Surface Mesh Optimization, Adaption, and Untangling with High-Order Accuracy

Bryan Clark, Navamita Ray, and Xiangmin Jiao*

Department of Applied Mathematics and Statistics,
Stony Brook University, Stony Brook, NY 11794, USA
xiangmin.jiao@stonybrook.edu

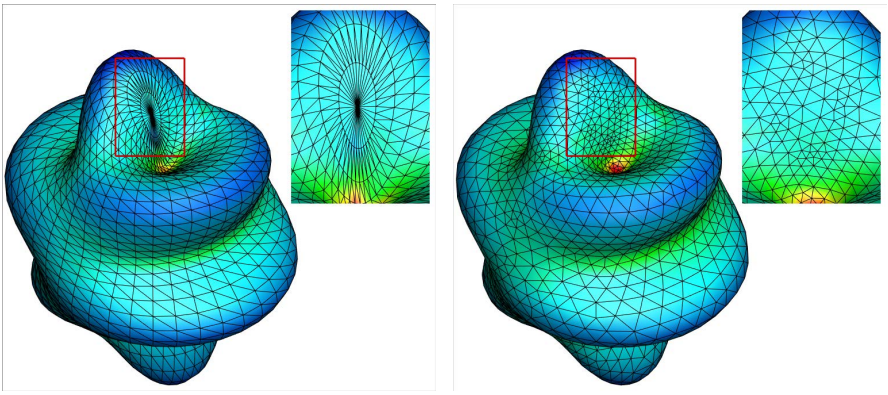
Summary. We investigate the problem of optimizing, adapting, and untangling a surface triangulation with high-order accuracy, so that the resulting mesh has sufficient accuracy for high-order numerical methods, such as finite element methods with quadratic or cubic elements or generalized finite difference methods. We show that low-order remeshing, which may preserve the "shape" of the surface, can undermine the order of accuracy or even cause non-convergence of numerical computations. In addition, most existing methods are incapable of accurately remeshing surface meshes with inverted elements. We describe a remeshing strategy that can produce high-quality triangular meshes, while untangling mildly folded triangles *and* preserving the geometry to high-order accuracy. Our approach extends our earlier work on high-order surface reconstruction and mesh optimization. We present the theoretical framework of our methods, experimental comparisons against other methods, and demonstrate its utilization in accurate solutions for geometric partial differential equations on triangulated surfaces.

Keywords: mesh generation, mesh adaption, curves and surfaces, accuracy and stability, high-order methods.

1 Introduction

Surface meshing and remeshing are important subjects in geometry processing, mesh generation, and numerical solutions of partial differential equations with complex geometry. In the past, researchers have mostly focused on generating piecewise linear (or bilinear) meshes (such as [5]), which have only up to second-order accuracy. In recent years, high-order (third- or higher-order) surface meshing and remeshing (using piecewise quadratic or higher-degree elements or polynomial patches) have become increasingly important. The fundamental reason behind this trend is simple: high-order numerical methods require the same or higher order accuracy for the geometry. With only piecewise linear approximations, the discretizations of differential quantities (such as normals or curvatures) or the solutions of the

* Corresponding author.



(a) Initial mesh with high-valence vertices. (b) Optimized mesh with high accuracy.

Fig. 1. An example of remeshing a spherical harmonic surface to produce a high-quality mesh while preserving geometric accuracy. Images are color coded by mean curvatures. The initial mesh (left) has very high-valence vertices (as highlighted in the inlet, at the poles of the spherical coordinate system) and a minimum angle of 3.2 degrees. The optimized mesh (right) has a minimum angle of 29.9 degrees, with accurate surface normals and curvatures.

differential equations may not achieve the desired order of accuracy, and sometimes may not converge at all. The problem is even more demanding in the problems with evolving geometry, where the surface mesh must be adapted, and sometimes needs to be untangled to remove mildly (or nearly) folded triangles.[1]

In this paper, we investigate the problem of optimizing, adapting, and untangling a surface mesh with high-order accurate nodal positions,[2] so that the new mesh can preserve the order of accuracy of numerical methods, such as finite element methods with quadratic or cubic elements and generalized finite difference methods [1, 13]. We perform all these operations with only the mesh as input (i.e., without accessing a CAD model). Fig. 1 shows an example of optimizing a smooth surface while maintaining the number of vertices. Our procedure not only improves the mesh quality (in this case, eliminating the high-valence[3] vertices and improving the

[1] On a smooth surface mesh, we consider a triangle to be *folded* (and sometimes said to be *inverted*) if its normal direction is more than 90 degrees off from its neighborhood. In a local parameterization, such triangles tend to have a negative Jacobian, which could undermine the accuracy and stability of numerical computations. We consider a folding is "mild" if the area of folded triangle is small compared to the area of its one-ring neighborhood.

[2] By high-order accuracy, we mean that the errors in the vertex position converge at a rate of third or higher order with respect to some measure of edge lengths under mesh refinement. Note that high-order accuracy does not imply high degree of continuity, and conversely high degree of continuity does not imply high-order accuracy either.

[3] The valence of a vertex is the number of its incident edges, which is equal to the number of incident triangles for a closed surface mesh.

minimum angle from 3.2 degrees to 29.9 degrees), but also preserves the geometry to third-order accuracy, so that both surface normals and curvatures can converge after remeshing.

This work is based on, and also a major extension of, our previous works on variational mesh optimization [11] and high-order surface reconstructions [10]. The contributions of the work are as follows. First, we combine our mesh optimization technique with mesh flipping to improve mesh quality even further, along with edge splitting and edge contraction to refine or coarsen meshes. Second, we couple our high-order surface reconstruction techniques with mesh optimization and adaption, to preserve the accuracy of geometry to high-order accuracy. Third, we improve our method to allow untangling mildly folded triangles, to improve the accuracy, stability, and robustness of mesh-based geometry processing and numerical computations. Finally, we verify high-order convergence of various numerical computations with our adapted meshes, and perform a quantitative comparison with other alternatives.

