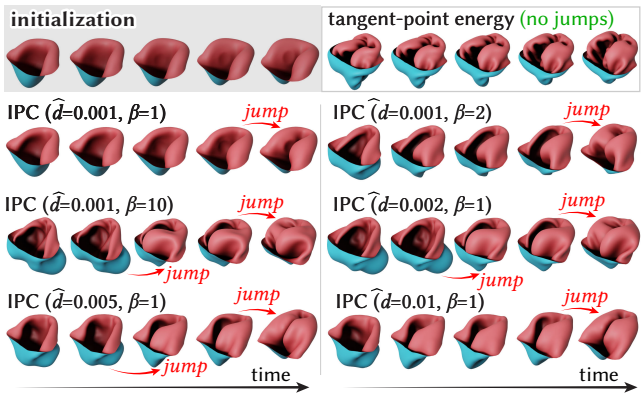


Fig. 20. When using the IPC barrier as potential energy in our framework, we struggle to find geodesics without sudden "jumps" (more obvious in supplementary video), even with good initialization *(top left)*. Here using the IPC barrier energy for repulsion fails to yield a smooth surface eversion, even after extensive parameter tuning *(bottom three rows)*. A TPE-based formulation easily finds smooth trajectories without jumps *(top right)*.

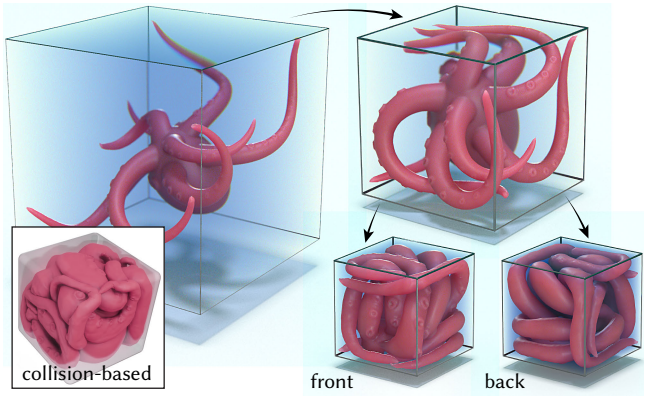


Fig. 21. Progressively packing an octopus *(top left)* into a small box *(bottom right)* with a long-range repulsive potential yields a well-separated packing that nicely preserves local geometric features. The result is qualitatively different from a collision-based packing *(inset, using reference IDP code)*, which prevents interpenetration but may not yield a nice global arrangement.

barrier

$$\mathcal{E}_{\text{sphere}}(\mathbf{x}) := \sum_{v \in V} \frac{1}{(r - |\mathbf{x}_v|)^2}.$$

In both cases we then reduce the size of the bounding domain step-by-step and thus get a tight packing of the shapes into successively smaller regions. Unlike Yu et al. [2021a], which "forgets" all surface detail from the original domain, this strategy nicely preserves surface detail. Moreover, in contrast to a surface packing via artificial time integration of physical dynamics, which merely forbids collisions, a global repulsive energy seeks to *maximize* separation distance between all points—uniformly distributing gap sizes throughout the domain. Compare for instance Figure 21, *bottom left*, computed via the code from *interactive deformation processing (IDP)* [Fang et al. 2021], versus our results in the same figure. Elastic

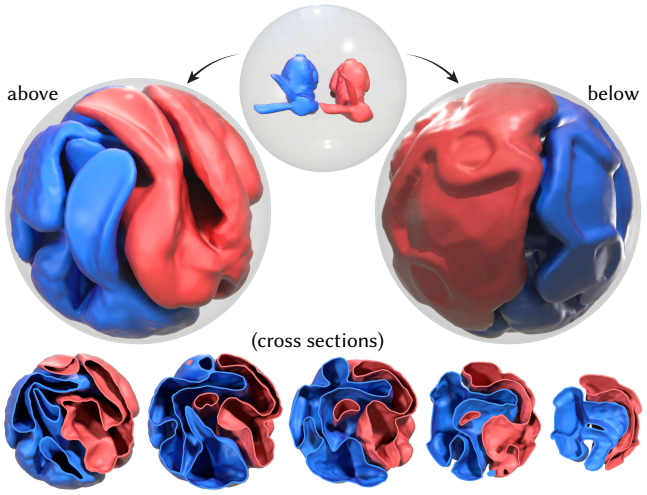


Fig. 22. Progressively packing two bunnies into a smaller and smaller sphere *(top)* yields surfaces that are not only intersection-free, but also nicely distributed throughout the volume *(bottom)*.

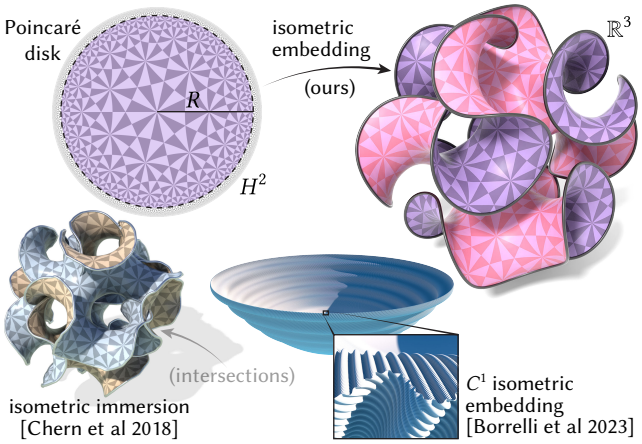


Fig. 23. We can use our framework to faithfully visualize abstract metrics. Here we isometrically embed a large piece of the hyperbolic plane H^2, obtaining an surface that is both intersection-free and more regular than previous embeddings.

surface packing with large inter-surface "padding" may be particularly interesting for digital fabrication, *e.g.*, printing pre-stressed deformable objects in a silicone mold [Alderighi et al. 2018].

