

Discrete Quadratic Curvature Energies

Max Wardetzky^a Miklós Bergou^b David Harmon^b Denis Zorin^c
Eitan Grinspun^b

^a*Freie Universität Berlin*

^b*Columbia University*

^c*New York University*

Abstract

We present a family of *discrete isometric bending models* (IBMs) for triangulated surfaces in 3-space. These models are derived from an axiomatic treatment of discrete Laplace operators, using these operators to obtain linear models for discrete mean curvature from which bending energies are assembled. Under the assumption of *isometric surface deformations* we show that these energies are quadratic in surface positions. The corresponding linear energy gradients and constant energy Hessians constitute an efficient model for computing bending forces and their derivatives, enabling fast time-integration of cloth dynamics with a two- to three-fold net speedup over existing nonlinear methods, and near-interactive rates for Willmore smoothing of large meshes.

Key words: cloth simulation, thin plates, Willmore flow, bending energy, discrete Laplace operator, discrete mean curvature, non-conforming finite elements.

1 Introduction

Curvature-based energies play a principal role in the description of both physical and non-physical systems: simulations of cloth and garments rely on bending energies that govern the mechanics of elastic thin plates and shells [14,47]; geometry processing operations such as smoothing, denoising, and hole-filling rely on curvature functionals that measure surface quality [8,15,18,29,43]. At the same time, computation of curvature-based energies—and in particular their derivatives—is considered costly in simulations of deforming meshes.

In general a curvature energy may be expressed in terms of principal curvatures (eigenvalues of the shape operator) of a surface, S . Desired symmetries often lead to expressions in terms of the elementary symmetric functions: mean

($H = \kappa_1 + \kappa_2$), Gaussian ($K = \kappa_1\kappa_2$), and total ($\kappa_1^2 + \kappa_2^2$) curvature. As a model problem consider the bending energy functional

$$E_b = \frac{1}{2} \int_S H^2 dA . \quad (1)$$

This energy, closely related to the Willmore energy [52] of a surface and the Canham-Helfrich energy [12,28] of thin bilipid membranes, is invariant under the group of Möbius transformations of ambient 3-space [50] and in particular under rigid motions and uniform scaling of the surface. Invariance of the energy under rigid motions leads to conservation of linear and angular momenta (Noether's theorem), and invariance under uniform scaling plays a role in setting the size of wrinkles and folds of elastic surfaces. For these reasons, we require *discrete bending energies to be invariant under rigid motions and uniform scaling of the surface S* .

Typically, computations in cloth simulations and surface smoothing applications require both the energy gradient and Hessian. Since the gradient of E_b is a nonlinear operator of the embedding, some authors consider linearizations of the gradient. While this greatly accelerates computation, the linearization destroys invariance under rigid-motion and uniform scaling. Our central observation in the continuous setting is that E_b is *a priori* quadratic in positions under isometric deformations of the surface S , with linear gradient and constant Hessian: *there is no need for artificial linearization*.

To see that E_b is quadratic in positions, first recall that a Riemannian metric (first fundamental form) of a smooth surface is defined by assigning a smoothly varying inner product to each tangent plane of the surface. If the surface is embedded into Euclidean 3-space, we refer to this metric as the metric *inherited* from \mathbb{R}^3 . A Riemannian metric gives rise to intrinsic notions of gradient and divergence, and hence a Laplace-Beltrami operator $\Delta = -\text{div grad}$ on the surface, see [19]. Now consider *deformations* of some fixed reference surface S . A deformation corresponds to changing the embedding $\mathbf{x} : S \rightarrow \mathbb{R}^3$; it is called *isometric* if the metric inherited by $\mathbf{x}(S)$ is the same as that of the reference surface S . Assuming isometric deformations, the mean curvature vector¹ of $\mathbf{x}(S)$ is given by $\mathbf{H}(\mathbf{x}) = \Delta \mathbf{x}$, where Δ is the Laplace-Beltrami operator on the reference surface. We can hence rewrite equation (1) as

$$E_b = E_b(\mathbf{x}) = \frac{1}{2} \int_S \langle \Delta \mathbf{x}, \Delta \mathbf{x} \rangle_{\mathbb{R}^3} dA , \quad (2)$$

where $\langle \cdot, \cdot \rangle_{\mathbb{R}^3}$ denotes the standard inner product of \mathbb{R}^3 ; here and henceforth vector quantities are typeset in boldface. Since Δ remains unaltered under isometric deformations, the energy $E_b(\mathbf{x})$ is indeed *quadratic* in positions \mathbf{x} , and

¹ The mean curvature vector is defined as the vector field normal to $\mathbf{x}(S)$ with magnitude pointwise equal to mean curvature.



Fig. 1. Snapshots from our simulation of a billowing flag. Despite its economy of cost, the proposed bending model achieves qualitatively the same dynamics as popular nonlinear models.

equation (2) together with the assumption of isometric surface deformations is henceforth called the *isometric bending model* (IBM). The assumption of (quasi)isometric deformations is justified when simulating nearly inextensible materials such as cloth since internal stretching forces exceed bending forces by orders of magnitude and keep the surface deformation nearly isometric. We list the essential properties of our isometric bending model in the following box:

- (E1) E_b is invariant under rigid motions and reflections of the surface.
- (E2) E_b is invariant under uniform scaling of the surface.
- (E3) E_b is quadratic in positions under isometric deformations.
- (E4) $E_b = 0$ if the surface is part of a plane.

Discrete theory. In the discrete setting, we replace smooth surfaces by meshes comprised of flat Euclidean triangles. When dealing with such discrete meshes, one requires a corresponding *discrete bending energy*, which may be derived either by a process of discretization—for example using finite elements and numerical quadrature—or by taking a discrete viewpoint *ab initio*, such as provided by the various families of discrete differential operators proposed in [7,16,17,36,37,40,51]. The literature offers multiple examples where the latter approach enables fast, robust computations for simulation and geometric modeling [8,18,25,26,29,41,43,44,48]. Our approach also belongs to this second category: we discuss discrete isometric bending models that are constructed from discrete building blocks rather than from discretizing a continuous model. We derive a variety of discrete IBMs satisfying (E1)–(E4) from very general structural properties of discrete Laplace operators (Section 2), and we illustrate how these models can be applied to significantly accelerate physically plausible simulations of cloth dynamics as well as Willmore fairing applications (Section 3).

2 Discrete IBMs

Our goal is to derive discrete versions of (2) *such that* properties (E1)–(E4) remain valid. To this end, we analyze the structure of discrete isometric bending models by presenting an axiomatic approach to discrete Laplace-Beltrami operators. Together with an axiomatic treatment of mass matrices, discrete Laplacians are used to define notions of discrete mean curvature vectors, from which we obtain discrete IBMs in analogy to (2). In particular, we show in Theorem 1 that the energy axioms (E1)–(E4) are fulfilled in the discrete case *if* the corresponding discrete Laplacians satisfy a set of properties (L1)–(L4) and the mass matrices satisfy another set of properties (M1)–(M3).

The metric of a polyhedral surface is uniquely determined by the lengths of its edges, and the terms *isometric embedding* and *isometric bending* should be understood accordingly: for polyhedral surfaces, an embedding is called isometric if it leaves edge lengths unaltered. Moreover, edges serve as hinges across which stiff triangles can bend. Notice that surfaces may become rigid if isometry is strictly enforced (*e.g.*, for convex bodies), but small deviations from isometry are sufficient for eliminating rigidity. Our approach is to derive models which are exact for isometric deformations and use them as approximations for deformations sufficiently close to isometric.

2.1 Structure of Laplace operators; mass matrix; discrete mean curvature

We define discrete Laplace operators by requiring that core properties from the continuous theory carry over to the discrete setting. In the continuous case, consider the solution, u , to the Dirichlet problem: $\Delta u = 0$ in the interior of S , with prescribed boundary condition $u = g$ along the boundary, ∂S . Then u is the minimizer of the *Dirichlet energy*,

$$D(u) = \int_S \langle \nabla u, \nabla u \rangle dA, \quad (3)$$

with fixed boundary condition $u = g$ on ∂S . Since D is defined via a surface integral we refer to it as an *integrated quantity*. Dirichlet energy is a *quadratic form* which satisfies the following three main properties: (D1) D is non-negative. (D2) $D(u)$ is zero if and only if u is a constant function. (D3) D is scale-invariant in dimension two, *i.e.*, if the surface is uniformly scaled by a factor λ then the Dirichlet energy remains unaltered because the area element, dA , scales as λ^2 and the gradient, ∇ , scales as $1/\lambda$.

Discrete Laplace operators. In the continuous case, the Laplace-Beltrami operator is the variational derivative of Dirichlet energy. We want to keep this property in the discrete case. Dirichlet energy is a quadratic form, which in

the discrete case can be written as

$$D(u) = \sum_{m,n} L_{mn} u_m u_n ,$$

where L is a real symmetric matrix. Here, the set $\{u_n\}$ is a finite set of samples of a (sufficiently smooth) function u .² We note that the symmetric matrix L takes the role of discretizing the Sobolev H^1 semi-norm $|u|_1^2 = \int_S \langle \nabla u, \nabla u \rangle dA$.

We now translate properties (D1)–(D3) from the continuous to the discrete case. First, we make the natural assumption that L be *intrinsic*, that is, it does not change if a different isometric embedding of the mesh is chosen. Secondly, observe that (D1) implies that L must be a positive semi-definite matrix. Next, observe that (D2) states that D vanishes on constants. The following argument shows that this implies that constants are in the kernel of the L . Every quadratic form gives rise to a symmetric bilinear form via *polarization*: $2D(u, v) = D(u + v) - D(u) - D(v)$. Since D vanishes on constants, we obtain

$$D(u + c) = D(u) + D(c) + 2D(u, c) = D(u) + 2D(u, c) .$$

This implies that $D(u, c) = 0$.³ Since $D(u, c) = 0$ for all u , we conclude that $Lc = 0$, so that constants are indeed in the kernel of L . Moreover, property (D2) states that D vanishes *only* on constants. We therefore additionally require that the kernel of L consists *only* of constants. Finally, (D3) states that D is scale-invariant, and we require that the matrix L remains unaltered if the mesh is uniformly scaled. In summary, L must satisfy the following conditions:

- (L1) L is symmetric positive semi-definite.
- (L2) L has a 1-dimensional kernel, given by the constants.
- (L3) L is invariant under uniform scaling of the mesh.

We consider a fourth requirement:

- (L4) If S is part of a plane then $\sum_n L_{mn} \mathbf{x}_n = 0$ at each inner node m .

This property is motivated by a *geometric observation* in the smooth case: the standard Laplacian on the Euclidean plane, $\Delta = \partial_x^2 + \partial_y^2$, has linear functions in its kernel. This implies that planar surfaces have zero mean curvature, $\mathbf{H} = \Delta \mathbf{x} = 0$, since \mathbf{x} is a *linear* embedding of surface positions. Therefore,

² For the remainder of this section, we treat $\{u_n\}$ as interpolating samples of the smooth function u . In many settings, samples approximate but do not interpolate smooth functions; under careful assumptions, the development presented here accommodates the approximating case.

