Simple and Effective Variational Optimization of Surface and Volume Triangulations

Xiangmin Jiao[1,⋆], Duo Wang[1], and Hongyuan Zha[2]

[1] Department of Applied Mathematics, Stony Brook University, Stony Brook, NY 11794
[2] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332
jiao@ams.sunysb.edu

Summary. Optimizing surface and volume triangulations is critical for advanced numerical simulations. We present a simple and effective variational approach for optimizing triangulated surface and volume meshes. Our method minimizes the differences between the actual elements and ideal reference elements by minimizing two energy functions based on conformal and isometric mappings. We derive simple, closed-form formulas for the values, gradients, and Hessians of these energy functions, which reveal important connections of our method with some well-known concepts and methods in mesh generation and surface parameterization. We then introduce a simple and efficient iterative algorithm for minimizing the energy functions, including a novel asynchronous step-size control scheme. We demonstrate the effectiveness of our method experimentally and compare it against Laplacian smoothing and other mesh optimization techniques.

Keywords: Mesh optimization; isometric mapping; conformal mapping; inverse mean ratio; variational methods.

1 Introduction

Optimization of mesh quality is a fundamental problem for numerical simulations in science and engineering. It has become increasingly important in recent years, because modern computational applications often involve dynamic interfaces or moving boundaries. These applications require on-line optimization of the meshes to maintain the validity and quality within the simulation codes as the physical domains deform. Therefore, the mesh-optimization algorithms must be *effective* and *robust* without user intervention, and at the same time be *efficient* and *easy to implement*. These requirements pose decidedly nontrivial new challenges. For problems with very large deformation or motion, one may have to change the mesh connectivity by adding or removing points or flipping edges for best robustness. However, it is often desirable to maintain the mesh connectivity as far as possible to minimize numerical errors, such as in the Arbitrary Lagrangian Eulerian (ALE) methods [15]. In this paper, we focus on mesh optimization with fixed connectivity, i.e. the *mesh smoothing* problem.

⋆ Corresponding author.

1.1 Related Work

Mesh smoothing has a vast amount of literature. The most popular methods for optimizing two-dimensional meshes are probably Laplacian smoothing and its variations [6, 13], which move each point to a weighted average of its neighbors. Their popularity is largely due to their simplicity and efficiency, though they are ineffective at concave regions. A simple but more effective approach is the angle-based smoothing in [27], which provides no theoretical guarantee either but often works well in practice. The most effective methods are optimization based, such as those in [21, 24]. However, these methods are much more expensive and difficult to implement than other simpler alternatives. An ideal method should be simple, efficient, and effective, which we strive to achieve in this paper.

Most methods for optimizing triangular meshes are designed for meshes in $\mathbb{R}^2$. They can be applied to surface meshes by first parameterizing the surface locally or globally and then optimize the flattened mesh (see e.g. [12, 19]). To preserve geometry, these methods typically require a smooth or discrete CAD model and associated point-location procedure to project the points, which increase implementation complexity and computational cost. We propose a simpler approach in this paper.

Mesh smoothing of volume or even higher-dimensional meshes is far more complex than smoothing 2-D meshes. Many methods fail to generalize. For example, Laplacian smoothing often produces inverted elements in 3-D, angle-based method [27] no longer applies, and condition-number based optimization [8, 21] becomes increasingly cumbersome. One of more successful methods is the minimization of the inverted mean ratio [10, 23]. A side product of this paper is to derive the inverted mean ratio from a differential geometry point of view, shows its intimate connection with the aspect ratio of tetrahedra [25], and in turn provide a justification and a simpler and more efficient algorithm for minimizing it.

In mesh optimization, there have been some notable endeavors to *unify* different methods and concepts, such as the study in [1] from an algorithmic point of view, the algebraic framework in [19, 20], and the finite-element-based methods in [14]. One of the objectives of this paper is also such a unification, but in the aspect of unifying the optimization methods for different dimensions. A notable benefit of such a unification is to enable simultaneous optimization of surface and volume meshes, which we will demonstrate to be critical for effective mesh optimization.

1.2 Contributions and Overview

In this paper, we strive to meet the needs of mesh smoothing within numerical simulations. We make contributions in both theoretical and practical aspects. Theoretically, our first contribution is to establish a unified framework of variational mesh optimization in arbitrary dimensions (including surfaces and manifolds) based on the theory of isometric mappings in differential geometry. The core of this framework is two energy functions that capture angle and volume preservation, respectively. We show the connection of the angle-preservation energy with the well-known harmonic smoothing in mesh generation (e.g., [17]) and conformal parameterization in computer graphics (e.g., [7]).

Our second theoretical contribution is a set of simple, closed-form formulas for the gradient and Hessian of the energy functions for surface and volume meshes. These formulas provide us direct proofs of the convexity of our energy functions and enable efficient algorithms and simple implementations for solving the minimization problems. In addition, they reveal the equivalence of angle-preserving energy with the inverse mean ratio in Euclidean space and establish an important connection of these energies with the aspect ratios of tetrahedra [25]. These results substantially enhance the fundamental understandings of variational mesh optimization.

In the practical aspect, our main contribution is a new algorithm for mesh optimization. The most notable features of our method include 1) its ability to smooth surface and volume meshes simultaneously for improved effectiveness, and 2) a simple asynchronous step-size control scheme. Other virtues of our algorithm include ease of implementation and parallelization (comparable with Laplacian smoothing), effectiveness (at least as comparable as inverse mean ratio), and high efficiency (significant improvement in computational cost and memory requirements compared to other optimization-based methods). These properties make our algorithm especially well suited for direct implementation within numerical simulation codes.

The remainder of this paper is organized as follows. Section 2 defines the notation and reviews some fundamental concepts for variational mesh optimization. Section 3 describes our formulation for optimizing surface meshes. Section 4 generalizes our method to volume meshes. Section 5 contains some experimental results of our method and comparisons with other methods for optimizing static and dynamic meshes. Section 6 concludes the paper with discussions of future research directions.

2 Unified Variational Framework for Mesh Optimization

In mesh optimization, there are typically desired shapes and/or sizes of the elements in the resulting mesh. Therefore, we envision each element has a corresponding "ideal" reference element of the desired shape and size. We focus on isotropic (instead of anisotropic) meshes. In this setting, the desired shape is typically regular (i.e., equilateral), and the desired size may be determined by a density function over the domain. The elements of an "optimal" mesh should be as "close" to their corresponding ideal elements as possible. To achieve this, we must define the measures for the "closeness" and then optimize them over the mesh. We seek inspirations from the continuous mappings between surfaces (or manifolds in general), which have been well studied in differential geometry.

2.1 Harmonic, Conformal, and Isometric Mappings

Mappings between surfaces (or manifolds in general) are central in differential and Riemannian geometry (see e.g. [4, 5, 22]). Before delving into the details, we first review the intuitive meanings of three types of mappings to motivate our method.

The most well-known type of mapping is probably *harmonic mappings*. Such mappings are "smooth," as they satisfy some elliptic partial differential equation (PDE),

namely the Laplace equation. Harmonic mappings are popular especially for optimizing structured meshes for their simplicity and efficiency [17], but unfortunately less so for unstructured meshes.

Another important type is *conformal mappings*, which preserve angles locally within a infinitesimal neighborhood of every point on the surface. Conformal mapping is a special type of harmonic mapping. The two types are equivalent under some conditions (such as for simply connected surfaces without boundaries and holes).