7.3.2 Isometric Embeddings. Many phenomena occurring in nature experience internal growth that leads to an external change in shape—consider for instance a growing leaf, where local division and growth of cells induces buckling and rippling in the observed geometry. Here, embeddings of so-called *hyperbolic metrics* can help explain growth patterns of "frilly" plants in nature [Yamamoto et al. 2021]. More generally, the question of how an abstract surface with an intrinsic notion of length can be faithfully embedded in space is a fundamental challenge in natural sciences, engineering,

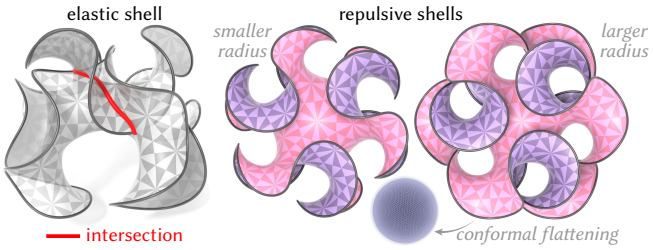


Fig. 24. Without the repulsive term, attempts at hyperbolic embedding yield large self-intersections *(left)*. Long-range repulsive forces also help to encourage global symmetry *(center, right)*. A conformal flattening of the final embedded mesh helps verify that the embedding is indeed nearly isometric *(bottom right)*.

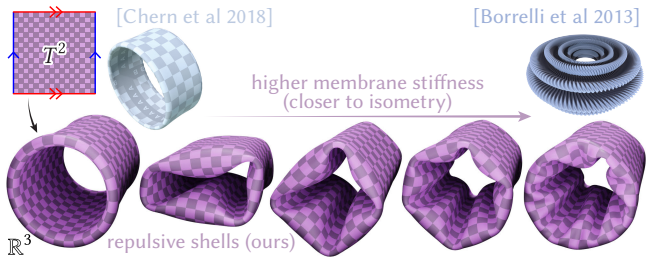


Fig. 25. Here we compute near-isometric embeddings of a flat metric on the torus T^2. As we increase membrane stiffness (hence get closer to isometry), corrugations naturally arise *(bottom)*—reminiscent of a Nash–Kuiper embedding *(top right)*, and suggesting better agreement with the target metric than Chern et al. [2018] *(top left)*.

and design [Van Rees et al. 2017]—as well as a variety of computer graphics and geometry processing algorithms [Chern et al. 2018]. Formally, given an abstract surface M with a Riemannian metric g, we seek an embedding $\mathbf{x}\colon M \to \mathbb{R}^3$ that exhibits the same metric, *i.e.*, $g_p(u,v) = \langle \mathrm{d}_p\mathbf{x}(u), \mathrm{d}_p\mathbf{x}(v)\rangle$ for all tangent vectors $u, v \in T_pM$ at all points p.

A number of algorithms have been developed to solve the isometric embedding problem. For instance, inspired by the *Nash–Kuiper theorem* [Nash 1954; Kuiper 1955], Borrelli et al. [2012] developed an algorithm based on convex integration, producing beautiful images of isometric embeddings, including the hyperbolic plane [Borrelli et al. 2023]. However, these surfaces are only C^1, achieved by making infinitely-fine corrugations (see Figure 23, *(bottom right)*). Chern et al. [2018] consider smoother isometric *immersions*, *i.e.*, maps $x\colon M \to \mathbb{R}^3$ that locally induce the right metric, but can exhibit significant self-intersections. The combination of a repulsive and elastic energy enables us to obtain maps that are *embedded* and also much smoother than just C^1. Like Chern et al. [2018], we make no claim that such maps can always be found.

Isometric embedding becomes more concrete in the discrete setting: given only mesh connectivity $\mathsf{M} = (\mathsf{V}, \mathsf{E}, \mathsf{T})$ and an assignment of edge lengths $\ell\colon \mathsf{E} \to \mathbb{R}_{>0}$ satisfying the usual triangle inequalities, we seek vertex positions $x_i \in \mathbb{R}^3$ such that $|x_i - x_j| = \ell_{ij}$ for all $ij \in E$. We find embedded surfaces that approximate this condition by minimizing the sum of tangent-point energy, a small bending energy which seeks dihedral angles as close as possible to zero, and a membrane term that tries to match the given edge lengths. Here we use a much larger membrane weight than usual, to obtain near-isometric embeddings. The tangent-point energy also serves as an implicit bending term that provides further regularization. Figure 23 shows one example where we embed a large disk from the hyperbolic plane H^2. To aid optimization we start with a small disk in the plane, and progressively grow the radius R (by uniformly scaling the domain vertices). To verify that the embedding is close to isometric, we compute a conformal flattening with minimal area distortion via [Sawhney and Crane 2017], yielding the expected scale factors over the *Poincaré disk* (Figure 24).

