Robust Fairing via Conformal Curvature Flow

Keenan Crane
Caltech

Ulrich Pinkall
TU Berlin

Peter Schröder
Caltech

Abstract

We present a formulation of Willmore flow for triangulated surfaces that permits extraordinarily large time steps and naturally preserves the quality of the input mesh. The main insight is that Willmore flow becomes remarkably stable when expressed in curvature space – we develop the precise conditions under which curvature is allowed to evolve. The practical outcome is a highly efficient algorithm that naturally preserves texture and does not require remeshing during the flow. We apply this algorithm to surface fairing, geometric modeling, and construction of constant mean curvature (CMC) surfaces. We also present a new algorithm for length-preserving flow on planar curves, which provides a valuable analogy for the surface case.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

Keywords: digital geometry processing, discrete differential geometry, geometric modeling, surface fairing, shape spaces, conformal geometry, quaternions, spin geometry

Links: DL PDF

1 Introduction

At the most basic level, a *curvature flow* produces successively smoother approximations of a given piece of geometry (*e.g.*, a curve or surface), by reducing a *fairing energy*. Such flows have far-ranging applications in fair surface design, inpainting, denoising, and biological modeling [Helfrich 1973; Canham 1970]; they are also the central object in mathematical problems such as the Willmore conjecture [Pinkall and Sterling 1987].

Numerical methods for curvature flow suffer from two principal difficulties: (I) a severe time step restriction, which often yields unacceptably slow evolution and (II) degeneration of mesh elements, which necessitates frequent remeshing or other corrective devices. We circumvent these issues by (I) using a curvature-based representation of geometry, and (II) working with *conformal* transformations, which naturally preserve the aspect ratio of triangles. The resulting algorithm stably integrates time steps orders of magnitude larger than existing methods (Figure 1), resulting in substantially faster real-world performance (Section 6.4.2).

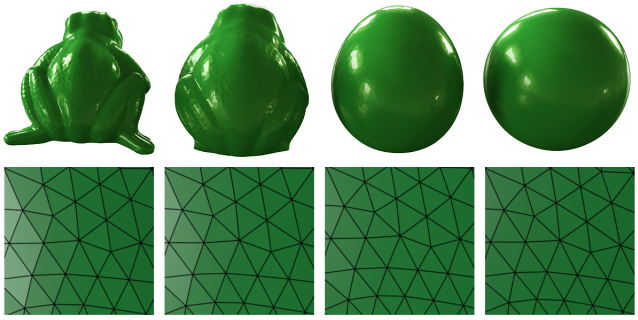


Figure 1: *A detailed frog flows to a round sphere in only three large, explicit time steps* (top). *Meanwhile, the quality of the triangulation* (bottom) *is almost perfectly preserved.*

The success of our method results from a judiciously-chosen *change of variables*: instead of positions, we work with a quantity called *mean curvature half-density*. Not surprisingly, curvature-based energies become easier to minimize when working directly with curvature itself! However, we must now understand the precise *integrability conditions* under which curvature variables remain valid, *i.e.*, when can curvature be integrated to recover position? Kamberov *et al.* [1998] and later Crane *et al.* [2011] investigate this question for topological spheres; we complete the picture by establishing previously unknown integrability conditions for surfaces of arbitrary topological type. In this paper we focus on curvature flow, providing a drop-in replacement for applications involving surface fairing and variational surface modeling – in particular, we show how to express Willmore flow via gradient descent on a quadratic energy subject to simple linear constraints. These insights are not specific to curvature flow, however, and can be applied to any geometry processing application where preservation of the texture or mesh is desirable.

2 Preliminaries

We adopt two essential conventions from Crane *et al.* [2011]. First, we interpret any surface in $\mathbb{R}^3$ (*e.g.*, a triangle mesh) as the image of a *conformal immersion* (Section 2.2.1). Second, we interpret three-dimensional vectors as *imaginary quaternions* (Section 2.3). Proofs in the appendix make use of quaternion-valued differential forms; interested readers may benefit from the material in [Kamberov et al. 2002; Crane 2013].

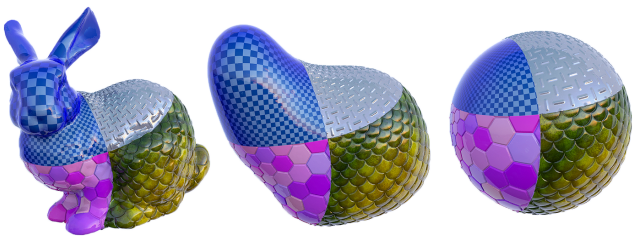


Figure 2: *Our flow gracefully preserves the appearance of texture throughout all stages of the flow.*

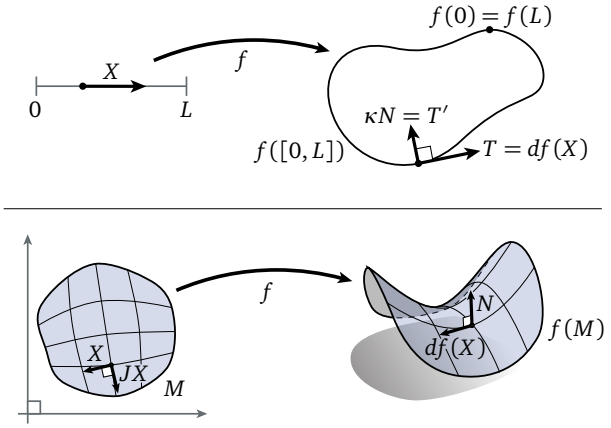


Figure 3: *The basic principles applied in the curve* (top) *and surface* (bottom) *cases are the same.*

2.1 Notation

Single angle brackets $\langle \cdot, \cdot \rangle$ and bars $|\cdot|$ denote the Euclidean inner product and norm (respectively) on vectors in $\mathbb{R}^n$; double brackets $\langle\langle \cdot, \cdot \rangle\rangle$ and bars $||\cdot||$ denote the $\mathscr{L}^2$ inner product and norm on functions. If $\varphi(t)$ is a time-varying quantity, then $\dot{\varphi}$ denotes the time derivative of φ at time zero, *i.e.*, $\dot{\varphi} := \frac{d}{dt}\varphi|_{t=0}$. For any quantity $\varphi(s)$ defined along a 1D curve, φ' denotes the spatial derivative $\frac{d}{ds}\varphi$. The symbol $\mathbb{1}$ denotes a function equal to one at every point, or a vector containing a one in each entry.

2.2 Curves and Surfaces

A unified description of curves and surfaces helps illuminate the relationship between algorithms in 2D and 3D. In particular, a planar curve can be viewed as the image of some map f from an interval $[0, L] \subset \mathbb{R}$ into $\mathbb{R}^2$ (Figure 3, top). Likewise, any surface can be viewed as the image of some map f from a domain M into $\mathbb{R}^3$ (Figure 3, bottom). Note that M need not be a piece of the plane – in our case it can be any compact, orientable topological surface. The *differential* df maps a tangent vector X in the domain to the corresponding embedded tangent vector $df(X)$. If df maps nonzero vectors to nonzero vectors then f is an *immersion*. (Note that immersions may still have self-intersections.)

Throughout we use Δ to denote the Laplace-Beltrami operator induced by f. The quantity $\Delta f/2$ is called the *curvature normal*, equal to κN for curves and HN for surfaces, where κ and H are the *curvature* and *mean curvature*, respectively, and N is the unit normal.

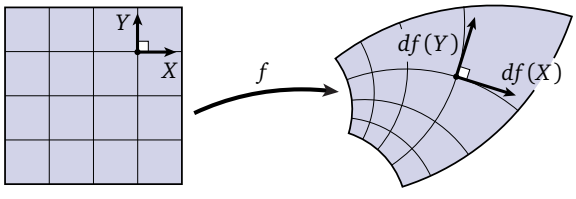


Figure 4: *We work with angle-preserving or* conformal *maps, which simplify analytical expressions and result in numerical robustness at the computational level.*

2.2.1 Isometric and Conformal Immersions

Expressions involving curves are simplified by using an *arc-length* or *isometric* parameterization, meaning that the length of any tangent vector X is preserved: $|df(X)| = |X|$. Similarly, expressions involving surfaces are simplified by assuming that f is *conformal*, meaning that angles are preserved (Figure 4), or equivalently, that tangent spaces are scaled uniformly (*i.e.*, $|df(X)| = e^u|X|$ for some real function u). We can then use $|df|$ and $|df|^2$ to denote the length element and area element induced by f, respectively. The quantity $\mu := H|df|$ is called *mean curvature half-density*, a scale-invariant version of mean curvature. Note that *any* surface can be viewed as the image of a conformal immersion, even if one does not explicitly construct the map f. Hence, any fact we obtain about conformal immersions can also be applied to triangle meshes in $\mathbb{R}^3$, which is the perspective adopted here.

