OPTIMIZATION-BASED QUADRILATERAL AND HEXHEDRAL MESH UNTANGLING AND SMOOTHING TECHNIQUES*

by
Xiang-Yang Li**
UINVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, IL

Lori A. Freitag
MCS DIVISION
ARGONNE NATIONAL LABORATORY

*Work performed at Argonne National Laboratory, a contract Laboratory of the United States Department of Energy.

**Participant in the Summer 1999 Energy Research Undergraduate Laboratory Fellowship Program. This program is coordinated by the Division of Educational Programs.

Optimization-Based Quadrilateral and Hexahedral Mesh Untangling and Smoothing Techniques

Xiang-Yang Li [*] Lori A. Freitag [†]

Abstract. *The accuracy and efficiency of the solution to numerical systems depends on the quality of the mesh used. Automatic mesh generation and adaptive refinement methods can, however, produce poor quality elements, even invalid elements, for quadrilateral or hexahedral mesh. We present an optimization-based approach for quadrilateral or hexahedral mesh untangling that maximizes the minimum area or volume of affected simplicial of an element in a local submesh. We formulate the problem as a linear program that is solved by using the computationally inexpensive simplex method. By proving that the function level sets are convex regardless of the position of the free vertex, we show that the local subproblem is guaranteed to converge. We combine the mesh untangling technique with optimization-based mesh improvement techniques. Typical results showing the effectiveness of the combined untangling and smoothing techniques are given for both quadrilateral and hexahedral meshes.*

Keywords. unstructured mesh, mesh untangling, mesh smoothing, quadrilateral mesh, hexahedral mesh.

1 Introduction

In the numerical solution of many scientific and engineering applications, the finite element and finite volume methods are two of the commonly used effective tools. Both techniques require a spatial decomposition of the computational domain into a mesh, i.e., a union of simple geometric elements such as triangles or quadrilaterals in two dimensions and tetrahedra or hexahedra in three dimensions. This is the problem of mesh generation. Unfortunately, meshes generated by automatic mesh generation tools, especially quadrilateral meshes and hexahedral meshes, can contain poorly shaped or distorted elements, which causes numerical difficulties during the solution process. For example, we know that the discretization error in the finite element solution increases as element dihedral angles becomes too large[2]; and the condition number of the element matrices increases as angles becomes too small[12]. Thus, for meshes with highly distorted elements, the solution is both less accurate and more difficult to compute. Needless to say, we need a topologically correct mesh for numerical systems solution, in other words, the elements of the mesh should be untangled. Compared with triangular meshes and tetrahedral meshes, quadrilateral meshes and hexahedral meshes can be distorted to poor quality in more ways than triangles or tetrahedra can.

In this paper, we study the mesh untangling problem which moves the vertices of a geometrically invalid but topologically valid mesh to make it also geometrically valid; and we also study the mesh smoothing problem for quadrilateral and hexahedral mesh. Local mesh smoothing algorithms have been shown to be effective in repairing distorted elements in automatically generated meshes. Researchers [3, 4, 5, 10, 7, 8, 9, 15] have developed several methods to improve element quality if the mesh is geometrically valid (i.e., untangled).

[*]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. The work is done when the author visited Argonne National Lab. The email: xli2@cs.uiuc.edu.

[†]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, 60439. The email: freitag@mcs.anl.gov.

These techniques include local reconnection methods, such as edge or face swapping; the refinement and coarsening methods that split or merge elements of the mesh, add or delete the mesh vertices; and vertex adjustment methods such as mesh smoothing. The simplest smoothing algorithm is Laplacian smoothing, which moves a free vertex to the geometric center of incident vertices. It tries to place the vertices evenly in space. Unfortunately, this method operates heuristically and can create invalid meshes or elements of worse quality than those contained in the original mesh. For example, even if the vertices of a tetrahedral mesh are evenly spaced, the mesh can still contain slivers. For quadrilateral or hexahedral mesh, the same problem exists. In contrast, optimization-based methods are designed to maximize some measure, such as the angle (dihedral angle), of mesh quality and are very effective at eliminating extremal angles in the mesh.

To achieve an overall improvement in the mesh, smoothing algorithms often perform some number of sweeps over the adjustable vertices. These techniques cannot, however, always be used when starting with a mesh with inverted elements. To address this problem, in this paper we present a local mesh "smoothing" technique which is designed specifically for mesh untangling. This method assumes that the mesh has valid topological connectivity but that the vertices positions are such that some of the elements are tangled or inverted. Notice here, a quadrilateral element is tangled if there are two edges intersect on a point which is not the vertex of the quadrilateral element, or the order of the vertices does not satisfy the right-hand rule. The tangled hexahedral element is defined similarly. Figure 1 shows some example of the inverted quadrilateral or hexahedral elements.

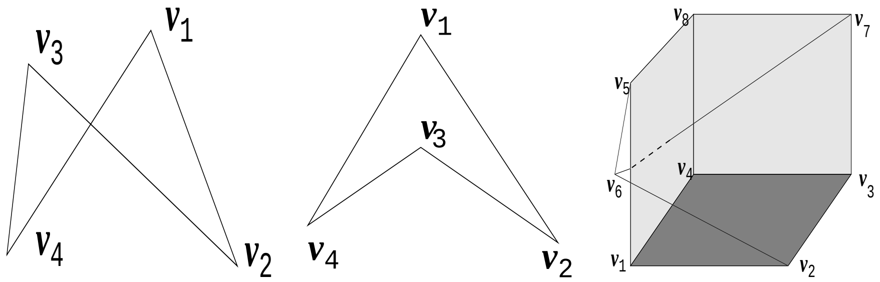


Figure 1: Some examples of inverted elements of a mesh. The first is an inverted quadrilateral element with two edges intersected; the second is an inverted quadrilateral with incorrect orientation; and the third is an inverted hexahedral element.

Mesh untangling and smoothing techniques often operate locally. In other words, it uses submesh data from the elements incident to the mesh vertex that is being adjusted. See Figure 2 for an example of such a two-dimensional quadrilateral submesh. The local submesh of the free mesh vertex v consists eight fixed vertices $v_1, \ldots, v_8$, and four incident quadrilateral elements $q_1, \ldots, q_4$: q_1 has four vertices v, v_6,v_7,v_8; q_2 has four vertices v, v_8,v_1,v_2; q_3 has four vertices v, v_2,v_3,v_4; q_4 has four vertices v, v_3,v_4,v_5. The *feasible region* of free vertex v is defined as the set of possible locations for v for which all of the incident elements are not tangled. The shaded region in each submesh is the feasible region for v. Notice that the feasible region can be empty for some local mesh configurations. As proved later, the feasible region of a free vertex v is always convex.