The former two types are more widely used. However, the most classical type is in fact *isometric mapping*, which is the foundation of intrinsic differential geometry. Such mappings preserve the first fundamental forms, and in turn preserve the length measures along all directions. In other words, an isometric mapping preserves angles and areas/volumes locally at all points, so it is a special type of conformal mapping and also of harmonic mapping.

We develop our method based on conformal and isometric mappings, as they are well suited for controlling the shapes and/or sizes of isotropic meshes. As we will explain, isometric mappings provide a powerful and flexible framework that can supersede conformal mappings in a variational framework.

2.2 Computational Aspects of Conformal and Isometric Mappings

We now give a more detailed and rigorous treatment of the mappings. Consider two m-dimensional manifolds Γ_1 and Γ_2 embedded in $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, where $n_1 \geq m$ and $n_2 \geq m$. Let $\boldsymbol{u}(\boldsymbol{\xi}) : \Omega \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^{n_1}$ and $\boldsymbol{x}(\boldsymbol{\xi}) : \Omega \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^{n_2}$ denote the corresponding coordinate functions of two parametric patches on Γ_1 and Γ_2, where $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_m]^T$, $\boldsymbol{u} = [u_1, \ldots, u_{n_1}]^T$ and $\boldsymbol{x} = [x_1, \ldots, x_{n_2}]^T$. Let $\boldsymbol{J}_1(\boldsymbol{\xi})$ and $\boldsymbol{J}_2(\boldsymbol{\xi})$ denote the Jacobian matrices of $\boldsymbol{u}(\boldsymbol{\xi})$ and $\boldsymbol{x}(\boldsymbol{\xi})$, respectively. For brevity, we will omit the argument $\boldsymbol{\xi}$ in the following discussions. The *first fundamental tensors* of Γ_1 and Γ_2 are $\boldsymbol{G}_1 = \boldsymbol{J}_1^T \boldsymbol{J}_1$ and $\boldsymbol{G}_2 = \boldsymbol{J}_2^T \boldsymbol{J}_2$, respectively. Let $\boldsymbol{G}$ denote the matrix $\boldsymbol{G}_2 \boldsymbol{G}_1^{-1}$, which is in general nonsymmetric. Let $\boldsymbol{F}$ denote the mapping from $\boldsymbol{u}(\boldsymbol{\xi})$ to $\boldsymbol{x}(\boldsymbol{\xi})$ over the local patches. The mapping $\boldsymbol{F}$ is said to be *isometric* if $\boldsymbol{G}_2 = \boldsymbol{G}_1$ or equivalently $\boldsymbol{G} = \boldsymbol{I}$, i.e., if the *first fundamental tensor* is preserved by the mapping. Similarly, the mapping $\boldsymbol{F}$ is *conformal* if $\boldsymbol{G}_2 = c\boldsymbol{G}_1$ or equivalently $\boldsymbol{G} = c\boldsymbol{I}$, where $c > 0$ and in general varies from point to point. These definitions of the mappings are classical but inconvenient for numerical computations. It turns out to be revealing to consider the eigenvalues of $\boldsymbol{G}$.

Lemma 1. *Let $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ denote the first fundamental tensors of local patches on Γ_1 and Γ_2 with parameterization $\boldsymbol{u}(\boldsymbol{\xi}) : \Omega \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^{n_1}$ and $\boldsymbol{x}(\boldsymbol{\xi}) : \Omega \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^{n_2}$, respectively. Let $\boldsymbol{F}$ denote the mapping from $\boldsymbol{u}(\boldsymbol{\xi})$ onto $\boldsymbol{x}(\boldsymbol{\xi})$. (a) The eigenvalues λ_i of $\boldsymbol{G} = \boldsymbol{G}_2 \boldsymbol{G}_1^{-1}$ are all positive. (b) $\boldsymbol{F}$ is conformal iff $\lambda_1 = \cdots = \lambda_m$. (c) $\boldsymbol{F}$ is isometric iff $\lambda_i = 1$ for $i = 1, \ldots, m$.*

Due to length limitations we omit all the proofs in this paper. The presence of eigenvalues in Lemma 1 may appear to be daunting computationally in high dimensions. Note that the sum of the eigenvalues of a matrix is equal to its trace (i.e., $\sum_{i=1}^m \lambda_i = \text{tr}(\boldsymbol{G}_2 \boldsymbol{G}_1^{-1})$), and the product of the eigenvalues is equal to its determinant (i.e.,

$\prod_{i=1}^{m} \lambda_i = |\boldsymbol{G}_2\boldsymbol{G}_1^{-1}|$). We use $|\boldsymbol{A}|$ to denote the determinant of a matrix $\boldsymbol{A}$ in this paper. We then obtain the following lemma.

Lemma 2. *The mapping $\boldsymbol{F}$ is conformal if and only if*

$$\frac{tr(\boldsymbol{G}_2\boldsymbol{G}_1^{-1})}{\sqrt[m]{|\boldsymbol{G}_2|/|\boldsymbol{G}_1|}} = m. \tag{1}$$

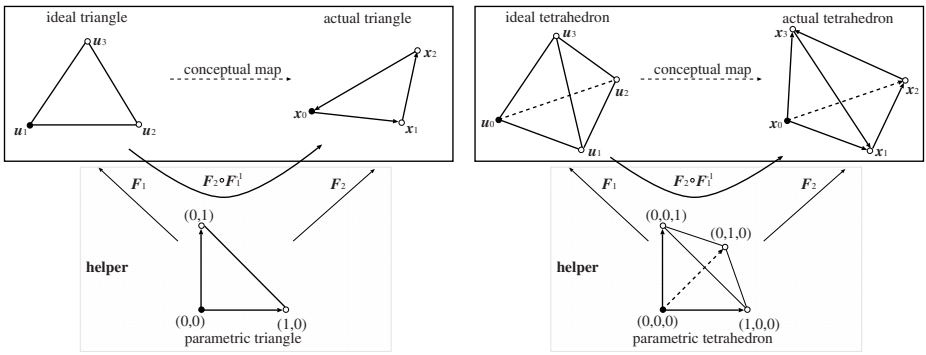
$\boldsymbol{F}$ is isometric if and only if $\boldsymbol{F}$ is conformal and

$$|\boldsymbol{G}_2| = |\boldsymbol{G}_1|. \tag{2}$$

Based on Lemma 2, we can verify whether a mapping is conformal or isometric from $\mathrm{tr}(\boldsymbol{G})$, $|\boldsymbol{G}_2|$, and $|\boldsymbol{G}_1|$, which are easy to compute as we will explain in later sections. Eqs. (1) and (2) are essentially measures for angle- and area-preservation, respectively. This separation of concerns will prove to be very convenient.

2.3 Variational Formulation for Discrete Isometric Mapping

Our preceding discussion focused on continuous surfaces. In meshing or more generally in most numerical computations, the domain is discretized (i.e., triangulated or tessellated), so the surfaces are discrete. Fortunately, variational calculus provides us exactly what we need for generalizing the formulations from continuous to discrete surfaces. The basic idea is to define the mappings over each element, where the formulas for the continuous surfaces apply. We refer to the mapping for each element as the *elemental mapping*, which maps from an *ideal element* to the *actual element* through a *parametric element*. As in the continuous case, we use $\boldsymbol{u}$, $\boldsymbol{x}$, and $\boldsymbol{\xi}$ to denote the coordinates of these elements, respectively. Fig. 1 illustrates the elemental mappings between triangles in 2-D and tetrahedra in 3-D.