2.3 Geometry in the Quaternions

The *quaternions* $\mathbb{H}$ are a number system well-suited to three-dimensional geometry, and provide a natural algebraic language for conformally immersed surfaces [Kamberov et al. 2002]. As usual, we use $\{1, i, j, k\}$ to denote a basis for $\mathbb{H}$; vectors $(a, b, c) \in \mathbb{R}^3$ can then be identified with elements $v = ai + bj + ck$ of the *imaginary quaternions* $\mathrm{Im}\,\mathbb{H} := \mathrm{span}\{i, j, k\}$. Letting $\bar{q}$ denote conjugation, the expression $\bar{q}vq$ represents rotation of a vector v for any *unit* quaternion q, or rotation and scaling if $|q| \neq 1$. Explicitly, $q = a(\cos(\theta/2) - \sin(\theta/2)w)$ yields scaling by a^2 and rotation by an angle θ around the unit axis $w \in \mathrm{Im}\,\mathbb{H}$.

3 Related Work

There are a wide variety of surface fairing procedures for triangle meshes, which at first glance appear to be quite disparate. Yet most fairing algorithms can be viewed as numerical minimization of either the *membrane energy* E_A, or the *Willmore energy* E_W. To simplify discussion, we will assume that the surface M has no boundary – in this case, membrane energy is just the surface area

$$E_A(f) := \int_M dA,$$

and Willmore energy is the squared $\mathscr{L}^2$ norm of mean curvature

$$E_W(f) := \int_M H^2 dA,$$

where dA denotes the area element. Minimizing E_A via gradient descent leads to *mean curvature flow* $\dot{f} = -HN$, where the surface moves in the normal direction with speed proportional to curvature. Noting that $\Delta f = 2HN$, we can also write this flow as

$$\dot{f} = -\tfrac{1}{2}\Delta f.$$

Similarly, we can write Willmore energy as $E_W(f) = \frac{1}{4}\langle\langle \Delta f, \Delta f \rangle\rangle = \frac{1}{4}\langle\langle \Delta^2 f, f \rangle\rangle$. If we ignore the dependence of Δ on f when taking the gradient of E_W, we get the *bi-Laplacian flow*

$$\dot{f} = -\tfrac{1}{2}\Delta^2 f,$$

which approximates the fully nonlinear *Willmore flow*

$$\dot{f} = -\nabla E_W(f).$$

A bi-Laplacian term also appears in *surface diffusion flow* [Schneider and Kobbelt 2001]. Importantly, all these flows are *nonlinear* PDEs since the operator Δ is itself a function of the immersion f.

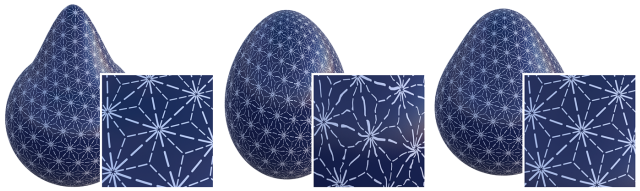


Figure 5: Left: *a bump representative of small surface detail.* Center: *standard fairing distorts texture, even with tangential smoothing (method of Yoshizawa & Belyaev [2002] applied to the method of Desbrun* et al. *[1999]).* Right: *conformal fairing preserves texture while producing a pleasing geometric shape.*

From this perspective, many algorithms for surface fairing arise from different choices of spatial and temporal discretization: Brakke's *Surface Evolver* [1992] minimizes membrane energy via explicit gradient descent, corresponding to mean curvature flow via the forward Euler method; Taubin's $\lambda|\mu$ algorithm [1995] amounts to bi-Laplacian flow via forward Euler when $\lambda = -\mu$; and the *implicit fairing* method of Desbrun *et al.* [1999] corresponds to mean curvature flow via backward Euler. More recently, Bobenko and Schröder [2005] and Wardetzky *et al.* [2007] investigate discrete Willmore flow, using a semi-implicit quasi-Newton scheme to cope with nonlinearity. Common to all these methods are time step restrictions based on the smallest edge length h – at a practical level performance degrades rapidly as resolution increases or elements degenerate. *Explicit* methods typically exhibit restrictions of $O(h^2)$ and $O(h^4)$ for mean curvature and Willmore flow, respectively (see [Olischläger and Rumpf 2009] for a proposal to ameliorate this restriction). Implicit integrators such as backward Euler improve the situation, but do not guarantee unconditional stability since flows are inherently nonlinear, due to the dependence of Δ on f (*cf.* [Desbrun et al. 1999]). In light of this situation, it is rather remarkable to find a change of variables that evades this restriction.

One can also compare geometric qualities of these methods. For example, mean curvature flow can develop sharp singularities which undermine the fairing process, even in the continuous setting [Colding and Minicozzi 2012]. Kazhdan *et al.* propose a modification that helps avoid degeneracy but can still produce sharp features; in contrast, Willmore or bi-Laplacian flow tend to produce rounder, more æsthetic shapes (Figure 6). These flows also permit tangent constraints at the boundary, valuable for geometric modeling [Celniker and Gossard 1991; Welch and Witkin 1994; Clarenz et al. 2004].

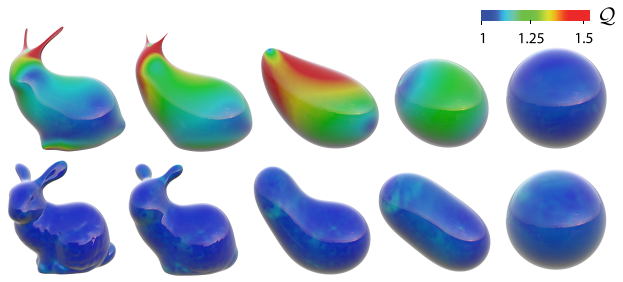


Figure 6: Top: *the modified mean curvature flow of Kazhdan* et al. *yields a conformal map to the sphere, but exhibits conformal distortion at intermediate steps; it also produces sharp features,* e.g.*, at the tips of the ears.* Bottom: *our flow produces smooth geometric features and low conformal distortion throughout.*

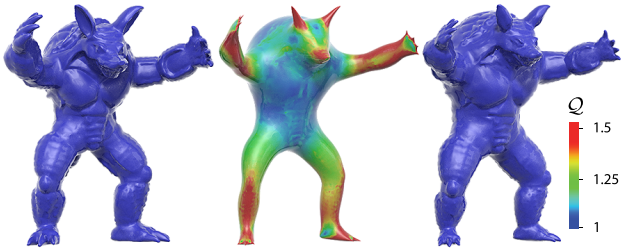


Figure 7: *Fairing a triangle mesh* (left) *via standard methods such as implicit mean curvature flow* (center) *produces significant conformal distortion ($\mathcal{Q}$) and sharp geometric discontinuities. Projecting onto an angle-preserving deformation does not help, since in the discrete case such deformations are rigid and one simply recovers the initial surface* (right).

Existing flows also produce unwanted metric distortion which degrades texture and mesh elements, and further exacerbates stability issues. Simple corrective devices do not quite work as desired. For instance, Laplacian-based tangential smoothing helps avoid degenerate elements at the cost of distorting texture (Figure 5), and suffers from the same stability issues as curvature flow itself [Yoshizawa and Belyaev 2002]. Similarly, adaptive remeshing helps maintain element quality [deGoes et al. 2008], but neither prevents texture distortion nor improves asymptotic stability. Another tempting idea is to project onto the nearest angle-preserving deformation, yet this approach is far too rigid since neighboring triangles are forced to have identical scale. Figure 7 demonstrates that projections via *ShapeUp* [Bouaziz et al. 2012] very nearly restore the original surface, reverting any fairing that may have occurred.

Finally, conformal flows of the *metric* (*e.g.*, Yamabe) have proven invaluable for applications like parameterization [Gu et al. 2011]. However, these methods work with *intrinsic* (Gaussian) curvature which is insufficient to determine the three-dimensional geometry of a surface, and hence cannot be used for fairing. We present the first method for *extrinsic* conformal flow, which is naturally suited to surfaces in $\mathbb{R}^3$.

