An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes

Scott A. Canann, Joseph R. Tristano, and Matthew L. Staten
ANSYS, Inc.
275 Technology Drive
Canonsburg, PA 15317
{scott.canann, joe.tristano, matt.staten}@ansys.com

Abstract. Automatic finite element mesh generation techniques have become commonly used tools for the analysis of complex, real-world models. All of these methods can, however, create distorted and even unusable elements. Fortunately, several techniques exist which can take an existing mesh and improve its quality. Smoothing (also referred to as mesh relaxation) is one such method, which repositions nodal locations, so as to minimize element distortion. In this paper, an overall mesh smoothing scheme is presented for meshes consisting of triangular, quadrilateral, or mixed triangular and quadrilateral elements. This paper describes an efficient and robust combination of constrained Laplacian smoothing together with an optimization-based smoothing algorithm. The smoothing algorithms have been implemented in ANSYS® and performance times are presented along with several example models.

Keywords. Smoothing, Laplacian smoothing, optimization-based smoothing, triangular, quadrilateral, quad-dominant

1. Introduction

1.1 Importance of work

As mesh generation becomes more automated and as mesh sizes increase, the need to create meshes completely free of unusable or geometrically distorted elements increases. That is, for isotropic solutions, geometrically distorted elements give poor solutions and/or ill-conditioned matrices. For large meshes, visually checking the quality of a mesh can be difficult at best, not to mention error-prone. In order to minimize the user time required to validate that the mesh is acceptable, a priori distortion metrics and automatic correction procedures are needed.

Geometrically distorted elements can be caused by many things, including:

- Model size -- coarse meshes, by their very nature, can result in geometrically distorted elements;
- Tough topological and geometrical configurations common in real-world CAD geometries, including sliver areas, sharp corners, small areas, small holes, thin sections, and areas of high curvature; and
- Algorithmic flaws in the meshing algorithm.

A variety of mesh improvement techniques has been developed to improve the quality of meshes created by automatic techniques. Some of the existing techniques for improving the quality of an existing surface mesh include:

- Topological and quality-based operators – including:
  - Node insertion or local refinement techniques [4,38,46,56,57,70];
  - Edge/face swaps [2,15,16,17,22,26,37,38,44 ,70,73]; and
  - Node removal or element deletion [9-11,15-17,44,70,73].

- Smoothing – modifying node placement so as to improve the elements shape without modifying the mesh connectivity. This can be done for:
  - Corner nodes [1,5,7,14,21,23,25,27,29,30,32,34,35,39-41,55,72]; or
  - Midnodes [62-65] – where the corner nodes of quadratic elements are held fixed and the mid-nodes are repositioned, so as to improve the element's quality.

While ANSYS uses a combination of the techniques listed above to improve the quality of a mesh, this paper will only cover the implementation of the smoothing portion of the overall mesh improvement process.

Since smoothing is independent of the mesh generation technique used to create a mesh, the ideas discussed in this paper can be used to compliment any mesh generation technique.

In addition to being used as an improvement step after creating a mesh, smoothing can also be used to:

- Smooth local collections of elements between steps in an advancing front meshing approach [9-11,52];
- Locally improve elements after each quality-based mesh cleanup operation [2,9-11,26,37,70,73], such as node insertions, deletions and diagonal swaps;
- Improve a mesh after local mesh refinement [4,38,70]; and
- Aid in shape optimization [42].

1.2 Previous work

A significant amount of work has been done in the area of mesh smoothing. This section will present an overview of some of that work, including in the areas of Laplacian smoothing, optimization-based smoothing, and physics-based smoothing. In addition to the overview of smoothing given here, another overview can be found in George [31,32].

1.2.1 Laplacian Smoothing

Laplacian smoothing is by far the most common smoothing technique. Laplacian smoothing, in its simplest form, consists of recursively placing each node at the average of the nodes connected to it. This technique generally works quite well for meshes in convex regions. However, it can result in distorted or even inverted elements near concavities in the model. Many researchers have used and extended the capabilities of Laplacian smoothing [9,10,11,13,18,23,30,32,34,35,39,40]

Some variations on the basic Laplacian technique, include:

- Weighting the contribution of each neighboring node in the averaging function [9,10,11,32,40] by edge length, element area, or other similar criteria;
- Constraining the node movement, so as to avoid the creation of inverted elements [18,23];
- Developing methods to extend it to anisotropic (stretched) meshes [13,34]; and
- Generating a means of combining it with the similar *isoparametric* smoothing technique [35].