(a) (b)

Fig. 1. Schematic of elemental mapping for (a) triangles and (b) tetrahedra

Ideally, we hope all of the elemental mappings are isometric or conformal. Unfortunately, these requirements in general are over-determined and in turn would not yield a solution. We solve the problem in an approximate sense by minimizing a potential energy function (or objective function). This is in spirit similar to solving an over-determined linear system using linear least squares approximations, except that our equations are more complex and nonlinear.

A fundamental question in a variational method is the definition of the energies. For best effectiveness, we require an energy to have the following properties:

Optimality: It must be positive and be minimized for an "ideal" mesh;
Barrier: It should be infinite for degenerate elements with zero area/volume;
Convexity: It should be free of local minimum;
Simplicity: It should be simple and efficient to evaluate and differentiate.

Based on Lemma 2, for each element τ we define two energies E_θ and E_l for angle preservation and area/volume preservation:

$$E_\theta(\tau) = \frac{\mathrm{tr}(\boldsymbol{G}_2\boldsymbol{G}_1^{-1})}{\sqrt[m]{|\boldsymbol{G}_2|/|\boldsymbol{G}_1|}} \tag{3}$$

$$E_l(\tau) = (|\boldsymbol{G}_2|/|\boldsymbol{G}_1|)^p + (|\boldsymbol{G}_1|/|\boldsymbol{G}_2|)^p\,. \tag{4}$$

In E_l, $p > 0$, and the larger p is, the steeper the gradient of E_l becomes. In particular, we choose $p = 0.5$, as we explain later. Per Lemma 2, E_θ is minimized for conformal mappings, and E_θ and E_l are minimized simultaneously for isometric mappings. If the volume of an element is zero, then $|\boldsymbol{G}_2| = 0$ and in turn $E_\theta = E_l = \infty$. Therefore, the first two requirements are satisfied. The convexity and simplicity of these functions are far more involved, which we address in later sections.

We remark that in Euclidean spaces, the *inverse mean ratio* [10, 23] is

$$\frac{\|\boldsymbol{J}_2\boldsymbol{J}_1^{-1}\|_F^2}{m\left(|\boldsymbol{J}_2|/|\boldsymbol{J}_1|\right)^{2/m}}, \tag{5}$$

which is equal to E_θ/m, because $\mathrm{tr}(\boldsymbol{G}_2\boldsymbol{G}_1^{-1}) = \mathrm{tr}(\boldsymbol{J}_1^{-T}\boldsymbol{J}_2^T\boldsymbol{J}_2\boldsymbol{J}_1^{-1}) = \|\boldsymbol{J}_2\boldsymbol{J}_1^{-1}\|_F^2$ and $|\boldsymbol{G}_i| = |\boldsymbol{J}_i|^2$. However, the energy (3) is more general as it is directly applicable to surfaces and manifolds. Given the energy of elemental mappings, we define the total energy over a mesh M as

$$E(\Gamma, \mu) = \sum_{\tau \in M} \left(\mu E_\theta(\tau) + (1-\mu)E_l(\tau)\right), \tag{6}$$

where $\mu \in [0, 1)$ controls the significance of angle versus volume conservation.

Overall, our method computes the energy and their first and second derivatives with respect to the vertex positions, and then moves each vertex to reduce the energy subject to constraints (such as boundary conditions, validity of the mesh, and preservation of sharp features). We repeat this process for a number of iterations or until the mesh quality no longer improves. This framework applies to meshes in any dimensions. It produces approximations to special types of harmonic maps, so it is justified to refer to

our method as *variational mesh optimization* or *mesh smoothing*. However, we have not yet to derive efficient formulas for evaluating and differentiating the energy functions and define simple and effective numerical solutions for solving the resulting nonlinear optimization. We address these issues for surface and volume meshes in the next two sections, respectively.

3 Optimizing Triangular Surface Meshes

We now present our formulation and algorithm for optimizing a triangulated surface mesh embedded in $\mathbb{R}^3$. Obviously, this algorithm would also be applicable to 2-D meshes, but we derive it directly for curved surfaces for generality. As illustrated in Fig. 1(a), we construct a mapping from the ideal to the actual triangle through a parametric triangle. In this specific setting, $\boldsymbol{x}$ has three components, namely $\{x, y, z\}$, while $\boldsymbol{u}$ and $\boldsymbol{\xi}$ has two components, namely $\{u, v\}$ and $\{\xi, \eta\}$, respectively.

3.1 Energy for Angle Preservation

We first derive the formulas for E_θ over a triangle $\tau = \boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_2$. For efficient minimization of the energy over the mesh, we need to evaluate its value and compute its gradient and Hessian with respect to the vertex positions of the triangle. Fortunately, these can be computed efficiently from the angles (or the first fundamental tensor) of the ideal triangle as well as the edge lengths and area of the actual triangle. As illustrated in Fig. 2, let θ_i denote the angle at the ith vertex of the ideal triangle. Let $a = \sqrt{|\boldsymbol{G}_1|}$ and $A = \sqrt{|\boldsymbol{G}_2|}$, i.e., twice of the areas of the ideal and actual triangle, respectively. Let $\boldsymbol{l}_i$ denote the opposite edge of the ith vertex of the actual triangle, $\hat{\boldsymbol{n}}$ is its unit normal in $\mathbb{R}^3$, and $\boldsymbol{l}_i^\perp = \hat{\boldsymbol{n}} \times \boldsymbol{l}_i$, the 90° counter-clockwise rotation of $\boldsymbol{l}_i$. We define the shorthands

$$i+ = \begin{cases} i+1 & i < 3 \\ 1 & i = 3 \end{cases} \text{ and } i- = \begin{cases} i-1 & i > 1 \\ 3 & i = 1 \end{cases}. \tag{7}$$

Given a function $\boldsymbol{f} = [f_1|f_2|f_3]^T$, let $\nabla_{\boldsymbol{x}_1}\boldsymbol{f} = [\nabla_{\boldsymbol{x}_1} f_1|\nabla_{\boldsymbol{x}_1} f_2|\nabla_{\boldsymbol{x}_1} f_3]$. Let $\|\boldsymbol{x}\|$ denote the 2-norm of a vector $\boldsymbol{x}$. We obtain the following theorem.