4 Curvature Flow in Curvature Space

Our main point of departure from existing algorithms is that we manipulate curvature directly – for now, let u denote a generic curvature variable equal to either κ for curves or μ for surfaces. Fairness is measured via the quadratic energy $E(u) := ||u||^2$, leading to the simple gradient flow

$$\dot{u} = -2u. \tag{1}$$

However, we must ensure that curvature remains *integrable* as it evolves, *i.e.*, at any point in time there must be positions f that exhibit the given curvature u. This seemingly difficult condition can be expressed as a set of *linear* constraints $\langle\langle \dot{u}, c_i \rangle\rangle = 0$ for some collection of easily computed constraint functions c_i. In practice, we integrate Eq. (1) using the forward Euler scheme

$$u^{k+1} = u^k - 2\tau u^k \tag{2}$$

for some fixed time step $\tau > 0$; constraints are enforced by building an orthonormal basis $\{\hat{c}_i\}$ via the Gram-Schmidt process and augmenting the initial flow direction $v := -2u$ via $v \leftarrow v - \sum_i \langle\langle v, \hat{c}_i \rangle\rangle \hat{c}_i$. To recover the final geometry we integrate curvature to get tangents, then integrate tangents to get positions, as discussed in Sections 5 and 6. The overall procedure is outlined in Table 1.

Step	Description	Curve	*Section*	Surface	*Section*
I.	Evaluate curvature.	$\kappa \leftarrow \frac{1}{2}\langle N, \Delta f \rangle$	5.2	$H \leftarrow \frac{1}{2}\langle N, \Delta f \rangle$	6.1
II.	Pick a desired flow direction.	$\dot{\kappa} \leftarrow -\nabla E_C(\kappa)$	5	$\dot{\rho} \leftarrow -\nabla E_W(\rho)$	6.1
III.	Build a constraint basis.	Gram-Schmidt $\{\mathbb{1}, f^x, f^y\}$	5	Gram-Schmidt $\{\mathbb{1}, N^{x,y,z}, Z_i^{x,y,z}\}$	6.2
IV.	Project flow onto constraints.	$\dot{\kappa} \leftarrow \dot{\kappa} - \sum_i \langle\langle \dot{\kappa}, \hat{c}_i \rangle\rangle \hat{c}_i$	4	$\dot{\rho} \leftarrow \dot{\rho} - \sum_i \langle\langle \dot{\rho}, \hat{c}_i \rangle\rangle \hat{c}_i$	4
V.	Take an explicit Euler step.	$\kappa \leftarrow \kappa + \tau\dot{\kappa}$	4	$\rho \leftarrow \rho + \tau\dot{\rho}$	4
VI.	Recover tangents.	$\tilde{T} \leftarrow$ Integrate κ	5.1,5.2	Solve $(D-\rho)\lambda = \gamma\lambda$, $\tilde{T} \leftarrow \tilde{\lambda} T \lambda$	6
VII.	Recover positions.	Solve $\Delta \tilde{f} = \nabla \cdot \tilde{T}$	5.2	Solve $\Delta \tilde{f} = \nabla \cdot \tilde{T}$	6

Table 1: *Our algorithms for curve* (left) *and surface* (right) *flows are nearly identical in structure.*

This formulation has a number of valuable consequences. First, since only curvatures are prescribed, we are free to reconstruct positions that preserve lengths (Figure 9) or angles (Figure 1). Unlike constraint- or penalty-based methods, these quantities are preserved *by construction*. Second, we obtain greatly improved stability, chiefly because the flow we want to integrate (Eq. (1)) involves *no spatial derivatives*. Therefore, our one and only stability criterion is that $|1-2\tau| < 1$, or equivalently, $\tau < 1$. (The addition of constraints only *improves* stability, since projection onto the constraint set *reduces* the norm of the speed function.) Experiments agree perfectly with this analysis: setting τ just above 1 yields an unstable flow; setting it just below 1 produces a stable flow, *independent of mesh quality or resolution* (see Figure 18).

Note that the flow $\dot{u} = -2u$ gives an appearance different from traditional smoothing, since we take the gradient of E with respect to a nonstandard metric ($\mathscr{L}^2$ norm on curvature rather than position). In particular, large features shrink at the same rate as small bumps – see Figure 19 and Appendix F. To achieve more typical behavior, one can simply filter the flow direction v – we use the spectral filter

$$v \leftarrow v - (\mathrm{id} - \sigma\Delta^k)^{-1} v,$$

which damps low-frequency motion by subtracting a regularized version of v from itself. The parameters $\sigma > 0$ and $k \in \mathbb{Z}$ control the degree of regularization and the filter shape, respectively; evaluating the filter amounts to solving a scalar Poisson equation (see Taubin [1995] for a more thorough discussion of this approach to filtering). For $k = 2$ this flow closely approximates traditional position-based Willmore flow (Figure 19), since Δ^{-2} approximates the norm on curvature induced by the $\mathscr{L}^2$ norm on position. Figure 8 demonstrates $k = 0, 1, 3$.

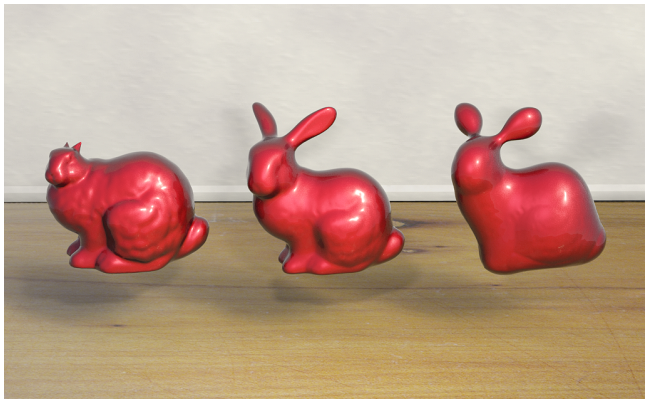


Figure 8: *By augmenting the descent direction we achieve a wide variety of flows. Here we apply frequency-space filters to smooth out features at different scales (from left to right, $k=0,1,3$).*

5 Isometric Curve Flows

Let $f : [0, L] \to \mathbb{R}^2$ be an isometrically immersed curve with unit tangents $T = f'$ and curvature normal $\kappa N = T'$. A natural fairing energy is just the squared $\mathscr{L}^2$ norm of curvature

$$E_C(\kappa) := \int_0^L \kappa^2 d\ell = ||\kappa||^2 .$$

Note that this energy would be rather difficult to express in terms of the positions f, involving second derivatives and a high degree of nonlinearity.

For open curves, the resulting flow can be integrated without modification. In the case of a closed regular curve, however, $\dot{\kappa}$ must satisfy two additional constraints: endpoints must meet ($f(0) = f(L)$), and tangents must agree at endpoints ($T(0) = T(L)$). How do we express these constraints in terms of $\dot{\kappa}$? The latter condition implies that the tangent turns around a whole number of times as we walk around the curve, or equivalently, that the total curvature $\int_0^L \kappa d\ell$ equals $2\pi k$ for some fixed *turning number* $k \in \mathbb{Z}$. If κ satisfies this condition at time $t = 0$, it will remain satisfied as long as the total curvature does not change, *i.e.*, as long as $\int_0^L \dot{\kappa} d\ell = 0$. The condition $f(0) = f(L)$ is not quite as easy to reformulate, but a fairly straightforward derivation (Appendix A) reveals another simple linear condition, namely $\int_0^L \dot{\kappa} f \, d\ell = 0$. A concise way to write the whole collection of constraints is then

$$\langle\langle \dot{\kappa}, \mathbb{1} \rangle\rangle = \langle\langle \dot{\kappa}, f^x \rangle\rangle = \langle\langle \dot{\kappa}, f^y \rangle\rangle = 0,$$

where $f^x, f^y : [0, L] \to \mathbb{R}$ are the current x- and y-coordinate functions of the curve, respectively. These constraints are enforced as described in Section 4. Figure 10 demonstrates that curves produced by the resulting flow remain smooth; in contrast, 1D mean curvature flow $\dot{f} = -2\kappa N$ develops sharp cusps that can be a source of numerical instability. On a curve of 2k vertices, each step of our flow takes about 1.4ms; a global minimum can be found in about six time steps.

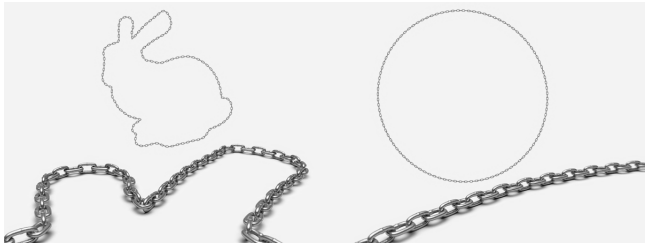


Figure 9: *The silhouette of a bunny* (top left) *flows to a perfectly round circle* (top right) *while preserving edge lengths* (bottom).