³ Assume $D(u, c) \neq 0$. Choose α such that $D(u) + \alpha 2D(u, c) < 0$. Then $D(u + \alpha c) = D(u) + \alpha 2D(u, c) < 0$, a contradiction to the requirement that D be non-negative.

the smooth bending energy vanishes for flat surfaces. We require the same to hold in the discrete case: the x - and y -coordinates of the vertices of a planar mesh are linear functions over the plane, and (L4) requires those to be in the kernel of L . This will ensure that $E_b = 0$ for planar meshes.

Before presenting examples of discrete Laplacians fitting into the above paradigm, we discuss two important concepts related to discrete Laplacians: (1) integrated vs. pointwise discrete operators, and (2) discrete mean curvature.

Pointwise vs. integrated objects: the role of the mass matrix. Dirichlet energy is an integrated quantity which is scale-invariant. As a consequence, by (L3), if the entire mesh is uniformly scaled with λ then L_{mn} remains invariant. In contrast, the smooth *pointwise* Laplace-Beltrami operator, $\Delta = -\text{div grad}$, scales with $1/\lambda^2$ in this case. In order to obtain a similar discrete *pointwise Laplacian* with the same scaling behavior, we consider the *Poisson problem*, $\Delta u = f$ on S , with $u = 0$ on ∂S . The corresponding variational formulation

$$D(u, v) = (f, v)_{L^2(S)} \quad \text{for all } v \text{ with } v|_{\partial S} = 0 ,$$

has the following discrete analogue:

$$\sum_{m,n} L_{mn} u_m v_n = \sum_{m,n} M_{mn} f_m v_n \quad \text{for all } v \text{ with } v|_{\partial S} = 0 .$$

Here the mass matrix, M , takes the role of *approximating the L^2 inner product* $(f, v)_{L^2(S)} = \int_S f v \, dA$ on S .⁴ Discarding boundary effects, the discrete Poisson problem can be written as $Lu = Mf$, and we may consequently regard the operator $M^{-1}L$ as a discrete pointwise analogue to the smooth pointwise operator Δ . This is immediately reminiscent of the Galerkin-Ritz method; however we stress that the presentation so far makes *no* use of test functions, and that our axiomatic view indeed accommodates for a broader class of mass matrices, such as those corresponding to, *e.g.*, Voronoi regions.

Consider now the indicator function χ_Ω of a set $\Omega \subset S$. Then $(\chi_\Omega, \chi_\Omega)_{L^2(S)}$ is equal to the area $|\Omega|$. Consequently, requiring that M approximate the L^2 -product implies that $\sum_{m,n \in \Omega} M_{mn}$, the sum taken over all nodes inside the set Ω , should be an approximation of $|\Omega|$. This discussion motivates calling M a *mass matrix* if it satisfies the following axioms:

- (M1) The mass matrix M is symmetric and positive definite.
- (M2) If the mesh is uniformly scaled by λ then M scales by λ^2 .
- (M3) The sum $\sum_{m,n \in \Omega} M_{mn}$ converges to the area $|\Omega|$ as the discretization of S is refined and $\Omega \subset S$ stays fixed.

⁴ A closely related interpretation of the mass matrix is given in terms of the discrete Hodge star operator, see [17]. This is due to the relation $(u, v)_{L^2} = \int u \wedge (\star v)$.

Properties (M1) and (M2) imply: if a discrete Laplacian, L , satisfies (L1)–(L3) and M is a mass matrix then the operator $M^{-1}L$ satisfies (L1) and (L2) and scales like the pointwise Laplace-Beltrami operator, Δ , in dimension two.

Discrete linear mean curvature. In analogy to the smooth case where $\mathbf{H} = \Delta \mathbf{x}$, we define the discrete mean curvature vector as $\mathbf{H} = L\mathbf{x}$. From the above discussion we conclude that $L\mathbf{x}$ is an array of integrated quantities. In the smooth case, integrated mean curvature corresponds to the action of \mathbf{H} on smooth functions φ with compact support, $\int \varphi \mathbf{H} = \int \varphi \Delta \mathbf{x}$. In contrast, $\Delta \mathbf{x}$ is a pointwise quantity in the smooth case. To obtain a similar pointwise version in the discrete case, we transform $L\mathbf{x}$ by the inverse mass matrix. Consequently, a pointwise version of the discrete mean curvature vector is given by $M^{-1}L\mathbf{x}$. The situation is summarized in the following table:

	Dirichlet energy (integrated)	L^2 -metric (integrated)	Laplacian (pointwise)	\mathbf{H} (int.)	\mathbf{H} (pntw.)
discrete	$\sum L_{mn} u_m u_n$	$\sum M_{mn} u_m v_n$	$M^{-1}L$	$L\mathbf{x}$	$M^{-1}L\mathbf{x}$
smooth	$\int \nabla u \cdot \nabla u$	$\int uv$	Δ	$\int \varphi \Delta \mathbf{x}$	$\Delta \mathbf{x}$
scaling	1	λ^2	$1/\lambda^2$	λ	$1/\lambda$

Here and henceforth $A\mathbf{x}$ refers to the usual matrix-vector product, $(A\mathbf{x})_m = \sum_n A_{mn} x_n$.

2.2 Laplacian Zoo

Discrete Laplace operators on meshes have been studied in abundance by the graphics community, *e.g.*, for geometric modeling [1,34,43,44], geometric flows [18,29,37,40], mesh parameterization [22,24], as well as mesh compression [5,32,45]. Two common discretizations of Laplacians found in the literature result from (1) the finite element method (FEM) and (2) a purely combinatorial viewpoint. We are first going to review the FEM approach and then embed it into a broader class of new geometric Laplacians recently introduced by Glickenstein [23]. Finally, we will shortly review combinatorial Laplacians in the framework of electrical networks.

The picture in terms of linear finite elements. In the Galerkin-Ritz approach, discretizations of linear PDEs, such as the Laplace and Poisson equation, are obtained by choosing a suitable finite dimensional *approximation space* consisting of the *test functions* u_h for which a corresponding finite-dimensional problem is solved. The finite element method (FEM) adopts locally-supported basis functions defined by their (polynomial) restriction to mesh triangles. Finite elements give a family of definitions for the pair of stiff-

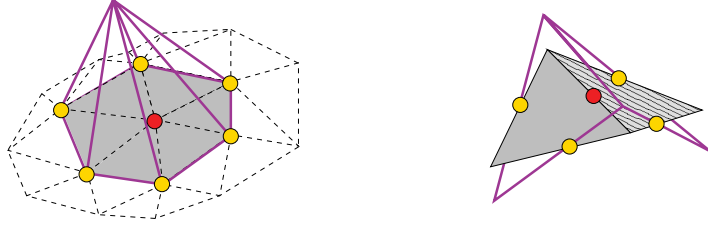


Fig. 2. *Left*: Vertex-based linear Lagrange basis function equaling 1 at a single vertex (marked red) and 0 at all others. *Right*: Edge-based linear Crouzeix-Raviart basis function equaling 1 at a single edge midpoint (marked red) and 0 at all others.

ness and mass matrices, L and M , and a unique definition comes from the choice of basis $\{\Phi_m\}$ via

$$L_{mn} = \int_S \nabla \Phi_m \cdot \nabla \Phi_n dA \quad \text{and} \quad M_{mn} = \int_S \Phi_m \cdot \Phi_n dA, \quad (4)$$

respectively. In the simplest case, approximation spaces are constructed from piecewise linear basis functions, although higher order ones have been explored in graphics [2,49]. Here we quickly review details for a vertex-based and an edge-based set of such linear basis functions.

The space spanned by *linear Lagrange basis functions*, denoted by (Φ_p) , with p running over all *vertices*, yields the so-called cotan formula [40]. Each basis function Φ_p is a linear function on the mesh such that

$$\Phi_p(p) = 1 \quad \text{and} \quad \Phi_p(q) = 0 \text{ if } q \neq p.$$

Similarly, linear finite elements associated with *edges* are spanned by the *Crouzeix-Raviart basis functions*, denoted by (Φ_i) , with i running over all edges. Here each Φ_i is linear on triangles and

$$\Phi_i(i) = 1 \quad \text{and} \quad \Phi_i(j) = 0 \text{ if } j \neq i,$$

where $\Phi_i(j)$ denotes the value of Φ_i at the *midpoint* of edge j , compare Figure 2. Note that the space spanned by linear Lagrange elements is contained in the space spanned by Crouzeix-Raviart elements since

$$\Phi_p = \frac{1}{2} \sum_{e_i \sim p} \Phi_i, \quad (5)$$

summing over all edges e_i incident to p . However, functions in the Crouzeix-Raviart approximation space are *non-conforming*: they are only continuous at edge-midpoints but are in general no longer continuous everywhere. Crouzeix-Raviart elements have been used to derive a discrete Hodge-decomposition of vector fields on meshes [41], as well as to obtain an edge-based linear mean curvature vector [29].

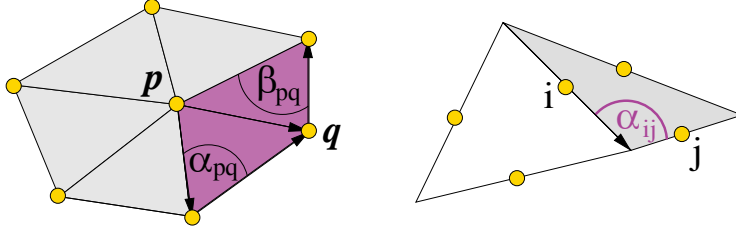


Fig. 3. *Left*: Vertex-based discrete Laplacian with $L_{pq} = -1/2(\cot \alpha_{pq} + \cot \beta_{pq})$, *Right*: Edge-based discrete Laplacian with $L_{ij} = -2 \cot \alpha_{ij}$, where $\alpha_{ij} = \angle e_i, e_j$.

Applying (4) to both the Lagrange and Crouzeix-Raviart elements, it is straightforward to check that the corresponding mass- and stiffness matrices satisfy properties (M1)–(M3) and (L1)–(L4). In particular, to verify (M3), note that

$$\sum_{q \sim p} M_{pq} = \frac{1}{3} A_p \quad \text{and} \quad M_{ii} = \frac{1}{3} A_i ,$$

where A_p and A_i denote the total area of triangles incident to p and i , respectively. Moreover, (M_{ij}) is always a diagonal matrix whereas (M_{pq}) is usually not. In general, replacing M by a diagonal matrix M_{lump} with entries corresponding to the sum of the entries of each row of M is called *mass lumping*. Note that the lumped version of the vertex-based mass matrix (M_{pq}) is again a mass matrix satisfying (M1)–(M3).

Finally, a careful calculation shows that if p and q share an edge (resp. e_i and e_j share a triangle) then, referring to the notation of Figure 3,

$$L_{pq} = -\frac{1}{2}(\cot \alpha_{pq} + \cot \beta_{pq}) \text{ if } q \neq p \quad \text{and} \quad L_{ij} = -2 \cot \angle e_i, e_j \text{ if } j \neq i .$$

The form of L_{pq} is precisely what is referred to as the *cotan representation* [40].