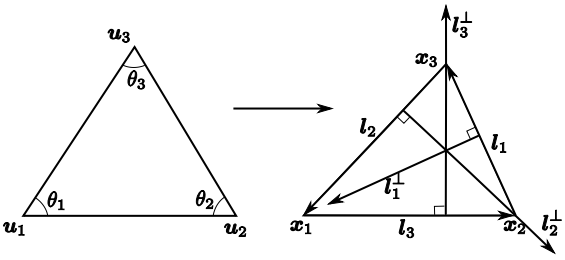


Fig. 2. Naming convention for triangles

Theorem 1. *The energy $E_\theta(\tau)$ in (3) for a triangle $\tau = \boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ in $\mathbb{R}^3$ is*

$$E_\theta = \frac{1}{A}\sum_{i=1}^{3}\omega_i\|\boldsymbol{l}_i\|^2, \tag{8}$$

where $\omega_i = \cot\theta_i$. Its gradient $\nabla_{\boldsymbol{x}_i}E_\theta$ and Hessian $\nabla^2_{\boldsymbol{x}_i}E_\theta$ with respect to the xyz coordinates of vertex $\boldsymbol{x}_i$ for $i = 1, 2, 3$ are

$$\nabla_{\boldsymbol{x}_i}E_\theta = \frac{2}{A}(\omega_{i+}\boldsymbol{l}_{i+} - \omega_{i-}\boldsymbol{l}_{i-}) - \frac{E_\theta}{A}\boldsymbol{l}_i^\perp, \tag{9}$$

$$\nabla^2_{\boldsymbol{x}_i}E_\theta = \frac{2}{A}(\omega_{i+} + \omega_{i-})\,\boldsymbol{I} - \frac{1}{A}\boldsymbol{B}_i - \frac{E_\theta\|\boldsymbol{l}_i\|^2}{A^2}\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T, \tag{10}$$

where $\boldsymbol{I}$ is the 3×3 identity matrix and $\boldsymbol{B}_i = (\nabla_{\boldsymbol{x}_i}E_\theta)\,\boldsymbol{l}_i^{\perp T} + \boldsymbol{l}_i^\perp\,(\nabla_{\boldsymbol{x}_i}E_\theta)^T$.

We note that in $\mathbb{R}^2$, the energy E_θ has re-occurred many times in the literature: It is essentially equivalent to the functional for harmonic mapping in [17], the "MIPS energy" for conformal parameterization in [16], and the condition-number-based metric in [12]. The notation using cotangent is due to Pinkall and Polthier [26], but we generalized it to triangles in $\mathbb{R}^3$. Our formulas for the gradient and Hessian in (9) and (10) appear to be new. In (13), the $\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T$ term has no effect if the surface is flat and is negligible if a vertex moves only nearly tangentially. In surface mesh optimization, all vertices move nearly tangentially, so we can omit this term. Under this assumption, for any reference triangle with acute angles, $-\boldsymbol{B}_i$ is positive definite, so is $\nabla^2_{\boldsymbol{x}_i}E_\theta$, and E_θ is therefore convex. A different analysis in $\mathbb{R}^2$ can be found [23].

For isotropic meshing, the ideal triangle is regular, and $\omega_i = \cot 60^\circ = 1/\sqrt{3}$ for $i = 1, 2, 3$. It is more convenient to use $\tilde{E}_\theta = \sqrt{3}E_\theta$ instead of E_θ for the energy. In this case, the energy is simply the ratio between the sum of the squared edge lengths versus twice of the area, and we then have the following simplified formulas:

$$\tilde{E}_\theta = \frac{1}{A}\sum_{i=1}^{3}\|\boldsymbol{l}_i\|^2, \tag{11}$$

$$\nabla_{\boldsymbol{x}_i}\tilde{E}_\theta = \frac{1}{A}(2\boldsymbol{l}_{i+} - 2\boldsymbol{l}_{i-} - \tilde{E}_\theta\boldsymbol{l}_i^\perp), \tag{12}$$

$$\nabla^2_{\boldsymbol{x}_i}\tilde{E}_\theta \approx \frac{1}{A}\left(4\boldsymbol{I} - \tilde{\boldsymbol{B}}_i\right), \tag{13}$$

where $\tilde{\boldsymbol{B}}_i = \left(\nabla_{\boldsymbol{x}_i}\tilde{E}_\theta\right)\boldsymbol{l}_i^{\perp T} + \boldsymbol{l}_i^\perp\left(\nabla_{\boldsymbol{x}_i}\tilde{E}_\theta\right)^T$.

3.2 Energy for Area Preservation

For triangles, our definition of E_l in (4) reduces to

$$E_l = \frac{A^{2p}}{a^{2p}} + \frac{a^{2p}}{A^{2p}}, \tag{14}$$

where $a = \sqrt{|\boldsymbol{G}_1|}$, $A = \sqrt{|\boldsymbol{G}_2|}$, and $p > 0$. This energy controls the triangle areas. Assume a is independent of $\boldsymbol{x}_i$. It can be shown that the gradient and Hessian of E_l with respect to $\boldsymbol{x}_i$ are

$$\nabla_{\boldsymbol{x}_i} E_l = \frac{2p}{A}\left(\frac{A^{2p}}{a^{2p}} - \frac{a^{2p}}{A^{2p}}\right)\boldsymbol{l}_i^{\perp} = \frac{2p\left(A^{4p} - a^{4p}\right)}{a^{2p}A^{1+2p}}\boldsymbol{l}_i^{\perp}, \tag{15}$$

$$\begin{aligned}\nabla^2_{\boldsymbol{x}_i} E_l &\approx \left(\frac{8p^2A^{4p}}{a^{2p}A^{2+2p}} - \frac{(1+2p)2p\left(A^{4p} - a^{4p}\right)}{a^{2p}A^{2+2p}}\right)\boldsymbol{l}_i^{\perp}\boldsymbol{l}_i^{\perp T} \\ &= \frac{(4p^2-2p)A^{4p} + (4p^2+2p)a^{4p}}{a^{2p}A^{2+2p}}\boldsymbol{l}_i^{\perp}\boldsymbol{l}_i^{\perp T},\end{aligned} \tag{16}$$

where the residual in $\nabla^2_{\boldsymbol{x}_i} E_l$ is a negligible $\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T$ term. $\nabla^2_{\boldsymbol{x}_i} E_l$ is positive semi-definite if $p \geq 0.5$. For $p = 0.5$, the formulas simplify to

$$\nabla_{\boldsymbol{x}_i} E_l = \frac{\left(A^2 - a^2\right)}{aA^2}\boldsymbol{l}_i^{\perp} \text{ and } \nabla^2_{\boldsymbol{x}_i} E_l \approx \frac{2a}{A^3}\boldsymbol{l}_i^{\perp}\boldsymbol{l}_i^{\perp T}. \tag{17}$$

Therefore, we choose $p = 0.5$ in practice. This choice of E_l coincides with the energy used for area distortion for surface parameterization in [3].

3.3 Energy Minimization

The total energy is the sum of the elemental energies over all the triangles. The gradient and Hessian of the total energy with respect to a vertex $\boldsymbol{x}_v$ is equal to the summation of those in the incident triangles of the vertex, i.e.,

$$\nabla_{\boldsymbol{x}_v} E = \sum_{\{\tau | v \in \tau\}} \left(\mu \nabla_{\boldsymbol{x}_v} \tilde{E}_\theta(\tau) + (1-\mu)\nabla_{\boldsymbol{x}_v} E_l(\tau)\right), \tag{18}$$

$$\nabla^2_{\boldsymbol{x}_v} E = \sum_{\{\tau | v \in \tau\}} \left(\mu \nabla^2_{\boldsymbol{x}_v} \tilde{E}_\theta(\tau) + (1-\mu)\nabla^2_{\boldsymbol{x}_v} E_l(\tau)\right). \tag{19}$$