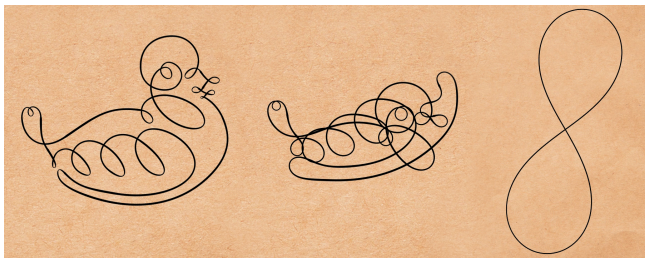


Figure 10: *Fairing a 1D duck. Our flow prohibits sharp cusps, ultimately flowing to the smoothest curve of equal turning number.*

Our analysis of curves illustrates the general method for establishing integrability conditions on curvature-based flow: we write down a constraint in terms of positions or tangents, differentiate it in time, then convert it to a constraint on curvature using established relationships. The only difference in the case of surfaces is that these relationships become more intricate (Section 6.2).

5.1 Recovering Position

After updating the curvature function κ, we can recover new unit tangents $\tilde{T}$ by integrating curvature. In particular, $\tilde{T}(s) = (\cos\theta(s), \sin\theta(s))$, where

$$\theta(s) = \theta_0 + \int_0^s \kappa \, d\ell. \tag{3}$$

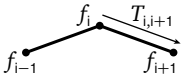
We then recover new positions $\tilde{f}$ by integrating tangents:

$$\tilde{f}(s) = \tilde{f}_0 + \int_0^s \tilde{T} \, d\ell. \tag{4}$$

Positions are determined only up to a global rotation and translation, determined by values $\theta_0 \in \mathbb{R}$ and $f_0 \in \mathbb{R}^2$; in practice we simply remove the mean change in position and angle.

In the smooth setting, length preservation follows from our isometry assumption. Numerically, however, we experience a small amount of discretization error, which we distribute uniformly by solving the optimization problem $\min_{\tilde{f}} |d\tilde{f} - \tilde{T}|^2$ for the vertex positions $\tilde{f}$ that best agree with our desired tangents $\tilde{T}$. Equivalently, we can solve the linear Poisson problem $\Delta\tilde{f} = \nabla \cdot \tilde{T}$. Figure 11 demonstrates that any remaining length distortion is quite small even for large time steps, and converges quadratically under temporal refinement.

5.2 Spatial Discretization



We discretize f as a collection of vertex coordinates $f_1, \ldots, f_m \in \mathbb{R}^2$, tangents $T_{ij} = f_j - f_i$ associated with edges, and pointwise curvatures $\kappa_i \in \mathbb{R}$ at vertices. Let $\ell_{ij} = |T_{ij}|$ be the edge lengths of the initial curve. To recover tangents we compute cumulative sums

$$\theta_k = \theta_0 + \sum_{i=1}^{k} \tfrac{1}{2}(\ell_{i-1,i} + \ell_{i,i+1})\kappa_i.$$

New tangent vectors are then given by $\tilde{T}_{ij} = \ell_{ij}(\cos\theta_i, \sin\theta_i)$. To recover positions we solve the discrete Poisson equation $\mathsf{L}\tilde{f} = b$ where L is an $m \times m$ matrix with off-diagonal entries $\mathsf{L}_{ij} = -1/\ell_{ij}$ for any edge (i,j) and diagonal entries $\mathsf{L}_{ii} = 1/\ell_{i-1,i} + 1/\ell_{i,i+1}$. The right-hand side b is the discrete divergence of the new tangent field, given by $b_i = \tilde{T}_{i-1,i}/\ell_{i-1,i} - \tilde{T}_{i,i+1}/\ell_{i,i+1}$ at each vertex. (Note that this system can be solved as a pair of *scalar* Poisson equations with either x- or y-components of $\tilde{T}$ on the right-hand side.)

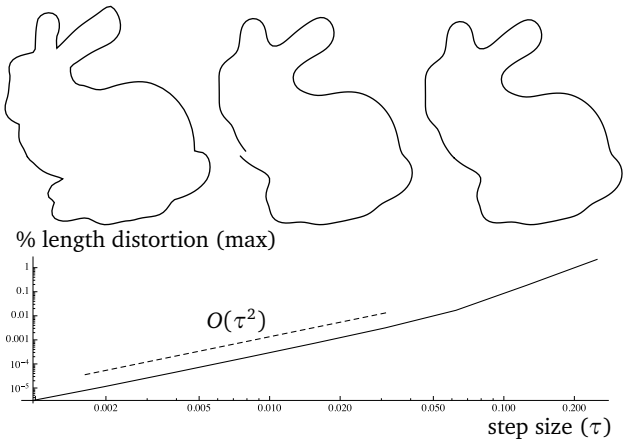


Figure 11: *Flow on a 1D bunny with 200 edges* (top left). *Even after taking a large time step, the failure to close is slight* (top center) *and yields little length distortion when projected onto the nearest closed curve* (top right). Bottom: *log-log plot of worst edge length distortion as a function of step size; the dashed line represents quadratic convergence.*

6 Conformal Surface Flows

Our treatment of surfaces closely parallels the curve case, except that instead of quantities f, T, and κ, we now work with analogous quantities f, λ, and ρ, respectively. The immersion $f : M \to \mathbb{R}^3$ again describes the position of our surface in space; the map $\lambda : M \to \mathbb{H}$ describes how tangents *change* from one surface to the next; likewise the map $\rho : M \to \mathbb{R}$ describes a *change* in curvature. More precisely, for a prescribed function ρ, we solve the *time-independent Dirac equation*

$$(D - \rho)\lambda = \gamma\lambda \tag{5}$$

for the smallest eigenvalue γ to recover the function λ, where D is the *quaternionic Dirac operator* on f [Crane et al. 2011]. We then use λ to compute new tangents $\tilde{T} = \bar{\lambda} T \lambda$ and solve the Poisson equation

$$\Delta\tilde{f} = \nabla \cdot \tilde{T} \tag{6}$$

for new positions $\tilde{f}$. A surface obtained this way will have curvature $\tilde{\mu} = \mu + (\rho - \gamma)|df|$, and will by construction be a conformal deformation of the original surface. We solve these equations as described in [Crane et al. 2011], except that for convenience we now specify ρ as a value per vertex – Appendix G gives a simple facewise construction of the final matrix used to solve Eq. (5).

6.1 Willmore Flow

The key motivation for this setup is that Willmore energy amounts to a simple quadratic function of mean curvature half density:

$$E_W(\mu) = \int_M H^2 |df|^2 = \int_M \mu^2 = \|\mu\|^2.$$

Gradient flow with respect to μ is then quite simply $\dot{\mu} = -2\mu$, or equivalently, $\dot{\rho} = -H$ (Appendix F). We again apply the forward Euler scheme $\rho^{m+1} = \rho^m - 2\tau H^m$, where H^m is the pointwise mean curvature of the current mesh computed via the cotan Laplacian [Desbrun et al. 1999]. In practice we express ρ with respect to the previous surface in the flow, hence $\rho^m = 0$ and we simply solve the smallest eigenvalue problem $(D + \tau H^m)\lambda = \gamma\lambda$.

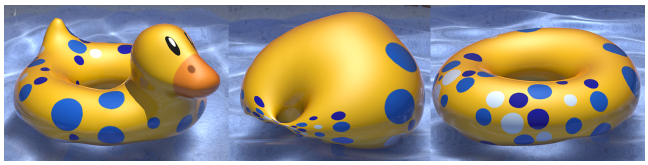


Figure 12: *Duck with nontrivial topology* (left). Center: *unconstrained flow yields distortion of both geometry and texture.* Right: *an exactness constraint prevents distortion.*

6.2 Constraints

For topological disks this flow can be integrated without further modification. For more general surfaces, however, our flow direction must satisfy the following linear constraints:

- **Total Curvature**: As with curves, total curvature must remain constant (Appendix B), which is enforced via the constraint $\langle\langle \dot{\rho}, \mathbb{1} \rangle\rangle = 0$.
- **Exactness**: For non-simply connected surfaces (*e.g.*, a torus or annulus) we must add the constraint $\langle\langle \dot{\rho}, Z_i^x \rangle\rangle = \langle\langle \dot{\rho}, Z_i^y \rangle\rangle = \langle\langle \dot{\rho}, Z_i^z \rangle\rangle = 0$ for Z_i solutions to $DZ_i = \upsilon_i$, where $\{\upsilon_i\}$ is a basis for harmonic vector fields on the surface (Appendix C).
- **Inversion**: sphere inversions preserve both Willmore energy and conformal structure, but may produce area distortion [Blaschke and Thomsen 1929]. To avoid inversions, we apply the constraint $\langle\langle \dot{\rho}, N^x \rangle\rangle = \langle\langle \dot{\rho}, N^y \rangle\rangle = \langle\langle \dot{\rho}, N^z \rangle\rangle = 0$, where $N^{x,y,z}$ are the scalar components of the normal N (Appendix D).
- **Möbius Balancing** (optional): We can also use Möbius transformations to *reduce* area distortion, in particular by *adding* $\langle a, N \rangle$ to $\dot{\rho}$, where $a \in \mathbb{R}^3$ is computed as in Appendix E.1.