For some quality measures, unlike to the triangular or tetrahedral mesh, moving the free vertex v will only affect part of the quality measure of a quadrilateral or hexahedral elements. For example, moving free vertex v will only change 3 angles among the 4 angles of a quadrilateral elements. Hence, if the quality of the unaffected angle is very poor, and optimization-based smoothing methods is applied to all angles of the elements, then the smoothing can not move the free vertex to improve the quality of the local submesh. In other words, wherever the free vertex is, the quality of the local submesh is determined by the very poor unaffected angle. Thus, any optimization-based methods that can be used to untangle or smooth the local submesh have to apply only to the quality that can be changed by moving the free vertex. We define the *affected simplex* as the simplex formed by free vertex v and all of the vertices, which are directly connected to v, of a quadrilateral element or hexahedral element. For example, in the second figure of Figure 2, the affected simplex of quadrilateral element q_4 is triangle $\nabla v v_4 v_6$.

The local submesh on the left of Figure 2 shows the free vertex not in the feasible region: quadrilateral elements q_1 and q_2 are tangled. The local mesh on the middle shows the free vertex in the feasible region, but the quality of the elements are very poor: quadrilateral elements q_3 and q_4 has very poor angle quality. But all of the elements in this mesh are valid because the free vertex lies inside the feasible region. In the right submesh, we show the same local submesh after a typical mesh smoothing operation. In local mesh smoothing techniques, the location of the free vertex is changed according to some rule or heuristic procedure based on information available at the mesh vertices of all incident quadrilateral (hexahedral) elements. Only the position of v is affected; adjacent vertex locations remain unchanged. The quality of elements q_3 and q_4 has significantly improved by moving v toward the middle of the feasible region.

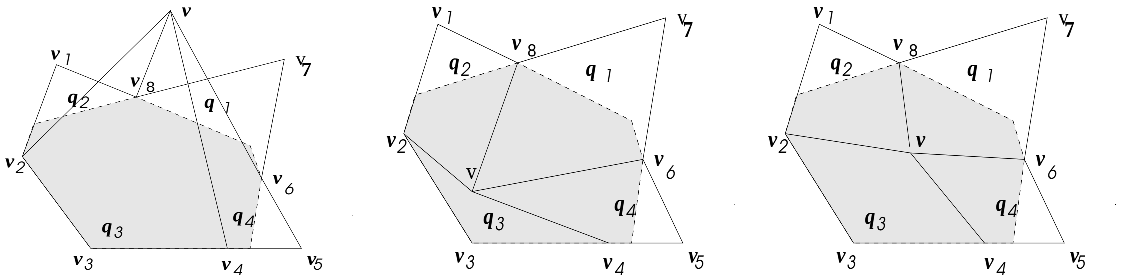


Figure 2: A local submesh consisting of a free vertex, v, to be moved and its incident vertices, $v_1, \ldots v_8$, and four quadrilateral elements, $q_1, \ldots, q_4$. The feasible region for the free vertex location is shaded gray in all three submeshes. The submeshes show three possible locations for v. The first results in an invalid mesh with inverted elements; the second in a valid but poor-quality mesh, and the third in a higher-quality valid mesh.

In previous papers [10, 7, 8, 9], the authors have developed an optimization-based approach to mesh untangling and improvement that involves minimizing a nonsmooth, composite function on a local submesh using a technique analogous to steepest descent. Given the convex function level sets in the feasible region and a feasible starting point, it had been shown [1] that this approach is equivalent to generalized linear programming techniques, and can be guaranteed to converge to an optimal solution. In Figure 3, we show the level sets for three geometric mesh quality metrics that can be used to create a composite function: minimum angle in the local submesh, minimum element area in the local submesh, and the minimum affected simplex area in the local submesh. Notice that the function level sets for the minimum element area and the minimum affected simplex area are convex regardless of the location of the free vertex. Thus, if the mesh contains inverted or tangled elements, optimization approaches using these metrics will be guaranteed to converge. Also notice that the level set of the minimum angle are convex only if the free vertex lies inside the feasible region, and the local submesh is not tangled. Thus, this metric can only be used to improve the quality of a valid local submesh.

In this paper, we present an optimization-based approach to untangle quadrilateral or hexahedral meshes based on maximizing the minimum affected area or volume of mesh elements by moving the free vertex in a local submesh. The paper is structured as following. The formulation and solution for mesh untangling are presented in Section 2. We prove that the level sets of function, which is used to untangle the mesh, are convex regardless of the position of the free vertex, and hence the local subproblem is guaranteed to converge. However, maximizing the minimum affected area or volume of mesh elements, although well suited for mesh untangling, is not ideal for mesh improvement because a small, but perfectly shaped element is likely to be distorted by maximizing its area, and in turn it may create invalid elements. We include a brief description of the mesh improvement method for completeness in Section 3, and expand previous results to show that the level sets for the quadrilateral mesh quality criterion minimum affected angle are convex in the feasible region. Typical results showing the effectiveness of the combined untangling and smoothing techniques for both two- and three-dimensional meshes are given in Section 4.

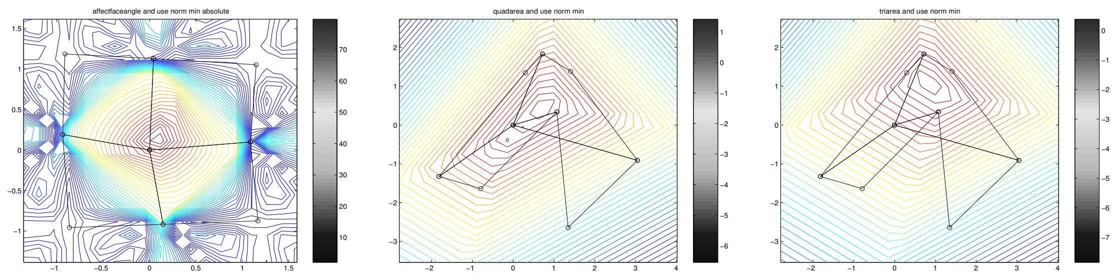


Figure 3: Level sets for the affected minimum angle, affected minimum triangle area, and the metrics. The minimum angle metric is nonconvex outside of the feasible region; the other two is convex everywhere.