Extending the FE view. As outlined in the previous paragraph, the cotan Laplacian satisfies properties (L1)–(L4). In fact, the cotan operator is a member of a whole family of discrete Laplacians, recently introduced by Glickenstein [23], satisfying these properties. Glickenstein’s Laplacians have a similar structure as the Laplacians previously introduced in the beautiful work of Mercat [36] who defines discrete Laplacian by considering simultaneously a *primal* and a corresponding *dual* mesh. In this view, discrete Laplacians are defined by the ratio of the lengths of dual to the lengths of primal edges. Mercat takes a combinatorial point of view, assigning *arbitrary positive* edge lengths to primal and dual edges—which does in general not yield Laplacians satisfying our geometric property (L4). In contrast, Glickenstein takes a geometric viewpoint, considering *orthogonal* dual edges with (signed) lengths given by intrinsic distances, allowing for positive *and* negative length assignments, and automatically satisfying our geometric condition (L4).

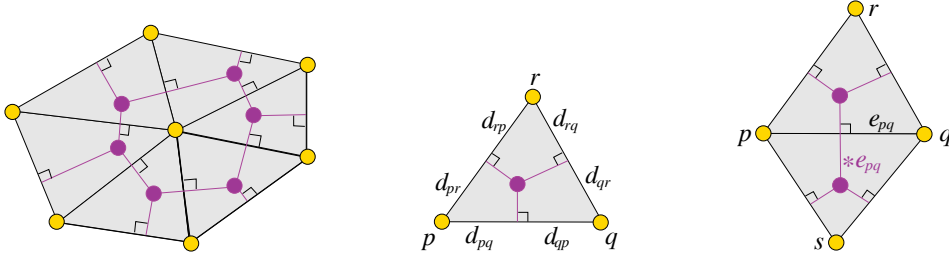


Fig. 4. *Left*: Dual (purple) and primal (black) edges are perpendicular. *Middle*: Dual edges partition primal edges such that $d_{pq}^2 + d_{qr}^2 + d_{rp}^2 = d_{qp}^2 + d_{rq}^2 + d_{pr}^2$. *Right*: Weighted triangulations give rise to Laplace operators $L_{pq} = -l(\star e_{pq})/l(e_{pq})$, where $\star e_{pq}$ is the dual edge to e_{pq} .

To construct these Laplacians, one associates with a given primal triangulation a dual graph with the condition that *all dual edges are perpendicular to primal edges*, see Figure 4 (left). There are certain *compatibility conditions*, which ensure that for each triangle $\{pqr\}$ the three lines perpendicular to the triangle's primal edges indeed meet in a single point, called the *center* C_{pqr} , see Figure 4 (middle). Centers are the vertices of the dual graph. The Laplacian associated with such a primal-dual structure is obtained by

$$L_{pq} = -\frac{l(\star e_{pq})}{l(e_{pq})}, \quad (6)$$

where $l(\star e_{pq})$ is the *signed* length of the dual edge $\star e_{pq}$. This length is obtained by taking the sum of the signed *Euclidean* distances of C_{pqr} to e_{pq} and C_{pqs} to e_{pq} , see Figure 4 (right). The sign of the distance between C_{pqr} and e_{pq} is positive if C_{pqr} and vertex r lie on the same side of e_{pq} and negative if they lie on different sides, and similarly for C_{pqs} .

Setting $L_{pp} = -\sum_{q \neq p} L_{pq}$, it is obvious that the Laplace matrix, L , satisfies condition (L3) and that constants lie in the kernel of L . Moreover, to ensure (L1)—the condition on non-negative Dirichlet energy—and (L2), namely that the kernel of L consists *only* of constants, it suffices to additionally assume that each center C_{pqr} is contained in the circumcircle of its respective triangle $\{pqr\}$, see [23]. Finally, Laplacians defined by (6) satisfy the *geometric property* (L4). To see this, identify the plane with the complex numbers \mathbb{C} , let $e_{pq} = \mathbf{x}_q - \mathbf{x}_p$, and let the dual edge be oriented such that the pair $(e_{pq}, \star e_{pq})$ has positive orientation if $l(\star e_{pq}) > 0$ and has negative orientation if $l(\star e_{pq}) < 0$. This orientation convention yields $\sum_q \star e_{pq} = 0$, so that

$$\sum_q L_{pq} \mathbf{x}_q = -\sum_q L_{pq} (\mathbf{x}_p - \mathbf{x}_q) = -\sum_q \frac{l(\star e_{pq})}{l(e_{pq})} e_{pq} = \imath \sum_q \star e_{pq} = 0,$$

where \imath denotes the unit imaginary number. This proves (L4).

Note that the vertex-based cotan operator discussed above is indeed a special

case of weighted triangulations: it corresponds to a weighted triangulation where all centers, C_{pqr} , coincide with the centers of circumcircles.

Electrical networks. In the view of discrete Laplacians associated with electrical networks (or weighted graphs, the umbrella operator being one example), see [20,21], one thinks of the edges of a polyhedral surface as carrying positive conductances $C_{pq} = -L_{pq}$ with potentials u_p living at vertices. Setting $L_{pp} = -\sum_q L_{pq}$ yields a *vertex-based* discrete Laplacian, and it is not hard to show that conditions (L1)–(L3) are satisfied. However, we issue warning that these Laplacians *usually fail to satisfy the geometric condition* (L4).

2.3 Discrete isometric bending model (IBM)

We follow the analogy to the smooth case, where the IBM was defined by equations (1) and (2). Using the fact that a pointwise version of the discrete mean curvature vector is given by $M^{-1}L\mathbf{x}$, and noting that by property (M3) the mass matrix plays the role of a discrete L^2 -metric, we define in the discrete case:

$$E_b(\mathbf{x}) = \frac{1}{2}(M^{-1}L\mathbf{x})^T M(M^{-1}L\mathbf{x}) = \frac{1}{2}\mathbf{x}^T (L^T M^{-1}L) \mathbf{x} .$$

Theorem 1 *Let \mathbf{x} be an isometric embedding of a discrete mesh. If L is a discrete intrinsic Laplace operator satisfying (L1)–(L4) and M is an intrinsic mass matrix satisfying (M1)–(M3), then the discrete bending energy*

$$E_b(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T (L^T M^{-1}L) \mathbf{x}$$

satisfies properties (E1)–(E4).

PROOF. Since L and M are intrinsic properties of the mesh (*i.e.*, they do not change if we choose another isometric embedding of the mesh), it follows that $E_b(\mathbf{x})$ is indeed quadratic in \mathbf{x} , which is property (E3). To show property (E1), first note that (L2) implies that E_b is invariant under translations of the mesh in \mathbb{R}^3 . To show invariance under rotations and reflections, note that

$$\begin{aligned} \mathbf{x}^T (L^T M^{-1}L) \mathbf{x} &= \sum_{m,n} (L^T M^{-1}L)_{mn} \langle \mathbf{x}_m, \mathbf{x}_n \rangle_{\mathbb{R}^3} \\ &= \sum_{m,n} (L^T M^{-1}L)_{mn} \langle R\mathbf{x}_m, R\mathbf{x}_n \rangle_{\mathbb{R}^3} \end{aligned}$$

for any element $R \in O(3)$ of the 3-dimensional orthogonal group. Property (E2) is a consequence of scaling: if \mathbf{x} scales with λ then L is scale-invariant by (L3) and M^{-1} scales with $1/\lambda^2$ by (M2), so that E_b is again scale-invariant. Finally, (E4) follows directly from (L4).

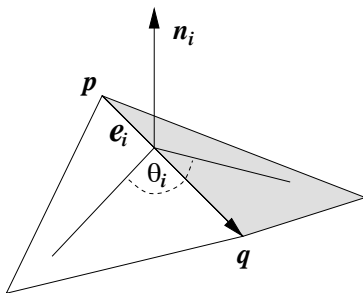
2.4 Comparison to other bending models

In common *edge-based* bending models, such as those of Baraff and Witkin [4], Bridson et al. [11], and Grinspun et al. [25], the (non-quadratic) bending energy, together with its (non-linear) bending forces and (non-constant) energy Hessian, is assembled from contributions of individual *hinge stencils* consisting of two triangles meeting at an interior edge. Here we sketch that—up to second order in the limit of infinitesimal displacements about the planar rest state—the (non-quadratic) discrete shells energy [25] *agrees* with the (quadratic) energy of a discrete IBM based on Crouzeix-Raviart elements. Consequently, if one substitutes models, then for small bending away from the flat state, the energy gradients are the same—and as we will see in Section 3.1—this implies that the elastic restoring forces are identical as well. Hence there is no need to readjust material coefficients when switching between these models, facilitating adoption of the discrete Crouzeix-Raviart based IBM.

To see this equality of models, we first rewrite the Crouzeix-Raviart based IBM (where the Laplace and mass matrix are assembled from non-conforming Crouzeix-Raviart elements) as a sum over hinge-based bending energies:

$$E_{CR} = \sum_i E_i \quad \text{with} \quad E_i = \frac{3\|e_i\|^2}{2A_i} \left(2 \cos \frac{\theta_i}{2}\right)^2, \quad (7)$$

where A_i is the combined area of the two triangles meeting at edge e_i , and θ_i is the dihedral angle at e_i . This can be derived from $E_i = 1/2 M_{ii}^{-1} \langle \mathbf{H}_i, \mathbf{H}_i \rangle$, with $M_{ii} = A_i/3$ as before, and $\mathbf{H}_i = (\mathbf{L}\mathbf{x})_i$ denoting the edge-based mean curvature vector at e_i . Note that in this form it is no longer obvious (but certainly still true) that the bending energy E_{CR} is quadratic in vertex positions under isometric deformations.



The *discrete shells bending energy* in [25] is given by

$$E_{DS} = \sum_i E_i \quad \text{with} \quad E_i = \frac{\|e_i\|}{h_i} (\pi - \theta_i)^2,$$

where θ_i is the dihedral angle and h_i is a third of the sum of the heights of the two triangles incident to e_i . Observing that $2A_i = 3h_i\|e_i\|$, and that $(\pi - \theta_i)$ and $2 \cos(\theta_i/2)$ agree up to first order as $\theta_i \rightarrow \pi$, we obtain:

Theorem 2 *The Crouzeix-Raviart-based discrete IBM agrees with the discrete shells bending model up to (and including) second order in the limit of small normal displacements of the plane ($\theta_i \rightarrow \pi$).*

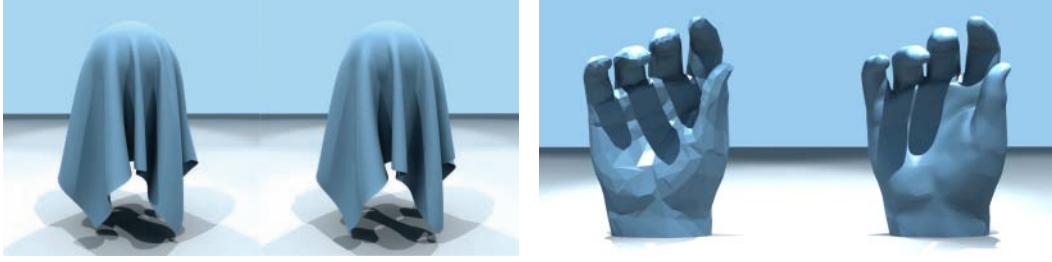


Fig. 5. *Left*: Final rest state of a cloth draped over a sphere, for (left) the proposed isometric bending model and (right) the widely-adopted nonlinear hinge model. Both models yield very similar final states. *Right*: Initial and final frames of Willmore flow applied to smoothing a 24192 triangle hand at interactive rates.