Overall we have a constraint basis $\{\mathbb{1}, N^{x,y,z}, Z_i^{x,y,z}\}$ – these constraints are enforced as described in Section 4. On a mesh N is the vector of vertex normals; the functions Z_i are computed as described in Appendix C.1. Möbius balancing must be applied after the inversion constraint (otherwise it will be ignored), and prior to any remaining constraints. Figure 12 compares surfaces of nontrivial genus obtained with and without our exactness constraint. Notice that this constraint is crucial not only for preserving texture, but also for preventing degenerate geometry. Figure 13 illustrates scale artifacts resulting from sphere inversions, and the improved area distribution resulting from Möbius balancing.

Figure 13: *Topologist's view a of a coffee cup.* Top: *Sphere inversions neither distort angles nor increase Willmore energy, but distort area in undesirable ways.* Bottom: *a simple linear constraint prevents unnecessary inversion, and improves area distortion by flowing toward a desired metric.*

6.3 Boundary Conditions

We extend the boundary conditions described by Crane *et al.* [2011], providing a more general treatment of prescribed vectors, and allowing position-based constraints.

6.3.1 Prescribed Vectors

The surface boundary has a natural coordinate frame consisting of the unit normal N, unit vector T tangent to the boundary, and binormal $B := T \times N$ (Figure 14). We generalize the boundary conditions of Crane *et al.* by prescribing a new direction $\tilde{V}$ for any linear combination V of N, T, and B. In particular, if θ is the angle between V and $\tilde{V}$, and $w := V \times \tilde{V}$, then the function $\lambda_\partial := (\cos\frac{\theta}{2} - w \sin\frac{\theta}{2})(a - b\tilde{V})$ maps V to $\tilde{V}$ for any pair of values $a, b \in \mathbb{R}$. (Numerical implementation mirrors [Crane et al. 2011, Sect. 5.7].) Bohle and Pinkall [2013] make the important observation that the initial direction V has a profound effect on the character of Eq. (5). In particular, this problem will be *elliptic* if V does not coincide with the directions $\pm N$, and *self-adjoint* if V is perpendicular to T; choosing V from the $N-B$ plane (*excluding* $\pm N$) therefore yields the best numerical behavior. Moreover, by setting $V = B$ and minimizing Willmore energy one can rapidly produce surfaces of constant mean curvature, sought after in applications such as architectural geometry [Pan et al. 2012].

6.3.2 Fixed Boundary

To force the boundary curve $f|_{\partial M}$ to match the boundary of a target surface $\hat{f}$, we find values λ_∂ along the boundary that map initial tangents T to the target tangents $\hat{T}$, and make initial normals N parallel with target normals $\hat{N}$, as in Section 6.3.1. We approximate T at vertices as the mean of incident edge vectors; N is any vertex normal orthogonal to T. Minimizing the residual $|(D - \rho)\lambda|^2$ subject to $\lambda|_{\partial M} = \lambda_\partial$ yields a standard linear least squares problem – in particular we build the matrix X described in Appendix G, moving boundary columns to the right-hand side. We recover the new surface by minimizing $|d\tilde{f} - \tilde{\lambda} T \lambda|^2$ subject to $\tilde{f}|_{\partial M} = \hat{f}_{\partial M}$, yielding a Poisson equation with fixed boundary values. Figure 15, right shows one example.

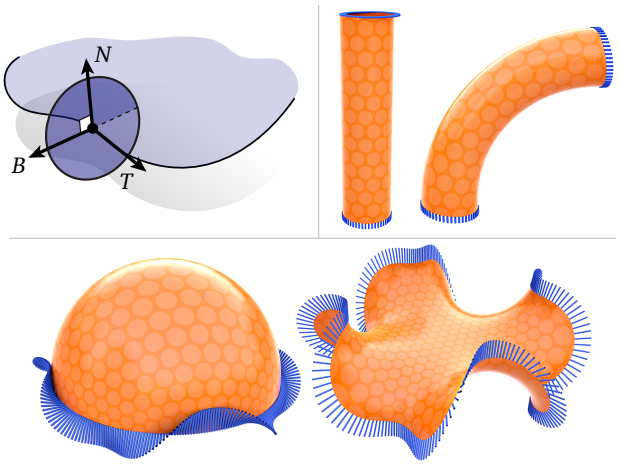


Figure 14: Top left: *prescribing new boundary data from the $N-B$ plane (excluding $\pm N$) results in a well-behaved boundary value problem. For instance, rotated binormals at the top of a cylinder produce a natural bend* (top right)*; oscillating binormals on the hemisphere yield a CMC surface* (bottom).

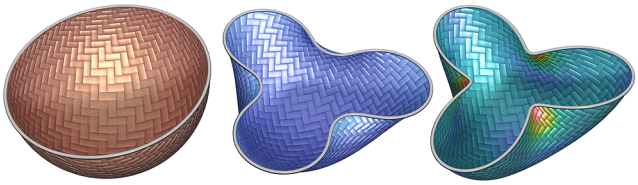


Figure 15: *The boundary of a hemisphere* (left) *is modified using constraints on either tangents* (center) *or positions* (right). *In the latter case a perfectly conformal deformation may not exist – we instead find the deformation with least distortion in the* $\mathscr{L}^2$ *sense.*

6.4 Evaluation

6.4.1 Convergence

As with curves (Section 5.1), integrability is not satisfied exactly due to spatial and temporal discretization error – the Poisson equation used to recover position (Eq. (6)) distributes this error over the domain. Even on coarse meshes the resulting conformal distortion is slight, and converges linearly under temporal and spatial refinement. Figure 16 plots conformal distortion for a flow of fixed duration using progressively smaller time steps. (Spatial refinement was investigated by Crane *et al.* [2011, Sect. 6.3].)

6.4.2 Comparison

We evaluated a variety of fairing methods in terms of numerical stability and conformal distortion. The quality of a map $\phi : f(M) \to \tilde{f}(M)$ was measured via the *quasi-conformal error* $\mathcal{Q}$, defined as the ratio of largest to smallest singular value of the differential $d\phi$ [Sander et al. 2001]. Ideally, $\mathcal{Q} = \mathbb{1}$. All methods were carefully optimized and run on a 2.4 GHz Intel Core 2 Duo machine; for each method we used appropriate solvers from *SuiteSparse* [Chen et al. 2008], re-using symbolic and numeric factorizations wherever possible. The main cost in our method is a sparse eigenvalue problem (Eq. (5)), solved via a simple inverse power method, *i.e.*, we prefactor X via CHOLMOD and repeatedly apply backsubstitution. Since consecutive time steps are quite

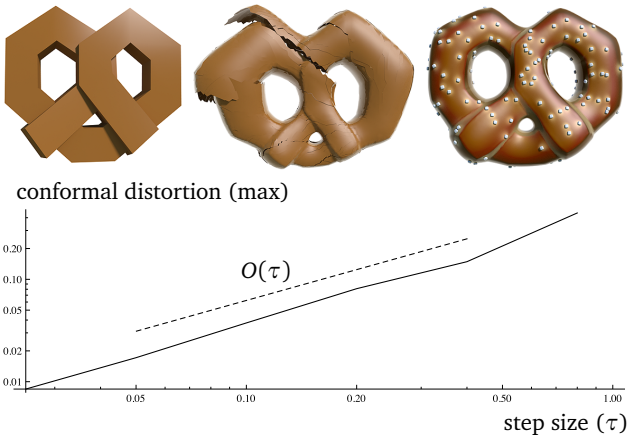


Figure 16: *Curvature flow on a genus-3 pretzel.* Top left: *initial surface.* Top center: *due to time discretization error the surface may fail to close perfectly – here we take an excessively large time step to exacerbate the effect.* Top right*: final surface recovered by our algorithm (salt added for taste).* Bottom: *log-log plot of* $\mathcal{Q} - 1$ *versus time step size; dashed line represents linear convergence.*

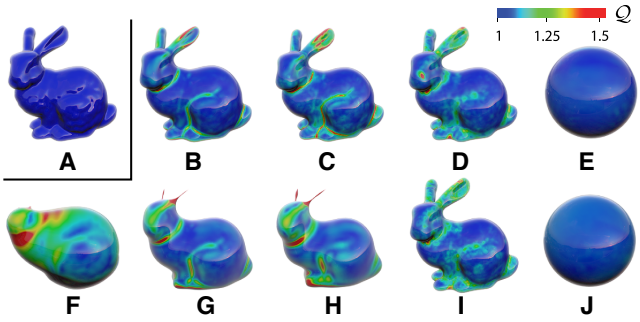


Figure 17: *Various flows applied to* (**A**) *for 10s of CPU time at the maximum stable time step. See Section 6.4.2 for a list of methods used. Note that the majority of methods do not make significant progress, exhibit significant conformal distortion, and/or demonstrate unwanted geometric artifacts (e.g., ears become sharp spikes).*

similar, we use only a single power iteration; the overall cost is therefore identical to a single linear solve. Note that we do *not* require a specialized eigensolver like ARPACK. For non-simply connected surfaces we must also solve for the functions Z_i; the matrix X can again be prefactored. Our implementation extends freely-available code for solving Eq. (5) and (6) [Crane 2012].