2 Optimization-Based Mesh Untangling

We formulate the local mesh untangling techniques in terms of the free vertex v to be adjusted; the other vertices of the quadrilateral or hexahedral elements that are incident to v, i.e., $adj(v)$; and the connectivity information $conn(v)$ of all these incident quadrilateral or hexahedral elements. The connectivity information $conn(v)$ is stored according to the right hand rule for each element. Suppose x is the position of the free vertex; then the general form of a local untangling algorithm is given by

$$x_{new} = Untangle(x, adj(v), conn(v)),$$

where x_{new} is the proposed new position of v. Ideally, x_{new} improves the local submesh in such a way that it is untangled or can be untangled in succeeding sweeps through the mesh.

In this section, we describe a method for performing the free vertex adjustment based on maximizing the minimum area or volume of affected simplexes contained in the local submesh. By showing that the level sets are convex regardless of the position of the free vertex, we prove that convergence of the local subproblem is guaranteed.

2.1 Formulation

Notice that the number of the affected simplexes is same as the number of quadrilateral (hexahedral) elements in the local mesh. The function that gives the minimum area or volume of an affected simplex in a local submesh is

$$f(\boldsymbol{x}) = \min_{1 \le t \le n} A_t(\boldsymbol{x}), \tag{1}$$

where n is the number of elements in the local submesh, $A_t()$ is the area (volume) function of the affected simplex defined by free vertex $\boldsymbol{x}$. In two dimensions, if triangle t is defined by the free vertex, $\boldsymbol{x}$, and two other vertices, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, then A_t can be expressed as a function of the Jacobian of the element

$$A_t(\boldsymbol{x}) = \frac{1}{2} det(\boldsymbol{x}_i - \boldsymbol{x}, \boldsymbol{x}_j - \boldsymbol{x}) = \frac{1}{2}(a_{xt}x + a_{yt}y + c_t), \tag{2}$$

where

$$a_{xt} = y_i - y_j, \quad a_{yt} = x_j - x_i, \quad c_t = x_i y_j - x_j y_i. \tag{3}$$

Notice that where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two vertices of a quadrilateral element that are directly connected to the free vertex v.

Similarly, in three dimensions, if the affected tetrahedron t is defined by the free vertex, $\boldsymbol{x}$, and three other vertices, $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, and $\boldsymbol{x}_k$, then the volume A_t is given by

$$A_t = \frac{1}{6}det(\boldsymbol{x}_i - \boldsymbol{x}, \boldsymbol{x}_j - \boldsymbol{x}, \boldsymbol{x}_k - \boldsymbol{x}) = \frac{1}{6}(a_{xt}x + a_{yt}y + a_{zt}z + c_t), \tag{4}$$

where

$$a_{xt} = -det\begin{bmatrix} 1 & 1 & 1 \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix}, \quad a_{yt} = -det\begin{bmatrix} x_i & x_j & x_k \\ 1 & 1 & 1 \\ z_i & z_j & z_k \end{bmatrix}, \quad a_{zt} = -det\begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{bmatrix},$$
$$and \quad c_t = -det\begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{bmatrix}.$$

In both two and three dimensions, A_t is a linear function of the free vertex position, $\boldsymbol{x}$. Notice that if the affected simplex is inverted, then $A_t(\boldsymbol{x})$ has negative value. We can use this fact to pose the solution of the optimization problem

$$\max \min_{1 \le t \le n} A_t(\boldsymbol{x})$$

as a linear programming problem. To do this, we first construct the dual problem and then solve the dual problem. For the detail of forming and solving the dual problem, the readers are referred to [11].

If after the untangling procedure, for all local submesh, the minimum area or volume of all affected simplexes is positive, then we know that all quadrilateral (hexahedral) elements of the mesh are not tangled, and the feasible region of free vertex for local submesh is convex.

However, maximizing the minimum area or volume of the affected simplex is actually aggressive to untangle the mesh: because, for some local submesh, although an affected simplex has negative area (or volume), the quadrilateral or hexahedral element can still be valid. See Figure 4. To address this problem, the other approach is to untangle the mesh by maximizing the actual area or volume of all elements of the local submesh. Notice, here, for tangled elements q, we define its area or volume A_q as following.

$$A_q = \delta * (A_t + A_o)$$

where A_t is the area or volume of the affected simplex, A_o is the area or volume of the remaining portion of the element; if the element is valid then δ is set to 1, else, δ is set to -1. Notice that A_o is independent of the position of free vertex v. In other words, we can view it as a constant. For example, in Figure 4, we assume that vertex v_3 is the free vertex, then A_t is the area of triangle $v_2v_3v_4$ and A_o is the area of triangle $v_2v_4v_1$. We still use the linear programming approach to find the location of free vertex that maximizes the minimum area (volume) of the incident elements in the local submesh.

Notice that by applying this approach to untangle the mesh, the final mesh can still have some vertices which have negative Jacobi. In other words, the quality of the elements are poor. Thus if we want combine the untangling and smoothing, we use the technique that maximizes the minimum area (volume) of affected simplex, because after the untangling, ideally, the quality at each is not very poor.

2.2 Checking Tangled Element

Notice that the untangling technique is not guaranteed to improve the quality of the local submesh. Thus, if the local submesh is not untangled, it is desirable not to apply the untangling procedure at all. [1] Hence, given the local submesh information, we need to decide whether a local submesh is untangled or not. We know that a triangle or a tetrahedra element is tangled if and only if the Jacobian of the element is negative, assuming that we always use the right-hand rule to number the vertices of an element. But for quadrilateral

[1]To get a global convergence, it may be desirable to use the untangling method by maximizing the minimum area (volume) of the affected simplex.

and hexahedral mesh elements, the above argument is not true, i.e., an element can have negative Jacobian at some vertices, but it is still untangled. Here the Jacobian at a vertex v of a mesh element is the Jacobian of the affected simplex defined by v. Figure 4 gives some examples of untangled element with negative Jacobian at some vertices: the Jacobian at vertex v_3 of the first figure is negative, but the quadrilateral element is still valid.

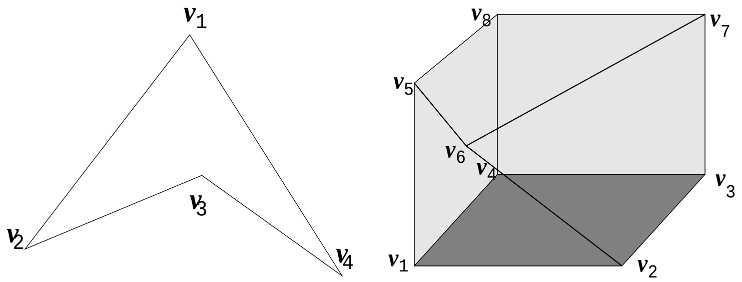


Figure 4: Some examples of noninverted elements of a mesh. The first is a noninverted quadrilateral element with negative Jacobian at one vertex v_3; and the second is an noninverted hexahedral with negative Jacobian at one vertex v_6.