The remainder of the paper is organized as follows. Section 2 reviews some background knowledge and related work, including high-order surface reconstruction and generalized finite-difference schemes over unstructured meshes. Section 3 describes our method for improving mesh quality by combining variational mesh optimization with edge flipping, while preserving the numerical accuracy to high order. Section 4 extends our method to adapt the surface mesh and untangle mildly or nearly folded triangles, while preserving the geometry to high-order accuracy. Section 5 presents some numerical experiments with our remeshing framework to demonstrate its effectiveness in improving mesh quality and preserving accuracy, and reports a comparative study with other methods. Section 6 presents an application of our method in accurate numerical solutions of a moving mesh problem. Section 7 concludes the paper with a discussion.

2 Background and Related Work

We start by reviewing some background for high-order surface remeshing. We first motivate the problem by giving a brief overview of two types of numerical methods over unstructured meshes: finite element methods and generalized finite difference methods. We then review techniques for high-order surface reconstruction, which are closely related to generalized finite difference methods.

2.1 High-Order Finite Element Methods

The most well-known numerical methods on unstructured meshes are probably finite element methods. When using isoparametric elements [8], these methods use quadratic or cubic interpolation within each element for both representing the geometry and approximating the solutions. Additional nodes are required within each element to construct these interpolations. For example in two dimensions, a quadratic interpolation requires six points and a cubic interpolation requires ten points, so a six-node triangle (three corners plus three edge mid-points) and a ten-node triangle

(three corners, two points along each edge, and the centroid) are used, respectively. Higher than cubic interpolation is possible but is rarely used in practice.

Generally speaking, a quadratic interpolation *can* approximate a function up to third-order accuracy, and a cubic interpolation *can* approximate a function up to fourth-order accuracy. However, this order of accuracy may not be attainable if any part of the algorithm is low order, including the geometry. In particular, if linear elements are used for approximating the surface, or if the nodal coordinates are only second-order (or third-order) accurate, then the overall numerical method will be limited to at most second-order (or third-order) accuracy, defeating the purpose of using quadratic (or cubic) elements. For this reason, any remeshing of the surface must be at least the same or higher order accurate than the order of the numerical methods to be used. In [10], we described a refinement scheme to generate quadratic or cubic surface meshes for high-order finite element methods from a given surface mesh with piecewise linear elements but high-order accurate nodal positions.

2.2 Generalized Finite Difference Methods

Besides finite-element methods, another class of high-order method is the generalized finite difference methods, which have been gaining popularity in recent years. Unlike the finite element methods, the generalized finite difference methods are based on weighted least-squares approximations rather than interpolation, so they have more flexibility in defining the stencils for numerical differentiation and in constructing local patches for numerical integration, which are the core computations in most numerical discretizations of differential equations.

The generalized finite difference schemes are often used in meshless methods [1, 13]. However, they also apply to unstructured meshes, where the mesh connectivity can be used as an aid for efficient construction of the stencils for computing least squares approximations. For example, Jiao and co-authors have used a generalized finite difference scheme for computing normals and curvatures of triangulated surfaces to high-order accuracy [12, 15]. Given a surface mesh, they constructed the stencils of a center vertex as its k-ring and $k.5$-ring neighborhood, where $k = 1, 2, 3, \ldots$ Let us define the 0-ring of a vertex as the vertex itself. Then, the k-ring vertices are those share an edge with a vertex in the $(k-1)$-ring, and the $k.5$-ring vertices are those share a face with two vertices in the k-ring.[4] Typically, it is advisory to use the $(d+1)/2$-ring for noise-free surfaces or $(d/2+1)$-ring for noisy surfaces, so that the number of points in the stencil are about 1.5 to two times of the number of coefficients for least-squares fittings. These stencils are based on mesh connectivity alone. In addition, each point has an associated weight in the stencil in the weighted least-squares formulation, which can be used to reduce the influence of points that are far away from the center vertex or to filter out vertices that are on the opposite side of a sharp feature. Compared to meshless methods, this approach allows much more efficient construction of stencils and also avoids the

[4] The definition of half-rings for triangular meshes was introduced in [12]. Our definition here is simpler than the original definition in [12], and it is better suited for quadrilateral meshes as the definition in [10].

issues of short circuiting of stencils. We refer readers to [12, 15] for further details of the stencils.

We emphasize that the generalized finite difference methods use the mesh connectivity for only defining the topology of the surface and for constructing the stencils for least squares approximations. This is a major departure from the finite-element methods, which uses the elements of the mesh to define the interpolation of the geometry. In this paper, we focus on remeshing for generalized finite difference methods. Our method produces high-quality, high-order accurate surface meshes, so they can be refined using the procedure in [10] to obtain valid, quadratic or cubic surface meshes for high-order finite element methods.

2.3 High-Order Surface Reconstruction

In order to obtain more accurate approximations to surfaces, we must first perform *high-order surface reconstructions*, i.e., to reconstruct a high-order surface representation from a given surface mesh, which is typically composed of triangles and/or quadrilaterals. This surface reconstruction problem is closely related to the generalized finite-difference schemes. As in the generalized finite difference methods, we consider the mesh only as the definition of the topology of the surface, rather than using its triangles or quadrilaterals as the definition of the geometry. Then at each vertex, a weighted least squares fitting is constructed using the stencil similar to generalized finite-difference schemes. However, one additional requirement of the surface reconstruction is the continuity of the constructed surface. In this paper, we will utilize the *Weighted Averaging of Local Fittings* (*WALF*) scheme [10], which blends the weighted least squares fittings using the finite-element shape functions to achieve C^0 continuity while preserving high-order accuracy. We will give more details about WALF in Section 3.2.