Figure 17 shows the result of each method after 10s of CPU time, using the maximum stable time step in each case. We use an isotropic mesh with near-uniform edge lengths [Botsch and Kobbelt 2004], a best-case scenario for traditional methods. Methods were (**B**) explicit mean curvature flow [Brakke 1992], (**C**) explicit bi-Laplacian flow [Taubin 1995]; (**D**) implicit discrete Willmore flow [Bobenko and Schröder 2005], (**E**) implicit modified mean curvature flow [Kazhdan et al. 2012], (**F**) implicit volume-controlled mean curvature flow [Eckstein et al. 2007], (**G**) implicit mean curvature flow [Desbrun et al. 1999], (**H**) implicit bi-Laplacian flow via backward Euler, (**I**) implicit Willmore flow based on isometric bending [Wardetzky et al. 2007], and (**J**) our conformal Willmore flow. The per time step cost of our method relative to implicit mean curvature flow was 4.06x, 4.27x, 3.60x, 6.33x, 6.41x, 3.29x, and 4.07x in Figures 1, 2, 5, 12, 13, 18 and 19, respectively, requiring about 0.8 seconds on a mesh of 13k triangles. Overall, however, our method exhibits a substantial net gain in performance, since we can take time steps orders of magnitude larger than traditional methods. A notable exception is (**E**), which rapidly flows to the unit sphere. However, this method produces conformal distortion and sharp geometric features during intermediate steps (Figure 6); moreover, it applies only to surfaces with spherical topology.

7 Conclusion

We have presented a method for curvature flow that exhibits extraordinary stability even on highly degenerate meshes. This type of robustness is especially valuable in computer graphics, where meshes come from many disparate sources (*e.g.*, simulation, artist modeling, *etc.*) which do not often provide guarantees on mesh quality. Our method is also the first to gracefully preserve the appearance of texture throughout the flow. Note that although we have developed this machinery in the context of Willmore flow, it could be applied to other flows of interest by simply changing the initial flow direction $\dot{u}$. A promising avenue for future work is a more thorough investigation of curvature-based filtering, as already hinted at in Section 4. We are also eager to investigate whether this type of approach can be used to help stabilize flows based on higher-order fairness energies.

Figure 18: *Curvature flow on a highly irregular mesh* (left) *with minimum edge length less than 0.2% of mesh diameter. As predicted, our flow remains stable for time steps $\tau < 1$.* Top: *first four steps of stable flow with $\tau = 1 - \epsilon$;* bottom: *unstable flow with $\tau = 1 + \epsilon$, where $\epsilon = 3 \times 10^{-1}$. Notice that curvature in unstable flow exhibits oscillatory behavior characteristic of forward Euler.*

Acknowledgments This research was supported by a Google PhD Fellowship, the Hausdorff Research Institute for Mathematics, BMBF Research Project GEOMEC, SFB / Transregio 109 "Discretization in Geometry and Dynamics," and the TU München Institute for Advanced Study, funded by the German Excellence Initiative. Meshes provided by the Stanford Computer Graphics Laboratory and the AIM@SHAPE Shape Repository.

References

Blaschke, W., and Thomsen, G. 1929. *Vorlesungnen über Differentialgeometrie III*. Springer, Ch. Invarianten der Kreisgeometrie von Möbius, 46–91.

Bobenko, A., and Schröder, P. 2005. Discrete Willmore Flow. In *Proc. Symp. Geom. Proc.*, 101–110.

Bohle, C., and Pinkall, U. 2013. Conformal Deformations of Immersed Discs in $\mathbb{R}^3$ and Elliptic Boundary Value Problems. *ArXiv e-prints* (Jan.).

Botsch, M., and Kobbelt, L. 2004. A Remeshing Approach to Multiresolution Modeling. In *Proc. Symp. Geom. Proc.*, 185–192.

Bouaziz, S., Deuss, M., Schwartzburg, Y., Weise, T., and Pauly, M. 2012. Shape-Up: Shaping Discrete Geometry with Projections. *Comp. Graph. Forum 31*, 5, 1657–1667.

Brakke, K. 1992. The Surface Evolver. *Experiment. Math. 1*, 2, 141–165.

Canham, P. B. 1970. The Minimum Energy of Bending as a Possible Explanation of the Biconcave Shape of the Human Red Blood Cell. *J. Th. Bio. 26*, 1, 61–81.

Celniker, G., and Gossard, D. 1991. Deformable Curve and Surface Finite-Elements for Free-Form Shape Design. *Comp. Graph. (Proc. of ACM/SIGGRAPH Conf.) 25*, 4, 257–266.

Chen, Y., Davis, T. A., Hager, W. W., and Rajamanickam, S. 2008. CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Trans. Math. Softw. 35*.

Clarenz, U., Diewald, U., Dziuk, G., Rumpf, M., and Rusu, R. 2004. A Finite Element Method for Surface Restoration with Smooth Boundary Conditions. *Comput. Aided Geom. Des. 21*, 5, 427–445.

Colding, T. H., and Minicozzi, II, W. P. 2012. Generic Mean Curvature Flow I; Generic Singularities. *Ann. Math. 175*, 2, 755–833.

Crane, K., Pinkall, U., and Schröder, P. 2011. Spin Transformations of Discrete Surfaces. *ACM Trans. Graph. 30*, 4, 104:1–104:10.

Crane, K., 2012. SpinXForm. http://multires.caltech.edu/~keenan/project_spinxform.html#sourcecode.

Crane, K. 2013. *Conformal Geometry Processing*. PhD thesis, Caltech.

deGoes, F., Goldenstein, S., and Velho, L. 2008. A Simple and Flexible Framework to adapt Dynamic Meshes. *Comp. & Graph. 32*, 2, 141–148.

Desbrun, M., Meyer, M., Schröder, P., and Barr, A. 1999. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In *Proc. ACM/SIGGRAPH Conf.*, 317–324.

Desbrun, M., Kanso, E., and Tong, Y. 2008. Discrete Differential Forms for Computational Modeling. In *Discrete Differential Geometry*, A. I. Bobenko, P. Schröder, J. M. Sullivan, and G. M. Ziegler, Eds., Vol. 38 of *Oberwolfach Seminars*. Birkhäuser Verlag, 287–324.

Eckstein, I., Pons, J.-P., Tong, Y., Kuo, C. J., and Desbrun, M. 2007. Generalized Surface Fows for Mesh Processing. In *Proc. Symp. Geom. Proc.*, 183–192.

Gu, X., Zeng, W., Luo, F., and Yau, S.-T. 2011. Numerical Computation of Surface Conformal Mappings. *Comp. Meth. & Fun. Theory. 11*, 2, 747–787.

Helfrich, W. 1973. Elastic Properties of Lipid Bilayers: Theory and Possible Experiments. *Z. Naturf. C 28*, 11, 693–703.

Kamberov, G., Pedit, F., and Pinkall, U. 1998. Bonnet Pairs and Isothermic Surfaces. *Duke Math. J. 92*, 3, 637–644.

Kamberov, G., Norman, P., Pedit, F., and Pinkall, U. 2002. *Quaternions, Spinors and Surfaces*, Vol. 299 of *Contemp. Math.* AMS.

Kazhdan, M., Solomon, J., and Ben-Chen, M. 2012. Can Mean-Curvature Flow Be Made Non-Singular? *Comp. Graph. Forum 31*, 5, 1745–1754.

Olischläger, N., and Rumpf, M. 2009. Two Step Time Discretization of Willmore Flow. In *Mathematics of Surfaces XIII*, Vol. 5654/2009 of *Lect. N. in Comp. Sc.* Springer, 278–292.