A quadrilateral or hexahedral element is valid if the intersections among the edges (or faces) are the lower dimensional elements of the mesh. For example, the geometric intersection of two edges should be either empty or a vertex of the mesh; the intersection of two faces of a hexahedra should be either empty or an edge of the mesh or a vertex of the mesh. By checking the possible valid configurations of quadrilateral element, the following lemma state the sufficient and necessary condition for a quadrilateral element to be valid.

Lemma 2.1 *A quadrilateral element is valid if and only if there is at most one negative Jacobian among four vertices.*

Proof: First, we prove that if a quadrilateral element is valid, then there is at most one negative Jacobian among four vertices. Assume that there is one negative Jacobian at vertex v_1. See the second figure of Figure 7. Then the feasible region for vertex v_3, which will not violate the validity of the quadrilateral element is shown as the shaded area in the second figure of Figure 7. It is easy to verify that the Jacobians at v_2, v_3 and v_4 are not negative.

If there is at most one negative Jacobian among four vertices of a quadrilateral element, then we show that the quadrilateral element is valid. If there is no negative Jacobian among four vertices, then it is easy to show that the quadrilateral element is valid. Otherwise, assume v_3 has negative Jacobian, as shown by the second figure of Figure 7. Because the Jacobian at vertex v_2 is not negative, then we know that v_3 must lie on the positive (right) half plan defined by line v_1v_2. Because the Jacobian at vertex $v4$ is not negative, then we know that v_3 must lie on the positive (right) half plan defined by line v_1v_4. In other words, v_3 must lie in the shaded quad-plane, i.e., the feasible region for vertex v_3. Notice that for any vertex v_3 in the shaded region, quadrilateral $v_1v_2v_3v_4$ is valid. □

Hence, whether a quadrilateral element is valid can be checked by just computing the Jacobian at it's four vertices. But unfortunately, it is more difficult to check the validity of a hexahedral element. We use a simple checking as following. First, we compute the Jacobian at all 8 vertices of the hexahedral element. If the number of the negative Jacobian is at most one, then we know that the hexahedra is valid. If there are more than five vertices with negative Jacobian, then we know that the hexahedra is invalid. Otherwise, we check all vertices with negative Jacobian, if there are adjacent vertex with negative Jacobian, then we know that the hexahedra is invalid, otherwise, we regard the hexahedra as valid.

Combine both the observation for quadrilateral and hexahedral elements, we use the following conjecture to check the validity of an element.

Conjecture 2.2 *A quadrilateral or hexahedral element is valid if and only if there are no two adjacent vertices both with negative Jacobian.*

Notice that the claim is true for quadrilateral element.

2.3 Convergence of Local Mesh Untangling

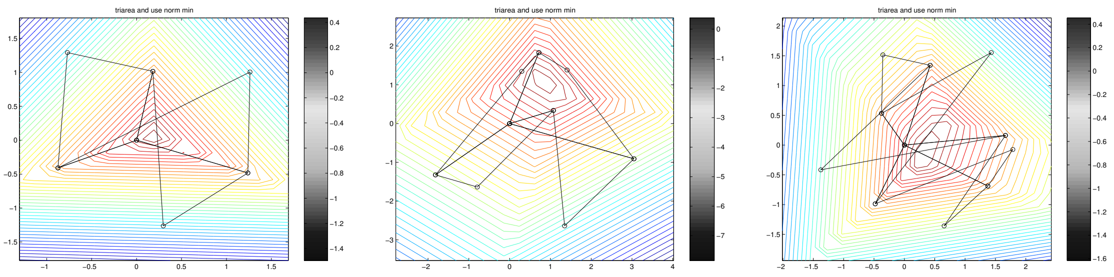


Figure 5: Level sets for the minimum affected simplex area function for three tangled local submeshes.

To guarantee convergence of the local optimization problem, the level sets of the composite function given in equation (1) must be convex and closed. Typical level sets for this function in two dimensions are shown in Figure 5 for three different tangled local submeshes. In each case, the level sets appear to be convex, and in this section we prove that they are convex in both two and three dimensions in the entire domain, using same technique as in [11], if the initial local submesh is not degenerate and the subproblem is well-posed. The area (or volume) function of the affected simplex can be written as

$$A_t(x) = \frac{1}{d} b_t h_t(x),$$

where $d = 2$ for two dimension, and $d = 3$ for three dimension, b_t is the length (area) of the base of simplex t, and $h_t(x)$ is the perpendicular distance between x and b_t. All triangles (or tetrahedra) with base b_t and area greater than c have a third (or fourth) vertex that lies in a half-plane with boundary parallel to b_t. The level set boundary for a local submesh is the intersection of the half planes defined by the individual affected simplex in that local submesh. As the intersection of half-plane is convex, these levels sets are convex. One can formally show that the level sets are convex in both two and three dimensions.

Theorem 2.3 *The level sets of the function*

$$f(x) = min_{1 \leq t \leq n} A_t(x)$$

are convex, where A_t is defined by equations (2) and (4) in two and three dimensions, respectively.

Proof: Note that component functions $A_t(x)$ as given in equations (2) and (4) are linear functions of the position x of the free vertex. Thus, these component functions are concave. We have that the minimum of concave functions is also concave, thus $f(x)$ is concave.

The level sets for any concave function are convex [6]. For any two points, x_1 and x_2, on the boundary of the set $S(\mu) = \{x | f(x) \geq \mu\}$, we have that for $\alpha \in [0, 1]$,

$$f(\alpha x_1 + (1 - \alpha)x_2) \geq \alpha f(x_1) + (1 - \alpha) f(x_2) = \mu.$$

Thus, the set $S(\mu)$ is convex and therefore, the level set of f are convex. □

We assume that the local subproblems are not degenerate and are well-posed so that $f(x)$ (the minimum area or volume of an affected simplex in the submesh) is bounded. As the level sets are convex, any local maximum of $f(x)$ is a global maximum. Thus, any optimization algorithm guaranteed to find a local maximum (such as the linear programming approach presented [11]) is guaranteed to determine the global maximum for the local subproblem. Because the level sets for $f(x)$ are convex on the entire domain, this approach converges to the global maximum from any starting point for the free vertex.

Notice that, a similar proof can show that the level set of the minimum area (or volume) of the elements of a local submesh is also convex. Thus untangling method based on maximizing the minimum area or volume can also be guaranteed to converge for any local submesh.