7.4 Interpolation and Extrapolation

We next evaluate our method's ability to generate natural shape interpolation and extrapolation, without use of a rig or any other additional information.

7.4.1 Interpolation. The most basic interpolation task is to find a geodesic between given configurations $\mathbf{x}^0$ and $\mathbf{x}^n$. For instance, Figure 30 shows an interpolation between two poses of the human body going from arms behind the back, to arms in front of the torso. Notice that, unlike purely elastic interpolation or interpolation of rig parameters, our formulation naturally and automatically finds a global intersection-free motion. Figure 11 shows a similar example where we find an intersection-free interpolation between two different interleavings of the hands. Since we maintain a near-constant level of repulsive energy throughout the trajectory, the hands remain close to each other, sliding almost tangentially while still exhibiting natural bending of the fingers. Again, no rig is used. As noted in Section 6.1.1, we initialize the path with a separated middle state to provide an initial "hint" to the optimizer. To generate Figure 14 we first pack the camel into a small tube *à la* Section 7.3.1; we then use this packed state as the endpoint $\mathbf{x}^n$ of a geodesic from the original camel $\mathbf{x}^0$.

7.4.2 Inversion. A classic challenge in mathematical visualization is to "turn a surface inside out," *i.e.*, to construct a continuous, regular family of surfaces between some given surface, and the same surface with opposite orientation. The most famous example is the so-called *sphere eversion* [Sullivan 1999], where the key trick is to minimize an energy starting from a symmetric *mid-surface*. Recently, Chern et al. [2018] also compute near-isometric eversions, where the metric is closely preserved. This past work largely considers families of *immersions* which permit self-intersections—which in the spherical case are unavoidable for topological reasons. For this reason, the problem has remained quite abstract for the general public, requiring significant exposition to even understand the rules of the game [Thurston et al. 1994].

Our repulsive shape space enables us to illustrate more broadly accessible surface eversions, where the surface cannot pass through itself in a nonphysical way. To do so, we must of course consider surfaces of different topology, such as disk-like surfaces. A basic example is shown in Figure 29 where a sphere with a hole is everted

Fig. 26. Turning laundry right side out. For each article of clothing we compute a geodesic between a mesh with original and reversed dihedral angles. Intricate, collision-free deformation, like fingertips being inverted as they are pulled through the glove, emerge naturally from the definition of our repulsive metric.

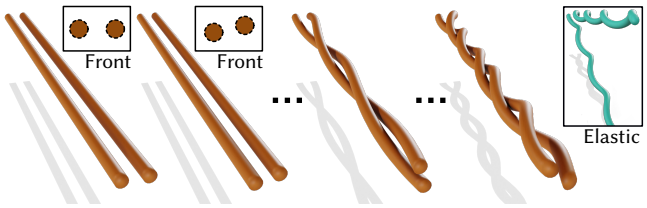


Fig. 27. Twisting two cylinders via extrapolation starting from the two left-most pairs. Without repulsive term, extrapolation would move the cylinders through each other and twist them apart (inset image). However, the repulsive shells exponential map twists the surface into tight configurations.

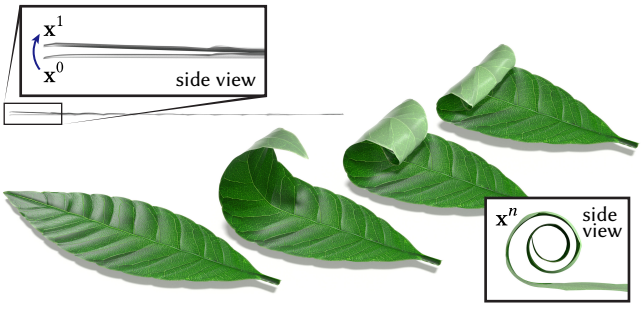


Fig. 28. A leaf naturally rolls up by extrapolating a tiny bend at the tip *(top left)*. Since the initial motion from $\mathbf{x}^0$ to $\mathbf{x}^1$ slightly *increases* repulsive energy, the exponential map gradually brings the leaf closer and closer to itself, approaching self-intersection only at time $t = \infty$. Meanwhile, the elastic part of the metric preserves surface detail and guides the overall motion.

using both purely an elastic shell geodesic (producing immersions) and a repulsive shell geodesic (producing intersection-free embeddings). Increasing membrane stiffness in our elastic energy drives this family closer and closer to an *isometric* eversion. An even more relatable example is shown in Figure 26, where we use our algorithm to solve the vexing problem of turning washed laundry outside-in.

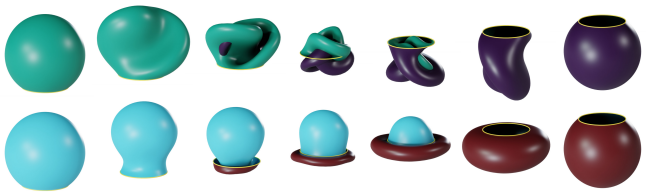


Fig. 29. Inversion of cut-open sphere. Top: Discrete geodesic interpolation without repulsion (green to purple). Bottom: Geodesic interpolation with repulsion (blue to red). The fixed surface boundary is shown as yellow curve.