Pan, H., Choi, Y.-K., Liu, Y., Hu, W., Du, Q., Polthier, K., Zhang, C., and Wang, W. 2012. Robust Modeling of Constant Mean Curvature Surfaces. *ACM Trans. Graph. 31*, 4.

Pinkall, U., and Sterling, I. 1987. Willmore Surfaces. *Math. Intell. 9*, 2, 38–43.

Sander, P. V., Snyder, J., Gortler, S. J., and Hoppe, H. 2001. Texture Mapping Progressive Meshes. In *Proc. ACM/SIGGRAPH Conf.*, 409–416.

Schneider, R., and Kobbelt, L. 2001. Geometric Fairing of Irregular Meshes for Free-From Surface Design. *Comput. Aided Geom. Des. 18*, 4, 359–379.

Taubin, G. 1995. A Signal Processing Approach to Fair Surface Design. In *Proc. ACM/SIGGRAPH Conf.*, 351–358.

Wardetzky, M., Bergou, M., Harmon, D., Zorin, D., and Grinspun, E. 2007. Discrete Quadratic Curvature Energies. *Comput. Aided Geom. Des. 24*, 8-9, 499–518.

Welch, W., and Witkin, A. 1994. Free-Form Shape Design Using Triangulated Surfaces. *Comp. Graph. (Proc. of ACM/SIGGRAPH Conf.) 28*, 247–256.

Yoshizawa, S., and Belyaev, A. G. 2002. Fair Triangle Mesh Generation with Discrete Elastica. In *Geo. Mod. & Proc.*, 119–123.

Appendices

For brevity, we omit the length element $|df|$ and area element $|df|^2$ from all integrals. We use JX to denote a quarter turn of X in the counter-clockwise direction, which means that $\star df(X) = df(JX)$ and hence $\langle\langle \alpha, \beta \rangle\rangle = \int_M \star\alpha \wedge \beta$ for any two 1-forms α, β. Finally, V, E, and F denote vertices, edges, and faces, respectively.

A Closure Condition for Curves

The endpoints $f(0)$ and $f(L)$ agree as long as the tangent field T integrates to zero. Note that since T has unit length, both $\dot{T}$ and T' are orthogonal to T. Differentiating in time therefore yields

$$0 = \int_0^L \dot{T} = \int_0^L \langle JT, \dot{T} \rangle JT.$$

Recalling that $f' = T$ and $T' = \kappa N$, we get

$$\begin{aligned} 0 &= J \int_0^L \langle JT, \dot{T} \rangle f' = -J \int_0^L (\cancelto{0}{\langle JT', \dot{T} \rangle} + \langle JT, \dot{T}' \rangle) f \\ &= -J \int_0^L (\langle JT, \dot{\kappa} N \rangle + \cancelto{0}{\langle JT, \kappa \dot{N} \rangle}) f = -J \int_0^L \dot{\kappa} f, \end{aligned}$$

where on the first line we apply partial integration and the fundamental theorem of calculus. Hence, $\langle\langle \dot{\kappa}, f^x \rangle\rangle = \langle\langle \dot{\kappa}, f^y \rangle\rangle = 0$.

B Total Mean Curvature Half Density

Let $f : M \to \operatorname{Im}\mathbb{H}$ be an immersion of a closed surface experiencing a conformal flow. At time $t = 0$, we have $\lambda(0) = 1$ and $\rho(0) = 0$, hence the time derivative of Eq. (5) is simply $D\dot{\lambda} = \dot{\rho}$. Integrating over M and applying Stokes' theorem we get

$$\int_M \dot{\rho} |df|^2 = -\int_M df \wedge d\dot{\lambda} = \int_M d(df\,\dot{\lambda}) = \int_{\partial M = \emptyset} df\,\dot{\lambda} = 0,$$

or in other words, $\langle\langle \dot{\rho}, \mathbb{1} \rangle\rangle = 0$. Integrating the relationship $\tilde{\mu} = \mu + \rho|df|$ over M we see that total curvature is conserved.

C Exactness

A transformation λ is integrable as long as the resulting differential $\beta := \bar{\lambda} df\, \lambda$ is *exact* ($\beta = d\tilde{f}$ for some new immersion $\tilde{f}$). When M is simply-connected, it is sufficient for β to be *closed* ($d\beta = 0$) – this condition leads directly to Eq. (5) (see [Crane et al. 2011, App. B]). When M is not simply connected, we must also ensure that β has no *harmonic component*. Here we seek the corresponding condition on curvature.

Let $\{\omega_i\}$, $i = 1, \ldots, 2g$ be a basis for *real*-valued harmonic 1-forms on a closed surface M, and note that $\mathbb{H}$-linear combinations of these bases span *all* harmonic 1-forms. To preserve integrability, then, we must ensure that $\langle\langle \dot{\beta}, \omega_i \rangle\rangle = 0$ for all i. Differentiating β with respect to time yields $\dot{\beta} = \dot{\bar{\lambda}} df + df\,\dot{\lambda} = 2\operatorname{Im}(df\,\dot{\lambda})$ which means that $\dot{\lambda}$ must satisfy $\langle\langle \operatorname{Im}(df\,\dot{\lambda}), \omega_i \rangle\rangle = 0$, or equivalently $\operatorname{Im} \int_M \star\omega_i \wedge df\,\dot{\lambda} = 0$ for all i. Evaluating the integrand yields

$$\begin{aligned} \star\omega_i \wedge df(X, JX) &= \omega_i(JX) df(JX) + \omega_i(X) df(X) \\ &= df(\underbrace{\omega_i(X)X + \omega_i(JX)JX}_{=:Y_i}) = df(Y_i) =: v_i, \end{aligned}$$

where Y_i is a harmonic *vector field* on M and v_i is its pushforward to the immersed surface. The condition on $\dot{\lambda}$ is then just $\operatorname{Im}\langle\langle \dot{\lambda}, v_i \rangle\rangle = 0$ for all harmonic bases v_i.

To find the equivalent condition on $\dot{\rho}$, consider the $2g$ solutions Z_i to $DZ_i = v_i$. A quaternion has no imaginary component if its scalar product with all vectors $a \in \mathbb{R}^3$ is zero, hence our condition on $\dot{\lambda}$ becomes

$$\begin{aligned} 0 &= \left\langle a, \int_M \dot{\lambda} v_i \right\rangle = -\int_M \left\langle v_i a, \dot{\lambda} \right\rangle = -\int_M \left\langle DZ_i a, \dot{\lambda} \right\rangle \\ &= -\int_M \left\langle Z_i a, D\dot{\lambda} \right\rangle = -\int_M \langle Z_i a, \dot{\rho} \rangle = \int_M \langle Z_i, a \rangle\, \dot{\rho}. \end{aligned}$$

where we have used the self-adjointness of D and the fact that $D\dot{\lambda} = \dot{\rho}$. In other words, $\dot{\rho}$ must be orthogonal to each of the three imaginary components of each of the functions Z_i:

$$\langle\langle \dot{\rho}, Z_i^x \rangle\rangle = \langle\langle \dot{\rho}, Z_i^y \rangle\rangle = \langle\langle \dot{\rho}, Z_i^z \rangle\rangle = 0.$$

C.1 Implementation

On a mesh, we compute the functions $Z_i \in \mathbb{H}^{|V|}$ as follows:

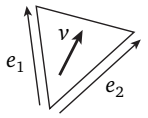
I. Pick $2g$ random discrete 1-forms $\alpha_i \in \mathbb{R}^{|E|}$.

II. Extract harmonic components ω_i via Hodge decomposition.

III. Orthogonalize $\{\omega_i\}$ via the Gram-Schmidt process.

IV. Construct corresponding vector fields $v_i \in \operatorname{Im}\mathbb{H}^{|F|}$.

V. Solve $\mathsf{X} Z_i = v_i$ for each harmonic vector field v_i.



For Hodge decomposition, we use the method of Desbrun *et al.* [2008]. The matrix X is given in Appendix G (here $\rho = 0$). To get the vector fields v_i, we construct appropriate vectors $v \in \operatorname{Im}\mathbb{H}$ in each triangle. Let e_1 and e_2 be two edge vectors (see inset), and let ω_i^1, ω_i^2 be the corresponding 1-form values. Then v must satisfy $\langle v, e_1 \rangle = \omega_i^1$ and $\langle v, e_2 \rangle = \omega_i^2$. If we express v as a linear combination $v = a_1 e_1 + a_2 e_2$, we get a 2×2 system

$$\begin{bmatrix} |e_1|^2 & \langle e_1, e_2 \rangle \\ \langle e_1, e_2 \rangle & |e_2|^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \omega_i^1 \\ \omega_i^2 \end{bmatrix},$$

for the coefficients $a_1, a_2 \in \mathbb{R}$.