We note that bending energy in discrete shells can be interpreted in terms of a (scalar) mean curvature model given by $H_i = (\pi - \theta_i)$, a version which in turn can be interpreted in terms of geometric measure theory, as shown by Cohen-Steiner and Morvan [16].

Finally, we sketch the relation to other common bending models. The model of Bridson et al. [11] replaces (up to a constant factor) the term $(\pi - \theta_i)\mathbf{n}_i$ by $\sin((\pi - \theta_i)/2)\mathbf{n}_i$ for evaluating bending forces in the limit of small displacements, where \mathbf{n}_i is the angle-bisecting edge normal to e_i . Consequently, in the limit $\theta_i \rightarrow \pi$, Bridson’s model agrees (up to a constant factor) with the discrete shells model, and hence our model (however, the models differ for $\theta_i \rightarrow 0$, where our model unfortunately has vanishing force, whereas Bridson’s model does not). Furthermore, Bobenko introduced a version of $\int H^2$, integrated squared mean curvature, and notes in [7] that his model is also intimately linked to discrete shells, and hence our model, in the limit of small displacements over the plane.

3 Applications

3.1 Application to Simulation of Cloth and Thin Plates

Simulations of flexible surfaces that bend without stretching (*inextensible thin plates*) find applications in manufacturing, computer animation, and fashion design. For specificity, we will focus on the simulation of inextensible textiles (see the recent surveys [13,35,39,54]); however, our observations about the role of the discrete IBM apply in general to inextensible thin plates. The behavior of a thin plate is governed by its elastic energy, which measures the deformation of the surface away from its flat undeformed configuration. Most models of cloth consider separately the bending energy, E_b , and the in-plane energy, E_p ,

$$E(\mathbf{x}) = E_b(\mathbf{x}) + E_p(\mathbf{x}) .$$

E_p penalizes change in first fundamental form with respect to the undeformed configuration, where the metric of the deformed surface is induced by the immersion, \mathbf{x} . The gradient of E_p is usually very large relative to that of E_b , since textiles tend to resist stretching more than bending [4,46]. Therefore, we may safely view the in-plane elastic response as a mechanism (in the spirit of penalty forces) that ensures that deformations preserve the metric. Henceforth, we accept that some in-plane model has been chosen, we assume that all deformations are isometric, and we focus on presenting the role of our discrete IBM in accelerating an existing cloth solver.

Forces. We first consider elastic (energy-preserving) forces, and then damping (energy-dissipating) forces. The elastic response of a deformed material is governed by a conservative force, *i.e.*, one which acts against the energy gradient:⁵

$$\mathbf{F}_e(\mathbf{x}) = -\nabla E(\mathbf{x}) = -\nabla E_b(\mathbf{x}) - \nabla E_p(\mathbf{x}) . \quad (8)$$

Most real materials dissipate energy during motion. *Rayleigh damping* is among the simplest models of dissipation used by the computational mechanics community [31,55]. In the Rayleigh view, the damping force, \mathbf{F}_d , is proportional to velocity, $\mathbf{v}(t)$:

$$\mathbf{F}_d(\mathbf{v}) = -\mathbf{K}_d \mathbf{v} , \quad \text{where} \quad \mathbf{K}_d = \alpha_1 \mathbf{M} + \alpha_2 \text{Hess}(E) . \quad (9)$$

The constant of proportionality is written as a linear combination⁶ of two tensors: the mass matrix (as discussed in §2.1), and the Hessian of elastic energy; the two tensors correspond to damping of low and high temporal frequencies, respectively. When using the discrete IBM, we may expand \mathbf{K}_d as

$$\mathbf{K}_d^{\text{IBM}} = \alpha_1 \mathbf{M} + \alpha_2 \text{Hess}(E_p) + \alpha_2 \text{Hess}(E_b) . \quad (10)$$

Computation of forces is efficient for the discrete IBM: The constant Hessian, $\text{Hess}(E_b)$, is pre-computed once, and forces are later computed by the matrix-vector products $\text{Hess}(E_b)\mathbf{x}$ and $\alpha_2 \text{Hess}(E_b)\mathbf{v}$, respectively.

Other bending models, such as those surveyed in [47], do not take advantage of isometry and they therefore involve forces nonlinear in positions. Methods exist to linearize the bending forces at every time step, and while these were applied to existing nonlinear models, they destroy properties (E1) and (E2).

Dynamics. In our classical mechanical system, the temporal evolution of position, $\mathbf{x} \equiv \mathbf{x}(t)$, and velocity, $\mathbf{v} \equiv \dot{\mathbf{x}}$, is governed by the equations of

⁵ Here $\nabla \equiv \nabla_{\mathbf{x}}$.

⁶ In this ad-hoc model, the constants α_1 and α_2 are endowed with the requisite units so that the final product has units of force.

motion:

$$\begin{pmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{v}}(t) \end{pmatrix} = \begin{pmatrix} \text{Id} & 0 \\ 0 & (\rho M)^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{F}_e(\mathbf{x}(t)) + \mathbf{F}_d(\mathbf{v}(t)) \end{pmatrix},$$

with initial conditions $\mathbf{x}(0)$ and $\mathbf{v}(0)$. Here the *physical mass matrix*, ρM , is given by the product of mass surface density and the geometric mass matrix of §2.1. Considering $\mathbf{q} = (\mathbf{x}, \mathbf{v})$ as a point in *phase space*, the above equation is simply an ordinary differential equation describing the flow of the point along a vector field $\mathbf{f}(\mathbf{q})$:

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}), \quad \text{given } \mathbf{q}(0). \quad (11)$$

Time integration. Time discretization of (11) is a well-studied problem (see [27] and references therein); approaches may be classified as explicit, implicit, or mixed implicit-explicit.⁷ As representative examples, consider the *explicit Euler* method, which uses the update rule $\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{f}(\mathbf{q}_k)$, where $t = hk$. Since the explicit method directly evaluates \mathbf{q}_{k+1} , it is easy to implement and fast to compute; however, explicit methods become unstable if the time step is not very small.

Implicit methods alleviate problems of instability, at the cost of a more expensive update step. The *implicit Euler* method searches for the root of

$$\mathbf{G}(\mathbf{q}_{k+1}) = \mathbf{q}_{k+1} - \mathbf{q}_k - h\mathbf{f}(\mathbf{q}_{k+1}) = 0.$$

Most often this system is treated by repeated Newton iterations until convergence (although *semi-implicit* approaches simply assume convergence after a single Newton iteration). The $(i+1)$ th Newton iteration is given by⁸

$$\mathbf{q}_{k+1}^{(i+1)} = \mathbf{q}_{k+1}^{(i)} - (\nabla \mathbf{G})^{-1} \mathbf{G}(\mathbf{q}_{k+1}^{(i)}). \quad (12)$$

If the function $\mathbf{G}(\cdot)$ is sufficiently smooth and the initial guess (which we take as $\mathbf{q}_{k+1}^{(0)} = \mathbf{q}_k$) is sufficiently close to the solution, then repeated Newton iterations converge to a root of $\mathbf{G} = 0$. Newton's method requires evaluation of the flow, \mathbf{f} , and the flow Jacobian, $\nabla \mathbf{f}$, to compute \mathbf{G} and $\nabla \mathbf{G}$, respectively,⁹

⁷ In mixed implicit-explicit (IMEX) time-integration, some forces are treated using the explicit method, and other forces are treated using the implicit method. In our tests we measured both an explicit and an implicit treatment of our bending model; since both modes of integration were accelerated, we expect acceleration also in the case of IMEX methods. For more details on implementing an IMEX solver we refer the reader to [27].

⁸ Here and henceforth $\nabla \equiv \nabla_{\mathbf{q}}$.

⁹ In computing terms of $\nabla \mathbf{G}$ associated to Rayleigh damping, it is standard practice to consider only derivatives with respect to \mathbf{v} but not \mathbf{x} .

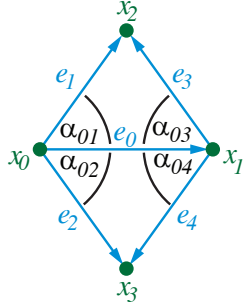
and in the case of the discrete IBM, the constant bending Jacobian

$$\nabla \mathbf{f}_{\text{IBM}} = \begin{pmatrix} 0 & \text{Id} \\ (\rho M)^{-1} \text{Hess}(E_b) & \alpha_2 (\rho M)^{-1} \text{Hess}(E_b) \end{pmatrix} \quad (13)$$

is pre-computed only once.

Implementation. We have presented general guidelines for incorporating the discrete IBM into an implicit or explicit cloth solver: We compute elastic and dissipative bending forces with the matrix-vector products $\text{Hess}(E_b)\mathbf{x}$ and $\alpha_2 \text{Hess}(E_b)\mathbf{v}$, respectively; the constant matrix $\text{Hess}(E_b)$ is pre-computed once. For implicit methods, expression (13) provides a flow Jacobian for the bending model; this too should be pre-computed once. These guidelines are independent of the choice of stretching model. In general the stretching model will have nonlinear time-varying forces and flow Jacobians. Therefore, even when the discrete IBM is used, the net forces are in general nonlinear, and the net flow Jacobian is in general nonconstant. However, all computations related to bending can be treated using the pre-computed Hessian, and these guidelines apply regardless of the choice of a specific discrete IBM.

We now describe a discrete IBM specifically tailored to cloth simulation, beginning with the intuition that bending is associated to pairs of triangles bending about an edge (a “hinge”). This immediately invokes the picture of the stencil of the Crouzeix-Raviart Laplacian (see §2.2). Recall that Crouzeix-Raviart elements have edge-based degrees of freedom (DOFs), and that functions defined in the edge basis are not everywhere continuous. If we were to represent the immersion in the edge-based DOFs, we would introduce mesh discontinuities. Therefore, we seek a way to combine the edge-based Laplacian with vertex-based DOFs. That such a matrimony is feasible, and indeed simple, follows from the observation that the vertex-based Lagrange approximation space is contained in the Crouzeix-Raviart space, see Equation (5). For each inner edge and with reference to the illustrated labeling convention, the mapping from vertex positions to mid-edge positions takes the form

$$\frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \mathbf{x}_0 + \mathbf{x}_1 \\ \mathbf{x}_0 + \mathbf{x}_2 \\ \mathbf{x}_0 + \mathbf{x}_3 \\ \mathbf{x}_1 + \mathbf{x}_2 \\ \mathbf{x}_1 + \mathbf{x}_3 \end{pmatrix} .$$


The diagram shows a diamond-shaped stencil with four vertices labeled x_0 (left), x_1 (right), x_2 (top), and x_3 (bottom). The edges are labeled e_1 (top-left), e_2 (bottom-left), e_3 (top-right), and e_4 (bottom-right). The central edge between x_0 and x_1 is labeled e_0 . At each vertex, an angle is indicated: α_{01} at x_0 (between e_1 and e_0), α_{02} at x_0 (between e_2 and e_0), α_{03} at x_1 (between e_0 and e_3), and α_{04} at x_1 (between e_0 and e_4).