2.4 Other Related Works

Besides WALF, in [10] Jiao and Wang investigated another approach called *continuous moving frames* (CMF). Both WALF and CMF are based on the assumptions that the vertices of the mesh accurately sample the surface, the faces of the mesh correctly specify the topology of the surface, and utilize the numerical techniques of weighted least squares approximations and piecewise polynomial fittings. They apply to surface meshes composed of triangles and/or quadrilaterals, and also to curves (such as ridge curves on a surface). We choose to base our method on WALF because of its simplicity and better efficiency, and because it delivers similar accuracy as CMF up to sixth order. In [10], they also compared WALF and CMF with some other alternatives, including moving least squares [4] and G^1 reconstructions [14]. Those methods did not deliver good accuracy in our tests. Therefore, we base our present work on WALF.

It is worth mentioning of a recent work of the isogeometric analysis [3], which uses smooth basis functions (such as NURBS) used in geometric modeling as basis functions for approximating the solutions. This idea shares many similarities with

isoparametric elements, but the the geometry and the solutions are C^1 continuous, which are particularly desirable for certain classes of problems, such as modeling of thin shells and contact problems. Our focus for this work is on finite element and generalized finite difference methods, so we consider only reconstructing high-order accurate polynomial patches, instead of NURBS for isogeometric analysis.

3 High-Order Surface Mesh Optimization

In this section, we describe our algorithm for improving mesh quality of a triangular surface mesh, while preserving its geometry to high-order accuracy.

3.1 Mesh Quality Improvement

Our approach is based on the variational optimization method described in [11], which minimizes a convex energy defined based on isometry. However, unlike the technique in [11], which maintained the same mesh connectivity, we also modify the mesh connectivity using edge flipping in this paper, so that the energy can be further decreased, and the mesh quality can be further improved.

Variational Mesh Optimization by Vertex Movement

To improve mesh quality, we minimize the total energy defined based on a discrete isometric mapping from an ideal reference triangle to an actual triangle. For simplicity, in this paper we consider the simplest case of isotropic surface meshes by using equilateral triangles as the reference to optimize only the angles. Given a triangle $\tau \equiv \boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ in $\mathbb{R}^3$, let θ_i denote the angle at the ith vertex of the ideal triangle, $\boldsymbol{l}_i$ denote the opposite edge of the ith vertex of the actual triangle, and A denote twice the area of triangle, respectively. The energy for this triangle is

$$E_\theta(\tau) = \frac{\omega}{A}\sum_{i=1}^{3} \|\boldsymbol{l}_i\|^2, \tag{1}$$

where $\omega = 1/\sqrt{3}$. To improve the quality of a mesh M, we minimize the total energy $\sum_{\tau\in M} E_\theta(\tau)$. We achieve this using an iterative procedure similar to the block-Jacobi solver for Newton's method. In particular, we compute the gradient and Hessian of E_θ over all the triangles, and obtain the gradient and Hessian at each vertex by adding their corresponding values at its incident triangles. For surface meshes, we then restrict the gradient and Hessian onto the tangent space, and then compute the displacements for each vertex using one step of Newton's method. To avoid mesh folding, we determine a relaxation factor α_v for each vertex and then add $\alpha_v \boldsymbol{d}_v$ to the vertex. For detailed description of the algorithm, see [11].

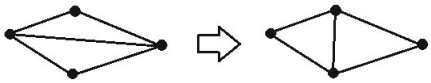


Fig. 2. Illustration of edge flipping

Connectivity Improvement by Edge Flipping

The energy (1) is convex with respect to the position of each vertex [11], and in practice the energy decreases close to the minimum quite rapidly. However, this minimum can be further reduced if the mesh connectivity is allowed to change. We modify the mesh connectivity using edge flipping, as illustrated in Fig. 2. In general, an edge flipping should be performed only if it preserves the topology of the surface (in particular, the new edge must not have already existed in the mesh). In addition, we use the following two additional criteria:

Energy-Reduction Edge Flipping: Flip an edge if it would decrease the sum of the energy of its incident triangles.

Valence-Improvement Edge Flipping: Flip an edge if after flipping the difference between the maximum and minimum valences among the vertices of the two triangles is smaller than that before flipping.

The first strategy is a local strategy, and it is easy to understand. Its goal is simply to reduce the energy for each edge flipping, and therefore the total energy would never increase. The second strategy is non-local, and it may be counter-intuitive, because such a flipping may in fact increase the energy. Its motivation is the following: For a vertex in a mesh for a smooth surface, if a vertex has a too high (> 7) or too low (< 5) valence, all its incident triangles are far from equilateral. Therefore, our strategy is to decrease the gap between the maximum and minimum valences, so that the energy in the whole neighborhood may be reduced. In our experience, repeatedly performing valence-improvement edge flipping tend to produce a mesh without high or low valences, and in turn allows much better mesh quality. In Fig. 1, the mesh was optimized by the combination of these flipping strategies with variational mesh optimization, to eliminate valence-40 vertices and obtain a mesh with valences between 5 and 7.

For each flipping strategy, we repeatedly perform the flipping until no further improvement is possible. Because valence-improvement edge flipping may not directly improve mesh quality, we perform it *before* the variational mesh optimization, and perform energy-reduction edge flipping *after* variational mesh optimization.

3.2 Achieving High-Order Accuracy

In our mesh optimization strategy, edge flipping does not change vertex position. However, in the variational mesh optimization step, the vertices are moved within

the tangent space, which is in general only second-order accurate. To achieve high-order accuracy, we must project the point onto a high-order surface reconstruction. We reconstruct the high-order surface using the *Weighted Averaging of Local Fittings* (*WALF*) scheme [10]. We now briefly describe the technique.

Local Polynomial Fitting

Our high-order reconstruction is based on local polynomial fittings and weighted least squares approximations. We first review local polynomial fittings, also known as *Taylor polynomials* in numerical analysis [7]. We are primarily concerned with surfaces, so the local fitting is basically an interpolation or approximation to a neighborhood of a point P under a local parametrization with parameters u and v, where P corresponds to $u = 0$ and $v = 0$. The polynomial fitting may be defined over the global xyz coordinate system or a local uvw coordinate system. In the former, the neighborhood of the surface is defined by the *coordinate function* $\boldsymbol{f}(u, v) = [x(u, v), y(u, v), z(u, v)]$. In the latter, assuming the uv-plane is approximately parallel with the tangent plane of the surface at P, each point in the neighborhood of the point can be transformed into a point $[u, v, f(u, v)]$ (by a simple translation and rotation), where f is known as the *local height function*.