D Inversions

We seek an explicit expression for a change in curvature $\dot{\rho}$ corresponding to a sphere inversion. Any inversion can be written as $(f^{-1}-c)^{-1}$, where $c \in \mathbb{R}^3$ is the inversion center. Consider a time-varying inversion with $c(0)=0$ and $\dot{c}=a \in \mathbb{R}^3$. Then

$$\tfrac{d}{dt}(f^{-1}-c)^{-1}|_{t=0} = f\,a\,f,$$

where we have used the identity $\frac{d}{dt}q^{-1} = -q^{-1}\dot{q}q^{-1}$. Hence, $d\dot{f} = df(af) + (fa)df$. But since $df(t) = \bar{\lambda}(t)df(0)\lambda(t)$ and $\lambda(0)=1$, we also have $d\dot{f} = df\,\dot{\lambda} + \dot{\bar{\lambda}}df$, implying that $\dot{\lambda} = af$.

The corresponding change in curvature $\dot{\rho}$ is determined via the relationship $D\dot{\lambda} = \dot{\rho}$. In particular, let $b \in \mathbb{R}^3$ be the component of a orthogonal to N so that $a = b + \langle a, N\rangle N$. Then

$$\begin{aligned} df \wedge b\,df(X,JX) &= df(X)b\,df(JX) - df(JX)b\,df(X) \\ &= df(X)bN\,df(X) - N\,df(X)b\,df(X) \\ &= df(X)(bN+Nb)df(X) = 0, \end{aligned}$$

where we have used the identities $N\,df(X) = df(JX)$, $N\,df(X) = -df(X)N$, and $bN = -Nb$. Hence $D(bf)=0$ and we find that any curvature change corresponding to an inversion has the form

$$\dot{\rho} = D\dot{\lambda} = -\frac{df \wedge \langle a,N\rangle N\,df}{|df|^2} = \frac{2N|df|^2}{|df|^2}\langle a,N\rangle N = -2\langle a,N\rangle$$

for some $a \in \mathbb{R}^3$. To avoid such motions, we therefore require

$$\langle\langle \dot{\rho}, N^x\rangle\rangle = \langle\langle \dot{\rho}, N^y\rangle\rangle = \langle\langle \dot{\rho}, N^z\rangle\rangle = 0$$

where N^x, N^y, N^z are the scalar components of the normal N.

E Möbius Balancing

We seek a sphere inversion that moves our current surface towards a desired distribution of area, up to uniform global scaling. If dA is the desired metric, then we have $|df|^2 = e^{2u}dA$ for some real function u. Since the metric of the transformed surface is $|d\tilde{f}|^2 = |\lambda|^4|df|^2$, the change in scale factor is $\dot{u} = \frac{1}{2}\frac{d}{dt}\log|\lambda|^4 = 2\,\mathrm{Re}(\dot{\lambda})$. For a sphere inversion, $\dot{\lambda} = af$ for some $a \in \mathbb{R}^3$ (Appendix D). Hence, $\dot{u} = -2\langle a, f\rangle$. The optimal direction for a is found by solving

$$\max_{|a|_\mu = 1} \langle\langle \dot{u}, \mathrm{mean}(u) - u\rangle\rangle,$$

where $|\cdot|_\mu$ is the appropriate norm, described below. In other words, we want to move away from our current scale factors u and toward uniform factors $\mathrm{mean}(u)$ as quickly as possible. Letting $u^0 := u - \mathrm{mean}(u)$, the objective can be expressed as

$$\langle\langle \langle a,f\rangle, u^0\rangle\rangle = \int_M \langle a,f\rangle u^0 = \langle a, \underbrace{\textstyle\int_M u^0 f}_{:=v\in\mathbb{R}^3}\rangle = \langle a, v\rangle,$$

where we omit a constant factor 2. Since the direction a determines a change in mean curvature half-density, the value of $|a|_\mu$ is given not by the usual Euclidean norm on $\mathbb{R}^3$ but rather the $\mathscr{L}^2$ norm of the induced function $\dot{\rho} = -2\langle a, N\rangle$. In particular,

$$|a|_\mu^2 := 4\int_M \langle a,N\rangle^2 = \langle Ba, a\rangle,$$

where $B \in \mathbb{R}^{3\times 3}$ equals $B = 4\int_M N \otimes N$. We then want to solve

$$\max_{a\in\mathbb{R}^3} \langle a, v\rangle \quad \text{s.t.} \quad \langle Ba, a\rangle = 1,$$

which amounts to solving $Ba = v$, then normalizing a w.r.t. B.

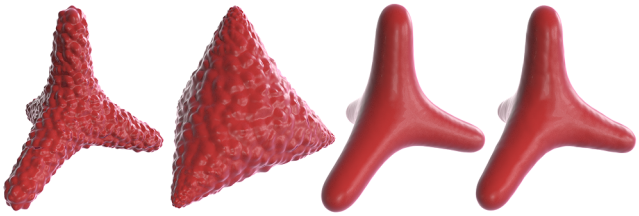


Figure 19: Left: *original surface. Gradient descent with respect to μ scales features uniformly* (center left). *Removing low-frequency components of the flow* (center right) *closely approximates standard (position-based) Willmore flow* (right).

E.1 Implementation

On a mesh with vertex areas $\mathcal{V}_i$, desired vertex areas $\mathcal{V}_i^*$, and vertex normals N_i the balancing direction a is computed as follows:

I. Compute scale factors $u_i \leftarrow \frac{1}{2}\log(\mathcal{V}_i/\mathcal{V}_i^*)$ at each vertex.

II. Remove the mean via $u_i^0 \leftarrow u_i - \sum_{i=1}^{|V|}\mathcal{V}_i u_i / \sum_{i=1}^{|V|}\mathcal{V}_i$.

III. Compute the vector $v \leftarrow \sum_i \mathcal{V}_i f_i u_i^0$.

IV. Solve $Ba = v$ where $B = 4\sum_i \mathcal{V}_i N_i N_i^T$.

V. Normalize $a \leftarrow a/\langle Ba, a\rangle^{1/2}$.

F Scaling Flow

Consider an immersed surface $f : M \to \mathbb{R}^3$. In a sufficiently small ball around any point $p \in M$ we can express f via the height function $h_p(q) = \langle N(p), f(q) - f(p)\rangle$. Noting that $N(p)$ and $f(p)$ are constant with respect to q, we have $\Delta h_p = \langle N(p), \Delta f\rangle = 2H\langle N(p), N\rangle$. In particular, when $q = p$ we get $H(p) = \frac{1}{2}(\Delta h_p)(p)$. Now consider an evolution of the surface that looks like $\dot{h}_p = -h_p$ at some point p, *i.e.*, an exponential scaling down of height over the tangent plane. Then differentiating $H(p)$ in time yields $\dot{H}(p) = \frac{1}{2}(\Delta\dot{h}_p)(p) = -\frac{1}{2}(\Delta h_p)(p) = -\frac{1}{2}H(p)$, *i.e.*, the change in H must be proportional to H itself. This evolution is the same as the evolution $\dot{\rho} = -\frac{1}{2}H$, since a small normal deformation $\tilde{f} = f + \epsilon hN$ over a plane is isometric (hence conformal) up to first order. In particular, $dN = 0$ for a plane, hence $\tilde{g}(u,v) = \left\langle d\tilde{f}(u), d\tilde{f}(v)\right\rangle = g(u,v) + \epsilon(dh(u)\langle N, df(v)\rangle + dh(v)\langle N, df(u)\rangle) + O(\epsilon^2)$, but $\langle N, df(u)\rangle = 0$ for all u.

G Facewise Construction of Dirac Operator

To compute the functions λ (Eq. (5)) and Z_i (Appendix C.1), we need to solve linear systems involving the Dirac operator D. By locally applying the discrete Dirac operator (plus the desired potential) followed by its adjoint, one finds that the final matrix $\mathsf{X} \in \mathbb{H}^{|V|\times|V|}$ can be expressed as a sum of matrices $\mathsf{X}_k \in \mathbb{H}^{|V|\times|V|}$ corresponding to each face, with nonzero entries

$$\mathsf{X}_{ij} = -\tfrac{e_i e_j}{4A_k} + \tfrac{1}{6}(\rho_i e_j - \rho_j e_i) + \tfrac{A_k}{9}\rho_i\rho_j$$

for every ordered pair (i,j) of vertices in the kth triangle, where A_k is the triangle area, $\rho_i \in \mathbb{R}$ is the value of ρ at vertex i, and $e_i \in \mathbb{H}$ is the edge vector opposite vertex i in triangle k.