Before detailing its proof, we present the final result, which makes reference to the hinge stencil illustrated above.

Theorem 3 *Using a diagonal mass matrix and discarding boundary edges, any discrete bending energy can be written as a single sum over contributions of inner edges, $E_b(\mathbf{x}) = \sum_i E_i(\mathbf{x})$. Specifically, using Crouzeix-Raviart elements, we obtain*

$$E_0(\mathbf{x}) = \frac{3}{2A_0} \sum_{p,q=0}^3 c_p c_q \langle \mathbf{x}_p, \mathbf{x}_q \rangle_{\mathbb{R}^3},$$

with respect to the above labeling, where A_0 is the combined area of the two triangles meeting at e_i , and

$$\begin{aligned} c_0 &= \cot \alpha_{03} + \cot \alpha_{04}, & c_1 &= \cot \alpha_{01} + \cot \alpha_{02}, \\ c_2 &= -\cot \alpha_{01} - \cot \alpha_{03}, & c_3 &= -\cot \alpha_{02} - \cot \alpha_{04}. \end{aligned}$$

In fact, this energy can be written in a more geometric way, see (7). Here we chose the above form because it illustrates how to *assemble the constant energy Hessian*, $\text{Hess}(E_b)$. In our case this is done in a single loop over all inner edges e_i , by adding the scalar factors in front of $\langle \mathbf{x}_p, \mathbf{x}_q \rangle$ from the above energy expression to $(\text{Hess}(E_b))_{pq}$.

PROOF. Discarding boundary edges (which cannot act as a “hinge” and are therefore excluded from consideration), the mass matrix M is a square matrix with rows and columns corresponding to inner edges, whereas the number of columns of the Laplace matrix L corresponds to all edges and the number of rows of L corresponds to the number of inner edges. We denote the mapping from the vertex- to the mid-edge representation by I , that is, $I_{jp} = 1/2$ if edge e_j contains vertex \mathbf{x}_p , and $I_{jp} = 0$ otherwise. Since the Crouzeix-Raviart mass matrix M is diagonal, we can write our bending energy as a sum over contributions from inner edges,

$$E_b(\mathbf{x}) = \frac{1}{2} \sum_i \frac{1}{M_{ii}} \left\| \sum_{j,p} (L_{ij} I_{jp}) \mathbf{x}_p \right\|_{\mathbb{R}^3}^2.$$

Observing that $L_{ij} = 0$ if e_i and e_j do not belong to a common triangle, and referring to the above labeling convention (fixing $i = 0$), we obtain

$$\begin{aligned} \sum_j L_{0j} I_{j0} &= -\frac{1}{2}(L_{03} + L_{04}), & \sum_j L_{0j} I_{j1} &= -\frac{1}{2}(L_{01} + L_{02}), \\ \sum_j L_{0j} I_{j2} &= \frac{1}{2}(L_{01} + L_{03}), & \sum_j L_{0j} I_{j3} &= \frac{1}{2}(L_{02} + L_{04}). \end{aligned}$$

For Crouzeix-Raviart elements one has $L_{0j} = -2 \cot \angle_{e_0, e_j}$ for $j \neq 0$, which completes the proof.

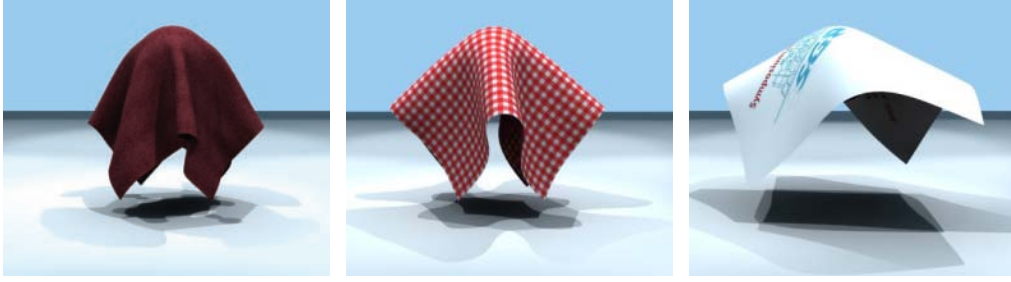


Fig. 6. The quadratic bending model is valid over the full range of bending to in-plane stiffness ratios, *e.g.*, (left to right) $10^{-5} : 1$, $10^{-3} : 1$, and $10^{-2} : 1$.

Experiments. In a series of experiments, we observed that pre-computation of the Hessian enables considerable acceleration of cloth simulations. We implemented the Crouzeix-Raviart discrete IBM, and for comparison we implemented a variant of the *nonlinear hinge* model, which measures the change in dihedral angle between two triangles; for popular variants of this model see [4,11], and for our specific implementation see [25].

In an evaluation of two solvers, two problem scenarios, two mesh types, and resolutions ranging from 400 to 25600 vertices, we observe a typical two- to three-fold speedup in simulation times compared to the nonlinear hinge. Figures 1 and 5-left provide a visual point of comparison, and Figure 7 summarizes our performance measurements. We observe a seven- to eleven-fold speedup in bending force computation. Since IBM’s Hessian is pre-computed, we can report only the negligible time required to add $\nabla \mathbf{f}_{\text{IBM}}$ to $\nabla \mathbf{G}$; in contrast, the repeated computation of the nonlinear hinge Hessian is costly. Overall speedup will depend on the fraction of total computation associated to bending; to estimate this we conducted several experiments. Our test platform includes two time integrators: an implicit solver framework of [4] and the explicit Euler method; the constant strain linear finite element for in-plane response [31,55]; collision detection using k-DOP trees [33] and response using Bridson’s framework [10]; the PETSc Newton solver and both direct and iterative linear solvers [3].

We simulated heavily-damped draping of a square sheet over a sphere (see Figures 5 and 6). The draped cloths are qualitatively similar in their final configuration and distribution of folds. Only the final draped shape is important; therefore, we used large Rayleigh coefficients, allowing larger time steps [31]. Next, we simulated the dynamics of a flag under wind (refer to Figure 1). The billowing motion of the IBM and nonlinear flag are qualitatively similar. We found no need to readjust material parameters when switching from the nonlinear to the IBM model; this is not surprising, in light of §2.4. We modeled wind by a constant homogeneous velocity field, with force proportional to the projection of the wind velocity onto the area-weighted surface normal.

Draping problem		regular mesh (resolution, in no. vertices)				irregular mesh (resolution, in no. vertices)			
		400	1600	6400	25600	450	2100	6500	22500
Gradient cost (ms)	nonlinear hinge	0.937	3.45	16.4	66.6	1.10	5.43	17.6	67.8
	quadratic IBM	0.081	0.338	2.19	9.15	0.098	0.494	2.32	9.68
Hessian cost (ms)	nonlinear hinge	12.8	54.2	218	890.	15.2	77.2	246	888
	quadratic IBM	0.237	0.963	3.87	15.7	0.266	1.28	3.99	13.6
Explicit step cost (ms)	nonlinear hinge	3.81	6.64	27.5	112.	2.16	9.53	31.4	140.
	quadratic IBM	2.63	2.90	11.9	48.8	0.964	4.35	15.2	76.5
Implicit step cost (ms)	nonlinear hinge	28.6	138	470.	1730	33.9	219	557	1880
	quadratic IBM	11.0	62.7	168	505	13.6	103	219	612

Flag problem		regular mesh (resolution, in no. vertices)				irregular mesh (resolution, in no. vertices)			
		400	1600	6400	25600	450	2100	6500	22500
Gradient cost (ms)	nonlinear hinge	0.975	3.99	16.0	64.0	1.10	5.43	17.8	68.7
	quadratic IBM	0.085	0.341	2.14	8.75	0.099	0.490	2.31	9.28
Hessian cost (ms)	nonlinear hinge	13.4	54.8	212	849	15.2	77.4	247	887
	quadratic IBM	0.251	0.974	3.79	14.99	0.267	1.30	3.96	13.7
Explicit step cost (ms)	nonlinear hinge	1.73	7.05	27.7	112.	1.97	9.80	32.7	134
	quadratic IBM	0.780	3.26	13.3	53.4	0.900	4.54	16.1	70.0
Implicit step cost (ms)	nonlinear hinge	27.6	106	420.	1680	33.5	155	513	1880
	quadratic IBM	9.53	32.9	127	490	12.5	50.4	166	608

Fig. 7. Computational cost per time step for two solvers, regular- and irregular-meshes, and multiple resolutions, comparing IBM to the nonlinear hinge, as measured on a Pentium D 3.4GHz, 2GB RAM. Time step cost includes collision handling.

3.2 Application to Geometric Modeling using Willmore Flow

As noted in [8], immersions which minimize Willmore energy are of interest in a range of areas, including the study of conformal geometry [6,50,52], physical simulation of fluid membranes [12,28], and geometric modeling. The *Willmore energy* of a surface is given as¹⁰

$$E_W(\mathbf{x}) = \int_S (H^2 - 4K) dA = \int_S (\kappa_1 - \kappa_2)^2 dA.$$

In contrast to our treatment of cloth and inextensible thin plates, in applications of Willmore energy the presence of an accompanying isometry-enforcing term is notably absent. Indeed, the deformations governed by Willmore flow are generally *not* isometric so that the energy gradient will not be linear, as it was for cloth. It may therefore seem surprising that our proposed Willmore flow application, while incorporating an isometry assumption, gains speed without paying in visual quality. The *inexact Newton Method* serves as the numerical framework in which this phenomenon can be explained.

Geometric flow. Our geometric flow application deals with surfaces without boundary and surfaces whose boundary is fixed up to first order, *i.e.*, positions and normals are prescribed. In both cases, the Willmore functional is variationally equivalent to the functional (1). The corresponding geometric flow sets surface velocity as a function of surface curvature:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = -\nabla E_b(\mathbf{x}).$$

¹⁰The Gaussian curvature factor of 4 is due to our use of $H = \kappa_1 + \kappa_2$ instead of $H = \frac{1}{2}(\kappa_1 + \kappa_2)$.