Let $\boldsymbol{u}$ denote $[u, v]^T$. Let $\varphi(\boldsymbol{u})$ denote a smooth bivariable function, which may be the local height function or the x, y, or z component of the coordinate function for a parametric surface. Let c_{jk} be a shorthand for $\frac{\partial^{j+k}}{\partial u^j \partial v^k}\varphi(\mathbf{0})$. Let d be the desired degree of the polynomial fitting, and it is typically small (typically no greater than 6 for stability reasons). If $\varphi(\boldsymbol{u})$ has $d + 1$ continuous derivatives, it can be approximated to $(d + 1)$st order accuracy about the origin $\boldsymbol{u}_0 = [0, 0]^T$ by

$$\varphi(\boldsymbol{u}) = \underbrace{\sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} c_{jk} \frac{u^j v^k}{j!k!}}_{\text{Taylor polynomial}} + \underbrace{\sum_{j,k\geq 0}^{j+k=d+1} \tilde{c}_{jk} \frac{\tilde{u}^j \tilde{v}^k}{j!k!}}_{\text{remainder}}, \tag{2}$$

where $\tilde{c}_{jk} = \frac{\partial^{j+k}}{\partial u^j \partial v^k}\varphi(\tilde{u}, \tilde{v})$, $0 \leq \tilde{u} \leq u$, and $0 \leq \tilde{v} \leq v$.

Suppose we have a set of data points (u_i, v_i, φ_i) for $i = 1, \ldots, m - 1$, sampled from a neighborhood near P on the surface. Substituting each given point into (2), we obtain an approximate equation

$$\sum_{p=0}^{d} \sum_{j,k\geq 0}^{j+k=p} \left(\frac{u_i^j v_i^k}{j!k!} \right) c_{jk} \approx \varphi_i, \tag{3}$$

which has $n = (d + 1)(d + 2)/2$ unknowns (i.e., c_{jk} for $0 \leq j + k \leq d$, $j \geq 0$ and $k \geq 0$), resulting in an $m \times n$ rectangular linear system. The local least squares polynomial fitting provides us the theoretical foundation for high-order reconstruction of surfaces, established by the following proposition [12]:

Proposition 1. *Given a set of points* $[u_i, v_i, \tilde{f}_i]$ *that interpolate a smooth height function* f *or approximate* f *with an error of* $O(h^{d+1})$*, assume the point distribution and the weighting matrix are independent of the mesh resolution, and the condition number of the linear system is bounded by some constant. The degree-*d *weighted least squares fitting approximates* c_{jk} *to* $O(h^{d-j-k+1})$.

Weighted Averaging of Local Fittings (WALF)

The *Weighted Averaging of Local Fittings* (*WALF*) obtains a continuous approximation to the surface by a weighted averaging of the local fittings at the vertices. For continuity, the weights used by the weighted averaging must be continuous over the mesh. One such a choice is the barycentric coordinates of the vertices over each triangle. Consider a triangle composed of vertices $\boldsymbol{x}_i$, $i = 1, 2, 3$, and any point $\boldsymbol{p}$ in the triangle. For each vertex $\boldsymbol{x}_i$, we obtain a point $\boldsymbol{q}_i$ for $\boldsymbol{p}$ from the local fitting in the local uvw coordinate frame at $\boldsymbol{x}_i$, by projecting $\boldsymbol{p}$ onto the uv-plane. Let ξ_i, $i = 1, 2, 3$ denote the barycentric coordinates of $\boldsymbol{p}$ within the triangle, with $\xi_i \in [0, 1]$ and $\sum_{i=1}^{3} \xi_i = 1$. We define

$$\boldsymbol{q}(\boldsymbol{u}) = \sum_{i=1}^{3} \xi_i \boldsymbol{q}_i(\boldsymbol{u}) \tag{4}$$

as the approximation to point $\boldsymbol{p}$.

WALF constructs a C^0 continuous surface, as can be shown using the properties of finite-element basis functions. Furthermore, under the same assumptions as Proposition 1, we obtain the following property of WALF [10]:

Proposition 2. *Given a mesh whose vertices approximate a smooth surface* Γ *with an error of* $O(h^{d+1})$*, the distance between each point on the WALF reconstructed surface and its closest point on* Γ *is* $O(h^{d+1} + h^6)$.

This proposition gives an upper bound of the error, and it shows that the error term is indeed high order for $d \geq 1$. The $O(h^6)$ term is due to the discrepancy of local coordinate systems at different vertices. However, in most applications we expect $d < 6$, so the total error would be dominated by the degree of polynomials used in the least squares fitting.

4 High-Order Surface Mesh Adaption and Untangling

The method we described in the previous section has two limitations. First, it does not change the number of vertices in the mesh. However, it may be necessary to change the number of vertices for an evolving geometry. Second, it requires the initial surface mesh to have no inverted elements (i.e., no mesh folding), because our energy function is infinite at a degenerate element, which forms a barrier to prevent inverted elements to be untangled. However, it is not uncommon for the input mesh to be mildly or nearly folded. As mentioned before we consider a triangle

to be *inverted* (i.e, *folded*) if its normal direction is more than 90 degrees off from its neighborhood. The folding is considered "mild" if the area of folded triangle is small compared to the area of its one-ring neighborhood. By untangling (or unfolding) we mean resolving these folded triangles so that there are no folded triangles in the resulting mesh. In this section, we further extend our techniques to accommodate mesh adaption (refinement and coarsening) and mesh untangling.

4.1 Mesh Refinement and Coarsening

We perform mesh adaption primarily using two operations: edge splitting and edge contraction. Since edge contraction can revert an edge splitting, we must consider the two operations together to ensure consistency.