3 Optimization-Based Mesh Smoothing

We formulate the mesh smoothing method in the same manner as the optimization-based untangling technique: given a free vertex v, and its adjacent vertices $adj(v)$ and elements connectivity $conn(v)$, the new position of the free vertex, x_{new}, is given by the general operation

$$x_{new} = Smooth(x, adj(v), conn(v)).$$

The new location of the free vertex is the point that improves the mesh according to some measure of mesh quality such as dihedral angle or the condition number or the element aspect ratio. The action of the function *Smooth* is determined by the particular algorithm chosen, and in this section we describe some methods for performing the vertex adjustment.

3.1 Laplacian and "Smart" Laplacian Smoothing

For Laplacian smoothing, the smoothing operator moves the free vertex to the geometric center of the adjacent vertices. Mathematically, Laplacian smoothing will minimize the summation of the square of all edges incident to free vertex v. No effort is made to ensure that mesh quality is improved. And at worst, it can generate poor quality even invalid elements: Laplacian smoothing will create an invalid mesh, if the computed free vertex position has moved outside the feasible region. However, the technique is computationally inexpensive, easy to implement, and therefore commonly used. A simple variant of Laplacian smoothing, which we call "smart" Laplacian smoothing, relocates the free vertex to the geometric center of the adjacent vertices only if the quality of the local mesh is improved according to some mesh quality measure.

3.2 Optimization-Based Smoothing

Like optimization-based mesh untangling, the optimization-based mesh improvement finds the position x^* that maximizes the composite function

$$f(x) = min_{1 \leq t \leq n} q_t(x), \tag{5}$$

where $q_t(x)$ is a mesh quality metric such as minimum angle, sine of the minimum angle, or aspect ratio of an element. [2] Typically, each $q_t(x)$ is a nonlinear, smooth, and continuously differentiable function, and multiple functions can obtain the minimum value of the composite function. Hence, the composite function $f(x)$ has discontinuous partial derivatives where the active set changes from one set of functions to another

[2]We also tried to minimize the function

$$\sqrt{\sum_{1 \leq t \leq n} q_t^2(x)}.$$

Ideally, if the function is minimized, quality measure q_t, $t = 1, 2, \ldots, n$ will have the same value. For example, if we use the sin of angle as the quality measure, ideally, after the local smoothing, every angle will have same sin value. But in practice, it did not. The reason is as follows: the change of the value of q_t is not independent (there are a lot of constraints among the quality measures q_t), and the summation of the quality q_t is not fixed. Hence, even if the function is minimized, quality measures q_t do not get same value. Some of the quality may be very small.

set. Let the minimum value of the functions evaluated at x be called the active value, and the set of functions that obtain that value, the active set, be denoted by $S(x)$.

We solve this nonsmooth optimization problem using an analogue of the steepest descent method for smooth functions. The search direction, s, at each step is the steepest descent direction derived from all possible convex linear combinations of the gradients in $S(x)$. This direction is computed by solving the following quadratic programming problem for β_i.

$$\min \overline{g}^T \overline{g}, \quad where \quad \overline{g} = \sum_{i \in S} \beta_i g_i(x)$$

$$subject \quad to \quad \sum_{i \in S} \beta_i = 1, \quad \beta_i \geq 0.$$

The line search subproblem along s is solved by predicting the points at which the active set $S(x)$ will change. These points are found by computing the intersection of the projection of a current active function in the search direction with the linear approximation of each $q_i(x)$ given by the first-order Taylor series approximation. The distance to the nearest intersection point from the current location gives the initial step length, α. The initial step is accepted if the actual improvement achieved by moving v exceeds 90 percent of the estimated improvement or the subsequent step results in a smaller function improvement. Otherwise, α is halved recursively until a step is accepted, or α falls below some minimum step length tolerance. The optimization-based smoothing algorithm is described in more detail elsewhere [10, 7]. We note that similar local optimization-based smoothing methods [3, 13, 15, 14] have been proposed for a variety of optimization procedures and mesh quality measures.

3.3 Convergence of Local Mesh Smoothing

To guarantee convergence of the local optimization problem, the level sets of the composite function must be convex and closed. Typical level sets for the minimum affected angle, the maximum affected angle and the 2-norm of the affected angles in two dimensions are shown in the first figure of Figure 6. In this section we prove that the level set of the most commonly used composite functions are convex in two dimensions in the entire domain if the initial local submesh is not degenerate and the subproblem is well-posed.

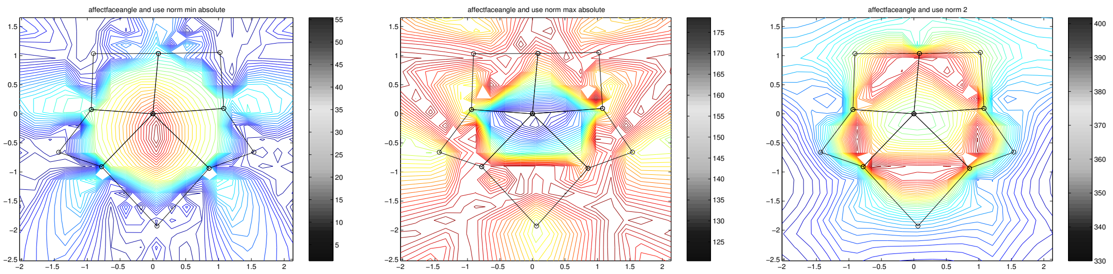


Figure 6: Level sets for the minimum affected angle, maximum affected angle, and the 2-norm of the affected angles. Each of these metrics is nonconvex outside of the feasible region.

In practice, the optimization process is terminated if one of the following conditions [11] apply: (1) the step size falls below the minimum step length with no improvement obtained; (2) the maximum number of iterations is exceeded; (3) the achieved improvement of any given step is less than some user-defined tolerance; or (4) the Kuhn-Tucker conditions of nonlinear programming are satisfied, indicating that we have found a local maximum $\mathbf{x}^*$.

First, we show that, after the untangling by maximizing the minimum area or volume of the affected simplex, we have the following fact.