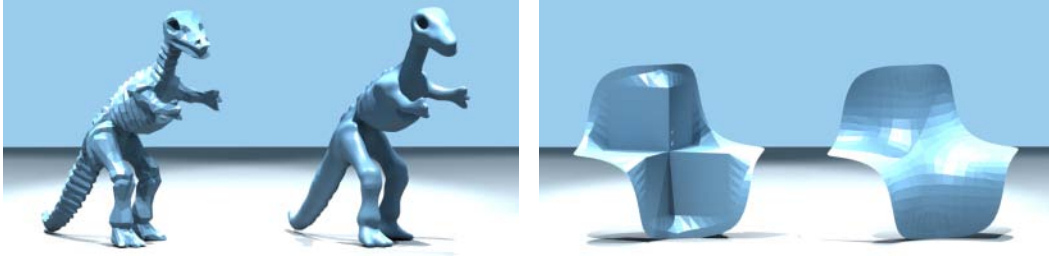


Fig. 8. Initial and final frames of Willmore flow applied to smoothing (*left*) a 44928 triangle dinosaur mesh and (*right*) to solve the hole filling problem posed in [8]. For the dinosaur, 16 smoothing steps require a total of 7.47s, with one-time factorization costing 8.77s. The solution of the hole-filling problem was obtained in 640ms, after 120ms for Jacobian pre-factorization. Images rendered with flat shading.

This flow has spurred many applications for surface fairing and surface restoration [8,15,43,53]: Hole-filling applications (see Fig. 8-right) integrate the flow to its stationary limit (when such a limit exists). Smoothing applications (see Figures 5-right and 8-left) integrate the flow over a prescribed duration, with longer integration times smoothing progressively coarser spatial frequencies.

Implementations of discrete Willmore flow were reported by several authors. Ken Brakke’s *Surface Evolver* [9,30] used a discretized version of mean curvature as a building block for Willmore energy. Yoshizawa et al. [53] discretized directly the energy gradient, $\mathbf{f}(\mathbf{x})$, using the cotangent formula. The latter introduced an additional tangential force to improve the quality of the evolving mesh. Clarenz et al. [15] discretized the variation of the Willmore energy in terms of linear Lagrange elements and treated the corresponding L^2 -flow by a coupled system of second order equations. Finally, Bobenko et al. [8] presented a discrete version of the fact that the integrand of the smooth Willmore energy is conformally invariant. However, existing approaches did not focus on the economy that arises from assuming (or rather pretending) that deformations are isometric.

To compute a geometric flow, one must integrate the flow trajectory over time. This may be achieved via explicit or implicit methods, as described in §3.1. Following the direction laid out by Desbrun and coworkers [18], who note that the stability of implicit integration methods improves the performance of geometric flows, we implemented an implicit method, in particular using the *inexact* Newton’s method. In the framework of this method we explain why the assumption of isometric deformations is indeed a powerful one.

The inexact Newton’s method. The fixed points of Newton’s method, $\mathbf{G}(\mathbf{x}_{k+1}) = \mathbf{x}_{k+1} - \mathbf{x}_k - h\mathbf{f}(\mathbf{x}_{k+1}) = 0$, remain unaltered when the Jacobian, $\mathbf{J} = \nabla\mathbf{G}$, is replaced by any invertible matrix, $\tilde{\mathbf{J}} \approx \mathbf{J}$. This observation justifies the *inexact* Newton’s method: Instead of a costly \mathbf{J} , use any good but efficient approximation $\tilde{\mathbf{J}}$ [38]. Approximating the Jacobian *may* affect (the rate or

radius of) convergence, but it will not affect the limit value of converging iterations. In contrast, approximating \mathbf{G} will affect the value of the fixed points; therefore, inexact Newton approximates only the Jacobian $\nabla\mathbf{G}$, not the function \mathbf{G} .

The isometric deformation assumption gives the constant approximant, $\tilde{\mathbf{J}}$, to the flow Jacobian, \mathbf{J} , where

$$\mathbf{J} = h \text{Hess}(E_b)(\mathbf{x}) - \text{Id} \approx h \text{Hess}(E_b)(\mathbf{x}_0) - \text{Id} = \tilde{\mathbf{J}} ,$$

and x_0 corresponds to the undeformed configuration. Equality $\mathbf{J} = \tilde{\mathbf{J}}$ would hold only if the space of permissible deformations were confined to isometries of the initial surface. The approximate flow Jacobian, $\tilde{\mathbf{J}}$ needs only be computed once as long as the time step, h , is constant. Along with this approximate Jacobian, we use the exact, nonlinear \mathbf{G} .

Implementation. Using the PETSc [3] and PARDISO [42] solver libraries, we implemented the backward Euler method with an inexact Newton solver. We found that a *semi-implicit* treatment, which stops Newton’s method after one iteration, exhibits the best trade-off between stable time step size and cost per time step, for a prescribed level of accuracy.

We briefly discuss the details that led to efficient Newton iterations. We rewrite the Newton iteration (12) to make clear the structure of the linear solve:

$$(\nabla\mathbf{G})(\mathbf{q}_{k+1}^{(i+1)} - \mathbf{q}_{k+1}^{(i)}) = -\mathbf{G}(\mathbf{q}_{k+1}^{(i)}) .$$

For the right hand side, we compute the full nonlinear expression $\nabla E_b(\mathbf{x})$; consult the appendix for a derivation. For the left hand side, we pre-compute $\nabla\mathbf{G}$ at program initialization. We used PARDISO’s LL^T solver, which factors $\nabla\mathbf{G}$ symbolically and numerically. Since $\nabla\mathbf{G}$ is known at program start, the symbolic factorization step—the bulk of the linear solver’s computation—can be executed just once at startup. The pre-factorization of $\nabla\mathbf{G}$, and the elimination of repeated matrix assembly, accounts for the speedup we observed.

To evaluate the performance of our method, we duplicated several problem scenarios presented by Bobenko et al. [8] (see Figures. 8 and 9). Whereas Bobenko’s work preserves the Möbius symmetries of the underlying continuous system, our focus is on rapid computation at nearly interactive rates, maintaining good surface quality, while retaining only a subset (E1 and E2) of the Möbius symmetries. Figure 9 demonstrates the evolution of a four-times subdivided icosahedron into a sphere. We report computation times for a single process running on a 2GHz notebook with 2GB RAM. The Jacobian was pre-factored in 200ms, and 10 smoothing steps were applied, each requiring 12ms of computation. The sphere diameter was 2 units and the flow time step $h = 0.1$. Figure 9-right illustrates the role of the mass matrix. One face of the

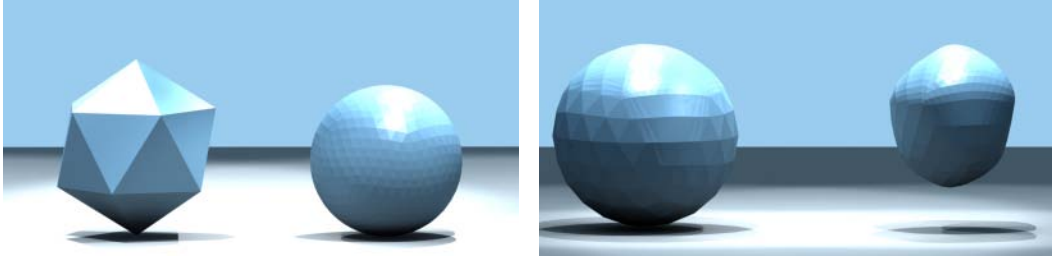


Fig. 9. Initial and final frames of Willmore flow applied to smooth a 4-times subdivided icosahedron into a sphere. (*left*) The sphere converged in 120ms (12ms \times 10 smoothing steps), with 200ms for Jacobian pre-factorization. (*right*) Comparison of flow results for a non-uniformly tessellated icosahedron, with and without a mass matrix, illustrating the importance of the scaling properties of the mass matrix.

icosahedron was subdivided a fifth time to induce a nonuniform tessellation of the domain. If the mass matrix is neglected, then the flow of the non-uniformly subdivided icosahedron does not converge to a sphere.

Figure 8-right shows the result of a six-sided hole filling problem (compare with [8]). In this problem setup, the boundary conditions are taken from a smooth Loop subdivision surface, and the interior triangles are initialized with a trivial non-smooth solution. We fix two rings of vertices to enforce the prescribed boundary conditions up to first order, and integrate the geometric flow until it reaches a stationary point. The solution required 760ms, which includes 120ms for Jacobian pre-factorization.

Due to its constant, pre-factored Jacobian, our method scales well to larger meshes. We applied Willmore flow to smooth several large meshes, including the dinosaur (45k triangles) and hand (24k triangles), shown in Figure 8 and Figure 5. Unlike Laplacian smoothing, for which fast implicit methods have been demonstrated [18], Willmore flow is derived from a scale-invariant energy, hence it is not biased toward shrinking the surface. The inexact Newton's method enabled us to accelerate computation by several orders of magnitude. The near-interactive times are reported using a notebook computer, suggesting that fully-interactive Willmore flow for large meshes is well within the reach of the discrete IBM.

4 Conclusion and Future Work

In this paper we derived a general class of discrete isometric bending models from an axiomatic treatment of discrete Laplace operators, linear mean curvature, and the mass matrix. We leave as open directions two generalizations of this idea. The first is to consider *anisotropic energies* of the form

$$E_{aniso} = \int_S (c_1 \kappa_1 + c_2 \kappa_2 + \dots + c_n \kappa_n)^2 dA , \quad (14)$$

where κ_i is the normal curvature along *material direction* \mathbf{v}_i (not a principal curvature direction). Observe that any directional curvature, κ_i , along material direction \mathbf{v}_i , is given by the vector-valued expression $\kappa_i \mathbf{n} = \frac{d^2}{ds_i^2} \mathbf{x}$, where \mathbf{n} is the surface normal, and derivatives are taken with respect to an arc length, s_i , of a curve with the direction \mathbf{v}_i at the point of interest. Considering only isometric deformations, $\kappa_i \mathbf{n}$ is linear in the embedding, \mathbf{x} . Furthermore, we can sum multiple such expressions corresponding to different material directions, as in (14): This is reasonable because only their magnitude (κ_i) but not their direction (\mathbf{n}) differs. Consequently, the energy is quadratic in the immersion, and one could carry out a similar procedure to that presented in the current paper to obtain a corresponding discrete model.

The second point for further exploration is to consider bending energies that are minimized by a *non-flat reference surface*; for example, consider

$$E_{nonflat} = \int_S (H - H_0)^2 \, dA ,$$

for a prescribed *spontaneous curvature* H_0 . In this case, the energy is no longer quadratic in the immersion of the deformed surface; we can see, however, that it is cubic under isometric deformations: $(H - H_0)^2 = \langle \mathbf{H}, \mathbf{H} \rangle - 2\langle \mathbf{H}, \mathbf{n}H_0 \rangle + H_0^2$; the familiar first term is quadratic and the last term is constant in the isometric immersion \mathbf{x} . The middle term is cubic: \mathbf{H} is linear in \mathbf{x} ; \mathbf{n} is quadratic under isometric deformations as is evident from $\mathbf{n} = \left(\frac{d}{ds_1} \mathbf{x}\right) \times \left(\frac{d}{ds_2} \mathbf{x}\right)$, where s_1 and s_2 are arclength parameterizations of any two orthogonal material directions \mathbf{v}_1 and \mathbf{v}_2 . That the non-flat energy is cubic can potentially be used to obtain simpler computations for gradients and Hessians of the discretized bending.