For each of these examples, we take a given mesh $\mathbf{x}^0$ and construct the reversed mesh $\mathbf{x}^n$ by flipping the sign of all dihedral angles, and recovering the corresponding embedding via the method of Wang et al. [2012]. We then compute a geodesic between $\mathbf{x}^0$ and $\mathbf{x}^n$. For more challenging examples, like the clothing, we also construct a flat mid-surface $\mathbf{x}^{n/2}$ by minimizing an elastic energy where the target edge lengths are equal to the initial ones, and all the dihedral angles are constrained to zero. We then compute a piecewise geodesic from $\mathbf{x}^0$ to $\mathbf{x}^{n/2}$, then to $\mathbf{x}^n$ which is coarse in time. This piecewise geodesic is then relaxed into a single geodesic minimizing overall path energy. Finally, by progressively reducing the bending stiffness and refining in time, we then obtain a smooth, near-isometric motion.

7.4.3 Rigid Motions. As noted in Section 4.2, the repulsive shell space factors out rigid motions (as do many other shape spaces), in order to make path energy minimization well-posed. Hence, we have effectively optimized the *shape*, but not its motion in $\mathbb{R}^n$. How to best add rigid motion to given surface dynamics is an interesting question [Gross et al. 2023]; we here take an elementary approach that simply penalizes translation and rotation in our path energy. In particular, we add terms

$$\mathcal{E}_{\text{trans}}(\mathbf{x}^0, \ldots, \mathbf{x}^n) = n \sum_{k=1}^{n} \|\bar{\mathbf{x}}^k - \bar{\mathbf{x}}^{k-1}\|^2, \tag{26}$$

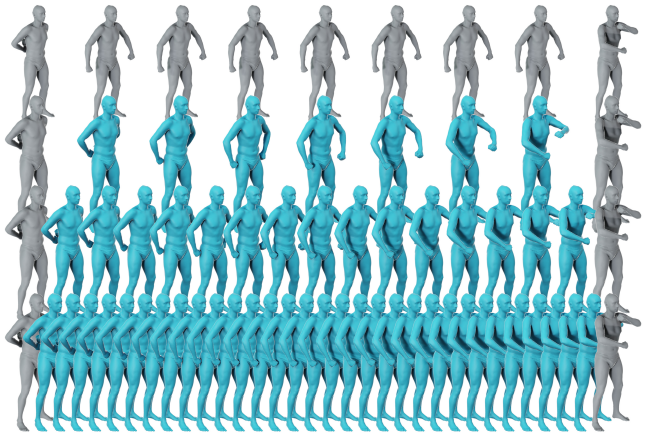


Fig. 30. Although performing space-time optimization of an all-pairs energy sounds expensive, a principled hierarchical scheme enables fast, predictive previews with a smaller number of time steps *(second row to bottom)* starting from piecewise constant initializations *(first row)*.

which penalizes movement of the barycenters $\bar{\mathbf{x}}$ and

$$\mathcal{E}_{\text{rot}}(\mathbf{x}^0, \ldots, \mathbf{x}^n) = n \sum_{i=1}^{n} \left\| \sum_{\mathsf{v} \in \mathsf{V}} (\mathbf{x}_\mathsf{v}^k - \mathbf{x}_\mathsf{v}^{k-1}) \times \mathbf{x}_\mathsf{v}^{k-1} \right\|^2, \qquad (27)$$

which penalizes the angular momentum of the deformations.

We use this, for example, in Figure 4, where we interpolate between two translations of a sphere. Furthermore, a barrier with a hole is placed between them. Here the barrier is incorporated in the path energy by adding a tangent-point barrier Φ^{barrier} (*à la* [Yu et al. 2021a]) to the usual repulsive penalty Φ. As a result, the sphere avoids contact with both the barrier and itself. Here, we also achieve different surface behavior by adjusting the elastic material parameters (Figure 31). Although the tangent-point energy also adds some additional bending-like energy, we did not observe this to have a noticeable effect on the choice of elastic parameters.

7.4.4 Extrapolation. Finally, we consider shape extrapolation via the exponential map. Note that extrapolated trajectories in the repulsive shell space are fundamentally different from dynamical trajectories in physics-based collision simulation: our trajectories continue to "go straight" no matter what, whereas dynamical elastic trajectories responding to a collision penalty slow down as the approach an obstacle and eventually turn around. For instance, if our initial velocity significantly increases or decreases the repulsive energy, this same rough rate of increase/decrease will continue along the entire motion. For instance, in Figure 27 the exponential map happily twists the surface into a tighter and tighter configuration—whereas a collision penalty will resist with greater and greater force as twisting increases. Figure 28 shows another interesting example, where an artist bends the tip of a leaf, and our repulsive exponential map naturally curls it into a dormant configuration. Figure 32 verifies that our exponential map is well-behaved even in situations of close contact—here we simply take two frames from our geodesic interpolation of hands, and extrapolate them forward in time to reproduce a close clasping pose.

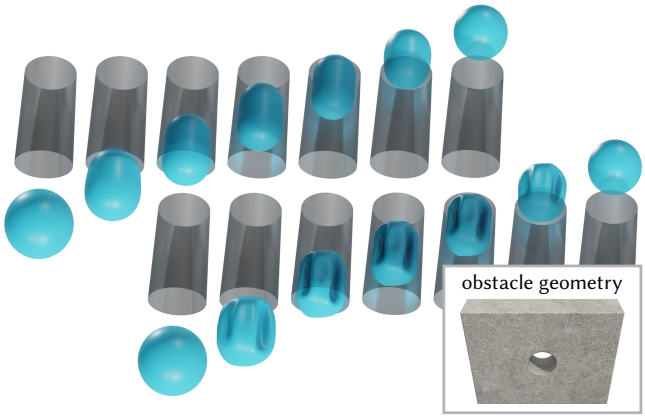


Fig. 31. The optimal trajectory of a surface depends on the interaction between elastic and repulsive forces—here we show variations of Figure 4 with stronger *(top)* and weaker *(bottom)* bending stiffness. The full obstacle geometry is shown in the inset.