Lemma 3.1 *The feasible region for each local submesh is convex.*

Proof: First, it is easy to show that for any edge (face) of a quadrilateral (hexahedral) element, the feasible region of a free vertex v such that connecting v does not violate the validity of an element is a half plane bounded by the edge (face). For example, if we have edges v_1v_2 and v_1v_4 fixed as shown in Figure 7, then the feasible region for v_3 is at the right half plane of edge v_1v_2 and v_1v_4. Then the feasible region for free vertex that does not violate the validity of a local quadrilateral or hexahedral element is the intersection of all half spaces. See Figure 7. Then the feasible region of free vertex for the local submesh is convex. Hence the intersection of all feasible regions for every elements is convex. Then the lemma follows. □

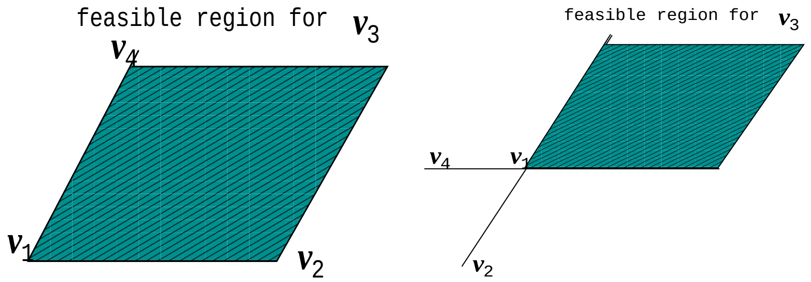


Figure 7: The feasible region for a free vertex that does do not violate the validity of a quadrilateral element. Left figure gives an example of quadrilateral with no vertex with negative Jacobian; the right figure gives an example with one vertex v_1 having negative Jacobian.

To guarantee the convergence of the mesh improvement methods, the level sets of the composite function used must also be convex. Typically, we prove that the level set for the composite function

$$f(x) = \min \sin(\theta_i(x))$$

are convex in the feasible region of a two dimensional quadrilateral mesh, using a similar approach by [11]. Thus the convergence of the local submesh is guaranteed.

Theorem 3.2 *The level sets, S, of the function $f(x) = \min_{1 \leq i \leq n} \sin \theta_i(x)$ are convex in the feasible region of a two-dimensional local submesh after untangling..*

Proof: Similar to the proof of [11], it is sufficient to show that the levels sets of the function

$$g(x) = \min_{1 \leq i \leq n} \theta_i(x)$$

are convex. The fact about the triangle had been shown previously [1, 11]. We use the almost same argument here.

There are two types of angles affected by the free vertex: one is the *external angles* at quadrilateral vertices other than the free vertex; the other is the *internal angles* formed at the free vertex v by two line segments connecting v to two vertices of the diagonal edge of the quadrilateral. See Figure 8.

Let $S_j(\theta)$ be the set of points in the feasible region, which form an internal (external) angle $\theta_j(x) \geq \theta$, if $S_j(\theta)$ is the internal (external) angle. The level set $S(\theta) = \{x | \theta \leq g(x)\}$ is the intersection of the sets $s_j(\theta)$ and the feasible region. Knowing that the feasible region for a free vertex is convex, we need only to prove that the level set of each $S_j(\theta)$ is convex.

For internal angle $S_j(\theta)$, using the same approach as in [11], we can prove that the level set of $S_j(\theta)$ is convex. For external angle $S_j(\theta)$, because the feasible region is at one side of the quadrilateral edge, hence the intersection of the points that form an angle $\theta_j(x) \geq \theta$ and the feasible region is also convex. Hence the level set of the quality function is convex. Then the lemma follows. □

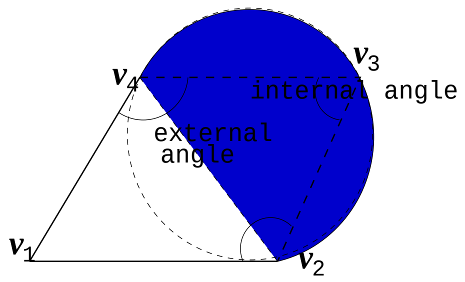


Figure 8: The internal and the external angle formed by the free vertex and the other three vertices of a quadrilateral element.

Experimental results [10, 8] have demonstrated the effectiveness of the optimization-based method compared with Laplacian smoothing for two- and three-dimensional simplicial meshes. In each case the optimization-based method with the quality metric minimum sine of an angle was effective at eliminating extremal angles from the mesh whereas the Laplacian smoother is often unable to significantly improve the most severely distorted elements. The corresponding increase in computational cost is approximately a factor of four in two dimensions and a factor of ten in three dimensions. These results also showed that three sweeps of the mesh were usually sufficient to improve the mesh quality; additional sweeps offered minimal incremental improvement.

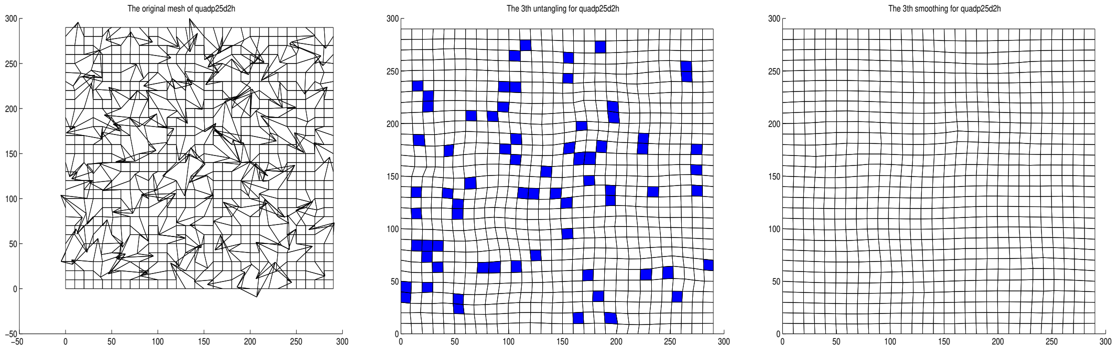
A number of approaches that combine Laplacian and optimization-based smoothing can be used to improve the mesh as effectively as optimization-based smoothing used alone at a fraction of the cost. See [7]. The approach used for the examples in Section 4 of this paper is to try smart Laplacian smoothing as the first step for every subproblem. If the active value in the local mesh after this step exceeds a user-defined threshold value (in this case, 30^o in two dimensions and 15^o in three dimensions), the local algorithm terminates; otherwise, optimization-based smoothing is performed. We note that other approaches[15, 8] exist for combining Laplacian and optimization-based smoothing.

4 Experimental Results

To show typical results for the mesh untangling problem, we start with a regular grid. We perturb a percentage of the vertices in these meshes a given distance to create new meshes that have valid connectivity, but invalid elements. In the discussion that follows, let P be the percentage of vertices perturbed and h be the average edge length in the initial mesh.

The initial two-dimensional mesh has 841 elements and an initial minimum angle of 90°. We perturb 25 percent of the vertices in this mesh a distance $2h$ to create a tangled mesh containing 276 invalid quadrilateral elements; which is shown in Figure 4 (a). The perturbed mesh is untangled by using the linear programming approach described in Section 2. The untangling process stops when all of the elements are valid or 40 sweeps through the mesh have been performed. In this case, only 3 sweeps through the mesh were required to eliminate all of the invalid elements. The second mesh in Figure 4 shows the untangled mesh. Although this mesh is valid, the untangling procedure results in meshes of poor quality; minimum angles of about 48.9°. We therefore follow the mesh untangling procedure with three passes of optimization-based smoothing using the combined approach with Laplacian method, and the quality metric maximizes the minimum affected angle. The resulted mesh are shown in the last mesh in Figure 4. The mesh has a minimum angle about 84.3°.