Acknowledgements. This work was supported in part by the DFG Research Center Matheon “Mathematics for key technologies” in Berlin, the NSF (MSPA Award No. IIS-05-28402, CSR Award No. CNS-06-14770, CAREER Award No. CCF-06-43268), Elsevier, and nVidia. We especially thank Konrad Polthier for facilitating the meeting leading to this work, Charles Han for his tireless production assistance including lighting and rendering, and Dave Glickenstein for discussing his weighted triangulations with us. Finally, we are indebted to the reviewers and editors for offering valuable feedback.

References

- [1] M. ALEXA, *Local control for mesh morphing*, Proc. Int. Conf. on Shape Modeling and Applications, (2001), pp. 209–215.
- [2] C. L. BAJAJ AND G. XU, *Adaptive fairing of surface meshes by geometric diffusion*, in Proceedings of the Fifth International Conference on Information Visualisation, 2001, pp. 731–737.

- [3] S. BALAY, W. D. GROPP, L. C. MCINNES, AND B. F. SMITH, *PETSc 2.0 users manual*, tech. report, Argonne National Laboratory, 1996.
- [4] T. BARAFF AND A. WITKIN, *Large steps in cloth simulation*, in Proceedings of ACM SIGGRAPH, 1998, pp. 43–54.
- [5] M. BEN-CHEN AND C. GOTSMAN, *On the optimality of spectral compression of mesh data*, ACM Trans. Graph., 24 (2005), pp. 60–80.
- [6] W. BLASCKE, *Vorlesungen über Differentialgeometrie*, Springer, 1929.
- [7] A. I. BOBENKO, *A conformal energy for simplicial surfaces*, Combinatorial and Computational Geometry, (2005), pp. 133–143.
- [8] A. I. BOBENKO AND P. SCHRÖDER, *Discrete Willmore Flow*, in Siggraph/Eurographics Sympos. Geom. Processing, 2005, pp. 101–110.
- [9] K. BRAKKE, *The surface evolver*, Exper. Math., 1 (1992), pp. 141–165.
- [10] R. BRIDSON, R. FEDKIW, AND J. ANDERSON, *Robust treatment of collisions, contact and friction for cloth animation*, ACM TOG, 21 (2002), pp. 594–603.
- [11] R. BRIDSON, S. MARINO, AND R. FEDKIW, *Simulation of clothing with folds and wrinkles*, SCA, (2003), pp. 28–36.
- [12] P. CANHAM, *The Minimum Energy of Bending as a Possible Explanation of the Biconcave Shape of the Human Red Blood Cell*, Journal of Theoretical Biology, 26 (1970), pp. 61–81.
- [13] K.-J. CHOI AND H.-S. KO, *Research problems in clothing simulation*, CAD, 37 (2005), pp. 585–592.
- [14] P. CIARLET, *Mathematical Elasticity, Vol III: Theory of Shells*, North-Holland, 2000.
- [15] U. CLARENZ, U. DIEWALD, G. DZIUK, M. RUMPF, AND R. RUSU, *A finite element method for surface restoration with smooth boundary conditions*, CAGD, (2004), pp. 427–445.
- [16] D. COHEN-STEINER AND J.-M. MORVAN, *Restricted Delaunay triangulations and normal cycle*, Sympos. Comput. Geom., (2003), pp. 312–321.
- [17] M. DESBRUN, A. HIRANI, M. LEOK, AND J. E. MARS DEN, *Discrete exterior calculus*, (2005). [arXiv:math.DG/0508341](https://arxiv.org/abs/math/0508341).
- [18] M. DESBRUN, M. MEYER, P. SCHRÖDER, AND A. H. BARR, *Implicit fairing of irregular meshes using diffusion and curvature flow*, in Proceedings of ACM SIGGRAPH, 1999, pp. 317–324.
- [19] M. DO CARMO, *Riemannian Geometry*, Birkhäuser, 1992.
- [20] P. DOYLE AND L. SNELL, *Random walks and electrical networks*, The Carus Mathematical Monographs, 1984.

- [21] R. DUFFIN, *Distributed and Lumped Networks*, Journal of Mathematics and Mechanics, 8 (1959), pp. 793–825.
- [22] M. S. FLOATER AND K. HORMANN, *Surface Parameterization: A Tutorial and Survey*, in Advances in Multiresolution for Geometric Modeling, 2005, pp. 157–186.
- [23] D. GLICKENSTEIN, *Geometric triangulations and discrete Laplacians on manifolds*, (2005). [arxiv:math.MG/0508188](https://arxiv.org/abs/math/0508188).
- [24] S. J. GORTLER, C. GOTSMAN, AND D. THURTON, *Discrete one-forms on meshes and applications to 3D mesh parameterization*, Computer Aided Geometric Design, 23 (2006), pp. 83–112.
- [25] E. GRINSBUN, A. N. HIRANI, M. DESBRUN, AND P. SCHRÖDER, *Discrete shells*, SCA, (2003), pp. 62–67.
- [26] X. GU AND S.-T. YAU, *Global conformal surface parameterization*, in Siggraph/Eurographics Sympos. Geom. Processing, 2003, pp. 127–137.
- [27] M. HAUTH, *Visual Simulation of Deformable Models*, PhD thesis, University of Tübingen, 2004.
- [28] W. HELFRICH, *Elastic Properties of Lipid Bilayers: Theory and Possible Experiments*, Zeitschrift für Naturforschung Teil C, 28 (1973), pp. 693–703.
- [29] K. HILDEBRANDT AND K. POLTHIER, *Anisotropic filtering of non-linear surface features*, Computer Graphics Forum, 23 (2004), pp. 391–400.
- [30] L. HSU, R. KUSNER, AND J. SULLIVAN, *Minimizing the squared mean curvature integral for surfaces in space forms*, Experiment. Math., 1 (1992), pp. 191–207.
- [31] T. J. R. HUGHES, *Finite Element Method - Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliffs, 1987.
- [32] Z. KARNI AND C. GOTSMAN, *Spectral compression of mesh geometry*, in Proceedings of ACM SIGGRAPH, ACM Press, 2000, pp. 279–286.
- [33] J. T. KLOSOWSKI, M. HELD, J. S. B. MITCHELL, H. SOWIZRAL, AND K. ZIKAN, *Efficient collision detection using bounding volume hierarchies of k-dops*, IEEE TVCG, 4 (1998), pp. 21–36.
- [34] Y. LIPMAN, O. SORKINE, D. LEVIN, D. COHEN-OR, C. RÖSSL, AND H.-P. SEIDEL, *Differential coordinates for interactive shape editing*, Proceedings of SMI, (2004), pp. 181–190.
- [35] N. MAGNENAT-THALMANN AND P. VOLINO, *From early draping to haute couture models: 20 years of research*, The Visual Computer, 21 (2005), pp. 506–519.
- [36] C. MERCAT, *Discrete Riemann surfaces and the Ising model*, Communications in Mathematical Physics, 218 (2001), pp. 177–216.

- [37] M. MEYER, M. DESBRUN, P. SCHRÖDER, AND A. H. BARR, *Discrete differential-geometry operators for triangulated 2-manifolds*, in Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds., 2003, pp. 113–134.
- [38] B. MORINI, *Convergence behaviour of inexact newton methods*, Mathematics of Computation, 68 (1999), pp. 1605–1613.
- [39] H. N. NG AND R. L. GRIMSDALE, *Computer graphics techniques for modeling cloth*, IEEE CG&A, 16 (1996), pp. 28–41.
- [40] U. PINKALL AND K. POLTHIER, *Computing discrete minimal surfaces and their conjugates*, Experim. Math., 2 (1993), pp. 15–36.
- [41] K. POLTHIER AND E. PREUSS, *Identifying vector field singularities using a discrete Hodge decomposition*, in Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds., Springer-Verlag, Heidelberg, 2003, pp. 113–134.
- [42] O. SCHENK AND K. GÄRTNER, *Solving unsymmetric sparse systems of linear equations with PARDISO*, Journal of Future Generation Computer Systems, 20 (2004), pp. 475–487.
- [43] R. SCHNEIDER AND L. KOBBELT, *Geometric Fairing of Irregular Meshes for Free-From Surface Design*, CAGD, 18 (2001), pp. 359–379.
- [44] O. SORKINE, *Laplacian mesh processing*, Eurographics STAR - State of The Art Report, (2005), pp. 53–70.
- [45] O. SORKINE, D. COHEN-OR, AND S. TOLEDO, *High-pass quantization for mesh encoding*, in Siggraph/Eurographics Sympos. Geom. Processing, 2003, pp. 42–51.
- [46] D. TERZOPOULOS, J. PLATT, A. BARR, AND K. FLEISCHER, *Elastically deformable models*, in Proceedings of SIGGRAPH, 1987, pp. 205–214.
- [47] B. THOMASZEWSKI AND M. WACKER, *Bending Models for Thin Flexible Objects*, in WSCG Short Communication proceedings, 2006.
- [48] Y. TONG, S. LOMBEYDA, A. N. HIRANI, AND M. DESBRUN, *Discrete multiscale vector field decomposition*, ACM TOC, 22 (2003), pp. 445–452.
- [49] K. WANG, WEIWEI, Y. TONG, M. DESBRUN, AND P. SCHRÖDER, *Edge subdivision schemes and the construction of smooth vector fields*, in Proceedings of ACM SIGGRAPH, 2006, pp. 1041–1048.
- [50] J. H. WHITE, *A global invariant of conformal mappings in space*, Proceedings of the American Mathematical Society, 38 (2000), pp. 162–164.
- [51] H. WHITNEY, *Geometric Integration Theory*, Princeton Univ. Press, 1957.
- [52] T. J. WILLMORE, *Surfaces in conformal geometry*, Annals of Global Analysis and Geometry, 18 (2000), pp. 255–264.
- [53] S. YOSHIZAWA AND A. G. BELYAEV, *Fair Triangle Mesh Generation with Discrete Elastica*, Proceedings of the 2nd Biennial International Conference on Geometric Modeling and Processing, (2002), pp. 119–123.

- [54] H. ZHU, X. JIN, J. FENG, AND Q. PENG, *Survey on cloth animation*, Journal of Computer Aided Design & Computer Graphics, 16 (2004), pp. 613–618.
- [55] O. C. ZIENKIEWICZ AND R. L. TAYLOR, *The finite element method: The basis*, vol. 1, Butterworth and Heinemann, 5th ed., 2000.

5 Appendix

Here we derive IBM’s fully non-linear forces. To compute these forces, one drops the assumption of isometric deformations and allows arbitrary variations of vertices. According to Equation (7), we can write IBM’s full bending energy as a sum over contributions from individual edges:

$$E_{CR} = \sum_i \frac{3\|\mathbf{e}_i\|^2}{2A_i} \left(2 \cos \frac{\theta_i}{2}\right)^2 = \sum_i \frac{3\|\mathbf{e}_i\|^2}{A_i} (1 + \cos \theta_i),$$

where A_i is the combined area of the two triangles meeting at edge \mathbf{e}_i , and θ_i is \mathbf{e}_i ’s dihedral angle.