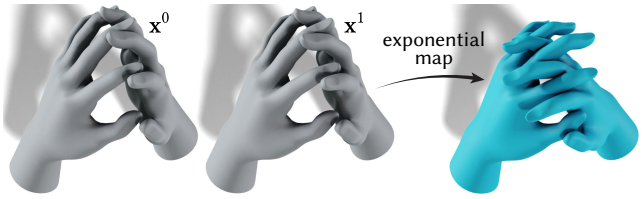


Fig. 32. Modeling intersection-free poses by hand can be challenging. Here we use extrapolation to get a tighter fit between fingers *(blue)*, starting with a pair of configurations that are much easier to model *(gray)*.

8 LIMITATIONS AND FUTURE WORK

Our framework provides the foundation for many useful extensions. For instance, our surface-based repulsion potential could be coupled with 3D volumetric elasticity (rather than a 2D shell model) to better model certain kinds of physical behavior. Or, in lieu of the full set of vertex positions, one could parameterize the shape space by degrees of freedom from a reduced model—such as the actuators for a soft robot, parameters in a learned latent space [Sharp et al. 2023], the parameters of a skeletal rig, or a few sparse handles [Sumner et al. 2005]. There are also a variety of potential improvements to the mathematical and numerical formulation detailed below.

Surfaces with Boundary. To date, the tangent-point energy has been shown to be repulsive only for surfaces without boundary [Strzelecki and von der Mosel 2013]. A pathological example is two disconnected planar components sharing a common plane. In practice, we find that both fixed and free boundary conditions tend to work well in practice (Figures 11, 23, 26, 29, 28). Alternatively, it may also be sufficient to add TPE of the boundary curve (*à la* Yu et al. [2021a]), and perhaps a boundary-surface term, to the energy.

Numerics. The elastic Hessian used in Section 6.1 is much simpler to implement than the fractional Sobolev inner product of Yu et al. [2021a], but adding the latter to our preconditioner (times the repulsion strength β) may improve performance in scenarios of close,

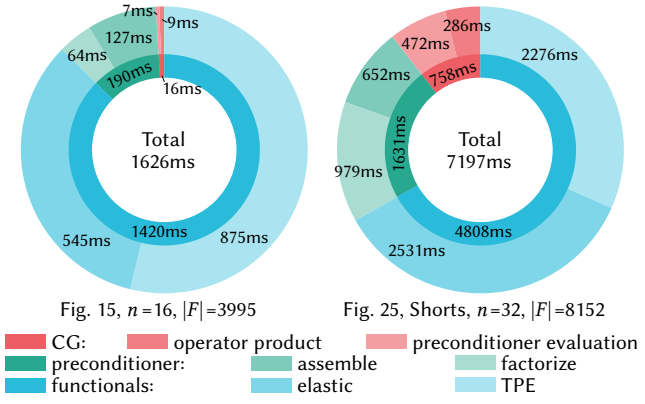


Fig. 15, $n=16$, $|F|=3995$ Fig. 25, Shorts, $n=32$, $|F|=8152$

Fig. 33. Breakdown of computational cost for two examples, showing timings for different steps of a trust region iteration and their various substeps. Timings were measured on an Intel i7 1260p laptop with 4 parallel threads. Notice that the cost of TPE is not much different from a local elastic energy.

high-dimensional contact (see Figure 33, *right*). For interpolation problems, initialization may require some care for situations involving nontrivial deformation and/or close contact. As detailed in Section 6.1.1, however, we found that simple initialization schemes already work quite well (*e.g.*, providing a single "middle" pose). Moreover, this challenge is not unique to our shape space: one runs into similar initialization challenges even with the standard (non-repulsive) space of elastic shells [Heeren 2017]. More intelligent initialization schemes would hence broadly benefit shape space methods. Finally, unlike Yu et al. [2021a] we do not perform adaptive remeshing, which would be helpful in scenarios of large tangential shearing (*e.g.*, global isotopies *à la* Yu et al. [2021a, Figure 1]).

Alternative Potentials. The tangent-point energy has known challenges. For instance, it adds non-local bending regularization that inhibits formation of sharp bends and wrinkles [Yu et al. 2021b, Section 3.2]. As mentioned in Section 1, however, there is no reason why we must use TPE for our shape space. For instance, although the standard IPC potential appears not to work well for the examples in this paper (Section 7.2.2), a recent, unpublished extension of IPC explores a continuous interpretation of IPC [Li et al. 2023]. It remains to be seen whether this energy provides a useful long-range potential for large values of the cutoff distance $\widehat{d}$. A more well-established potential is the *cosine energy* of Kusner and Sullivan [1994], which was designed for long-range surface untangling, and is based on a well-defined continuous *Möbius energy* [O'Hara 1991]. As explained by Freedman et al. [1994] and illustrated by Yu et al. [2021b, Figure 3], Möbius energy suffers from a "pull tight" phenomenon that may make it unsuitable for modeling physical surfaces—but our elastic term may make Möbius energy useful in the shape space context.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) via project 212212052, project 211504053 – Collaborative Research Center 1060, and Germany's Excellence Strategy project 390685813 – Hausdorff Center for Mathematics, an NSF CAREER Award (IIS 1943123), NSF Award IIS 2212290, a Packard Fellowship, the German-Israeli Foundation for Scientific Research and Development (grant number I-1339-407.6/2016), and gifts from Facebook Reality Labs, and Google, Inc. Furthermore, this project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101034255. Mesh diagrams were created with *Penrose* [Ye et al. 2020].