The initial three-dimensional mesh is shown in Figure 4. This mesh has 729 elements and an initial minimum dihedral angle of 90°. We again perturb this mesh in such a way that 10 percent of the vertices are moved a distance h from their original position, creating a mesh with 28 invalid elements. In this case, only one sweep through the mesh using the untangling procedure is required to create a valid mesh with



(a) (b) (c)

Figure 9: Typical results for quadrilateral mesh untangling and smoothing using the linear programming approach. (a) is the mesh by perturbing 25 percent of the vertices by $2h$; (b) the mesh after the untangling, the shaded quadrilateral corresponds to the tangled quadrilateral from original mesh; (c) the mesh after 3 passes of smoothing.

a minimum dihedral angle of 80.9^o. The untangling procedure is again followed by three passes of mesh smoothing using the combined approach and a mesh quality metric of minimum the dihedral angle, resulting in a final mesh with a minimum dihedral angle of 87.5^o.

To evaluate the performance of the mesh untangling procedure as a function of the number of invalid elements and the amount of vertex perturbation, as in [11], we created two series of perturbed meshes for each of the initial meshes. The perturbation direction for a particular mesh vertex is randomly chosen, and for each series of meshes the same vertices are perturbed in random directions. In the first series, the magnitude of the perturbation is fixed to be the average element edge length, h, and additional vertices are perturbed in each successive mesh in the series. In the second series of meshes, the number of vertices that are perturbed is constant, but the amount of the perturbation is increased in each successive mesh.

In Tables 1 and 2, we present untangling and improvement results for the two- and three- dimensional mesh series, respectively. In these tables, as in [11], P is the percentage of perturbed vertices, D is the distance they are perturbed, N is the resulting number of invalid elements, S_U is the number of sweeps required to untangle the mesh, $Min.\theta_U$ is the minimum (dihedral) angle in the mesh following untangling, and $Min.\theta_S$ is the minimum dihedral angle after mesh smoothing.

Table 1: Untangling results for a series of 2D quadrilateral meshes

			Untangling				Smoothing		
P	D	N	S_U	Total Time (s)	Time/ Call (s)	Min. θ_U	Total Time (s)	Time/ Call (s)	Min. θ_S
5	h	9	1	1.37e-01	1.75e-05	4.75e+01	1.64e-01	6.99e-05	8.63e+01
10	h	20	1	1.27e-02	1.61e-05	4.89e+01	1.71e-01	7.26e-05	8.43e+01
25	h	65	2	2.61e-02	1.66e-05	7.47e+01	1.81e-01	7.69e-05	8.49e+01
50	h	163	1	2.57e-02	3.28e-05	1.62e+01	1.68e-01	7.16e-05	7.79e+01
25	h	65	2	2.61e-02	1.66e-05	7.47e+01	1.81e-01	7.69e-05	8.53e+01
25	$2h$	276	3	6.23e-02	2.65e-05	6.89e+01	1.68e-01	7.15e-05	7.75e+01
25	$4h$	348	3	1.00e-01	4.26e-05	1.81e+01	1.64e-01	6.95e-05	5.95e+01
25	$8h$	385	5	1.78e-01	4.54e-05	2.12e+01	1.68e-01	7.13e-05	5.39e+01

These results clearly show that the amount a vertex is perturbed significantly increases the number of untangling passes required to create a valid mesh and decreases the effectiveness of three passes of mesh smoothing. We note that 3 passes of mesh smoothing were used, a result of about $\theta_S = 78.8^o$ was achieved for the last three-dimensional test case. The cost of mesh untangling per vertex is approximately half the

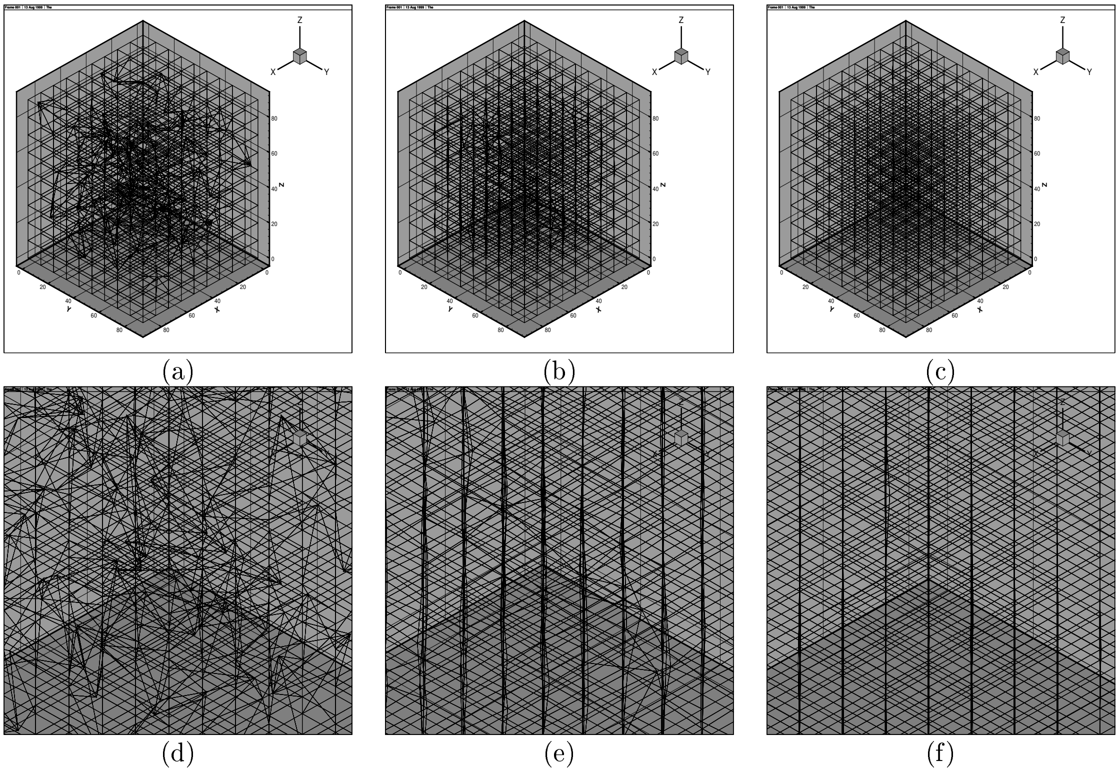


Figure 10: Typical results for hexahedral mesh untangling and smoothing using the linear programming approach. (a) is the mesh by perturbing 10 percent of the vertices by h; (b) the mesh after the untangling; (c) the mesh after 3 round of smoothing; (d) the zoom of the center of the initial mesh; (e) the zoom of the center of the untangled mesh; and (f) the zoom of the center of the smoothed mesh.

cost of mesh smoothing for three dimensions, and the ratio is about one third for two dimensions. In three dimensions, the mesh untangling cost and the mesh smoothing cost are significantly more than they are in two dimensions. This difference occurs due to an increase in the average number of iterations required to solve the linear program, and an increase in the average cost per iteration.