Edge Splitting and Edge Contraction

First, let us consider edge splitting, which is the easier of the two. Given two adjacent triangular elements, edge splitting inserts a new vertex on their shared edge. Similar to the approach in [9], we consider two criteria to determine whether an edge requires splitting:

Absolute Longness: The edge is the longest among its incident triangles and is longer than a provided threshold L.

Relative Longness: The edge is longer than a desired edge length l ($< L$), one of its opposite angles is close to π (greater than provided θ_l), and the shortest edge among its incident triangles is no shorter than a provided threshold s ($< l$).

The process of edge splitting abides by this criterion to help optimize element quality and consistency in size. The process of edge splitting occurs in decreasing order of edge lengths throughout the mesh.

For mesh coarsening, we perform edge contraction, which removes a vertex. As in [9], we consider the following four criteria to determine whether an edge should be contracted:

Absolute small angle: the opposite angle in an incident triangle of the edge in question is smaller than a threshold θ_s, and the triangle's longest edge is shorter than a desired edge length l.

Relative shortness: The edge in question is shorter than a fraction r of the longest edge of its incident triangles.

Absolute small triangle: The edge in question is the shortest in its incident triangles and the longest edge of its incident triangles is shorter than a given threshold S.

Relative small triangle: The longest edge in its incident triangles is shorter than a fraction R of the longest edge in the mesh and also shorter than the desired edge length l.

The process of edge contraction abides by this criterion to help optimize element quality and consistency in size. The process of edge contraction occurs in increasing order of edge lengths throughout the mesh. While the first two criteria help to

remove poor shaped triangles, the latter two criteria help to remove triangles that are too small, preserving the overall mesh quality as it evolves. When contracting an edge, its two incident vertices merge at a new location. To prevent mesh folding, we reject any contractions that would lead to topological changes or an inversion of normals on any triangle. Contracting the shortest edges first helps to avoid the need for such rejections. These parameters and criteria abide by Jiao et al. [9].

Preserving High-Order Accuracy

Just as mesh optimization, the mesh adaptation operations need to preserve the accuracy of the geometry. For edge splitting we first insert a new point onto an edge, and then we project the point onto a high-order reconstruction based on WALF. For edge contraction, we first replace the two vertices by a new point on the edge and then we project the point onto a high-order reconstruction based on WALF.

Uniform Mesh Refinement

Besides using edge splitting, we can also uniformly subdivide the triangles into four or ten triangles by inserting additional nodes to the edges and face centers. This strategy is particularly useful for generating quadratic and cubic elements for high-order finite element methods. We have explored this technique in [10], and readers are referred to it for further detail.

4.2 Mesh Untangling by Smoothing and Edge Flipping

For robustness, we must handle mildly folded meshes. For clarity, we emphasize that our untangling process attempts to resolve folded triangles of a smooth surface in a local fashion, and it does not attempt to resolve global self-intersections of a surface.

We resolve mesh folding using a combination of weighted Laplacian smoothing and edge flipping. In the weighted Laplacian smoothing, we move each vertex toward a weighted average of the centroids of its incident triangles, where the weight for each centroid is equal to distance from the vertex to the centroid. This is equivalent to a weighted averaging of neighboring vertices, where the weight for each vertex is equal to one third of the sum of the distances from the vertex to the centroids of adjacent triangles. This weighted Laplacian smoothing does not produce as good mesh quality as variational optimization. However, because it tends to move a vertex to a weighted average of its one-ring neighborhood, repeatedly applying the procedure tend to avoid mesh folding and even unfold a mesh. (This is the reason why we refer to this step as mesh smoothing, instead of mesh optimization.) In addition, we found the weighting based on lengths is more effective to untangle a mesh than using unweighted Laplacian smoothing.

Similar to variational optimization, when coupling the weighted Laplacian smoothing with edge flipping. We also use two heuristics for edge flipping. The first is valence improvement edge flipping, as we described above. The second criterion is based on the angles:

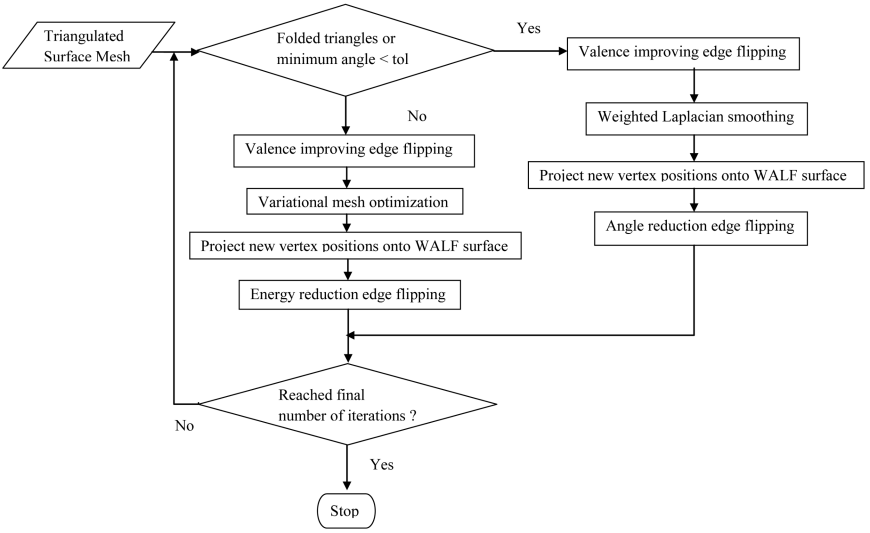


Fig. 3. Overview of the major steps in mesh optimization and untangling with high-order surface reconstruction

Angle-Reduction Edge Flipping: Flip an edge if the sum of opposite angles is smaller after flipping.