We focus on a single edge $i = 0$ and its hinge stencil, consisting of the two triangles meeting at \mathbf{e}_0 . We shall drop the subscript 0 wherever this causes no confusion. The non-linear forces arising from edge \mathbf{e}_0 with respect to variations associated with vertex \mathbf{x}_i are

$$\mathbf{f}_i^{nl} = \underbrace{-3(1 + \cos \theta) \nabla_{\mathbf{x}_i} \left(\frac{\|\mathbf{e}_0\|^2}{A_0} \right)}_{\mathbf{f}_i^P} + \underbrace{\frac{3\|\mathbf{e}_0\|^2}{A_0} \sin \theta \nabla_{\mathbf{x}_i} \theta}_{\mathbf{f}_i^B}. \quad (15)$$

Notice that we decompose the force \mathbf{f}_i^{nl} into a sum of two parts—a component \mathbf{f}_i^P , corresponding to in-plane deformations, and another component, \mathbf{f}_i^B , corresponding to pure bending modes. We now provide closed expression for these components.

Nonlinear bending forces With respect to the labels in Figure 10-left, we have

$$\cos \theta = -\langle \mathbf{n}^{(0)}, \mathbf{n}^{(1)} \rangle,$$

where $\mathbf{n}^{(0)}$ and $\mathbf{n}^{(1)}$ are triangle normals of unit length. The derivatives of these normals are obtained by expressing e.g. $\mathbf{n}^{(0)} = \mathbf{e}_0 \times \mathbf{e}_3 / \|\mathbf{e}_0 \times \mathbf{e}_3\|$ and differentiating with respect to e.g. \mathbf{x}_0 . This gives

$$\nabla_{\mathbf{x}_0} \mathbf{n}^{(0)} = \frac{\mathbf{e}_3 \times (\cdot)}{\|\mathbf{e}_0 \times \mathbf{e}_3\|} - \left\langle \mathbf{n}^{(0)}, \frac{\mathbf{e}_3 \times (\cdot)}{\|\mathbf{e}_0 \times \mathbf{e}_3\|} \right\rangle \mathbf{n}^{(0)} = \frac{\mathbf{e}_3 \times \mathbf{n}^{(0)}}{\|\mathbf{e}_0 \times \mathbf{e}_3\|} \otimes \mathbf{n}^{(0)}.$$

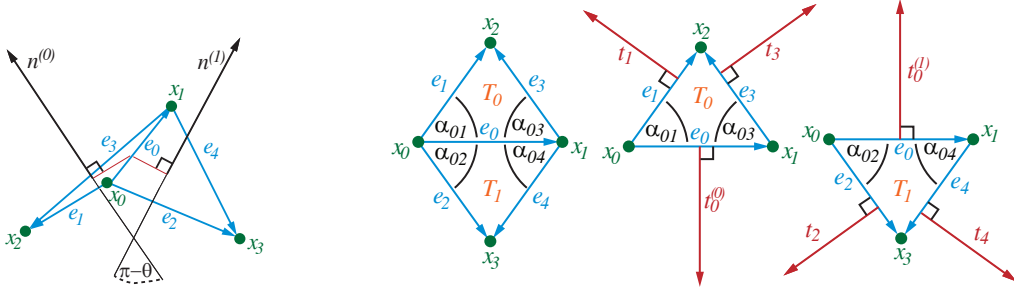


Fig. 10. Labeling of angles, edges, and normals in \mathbf{e}_0 's edge stencil.

where $a \otimes b = ab^T$. We thus obtain

$$\nabla_{\mathbf{x}_0} \cos \theta = \frac{[\mathbf{n}^{(0)}, \mathbf{e}_3, \mathbf{n}^{(1)}]}{\|\mathbf{e}_0 \times \mathbf{e}_3\|} \mathbf{n}^{(0)} + \frac{[\mathbf{n}^{(0)}, \mathbf{e}_4, \mathbf{n}^{(1)}]}{\|\mathbf{e}_0 \times \mathbf{e}_4\|} \mathbf{n}^{(1)},$$

where $[a, b, c] = \langle a \times b, c \rangle$ is the usual triple product. Using that $[a, b, c] = [c, a, b]$ as well as

$$\mathbf{n}^{(1)} \times \mathbf{n}^{(0)} = -\frac{\mathbf{e}_0}{\|\mathbf{e}_0\|} \sin \theta,$$

we arrive at

$$\nabla_{\mathbf{x}_0} \cos \theta = \frac{\sin \theta}{\|\mathbf{e}_0\|} \left(\cot \alpha_{03} \mathbf{n}^{(0)} + \cot \alpha_{04} \mathbf{n}^{(1)} \right),$$

where we use the labeling of Figure 10. Similarly, we derive variations with respect to \mathbf{x}_1 , \mathbf{x}_2 , and \mathbf{x}_3 . From $\nabla_{\mathbf{x}_i} \cos \theta = -\sin \theta \nabla_{\mathbf{x}_i} \theta$ we obtain

$$\begin{aligned} \nabla_{\mathbf{x}_0} \theta &= \frac{-1}{\|\mathbf{e}_0\|} \left(\cot \alpha_{03} \mathbf{n}^{(0)} + \cot \alpha_{04} \mathbf{n}^{(1)} \right), \\ \nabla_{\mathbf{x}_1} \theta &= \frac{-1}{\|\mathbf{e}_0\|} \left(\cot \alpha_{01} \mathbf{n}^{(0)} + \cot \alpha_{02} \mathbf{n}^{(1)} \right), \\ \nabla_{\mathbf{x}_2} \theta &= \frac{1}{\|\mathbf{e}_0\|} \left(\cot \alpha_{01} + \cot \alpha_{03} \right) \mathbf{n}^{(0)} = \frac{\|\mathbf{e}_0\|}{2A(T_0)} \mathbf{n}^{(0)}, \\ \nabla_{\mathbf{x}_3} \theta &= \frac{1}{\|\mathbf{e}_0\|} \left(\cot \alpha_{02} + \cot \alpha_{04} \right) \mathbf{n}^{(1)} = \frac{\|\mathbf{e}_0\|}{2A(T_1)} \mathbf{n}^{(1)}, \end{aligned}$$

where $A(T_i)$ denotes the area of triangle T_i . We note that the above formulas for $\nabla \theta$ are well-known in the literature, see e.g. Bridson [11]; indeed, they can be derived from the fact that $\nabla \theta$ causes no in-plane stretching and is orthogonal to all rigid body modes. This completes our derivation of \mathbf{f}^B .

Nonlinear in-plane forces To compute the gradient $\nabla(\|\mathbf{e}_0\|^2/A_0)$ in (15), which corresponds to in-plane deformations, we need to take derivatives of

$\|\mathbf{e}_0\|^2$ and of $1/A_0$. Referring to the labels of Figure 10, we let \mathbf{t}_i be the vector which is perpendicular to \mathbf{e}_i , is of the same length as \mathbf{e}_i , and lies in the same plane as the triangle which \mathbf{e}_i belongs to. With this notation, we see that

$$\nabla_{\mathbf{x}_0} A_0 = \frac{-\mathbf{t}_3 - \mathbf{t}_4}{2}, \quad \nabla_{\mathbf{x}_1} A_0 = \frac{-\mathbf{t}_1 - \mathbf{t}_2}{2}, \quad \nabla_{\mathbf{x}_2} A_0 = \frac{-\mathbf{t}_0^{(0)}}{2}, \quad \nabla_{\mathbf{x}_3} A_0 = \frac{-\mathbf{t}_0^{(1)}}{2}.$$

Moreover,

$$\nabla_{\mathbf{x}_0} \|\mathbf{e}_0\|^2 = -2\mathbf{e}_0, \quad \nabla_{\mathbf{x}_1} \|\mathbf{e}_0\|^2 = 2\mathbf{e}_0, \quad \nabla_{\mathbf{x}_2} \|\mathbf{e}_0\|^2 = 0, \quad \nabla_{\mathbf{x}_3} \|\mathbf{e}_0\|^2 = 0.$$

This yields

$$\begin{aligned} \nabla_{\mathbf{x}_0} \left(\frac{\|\mathbf{e}_0\|^2}{A_0} \right) &= \frac{-2}{A_0} \mathbf{e}_0 + \frac{\|\mathbf{e}_0\|^2}{2A_0^2} (\mathbf{t}_3 + \mathbf{t}_4), \\ \nabla_{\mathbf{x}_1} \left(\frac{\|\mathbf{e}_0\|^2}{A_0} \right) &= \frac{2}{A_0} \mathbf{e}_0 + \frac{\|\mathbf{e}_0\|^2}{2A_0^2} (\mathbf{t}_1 + \mathbf{t}_2), \\ \nabla_{\mathbf{x}_2} \left(\frac{\|\mathbf{e}_0\|^2}{A_0} \right) &= \frac{\|\mathbf{e}_0\|^2}{2A_0^2} \mathbf{t}_0^{(0)}, \\ \nabla_{\mathbf{x}_3} \left(\frac{\|\mathbf{e}_0\|^2}{A_0} \right) &= \frac{\|\mathbf{e}_0\|^2}{2A_0^2} \mathbf{t}_0^{(1)}. \end{aligned}$$

Discussion of bending gradient We compare the force \mathbf{f}^B associated with pure bending (but no in-plane stretching) from (15) to IBM's force vector \mathbf{f}^{IBM} (which is linear in positions for isometric mesh deformations). We have

$$\mathbf{f}_i^B = -\frac{3\|\mathbf{e}_0\|^2}{A_0} \nabla_{\mathbf{x}_i} \cos \theta \quad \text{and} \quad \mathbf{f}_i^{IBM} = -\left(\mathbf{x}^T \left(\mathbf{L}_0^T \mathbf{M}_0^{-1} \mathbf{L}_0 \right) \right)_i,$$

where \mathbf{L}_0 and \mathbf{M}_0 correspond to the stiffness and mass matrix of the hinge stencil of \mathbf{e}_0 , respectively, and $(\cdot)_i$ refers to the the i^{th} component of the force vector, *i.e.*, the force acting on vertex \mathbf{x}_i . Both, \mathbf{f}^B and \mathbf{f}^{IBM} , are orthogonal to rigid body modes. However, in general,

$$\mathbf{f}^B \neq \mathbf{f}^{IBM}.$$

This discrepancy is due to the fact that \mathbf{f}^B is *tangent to the manifold of isometric deformations*, whereas \mathbf{f}^{IBM} usually contains a component normal to the manifold of isometric deformations. In other words, \mathbf{f}^B causes no *in-plane deformations*—but this is in general not true for \mathbf{f}^{IBM} . What is important for our applications, though, is the fact that \mathbf{f}^{IBM} agrees with \mathbf{f}^B when projected to the subspace of infinitesimal isometric mesh deformations.