Because $\nabla^2_{\boldsymbol{x}_v} \tilde{E}_\theta(\tau)$ is positive definite and $\nabla^2_{\boldsymbol{x}_i} E_l$ is positive semi-definite, $\nabla^2_{\boldsymbol{x}_v} E$ is positive definite if $0 \leq \mu < 1$ and all the triangles have positive areas. In other words, E is convex locally at each vertex. In practice, we choose $\mu = 0$ for conformal optimization and $\mu = 0.5$ for isometric optimization. After obtaining the gradient and Hessian, one could apply one step of Newton's method to determine a displacement $\boldsymbol{d}_v$ for the vertex, i.e., by solving $\left(\nabla^2_{\boldsymbol{x}_v} E\right)\boldsymbol{d}_v = -\nabla_{\boldsymbol{x}_v} E$. To preserve the geometry, we must constrain the displacement to be nearly tangential. We compute the tangent vectors using an eigenvalue analysis as described in [18]. For a vertex on a smooth surface, let $\hat{\boldsymbol{t}}_1$ and $\hat{\boldsymbol{t}}_2$ denote two orthonormal tangent vectors at v, and let $\boldsymbol{T} = [\hat{\boldsymbol{t}}_1 \mid \hat{\boldsymbol{t}}_2]$. Instead of solving for $\boldsymbol{d}_v$ in $\mathbb{R}^3$, we reformulate the problem to solve for $\boldsymbol{d}_v = \boldsymbol{T}\boldsymbol{u}$, where $\boldsymbol{u} \in \mathbb{R}^2$. The gradient and Hessian of E with respect to $\boldsymbol{u}$ are then $\nabla_{\boldsymbol{u}} E = \boldsymbol{T}^T \nabla_{\boldsymbol{x}_v} E$ and $\nabla^2_{\boldsymbol{u}} E = \boldsymbol{T}^T\left(\nabla^2_{\boldsymbol{x}_v} E\right)\boldsymbol{T}$, respectively. Therefore, Newton's equation for minimizing E with respect to $\boldsymbol{u}$ becomes

$$\left(\boldsymbol{T}^T\left(\nabla^2_{\boldsymbol{x}_v} E\right)\boldsymbol{T}\right)\boldsymbol{u} = -\boldsymbol{T}^T \nabla_{\boldsymbol{x}_v} E,$$

and therefore

$$\boldsymbol{d}_v = -\boldsymbol{T}\left(\boldsymbol{T}^T\left(\nabla^2_{\boldsymbol{x}_v}E\right)\boldsymbol{T}\right)^{-1}\boldsymbol{T}^T\nabla_{\boldsymbol{x}_v}E. \tag{20}$$

Note that $\nabla^2_{\boldsymbol{u}}E$ is symmetric positive definite for any reasonable approximations of $\boldsymbol{T}$. For a point on a ridge curve, we replace $\boldsymbol{T}$ in (20) by the unit tangent vector to the ridge curve. If a point is at a corner, we fix the vertex and set $\boldsymbol{d}_v = \boldsymbol{0}$. This procedure preserves sharp features and preserve smooth surface geometry to second-order accuracy. If higher-order accuracy is required, the displacement $\boldsymbol{d}_v$ can be adjusted by projecting onto a high-order reconstruction of the surface as in [11].

3.4 Overall Algorithm

For the overall algorithm, it is most efficient and convenient to compute the gradient and Hessian of E_θ and E_l over all the triangles, accumulate them to vertices, and then compute the displacements for all the vertices concurrently. This is similar to a typical finite-element code. When applying the displacements, we determine a relaxation factor α_v for each vertex and then add $\alpha_v\boldsymbol{d}_v$ to the vertex, where the relaxation factor is chosen to ensure mesh validity, as we will explain shortly. Algorithm 1 summarizes the core of this variational algorithm, which is essentially a block-Jacobi solver for one step of Newton's method. Our overall algorithm repeatedly invokes this procedure for a desired number of iterations or until the mesh no longer improves.

Algorithm 1. One step of variational smoothing of surface triangulation.

```
Initialize vector grad and matrix H to zero for each vertex;
for each triangle τ do
    for each vertex x_i of τ do
        v ←vertex id of x_i
        grad[v] ← grad[v] + μ∇_{x_i} Ẽ_θ(τ) + (1 − μ)∇_{x_i} E_l(τ);
        H[v] ← H[v] + μ∇²_{x_i} Ẽ_θ(τ) + (1 − μ)∇²_{x_i} E_l(τ);
    end for
end for
for each vertex v do
    let T be composed of tangent vector to surface at v;
    x_v ← x_v − α_v T (T^T (H[v]) T)^{-1} T^T grad[v];
end for
```

In Algorithm 1, we compute the displacements for all the vertices concurrently for better efficiency and ease of implementation. However, concurrent vertex motions may lead to mesh folding if checking is not performed. To address this problem, we introduce a safeguard via an asynchronous step-size control: For each triangle $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$, we solve for the maximum $\alpha \leq 1$ such that the triangle $\boldsymbol{x}_1^{(\alpha)}\boldsymbol{x}_2^{(\alpha)}\boldsymbol{x}_3^{(\alpha)}$ does not fold, where $\boldsymbol{x}_i^{(\alpha)} = \boldsymbol{x}_i + \alpha\boldsymbol{d}_i$. We scale the displacement $\boldsymbol{d}_v$ at v by a factor α_v equal to the minimum α among the incident faces of v. After rescaling the displacement of all the vertices, we recompute α and repeat the rescaling process until $\alpha_v = 1$ for all vertices. Finally, we move all the vertices by the scaled displacement.

Note that this rescaling scheme can be used with any smoothing algorithms to prevent mesh folding. It is particularly effective for our method because the "barrier" property of the energy functions prevents the formation of poor-shaped elements. In practice, the rescaling is rarely invoked except during the first few iterations when optimizing an extremely poor-shaped input mesh.

4 Optimizing Tetrahedral Meshes

We now extend our algorithm to optimize tetrahedral meshes. We reuse the same framework, including the definitions of the energies and the control flow in Algorithm 1. In the elemental mapping, as illustrated in Fig. 1(b), all the coordinates, namely, $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{\xi}$, now have three components, denoted by $\{x, y, z\}$, $\{u, v, w\}$, and $\{\xi, \eta, \zeta\}$, respectively. Our main focus here will be on the efficient evaluation and differentiation of the energies.

4.1 Energy for Angle Preservation

For tetrahedral elements, the angle-preserving energy of the elemental map for a tetrahedron $\tau = \boldsymbol{x}_0\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ in $\mathbb{R}^3$ is

$$E_\theta(\tau) = \frac{\mathrm{tr}(\boldsymbol{G})}{|\boldsymbol{G}_2/\boldsymbol{G}_1|^{1/3}}. \tag{21}$$

As illustrated in Fig. 3, we label the edges as $\boldsymbol{l}_i$ and $\boldsymbol{r}_i$ for $i = 1, 2, 3$, where $\boldsymbol{l}_i = \boldsymbol{x}_i - \boldsymbol{x}_0$ and $\boldsymbol{l}_i = \boldsymbol{x}_{i-} - \boldsymbol{x}_{i+}$, where $i+$ and $i-$ are defined in (7). Let $\boldsymbol{n}_0 = \boldsymbol{r}_2 \times \boldsymbol{r}_1$, the normal to the opposite triangle of $\boldsymbol{x}_0$ pointing toward $\boldsymbol{x}_0$ with magnitude equal to twice of the triangle area. Let

$$\boldsymbol{G}_1 = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \quad \text{and} \quad \boldsymbol{G}_2 = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix},$$

where $g_{ij} = g_{ij}$ and $h_{ij} = h_{ji}$, and let $\boldsymbol{g}_j$ and $\boldsymbol{h}_j$ denote the jth columns of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively. Let $\nu = \sqrt{|\boldsymbol{G}_1|}$ and $V = \sqrt{|\boldsymbol{G}_2|} = -\boldsymbol{n}_0^T\boldsymbol{l}_i$, i.e., six times of the

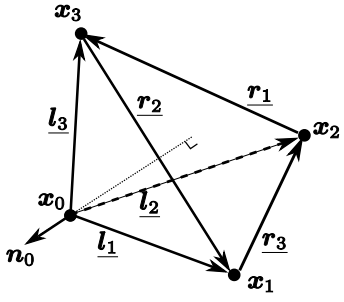


Fig. 3. Naming convention for tetrahedron. Underlined symbols indicate edges.

volumes of the ideal and actual tetrahedra, respectively. Let $\text{sum}(\boldsymbol{a}) = a_1 + a_2 + a_3$ for any $\boldsymbol{a} \in \mathbb{R}^3$. We obtain the following theorem.