REFERENCES

Thomas Alderighi, Luigi Malomo, Daniela Giorgi, Nico Pietroni, Bernd Bickel, and Paolo Cignoni. 2018. Metamolds: computational design of silicone molds. *ACM Trans. Graph.* 37, 4 (2018), 1–13.

Mishal Assif, Ravi Banavar, Anthony Bloch, Margarida Camarinha, and Leonardo Colombo. 2018. Variational collision avoidance problems on Riemannian manifolds. In *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2791–2796.

John M. Ball. 1981. Global invertibility of Sobolev functions and the interpenetration of matter. *Proceedings of the Royal Society of Edinburgh* 88A (1981), 315–328.

Jayanth R. Banavar, Oscar Gonzalez, John H. Maddocks, and Amos Maritan. 2003. Self-interactions of strands and sheets. *Journal of Statistical Physics* 110, 1-2 (2003), 35–50.

Jernej Barbič, Marco da Silva, and Jovan Popović. 2009. Deformable object animation using reduced optimal control. In *ACM SIGGRAPH 2009 papers*. 1–9.

Sören Bartels, Frank Meyer, and Christian Palus. 2022. Simulating Self-Avoiding Isometric Plate Bending. *SIAM Journal on Scientific Computing* 44, 3 (2022), A1475–A1496.

Rick Beatson and Leslie Greengard. 1997. A short course on fast multipole methods. *Wavelets, multilevel methods and elliptic PDEs* 1 (1997), 1–37.

M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes. 2005. Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. *International Journal of Computer Vision* 61, 2 (2005), 139–157.

Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. 2007. Tracks: toward directable thin shells. *ACM Trans. Graph.* 26, 3 (2007), 50–es.

Viktor Ivanovich Blagodatskikh and Aleksei Fedorovich Filippov. 1985. Differential inclusions and optimal control. *Trudy Mat. Inst. Steklov* 169 (1985), 194–252.

Simon Blatt. 2013. The Energy Spaces of the Tangent Point Energies. *Journal of Topology and Analysis* 5, 3 (2013), 261–270.

Vincent Borrelli, Roland Denis, Francis Lazarus, Mélanie Theillière, and Boris Thibert. 2023. The Hyperbolic Plane in $\mathbb{E}^3$. arXiv:math.DG/2303.12449

Vincent Borrelli, Said Jabrane, Francis Lazarus, and Boris Thibert. 2012. Flat tori in three-dimensional space and convex integration. *Proceedings of the National Academy of Sciences* 109, 19 (2012), 7218–7223.

Mario Botsch, Stephan Steinberg, Stephan Bischoff, and Leif Kobbelt. 2002. OpenMesh–a generic and efficient polygon mesh data structure. In *1st OpenSG Symposium*, Vol. 18.

Stephen P Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Gregory Buck and Jeremey Orloff. 1995. A simple energy function for knots. *Topology and its Applications* 61, 3 (1995).

Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2018. Shape from metric. *ACM Trans. Graph.* 37, 4 (2018), 1–17.

Stelian Coros, Sebastian Martin, Bernhard Thomaszewski, Christian Schumacher, Robert Sumner, and Markus Gross. 2012. Deformable objects alive! *ACM Trans. Graph.* 31, 4 (2012), 1–9.

Keenan Crane and Max Wardetzky. 2017. A Glimpse Into Discrete Differential Geometry. *Notices of the American Mathematical Society* 64, 10 (2017), 1153–1159.

T. A. Davis. 2006. *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA.

Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2021. DiffPD: Differentiable Projective Dynamics. *ACM Trans. Graph.* 41, 2 (2021), 1–21.

Marvin Eisenberger and Daniel Cremers. 2020. Hamiltonian Dynamics for Real-World Shape Interpolation. In *European Conference on Computer Vision*. 179–196.

Preston R Fairchild, Vaibhav Srivastava, and Xiaobo Tan. 2021. Efficient path planning of soft robotic arms in the presence of obstacles. *IFAC-PapersOnLine* 54, 20 (2021), 586–591.

Yu Fang, Minchen Li, Chenfanfu Jiang, and Danny M. Kaufman. 2021. Guaranteed Globally Injective 3D Deformation Processing. *ACM Trans. Graph.* 40, 4 (2021), 13.

Razvan Constantin Fetecau. 2003. *Variational methods for nonsmooth mechanics*. California Institute of Technology.

P. Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. 2004. Principal geodesic analysis for the study of nonlinear statistics of shape. In *IEEE Transactions on Medical Imaging*, Vol. 23. 995–1005.