We also experimented different quality metrics such as to maximize the 2-norm of all angles in the local submesh, in stead of maximizing the minimum angle. In other words, we try to maximize the following function:

$$f(x) = \sqrt{\sum_{i=1}^{m} \theta_i(x)^2}.$$

where m is the number of angles in the local submesh, $\theta_i(x)$ is the ith angle. Because we do not need to solve the quadratic programming to get the direction of the search steps. Each round of smoothing will cost less time than that by maximizing the minimal angle of the local submesh. But, on the other hand, it turns out need more rounds to smooth the mesh, because the optimal solution of this function does not imply that the minimal angle of the local submesh is maximized or at least is sufficiently large.

Notice that to compute an angle is expensive than to compute the cos or sin of the angle. So we would prefer the smoothing by maximizing the cos of the minimal angle θ other than by maximizing the minimum angle θ. But we found that the improvement of the angle condition by maximizing the minimal angle is better than that by maximizing the $cos(\theta)$. It is much significant than triangle or tetrahedra mesh. The reason is that the slope of $cos(\theta)$ near 90^o is very small. Hence, maximizing the minimum angle based

Table 2: Untangling results for a series of 3D hexahedral meshes

			Untangling				Smoothing		
P	D	N	S_U	Total Time (s)	Time/ Call (s)	Min. θ_U	Total Time (s)	Time/ Call (s)	Min. θ_S
5	h	10	1	2.22e-01	4.33e-04	8.09e+01	8.37e-01	5.45e-04	8.75e+01
10	h	28	1	4.34e-02	8.48e-05	6.46e+01	8.31e-01	5.41e-04	8.72e+01
25	h	119	2	1.29e-01	1.26e-04	6.86e+01	7.96e-01	5.18e-04	8.37e+01
50	h	318	2	2.90e-01	2.83e-04	4.13e+01	8.01e-01	5.21e-04	8.17e+01
25	h	119	2	1.29e-01	1.26e-04	6.86e+01	7.96e-01	5.18e-04	8.37e+01
25	$2h$	395	2	2.73e-01	2.67e-04	2.40e+01	7.93e-01	5.16e-04	8.27e+01
25	$4h$	479	4	4.14e-01	2.02e-04	7.59e+01	8.44e-01	5.49e-04	8.23e+01
25	$8h$	501	4	4.48e-01	2.19e-04	4.79e+01	8.28e-01	5.39e-04	7.88e+01

smoothing is recommended.

5 Conclusions

Optimization-Based approach for simplex mesh untangling and smoothing had been studied by [11]. In this paper we presented an optimization-based approach to quadrilateral and hexahedral mesh untangling that uses linear programming techniques to maximize the minimum area or volume of affected simplex in a local submesh. As with optimization-based smoothing techniques, this method should be used in combination with less expensive methods for mesh smoothing. In particular, our experience shows that Laplacian smoothing is often effective even though it offers no guarantee of mesh improvement. Thus, high-quality, valid meshes can be achieved at low computational cost by combining Laplacian smoothing, or its variants, with a strategic use of optimization-based techniques. Finally, we proved that the linear programming approach is guaranteed to converge for local submeshes, but the global convergence of both optimization-based mesh improvement and optimization-based mesh untangling is still an open question for general domains and mixed or hexahedral element types.

6 Acknowledgments

We acknowledge the MCS division at Argonne National Laboratory for providing both a stimulating work environment for the second author's Summer Internship visit.

References

[1] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. In *8th ACM-SIAM Symp. on Discrete Algorithms*, pages 528–537, 1997.

[2] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM J. Numer. Anal.*, 13(2):214–226, 1976.

[3] R. E. Bank and R. K. Smith. Mesh smoothing using a posteriori error estimates. *SIAM Journal on Numerical Analysis*, 34(3):979–997, June 1997.

[4] Scott Canann, Michael Stephenson, and Ted Blacker. Optismoothing: An optimization-driven approach to mesh smoothing. *Finite Elements in Analysis and Design*, 13:185–190, 1993.

[5] David A. Field. Laplacian smoothing and delaunay triangulations. *Communications and Applied Numerical Methods*, 4:709–712, 1988.

[6] R. Fletcher. *Practical Methods of Optimization (Second Edition)*. John Wiley and Sons, 1987.

[7] Lori Freitag. On combining laplacian and optimization-based smoothing techniques. In *Trends in Unstructured Mesh Generation*, volume 220, pages 37–44. ASME Applied Mechanics Division, 1997.

[8] Lori Freitag and Carl Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal of Numerical Methods in Engineering*, 40:3979–4002, 1997.

[9] Lori Freitag and Carl Ollivier-Gooch. A cost/benefit analysis of simplicial mesh improvement as measured by solution efficiency. Technical Report ANL/MCS-P722-0598, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1998. To appear in International Journal of Computational Geometry.

[10] Lori A. Freitag, Mark T. Jones, and Paul E. Plassmann. An efficient parallel algorithm for mesh smoothing. In *the Fourth International Meshing Roundtable*, pages 47–58. Sandia National Laboratories, 1995.

[11] Lori A. Freitag and Paul Plassmann. Local optimization-based simplicial mesh untangling and improvement. In *2nd Symposium on Trends in Unstructured Mesh Generation*, Boulder, Colorado, 1999.

[12] I. Fried. Condition of finite element matrices generated from nonuniform meshes. *AIAA Journal*, 10:219–221, 1972.

[13] Patrick Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities, part 1 - a framework for surface mesh optimization. Technical Report SAND 99-0110J, Sandia National Laboratories, 1999.

[14] Mark Shephard and Marcel Georges. Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering*, 32:709–749, 1991.

[15] Matthew L. Staten, Scott A. Canann, and Joseph R. Tristano. An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. In *the 7th International Meshing Roundtable*, pages 479–494. Sandia National Laboratories, 1998.