Theorem 2. *The energy $E_\theta(\tau)$ for a tetrahedron $\tau = \boldsymbol{x}_0\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ in $\mathbb{R}^3$ is*

$$E_\theta = \frac{1}{V^{2/3}} \sum_{i=1}^{3} \left(\alpha_i \|\boldsymbol{l}_i\|^2 + \beta_i \|\boldsymbol{r}_i\|^2 \right), \tag{22}$$

where $\alpha_i = \text{sum}(\boldsymbol{g}_{i+} \times \boldsymbol{g}_{i-})/\nu^{4/3}$ and $\beta_i = (g_{ii}g_{i-,i+} - g_{i,i-}g_{i,i+})/\nu^{4/3}$. Its gradient $\nabla_{\boldsymbol{x}_0} E_\theta$ and Hessian $\nabla^2_{\boldsymbol{x}_0} E_\theta$ with respect to $\boldsymbol{x}_0$ are

$$\nabla_{\boldsymbol{x}_0} E_\theta = -\frac{2}{V^{2/3}} \left(\sum_{i=1}^{3} \alpha_i \boldsymbol{l}_i \right) - \frac{2}{3V} E_\theta \boldsymbol{n}_0 \tag{23}$$

$$\nabla^2_{\boldsymbol{x}_0} E_\theta = \left(\frac{2}{V^{2/3}} \sum_{i=1}^{3} \alpha_i \right) \boldsymbol{I} - \frac{2}{3V} \boldsymbol{B} - \frac{2}{9V^2} E_\theta \boldsymbol{n}_0 \boldsymbol{n}_0^T, \tag{24}$$

where $\boldsymbol{I}$ denotes the 3×3 identity matrix and $\boldsymbol{B} = (\nabla_{\boldsymbol{x}_0} E_\theta)\, \boldsymbol{n}_0^T + \boldsymbol{n}_0 \left(\nabla_{\boldsymbol{x}_0} E_\theta\right)^T$.

The theorem gives only the derivatives with respect to $\boldsymbol{x}_0$, but it is easy to adapt the formulas for the other vertices by symmetry. To the authors' knowledge, these closed-form formulas are all new. For isometric mesh smoothing where the ideal tetrahedra are regular, $\alpha_i = \beta_i = 2^{-4/3}$, and it is more convenient to omit these constants and use the alternative function

$$\tilde{E}_\theta = \frac{1}{V^{2/3}} \sum_{i=1}^{3} \left(\|\boldsymbol{l}_i\|^2 + \|\boldsymbol{r}_i\|^2 \right), \tag{25}$$

i.e., the ratio between the sum of squared edge lengths versus $V^{2/3}$. The gradient and Hessian of $\tilde{E}_\theta$ with respect to $\boldsymbol{x}_0$ are then

$$\nabla_{\boldsymbol{x}_0} \tilde{E}_\theta = -\frac{2}{V^{2/3}} \sum_{i=1}^{3} \boldsymbol{l}_i - \frac{2}{3V} \tilde{E}_\theta \boldsymbol{n}_0, \tag{26}$$

$$\nabla^2_{\boldsymbol{x}_0} \tilde{E}_\theta = \frac{6}{V^{2/3}} \boldsymbol{I} - \frac{2}{3V} \tilde{\boldsymbol{B}} - \frac{2}{9V^2} \tilde{E}_\theta \boldsymbol{n}_0 \boldsymbol{n}_0^T, \tag{27}$$

where $\tilde{\boldsymbol{B}} = \left(\nabla_{\boldsymbol{x}_0} \tilde{E}_\theta\right) \boldsymbol{n}_0^T + \boldsymbol{n}_0 \left(\nabla_{\boldsymbol{x}_0} \tilde{E}_\theta\right)^T$. By plugging (26) into (27), it is easy to show that $\nabla^2_{\boldsymbol{x}_0} \tilde{E}_\theta$ is positive definite as long as $\tilde{E}_\theta > 0$.

Note that the aspect ratio of a tetrahedron [25] is defined as

$$\rho = \frac{c}{V} \left(\sum_{i=1}^{3} \left(\|\boldsymbol{l}_i\|^2 + \|\boldsymbol{r}_i\|^2 \right) \right)^{3/2},$$

where $c \approx 0.048113$. When a regular tetrahedron is used as the reference element, $\tilde{E}_\theta$ and the inverse mean ratio (5) are both a constant factor of $\rho^{2/3}$. This connection provides another justification for minimizing $\tilde{E}_\theta$ for isotropic meshes.

4.2 Energy for Volume Preservation

For tetrahedra, the energy for volume preservation is

$$E_l = \frac{V^{2p}}{\nu^{2p}} + \frac{\nu^{2p}}{V^{2p}} \tag{28}$$

for $p > 0$. For simplicity, we assume ν is independent of $\boldsymbol{x}_i$. The gradient and Hessian of E_l with respect to $\boldsymbol{x}_0$ are

$$\nabla_{\boldsymbol{x}_0} E_l = \frac{2p}{V}\left(\frac{V^{2p}}{\nu^{2p}} - \frac{\nu^{2p}}{V^{2p}}\right)\boldsymbol{n}_0 = \frac{2p\left(V^{4p} - \nu^{4p}\right)}{\nu^{2p}V^{1+2p}}\boldsymbol{n}_0, \tag{29}$$

$$\nabla^2_{\boldsymbol{x}_0} E_l = \frac{(4p^2 - 2p)V^{4p} + (2p + 4p^2)\nu^{4p}}{\nu^{2p}V^{2+2p}}\boldsymbol{n}_0\boldsymbol{n}_0^T. \tag{30}$$

As for surfaces, $\nabla^2_{\boldsymbol{x}_0} E_l$ is positive semi-definite if $p \geq 0.5$ and all the volumes are positive. We choose $p = 0.5$ in practice, which lead to simplified formulas

$$\nabla_{\boldsymbol{x}_0} E_l = \frac{\left(V^2 - \nu^2\right)}{\nu V^2}\boldsymbol{n}_0 \text{ and } \nabla^2_{\boldsymbol{x}_0} E_l = \frac{2\nu}{V^3}\boldsymbol{n}_0\boldsymbol{n}_0^T. \tag{31}$$

Let E be the total energy defined in (6) for tetrahedral meshes. $\nabla^2_{\boldsymbol{x}_i} E$ is symmetric positive definite for any tetrahedral mesh with positive volume, so E is convex locally at each vertex. To smooth a tetrahedral mesh, it is now a simple matter of plugging in the above formulas for E_θ and E_l into Algorithm 1, except that the triangles would be replaced by the tetrahedra. Regarding to the matrix $\boldsymbol{T}$ in the algorithm, we construct it from the tangent vectors for the vertices on the boundary but replace $\boldsymbol{T}$ by the identity matrix for interior vertices.