Michael H Freedman, Zheng-Xu He, and Zhenghan Wang. 1994. Mobius energy of knots and unknots. *Annals of mathematics* 139, 1 (1994), 1–50.
Shinji Fukuhara. 1988. Energy of a knot. In *A fête of topology*. Elsevier, 443–451.
Russell Gayle, Ming C Lin, and Dinesh Manocha. 2005a. Constraint-based motion planning of deformable robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 1046–1053.
Russell Gayle, Paul Segars, Ming C Lin, and Dinesh Manocha. 2005b. Path planning for deformable robots in complex environments. In *Robotics: science and systems*. 225–232.
Oscar Gonzalez and John H. Maddocks. 1999. Global curvature, thickness, and the ideal shapes of knots. *Proceedings of the National Academy of Sciences of the United States of America* 96, 9 (1999), 4769–4773.
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*. Eurographics Association, 62–67.
Oliver Gross, Yousuf Soliman, Marcel Padilla, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2023. Motion from Shape Change. *ACM Trans. Graph.* 42, 4 (2023), 1–11.
Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen V3. (2010). http://eigen.tuxfamily.org
David Harmon, Daniele Panozzo, Olga Sorkine, and Denis Zorin. 2011. Interference-aware geometric modeling. *ACM Trans. Graph.* 30, 6 (2011), 1–10.
Behrend Heeren. 2017. *Numerical methods in shape spaces and optimal branching patterns*. Ph.D. Dissertation. Universitäts-und Landesbibliothek Bonn.
Behrend Heeren, Martin Rumpf, Peter Schröder, Max Wardetzky, and Benedikt Wirth. 2014. Exploring the Geometry of the Space of Shells. *Comput. Graph. Forum* 33, 5 (2014), 247–256.
Behrend Heeren, Martin Rumpf, Peter Schröder, Max Wardetzky, and Benedikt Wirth. 2016. Splines in the Space of Shells. *Comput. Graph. Forum* 35, 5 (2016), 111–120.
Behrend Heeren, Martin Rumpf, Max Wardetzky, and Benedikt Wirth. 2012. Time-Discrete Geodesics in the Space of Shells. *Comput. Graph. Forum* 31, 5 (2012), 1755–1764.
Behrend Heeren and Josua Sassen. 2020. *The Geometric Optimization and Simulation Toolbox (GOAST)*. https://gitlab.com/numod/goast
Behrend Heeren, Chao Zhang, Martin Rumpf, and William Smith. 2018. Principal Geodesic Analysis in the Space of Discrete Shells. *Comput. Graph. Forum* 37, 5 (2018), 173–184.
Roland Herzog and Estefanía Loayza-Romero. 2022. A manifold of planar triangular meshes with complete Riemannian metric. *Math. Comp.* 92 (2022), 1–50. Issue 339.
Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial complex augmentation framework for bijective maps. *ACM Trans. Graph.* 36, 6 (2017).
Hermann Karcher. 2014. Riemannian center of mass and so called karcher mean. arXiv:1407.2087
Danny M Kaufman, Timothy Edmunds, and Dinesh K Pai. 2005. Fast frictional dynamics for rigid bodies. In *ACM SIGGRAPH 2005 Papers*. 946–956.
Martin Kilian, Niloy J Mitra, and Helmut Pottmann. 2007. Geometric modeling in shape space. *ACM Trans. Graph.* 26, 3 (2007), 64.
Holger Klein, Noémie Jaquier, Andre Meixner, and Tamim Asfour. 2023. On the Design of Region-Avoiding Metrics for Collision-Safe Motion Generation on Riemannian Manifolds. arXiv:2307.15440
Nicolaas H Kuiper. 1955. On C^1-isometric imbeddings. I, II. *Nederl. Akad. Wetensch. Proc. Ser. A.* **58** = *Indag. Math.* 17 (1955), 545–556,683–689.
Robert B Kusner and John M Sullivan. 1994. *Möbius Energies for Knots and Links, Surfaces and Submainfolds*. Citeseer.
Robert B Kusner and John M Sullivan. 1998. Möbius-invariant knot energies. *Ideal knots* 19 (1998), 315–352.
Lei Lan, Yin Yang, Danny Kaufman, Junfeng Yao, Minchen Li, and Chenfanfu Jiang. 2021. Medial IPC: accelerated incremental potential contact with medial elastics. *ACM Trans. Graph.* 40, 4 (2021).
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.* 39, 4 (2020), 20.
Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2023. Convergent Incremental Potential Contact. arXiv:2307.15908
Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph.* 40, 4 (2021), 24.
Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. 2022. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Trans. Graph.* 42, 1 (2022), 1–20.
Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
Michael Miller, Ayananshu Banerjee, Gary Christensen, Sarang Joshi, Navin Khaneja, Ulf Grenander, and Larissa Matejic. 1997. Statistical methods in computational anatomy. *Statistical methods in medical research* 6, 3 (1997), 267–299.
Jan Möbius and Leif Kobbelt. 2012. OpenFlipper: An Open Source Geometry Processing and Rendering Framework. In *Curves and Surfaces*. Lecture Notes in Computer Science, Vol. 6920. Springer, 488–500.
Jean J Moreau. 1988. Unilateral contact and dry friction in finite freedom dynamics. In *Nonsmooth mechanics and Applications*. Springer, 1–82.