This criterion is equivalent to the Delaunay edge flipping condition in 2-D, so it can help improve angles. Furthermore, this operation also tends to flip an edge of a folded triangle. We perform angle-reduction edge flipping after mesh smoothing. Similar to variational optimization, we perform valence improvement edge flipping before mesh smoothing, and perform angle-reduction edge flipping after mesh smoothing. In addition, the vertices are projected onto high-order surface reconstruction after mesh smoothing, to preserve the high-order accuracy. Note that during the algorithm, we do not need to check explicitly whether individual triangles are folded. When simply repeatedly operations until the mesh is far from folding (namely, with 5 degrees). Figure 3 shows the major steps involved in mesh optimization and untangling with high-order surface reconstruction. We omit mesh refinement and coarsening in the diagram for simplicity, as they can be performed in a separate step.

5 Numerical Experiments and Comparative Study

In this section, we demonstrate the accuracy of our proposed remeshing strategy, and compare it with some other methods in the literature. Because the point positioning for mesh smoothing and mesh adaption tend to use very different strategies in the literature, we compare them separately. Our comparisons focus on the accuracy and convergence of the results, because accuracy and convergence are often the ultimate

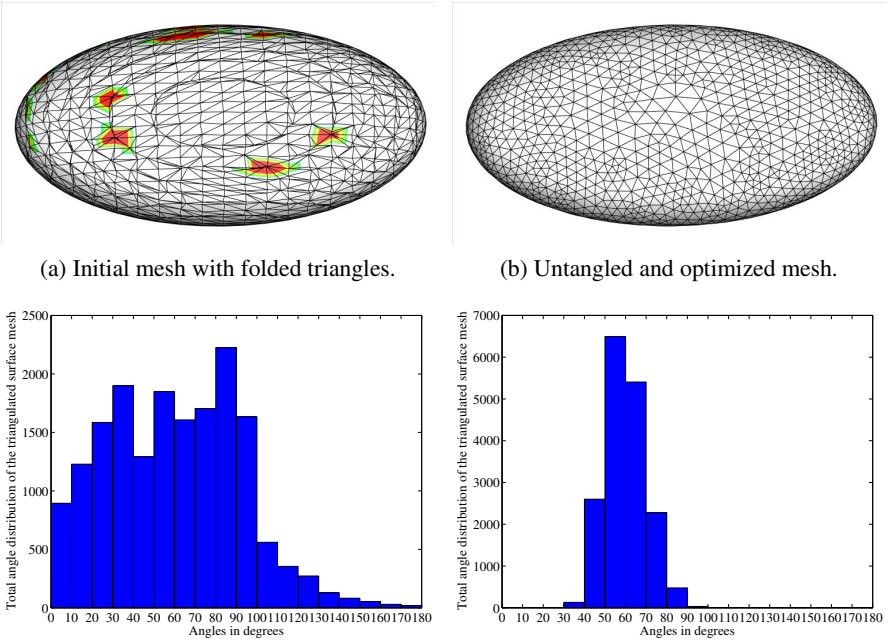
goals of numerical computations on meshes. For this reason, we use the same mesh smoothing and mesh adaption techniques for all methods, and only change the way how points are positioned.

5.1 Effectiveness of Mesh Optimization

We first report some results of our optimizing technique. An example is given in Fig. 1. In this case, the number of vertices is 1562, and the number of triangles is 3120, both before and after mesh optimization. The initial quadrilateral mesh was generated from subdividing the quadrilaterals in a 40×40 logically rectangular grid in the $\theta\phi$-domain of spherical coordinate system, which result in high-valence vertices at the poles. Before optimization, the minimum and maximum angles were 3.2 and 140.6 degrees, respectively. After repeatedly performing variational mesh optimization and edge flipping, the minimum and maximum angles were 29.9 and 109.1 degrees, respectively. We used quadratic fittings in the remeshing, so that the resulting points are third-order accurate. When the normal and curvature computation algorithms in [12] are used, the normal and curvature would be second- and first-order accurate, respectively. Because of the high mesh quality and high accuracy, the resulting mesh can be used for third-order generalized finite difference methods, and also can be subdivided to generate quadratic elements for third-order finite elements methods without producing any negative Jacobian. For even higher-order methods, we can simply replace the quadratic fitting by cubic or higher order fittings in our point-projection procedure.

5.2 Effectiveness of Mesh Untangling

One strength of our proposed methodology is the ability to untangle mildly folded triangles. In the context of high-order mesh generation, we observe that occasionally, even the small perturbation of projecting vertices onto the high-order surface reconstruction can cause a few very poor-quality triangles (which may be present in the initial mesh) to fold. To demonstrate the effectiveness of mesh untangling, we construct a much more severe case: We start with a very poor-quality triangular mesh for an ellipsoid mesh generated using marching cubes, randomly perturb the vertices by up to the length of the background grid, and then project the perturbed points back onto the ellipsoid. The initial mesh had 47 folded triangles, a few of which were high-lighted in the left image in Fig. 4. The combination of poor mesh quality, anisotropy of the geometry, and folded triangles makes the problem difficult to handle by ad hoc techniques. After only three iterations, our method untangled the mesh and produced a mesh with minimum angles of 18 degrees. After a few more iterations, the mesh converged to a high-quality triangulation with a minimum angle of 37.6 degrees and a maximum angle of 98.1 degrees, as shown in the right image in Fig. 4. In addition, the surface normal and curvatures of the resulting mesh are still accurate.



(a) Initial mesh with folded triangles.

(b) Untangled and optimized mesh.

(c) Angles of initial mesh.

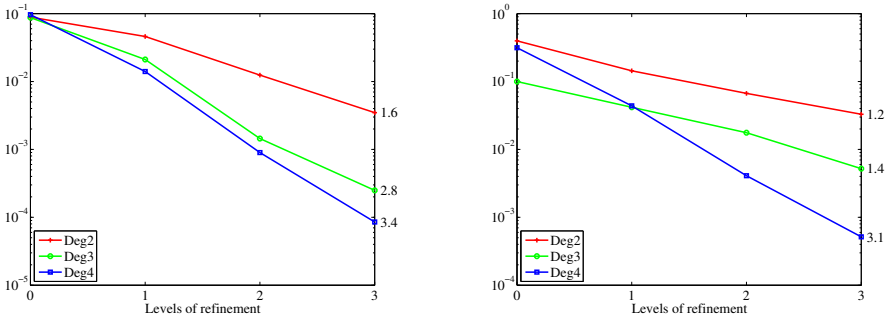
(d) Angles of optimized mesh.