5 Experimental Study

In this section, we present some experimental results using our methods for static and dynamic meshes. We report results only for tetrahedral meshes. Because our volume mesh optimization contains integrated surface mesh optimization, these results are also representative for surface meshes. To assess the methods, we measure mesh qualities using the maximum and minimum dihedral angles of tetrahedra. These measures are intuitive and can capture most bad elements, including slivers.

5.1 Optimizing Static Meshes

For static meshes, we used two meshes of a unit cube generated using GAMBIT of Fluent/ANSYS Inc., and also meshes of a hand and a dragon from [2], as shown in Fig. 4. The latter generated the meshes by enhancing some initial meshes through mesh modification and Laplacian smoothing. We optimized the meshes using conformal optimization with fixed boundary, conformal optimization with surface optimization, and

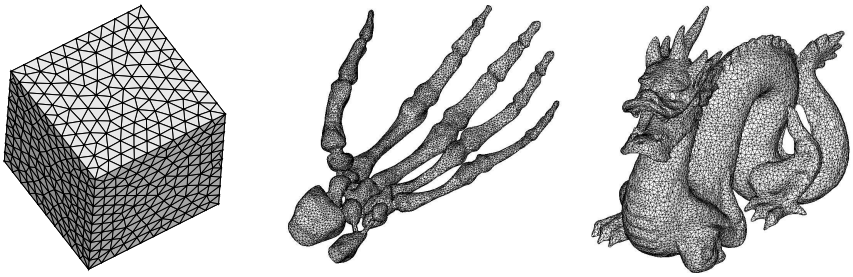


Fig. 4. Sample test tetrahedral meshes

Table 1. Optimization of tetrahedral meshes

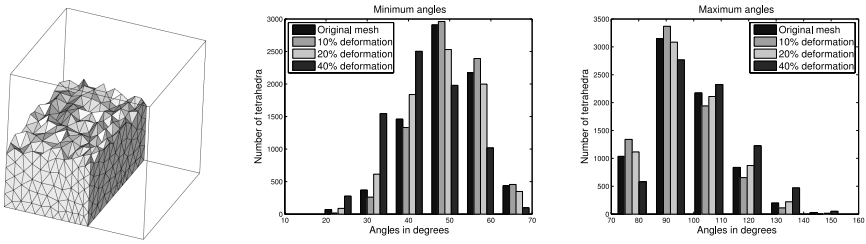
Case	#vertices	#tets	Min and max dihedral angles in degrees			
			Original	Fixed-boundary	Conformal	Isometric
Cube-1K	1,603	7,417	18.9 / 149.3	22.1 / 141.7	22.1 / 141.7	22.4 / 14.1.3
Cube-10K	10,246	53,068	14.4 / 157.0	19.6 / 152.7	19.6 / 152.7	19.9 / 151.9
Hand-10K	3,144	10,000	7.9 / 164.4	7.9 / 163.5	13.7 / 160.9	13.9 / 160.9
Hand-100K	16,649	99,995	3.6 / 174.6	3.6 / 174.6	6.1 / 171.0	6.4 / 170.8
Dragon-5k	1490	4,999	8.0 / 161.2	8.6 / 160.3	12.1 / 160.3	12.1 / 161.1
Dragon-100k	26,436	100,000	3.4 / 174.7	3.4 / 174.7	4.8 / 171.7	4.8 / 171.7

isometric optimization with surface optimization. In isometric optimization, for each tetrahedron we used the average volume of its neighbors as the desired volume.

Table 1 shows the quality measures of the initial and optimized meshes. We observe that the optimization with fixed boundary delivered virtually no improvement for the hand and dragon meshes. This is likely because the worst mesh qualities are constrained by the vertices on the boundary. On the other hand, simultaneous surface and volume mesh optimization improved the dihedral angles in all cases by about 3 to 7.7 degrees. Furthermore, we observe that isometric optimization produced roughly the same results as conformal optimization in terms of the dihedral angles while it provides better control of element sizes and improved the volume ratios by about 50% in our tests.

5.2 Optimizing Moving Meshes

One of the motivating applications of our methods is mesh motion. To demonstrate the effectiveness for such applications, we compressed the smaller mesh of the unit cube progressively up to 40% of its dimension along the x direction. We adapted the time steps to prevent mesh folding and performed up to 10 iterations of mesh smoothing per time step. Before smoothing, the worst minimum dihedral angle was 0.43° and the maximum was 179.3° among all the time steps, which were improved to 18.2° and 154.4° respectively after smoothing. Fig. 5(a) shows the cross-section view of the mesh after 40% compression along with the original unit cube. Fig. 5(b) and (c) show the profiles of the minimum and maximum dihedral angles of the original mesh and



(a) (b) (c)

Fig. 5. Cross section of cube after 40% compression (a) and profiles of dihedral angles (b-c)

after 10%, 20%, and 40% compressions. Our method delivered high quality meshes throughout the simulation. Furthermore, our optimized mesh after 10% compression eevn has a higher quality than the original undeformed mesh.

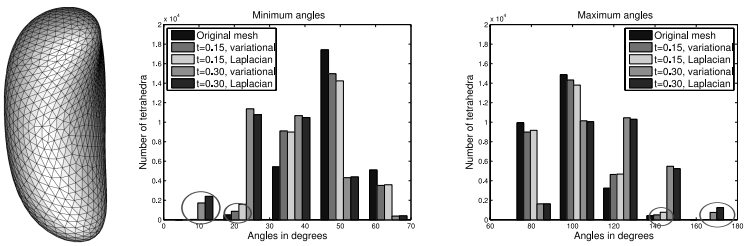
As another example, we deform a sphere of radius 0.15 centered at $(0.5, 0.75, 0.5)$ in a velocity field given by

$$u(x, y, z) = \sin^2(\pi x)(\sin(2\pi z) - \sin(2\pi y)), \tag{32}$$

$$v(x, y, z) = \sin^2(\pi y)(\sin(2\pi x) - \sin(2\pi z)), \tag{33}$$

$$w(x, y, z) = \sin^2(\pi z)(\sin(2\pi y) - \sin(2\pi x)). \tag{34}$$

We move the boundary vertices by integrating the velocity using the fourth-order Runge-Kutta scheme and move the interior vertices by mesh smoothing. Fig. 6(a) shows the deformed shape of the sphere at $t = 0.3$. As in the previous example, we constrained the time steps so that the tetrahedra do not become inverted. We compare our variational smoothing with Laplacian smoothing. Before smoothing, the worst minimum dihedral angle was as small as $0.004°$ and the maximum was as large as $179.987°$ among all the time steps, which were improved to $1.74°$ and $175.66°$ by Laplacian smoothing, respectively. In contrast, our variational smoothing improved the angles to $5.32°$ and $171.08°$, respectively. Fig. 6(b) and (c) show the histograms of the dihedral angles at times $t = 0.15$ and 0.3 along with the original mesh. As we highlighted in the figure,



(a) (b) (c)

Fig. 6. Sphere after moderate deformation (a) and profiles of dihedral angles (b-c)

Laplacian smoothing produced far more poor-shaped elements than variational smoothing, although both methods produced similar distributions for well-shaped elements. In this test, our variational smoothing was about twice slower than Laplacian smoothing. However, it allowed time steps twice as large as those allowed by Laplacian smoothing, so it can result in a net saving in the total computational time in a dynamic simulation, where the physics solvers dominate the overall computational cost.