Jean-Marie Morvan and Boris Thibert. 2002. On the approximation of a smooth surface with a triangulated mesh. *Computational Geometry* 23, 3 (2002), 337–352.
William Moss, Ming C Lin, and Dinesh Manocha. 2008. Constraint-based motion synthesis for deformable models. *Computer Animation and Virtual Worlds* 19, 3-4 (2008), 421–431.
Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air meshes for robust collision handling. *ACM Trans. Graph.* 34, 4 (2015), 1–9.
John Nash. 1954. C^1 Isometric Imbeddings. *The Annals of Mathematics* 60, 3 (1954), 383.
Jorge Nocedal and Stephen J. Wright. 2006. *Numerical optimization* (second ed.). Springer, New York. xxii+664 pages.
Jun O'Hara. 1991. Energy of a knot. *Topology* 30, 2 (1991), 241–247.
Stanley Osher and Ronald P Fedkiw. 2005. *Level set methods and dynamic implicit surfaces*. Vol. 1. Springer New York.
Samuel Rodriguez, Jyh-Ming Lien, and Nancy M Amato. 2006. Planning motion in completely deformable environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*. IEEE, 2466–2471.
Martin Rumpf and Benedikt Wirth. 2015. Variational time discretization of geodesic calculus. *IMA J. Numer. Anal.* 35, 3 (2015), 1011–1046.
Josua Sassen, Behrend Heeren, Klaus Hildebrandt, and Martin Rumpf. 2020a. Geometric optimization using nonlinear rotation-invariant coordinates. *Computer Aided Geometric Design* 77 (2020), 101829.
Josua Sassen, Klaus Hildebrandt, and Martin Rumpf. 2020b. Nonlinear Deformation Synthesis via Sparse Principal Geodesic Analysis. *Comput. Graph. Forum* 39, 5 (2020), 119–132.
Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Trans. Graph.* 37, 1 (2017), 5:1–5:14.
Henrik Schumacher. 2023. *Repulsor*. https://github.com/HenrikSchumacher/Repulsor
Nicholas Sharp, Cristian Romero, Alec Jacobson, Etienne Vouga, Paul Kry, David IW Levin, and Justin Solomon. 2023. Data-Free Learning of Reduced-Order Kinematics. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–9.
Paweł Strzelecki and Heiko von der Mosel. 2013. Tangent-point repulsive potentials for a class of non-smooth m-dimensional sets in $\mathbb{R}^n$. Part I: Smoothing and self-avoidance effects. *Journal of Geometric Analysis* 23, 3 (2013), 1085–1139.
John M Sullivan. 1999. "The Optiverse" and other sphere eversions. arXiv:math/9905020
Robert W Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. 2005. Mesh-based inverse kinematics. *ACM Trans. Graph.* 24, 3 (2005), 488–495.
William Thurston, Silvio Levy, Delle Maxwell, Tamara Munzner, Nathaniel Thurston, Stuart Levy, David Ben-Zvi, Daeron Meyer, Adam Deaton, Dan Krech, Matt Headrick, Mark Phillips, Celeste Fowler, Charlie Gunn, Stephanie Mason, Linus Upson, Scott Wisdom, Karen McNenny, and Paul de Cordova. 1994. Outside In. The Geometry Center.
Wim M Van Rees, Etienne Vouga, and Lakshminarayanan Mahadevan. 2017. Growth patterns for shape-shifting elastic bilayers. *Proceedings of the National Academy of Sciences* 114, 44 (2017), 11597–11602.
Christoph von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. 2015. Real-Time Nonlinear Shape Interpolation. *ACM Trans. Graph.* 34, 3 (2015), 34:1–34:10.
Yuanzhen Wang, Beibei Liu, and Yiying Tong. 2012. Linear surface reconstruction from discrete fundamental forms on triangle meshes. *Comput. Graph. Forum* 31, 8 (2012), 2277–2287.
Benedikt Wirth, Leah Bar, Martin Rumpf, and Guillermo Sapiro. 2011. A Continuum Mechanical Approach to Geodesics in Shape Space. *International Journal of Computer Vision* 93, 3 (2011), 293–318.
Chris Wojtan, Peter J Mucha, and Greg Turk. 2006. Keyframe control of complex particle systems using the adjoint method. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 15–23.
Kenneth K. Yamamoto, Toby L. Shearman, Erik J. Struckmeyer, John A. Gemmer, and Shankar C. Venkataramani. 2021. Nature's forms are frilly, flexible, and functional. *The European Physical Journal E* 44, 7 (2021), 95.
Katherine Ye, Wode Ni, Max Krieger, Dor Ma'ayan, Jenna Wise, Jonathan Aldrich, Joshua Sunshine, and Keenan Crane. 2020. Penrose: from mathematical notation to beautiful diagrams. *ACM Trans. Graph.* 39, 4 (2020), 144–1.
Laurent Younes. 2010. *Shapes and Diffeomorphisms*. Springer.
Chris Yu, Caleb Brakensiek, Henrik Schumacher, and Keenan Crane. 2021a. Repulsive Surfaces. *ACM Trans. Graph.* 40, 6 (2021), 1–19.
Chris Yu, Henrik Schumacher, and Keenan Crane. 2021b. Repulsive Curves. *ACM Trans. Graph.* 40, 2 (2021). arXiv:2006.07859
Yufeng Zhu, Jovan Popović, Robert Bridson, and Danny M. Kaufman. 2017. Planar interpolation with extreme deformation, topology change and dynamics. *ACM Trans. Graph.* 36, 6, Article 213 (2017), 15 pages.

Received January 2024