Fig. 4. Example of untangling and optimization of a poor-quality surface mesh. The initial mesh had 47 folded triangles (a few are highlighted in (a)) and a wide range of angles. After optimization, the mesh had excellent mesh quality, accurate curvatures, and no folding.

5.3 Numerical Accuracy with Mesh Optimization

We now report the orders of convergence for computing differential quantities (normal and mean curvature) of an optimized mesh. We use a torus (with inner radius 0.7 and outer radius 1.3) and an ellipsoid (with semi-axes 1, 2, and 3) as test geometries. For each geometry, we first generated a set of poor-quality meshes using marching cubes. These meshes were then optimized using our variational optimization, smoothing and edge flipping techniques. Finally, the differential quantities were computed at each vertex of the optimized meshes. For mesh convergence study, we generated four meshes for each set of our test meshes, and numbered these meshes from the coarsest (mesh 1) to the finest (mesh 4). The average edge lengths are approximately halved between adjacent mesh resolutions. Let v be the total number of vertices. Let $\tilde{\boldsymbol{n}}_i$ and $\hat{\boldsymbol{n}}_i$ denote the computed and exact unit normal at the ith vertex, and $\tilde{k}_i$ and k_i denote the curvatures at the ith vertex, respectively. We estimate the L_∞ errors in normals and (mean or Gaussian) curvatures as

$$\text{error in normal} = \max_i \|\tilde{\boldsymbol{n}}_i - \hat{\boldsymbol{n}}_i\|, \qquad \text{error in curvature} = \max_i |\tilde{k}_i - k_i|,$$



(a) L_∞ errors in normals. (b) L_∞ errors in mean curvatures.

Fig. 5. Errors and orders of convergence of normals and curvatures after mesh optimization for an ellipsoid

and compute the average convergence rate as

$$\text{average convergence rate} = \frac{1}{3}\log_2\left(\frac{\text{error of mesh 1}}{\text{error of mesh 4}}\right).$$

In Fig. 5, the horizontal axis corresponds to the level of mesh refinement, and the vertical axis corresponds to the L_∞ errors in logarithmic scale. In the legends, the "degree" indicates the degree of polynomial fittings used for point projection during mesh optimization. We show the average convergence rates along the right of the plots for each curve.

Theoretically, the order of convergence of normal and mean curvature (which are first and second order differential quantities) should be d and $d-1$, respectively for WALF reconstructed surface mesh using degree d polynomial fittings. Fig. 5 shows that the optimized meshes preserved the accuracy of the points and as a result achieved the theoretical orders of convergence for the differential quantities.

5.4 Accuracy Comparison of Mesh Adaption

For mesh adaptation, a common approach is to keep the original mesh during mesh smoothing/adaption, and project new vertices onto the faceted, piecewise linear geometries (see e.g., [6]). Such an approach has only second order accuracy. Another approach taken by Frey [5] was to construct a G^1 continuous surface using Walton's method [14], but our experiments have shown that Walton's method is at most second order accurate despite its G^1 continuity. We compare between different point projection methods for mesh adaption, namely,

1. Linear : Points are projected onto the linear triangle,
2. Walton's method : Points are projects onto a G^1 quartic patch [14], and
3. WALF : Points are projected onto WALF reconstructed surfaces with degrees 2, 3, and 4.

We use torus as a test geometry and compute the mean and Gaussian curvatures of the adapted mesh using the mentioned point projection methods. Table 1 compares the L_∞ errors of mean curvatures and Gaussian curvatures for the adapted mesh of a torus. It is evident that the mean curvatures did not converge in L_∞ error for linear reconstruction, and barely converged for Walton's method, whereas it converged to high order for WALF. The numbers of vertices of the three meshes are 544, 1896, and 7528, respectively.

6 Example Usage in Geometric PDEs

In the previous section, we verified our remeshing techniques through the computation of differential quantities. We now describe a usage of our techniques in the numerical solutions of geometric partial differential equations. Such problems in various applications, such as surface smoothing in computer-aided design [16] and the modeling of moving surfaces of materials [2]. As an example, we consider the solution of the mean-curvature flow over triangulated surfaces. The continuum formulations of these problems are as follows. Given a moving surface Γ, the coordinates $\boldsymbol{x}$ of points on Γ are functions of time t as well as some surface parametrization $\boldsymbol{u} = (u, v)$, which can be local instead of global parametrizations. Assume the surface is differentiable. The *mean-curvature flow* is a second-order nonlinear PDE modeling the motion of the surface driven by the mean curvature, given by

$$\frac{\partial \boldsymbol{x}}{\partial t} = M\hat{\boldsymbol{n}}, \tag{5}$$

where M denotes the mean curvature and $\hat{\boldsymbol{n}}$ denotes the unit normal vector. The vector $\hat{\boldsymbol{n}}$ involves first-order partial derivatives of $\boldsymbol{x}$ with respect to the parameters $\boldsymbol{u}$, whereas M involves second-order partial derivatives of $\boldsymbol{x}$ with respect to $\boldsymbol{u}$.