5.3 Comparison of Computational Cost

For tetrahedral meshes, computational cost of mesh smoothing is often a major concern. The most computational intensive steps are the evaluation and differentiation of the energies. We use the methods in [10], including the block coordinate decent method and inexact Newton method, as the basis of a comparison. The block coordinate decent method requires 480 floating-point operations (flops) to compute the gradient and Hessian of the elemental energy with respect to each vertex, which amounts to $480 \times 4 = 1920$ flops per element. The inexact Newton's method requires 20% extra operations. In contrast, our formulas require 312 flops per element to compute the gradient and Hessian of the elemental energy with respect to all its vertices (including 159 multiplications, 142 additions, and 1 power operation), which are more than six times faster than their counterparts in [10]. In terms of memory requirement, our method requires nine floating-point numbers per vertex, which is nearly an order of magnitude less than what is required by the inexact Newton's method. Besides these reductions in cost, our algorithm is easy to implement, without requiring any sophisticated packages. Therefore, we consider our algorithm as simple, efficient, and effective, and it is promising for direct integration into simulation codes.

6 Conclusions and Discussions

In this paper, we presented a new variational method for optimizing surface and volume meshes. We defined the energy functions based on conformal and isometric mappings between actual elements with ideal reference elements and derived closed-form formulas for their gradient and Hessian. In addition, we developed a simple and efficient algorithm for optimizing the surface and volume meshes by minimizing the energies. We compared our method with some existing methods and demonstrated its effectiveness for optimizing static and moving meshes.

Our method is not without limitations. First, it requires the input mesh to be valid (i.e., with no folding). This is not a severe constraint for numerical simulations where mesh validity is required by other numerical computations. However, it may limit the applicability of our method for other purposes. This limitation can be alleviated by applying a mesh untangling procedure, such as in [9]. For future research directions, we plan to extend our method to optimize quadrilateral meshes and hybrid volume meshes and extend it to anisotropic mesh optimization.

Acknowledgement. This work was supported by National Science Foundation under award number DMS-0809285 and also by a subcontract from the Center for Simulation

of Advanced Rockets of the University of Illinois at Urbana-Champaign funded by the U.S. Department of Energy through the University of California under subcontract B523819. We thank Dr. Eric Shaffer of UIUC for pointers to some test meshes.

References

[1] Amenta, N., Bern, M., Eppstein, D.: Optimal point placement for mesh smoothing. J. Algorithms 30, 302–322 (1999)
[2] Cutler, B., Dorsey, J., McMillan, L.: Simplification and improvement of tetrahedral models for simulation. In: Eurographics / ACM SIGGRAPH Symposium on Geometry Processing, pp. 93–102 (2004), http://people.csail.mit.edu/bmcutler/PROJECTS/SGP04/index.html
[3] Degener, P., Meseth, J., Klein, R.: An adaptable surface parameterization method. In: Proc. 12th Int. Meshing Roundtable, pp. 227–237 (2003)
[4] do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall, Englewood Cliffs (1976)
[5] do Carmo, M.P.: Riemannian Geometry. Birkhäuser, Boston (1992)
[6] Field, D.A.: Laplacian smoothing and Delaunay triangulation. Comm. Appl. Numer. Meth. 4, 709–712 (1988)
[7] Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: Advances in Multiresolution for Geometric Modelling, pp. 157–186. Springer, Heidelberg (2005)
[8] Freitag, L.A., Knupp, P.M.: Tetrahedral mesh improvement via optimization of the element condition number. Int. J. Numer. Meth. Engr. 53, 1377–1391 (2002)
[9] Freitag, L.A., Plassmann, P.: Local optimization-based simplicial mesh untangling and improvement. Int. J. Numer. Meth. Engr. 49, 109–125 (2000)
[10] Freitag Diachin, L., Knupp, P.M., Munson, T., Shontz, S.: A comparison of two optimization methods for mesh quality improvement. Engrg. Comput. 22, 61–74 (2006)
[11] Frey, P., Borouchaki, H.: Geometric surface mesh optimization. Comput. Visual. Sci, 113–121 (1998)
[12] Garimella, R.V., Shashkov, M.J., Knupp, P.M.: Triangular and quadrilateral surface mesh quality optimization using local parametrization. Comput. Meth. Appl. Mech. Engrg. 193, 913–928 (2004)
[13] Hansbo, P.: Generalized Laplacian smoothing of unstructured grids. Comm. Numer. Meth. Engrg. 11, 455–464 (1995)
[14] Hansen, G.A., Douglass, R.W., Zardecki, A.: Mesh Enhancement. Imperial College Press (2005)
[15] Hirt, C.W., Amsden, A.A., Cook, J.L.: An arbitrary Lagrangian-Eulerian computing method for all flow speeds. J. Comput. Phys. 14, 227–253 (1974); Reprinted in 135, 203–216 (1997)
[16] Hormann, K., Greiner, G.: MIPS: An efficient global parametrization method. In: Laurent, P.-J., Sablonniere, P., Schumaker, L.L. (eds.) Curve and Surface Design: Saint-Malo 1999, pp. 153–162 (2000)
[17] Ivanenko, S.A.: Harmonic mappings. In: Thompson, J.F., Soni, B.K., Weatherill, N.P. (eds.) Handbook of Grid Generation. CRC, Boca Raton (1999)
[18] Jiao, X.: Face offsetting: A unified approach for explicit moving interfaces. J. Comput. Phys. 220, 612–625 (2007)
[19] Knupp, P.M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part i: a framework for surface mesh optimization. Int. J. Numer. Meth. Engr. 48, 401–420 (2000)
[20] Knupp, P.M.: Algebraic mesh quality metrics. SIAM J. Sci. Comput. 23, 193 (2001)

[21] Knupp, P.M., Robidoux, N.: A framework for variational grid generation: conditioning the Jacobian matrix with matrix norms. SIAM J. Sci. Comput. 21, 20–29 (2000)
[22] Kreyszig, E.: Differential Geometry. Dover (1991)
[23] Munson, T.: Mesh shape-quality optimization using the inverse mean-ratio metric. Math. Program. Ser. A 110, 561–590 (2007)
[24] Parthasarathy, V.N., Kodiyalam, S.: A constrained optimization approach to finite element mesh smoothing. Finite Elem. Anal. Des. 9, 309–320 (1991)
[25] Parthasarathy, V.N., Graichen, C.M., Af, A.F.H.: A comparison of tetrahedron quality measures. Finite Elem. Anal. Des. 15, 255–261 (1993)
[26] Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Exper. Math. 2, 15–36 (1993)
[27] Zhou, T., Shimada, K.: An angle-based approach to two-dimensional mesh smoothing. In: Proc. 9th Int. Meshing Roundtable, pp. 373–384 (2000)