Table 1. L_∞ errors and orders of convergence of mean curvature and Gaussian curvature on a torus after mesh adaption using surface reconstructed based on WALF and other alternatives

Method	# of vertices	mean curvature			Gaussian curvature		
		Mesh 1	Mesh 2	Mesh 3	Mesh 1	Mesh 2	Mesh 3
Linear	L_∞ error	0.6	0.35	0.83	2.29	1.64	3.09
	Order	n/a	0.77	-1.2	n/a	0.48	-0.91
Walton's	L_∞ error	0.6	0.29	0.21	2.27	1.40	0.96
	Order	n/a	1	0.44	n/a	0.7	0.55
WALF deg 2	L_∞ error	0.6	0.27	0.11	2.29	1.36	0.52
	Order	n/a	1.1	1.3	n/a	0.74	1.4
WALF deg 3	L_∞ error	0.54	0.20	0.066	1.73	0.86	0.30
	Order	n/a	1.4	1.6	n/a	1	1.5
WALF deg 4	L_∞ error	0.45	0.085	0.0074	1.24	0.28	0.03
	Order	n/a	2.4	3.5	n/a	2.2	3.3

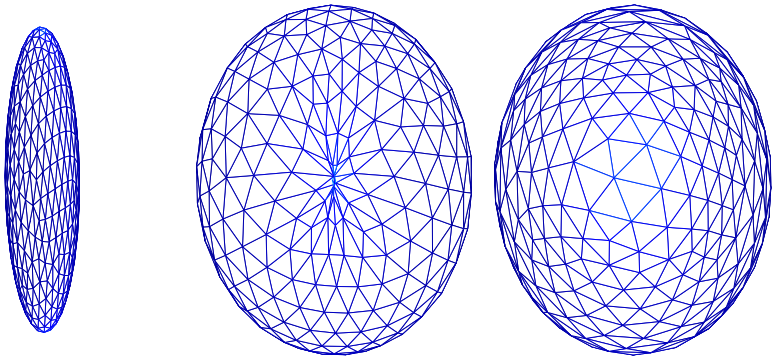


Fig. 6. Evolution of an initial mesh (left) of an ellipsoid under mean curvature flow. The center and the right images show the top of the surface meshes after 0.14 seconds of evolution without and with mesh adaption, respectively.

We discretize the problem in space using the generalized finite difference scheme, and discretize the equation in time using a semi-implicit scheme by evaluating the second-order terms over the new time step and evaluating the first-order terms over the current time step. As the surface evolves, the mesh may need to be adapted in order to maintain good spacing between the points. Utilizing adaptivity during evolution can help maintain mesh quality and ultimately increase the stability when trying to further evolve the mesh. Fig. 6 shows a comparison with and without mesh adaption for the evolution of an ellipsoid with semiaxes 1.5, 2, and 8. Without mesh improvement, the points become overly crowded at the top of the ellipsoid, which can severely undermine the time step requirement for the PDE solver. We optimize and adapt the mesh using our technique, so that the mesh quality and the order of accuracy are achieved simultaneously.

7 Conclusions

In this paper, we investigated the problem of remeshing a surface mesh to high-order accuracy. We demonstrated that lower-order remeshing, which may preserve the "shape" of the geometry, can destroy the order of accuracy or even convergence of numerical computations. We describe a new remeshing strategy that can not only substantially improve mesh quality, but also preserve the accuracy of the geometry to high order. In addition, we showed that our technique ensures that high-order algorithms for curvature computations on discrete surfaces still converge after remeshing, whereas some commonly used techniques may cause non-convergence of computations after remeshing.

Our proposed technique has many potential scientific and engineering applications, especially those involving high-order methods for solving differential equations. An example is the solution of geometric partial differential equations with

moving meshes. As a potential future research direction, we plan to couple our remeshing techniques with error estimations, to deliver error-based mesh adaption.

Acknowledgement. The second and third author are supported by the DOE NEUP program under contract #DE-AC07-05ID14517. The third author is supported also by DoD-ARO under contract #W911NF0910306.

References

1. Benito, J., Urena, F., Gavete, L.: Solving parabolic and hyperbolic equations by the generalized finite difference method. Journal of Computational and Applied Mathematics 209(2), 208–233 (2007)
2. Cahn, J.W., Taylor, J.E.: Surface motion by surface diffusion. Acta Metallurgica et Materialia 42(4), 1045–1063 (1994)
3. Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.: Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley, Chichester (2009)
4. Fleishman, S., Cohen-Or, D., Silva, C.T.: Robust moving least-squares fitting with sharp features. ACM Trans. Comput. Graph. (TOG) 24(3) (2005)
5. Frey, P.J.: About surface remeshing. In: Proceedings of 9th International Meshing Roundtable, pp. 123–136 (2000)
6. Garimella, R.: Triangular and quadrilateral surface mesh quality optimization using local parametrization. Comput. Meth. Appl. Mech. Engrg. 193(9-11), 913–928 (2004)
7. Heath, M.T.: Scientific Computing: An Introductory Survey, 2nd edn. McGraw-Hill, New York (2002)
8. Irons, B.M., Ahmad, S.: Techniques of Finite Elements. Ellis Horwood Ltd, Chichester (1980)
9. Jiao, X., Colombi, A., Ni, X., Hart, J.: Anisotropic mesh adaptation for evolving triangulated surfaces. Engrg. Comput. 26, 363–376 (2010)
10. Jiao, X., Wang, D.: Reconstructing high-order surfaces for meshing. Engineering with Computers (2011); Available online in September 2011. Preliminary version appeared in Proceedings of 19th International Meshing Roundtable
11. Jiao, X., Wang, D., Zha, H.: Simple and effective variational optimization of surface and volume triangulations. Engineering with Computers - Special Issue: 17th International Meshing Roundtable 27, 81–94 (2011); Preliminary version appeared in Proceedings of 17th International Meshing Roundtable
12. Jiao, X., Zha, H.: Consistent computation of first- and second-order differential quantities for surface meshes. In: ACM Solid and Physical Modeling Symposium, pp. 159–170. ACM (2008)
13. Prieto, F.U., Munoz, J.J.B., Corvinos, L.G.: Application of the generalized finite difference method to solve the advection-diffusion equation. Journal of Computational and Applied Mathematics 235(7), 1849–1855 (2011)
14. Walton, D.: A triangular g1 patch from boundary curves. Comput. Aid. Des. 28(2), 113–123 (1996)
15. Wang, D., Clark, B., Jiao, X.: An analysis and comparison of parameterization-based computation of differential quantities for discrete surfaces. Computer Aided Geometric Design 26(5), 510–527 (2009)
16. Xu, G., Zhang, Q.: A general framework for surface modeling using geometric partial differential equations. Computer Aided Geometric Design 25(3), 21 (2008)