1.2.2 Optimization-based Smoothing

A newer form of smoothing, that is receiving more attention lately, is optimization-based smoothing. Instead of moving nodes based on a heuristic algorithm, as is done in Laplacian smoothing, the nodes are moved so as to minimize a given distortion metric. Some of the developments in this area include [1,5,7,14,21,25,27-29,36,54,55]. While optimization-based smoothing is more expensive than Laplacian smoothing, it gives better results – especially near concave regions in the geometry. Several authors [21,27] (including this paper) have implemented schemes that use Laplacian smoothing when possible and only use optimization-based smoothing when necessary.

One of the first optimization-based smoothing algorithms was developed by de Cougny [21]. His technique was designed to improve the distorted tetrahedral elements that can be created near the boundary of a tet mesh generated using an octree technique [66]. An element distortion metric is presented that is basically the scaled ratio of an element's volume to it's face areas, for which he proves several properties, which indicate that the metric may be well-suited for optimization-based smoothing. He then finds the ideal location for the node to minimize the maximum distortion metric and does a search along the line from the current location to the potentially optimal location.

Parthasarathy [54] developed an optimization-based technique for triangular and tetrahedral meshes -- again created to repair elements created by quadtree and octree mesh generators. He solves a nonlinear, constrained, global optimization problem, using the element's aspect ratio as the objective function to be minimized. He also adds inequality constraints to ensure that none of the elements' surface area (or volume in 3D) drops below a certain threshold. He uses a modified version of the feasible directions algorithm to drive the optimization. Due to the fact that a global optimization was done, Parthasarathy found that as much time was spent in the smoothing process as was spent in the mesh creation phase. Encouraging results were obtained from the smoother.

Canann [14] developed a global optimization method that uses Oddy's distortion metric [51]. Although it was developed mainly for hex meshes, it can be easily extended to any other element type. This method, like that developed by Parthasarathy, is impractical for large meshes, because it uses global optimization. Recursive local optimization has proven to be more feasible. In addition, it was later determined that Oddy's metric was too lenient for angle distortion and too restrictive on aspect ratio distortion.

An approach developed by Freitag [7,25,27-29] works to maximize the minimum angle in triangular or tetrahedral meshes. Since this is a non-continuous function (i.e., the function does not have continuous derivatives defined everywhere), a *nonsmooth optimization* (NSO) is done, using an analogue of the steepest descent method for smooth functions. This approach has been shown to be parallelizable [25], gives good results, and is reasonably efficient. Freitag has also experimented with combining the approach with Laplacian smoothing in [27].

Amenta [1] presents theoretical results showing how some local triangle and tetrahedral shape optimizations can be solved in linear time using generalized linear programming. For the mesh smoothing problems that don't fit into that class of problems, other efficient algorithms are presented. Many distortion metrics are discussed and various optimization techniques are compared.

Other optimization-based methods include ones developed by Riccius [55], Bank [5], and Jacquotte [36]. An optimization-based smoothing algorithm, specifically designed for adaptively improving finite element triangulations by making use of a posteriori error estimates is presented by Bank [5]. Jacquotte [36] developed a distortion metric and optimization-based smoothing approach suitable for 2D & 3D structured grids.

In order to develop a robust optimization-based smoothing algorithm, much thought must go into selecting a good distortion metric. Among the many a priori distortion metrics that have been developed, a representative set can be found in [3,6,8,20,23,36,43,44,45,47-51,54,58-65,74].

1.2.3 Physics-based smoothing

Since a well-graded mesh can be similar to objects found in nature, several authors have developed techniques for smoothing that are based on solving simple physics problems. Lohner et al. [50] developed a smoothing algorithm that views the mesh as a system of springs between nodes, where the force between nodes is a ratio between the actual and desired grid sizes. His technique produces stretched elements based on density gradients. Shimada [67-69] and Bossen [12] developed mesh generation techniques that created the mesh, smoothed it, and cleaned it up, all as a part of a physics-based approach. Shimada viewed the nodes as being the center of bubbles, with each bubble interacting with each of the others. Bossen developed what he called a *pliant* method, which like Shimada's approach uses attraction and repulsion between nodes to determine nodal locations. Neither of these approaches is a true smoothing technique though, because both require node insertions or deletions when the mesh is not graded smoothly enough. Each of these physic-based techniques is capable of retaining stretched elements in an anisotropic mesh.

1.3 Paper Overview

In this work, a complete scheme is presented for smoothing 2D and 3D surface meshes. Some of the advantages include:

- Robustness – improves the mesh and never inverts elements;
- Efficiency – combines constrained Laplacian with selective optimization-based smoothing;
- Ability to repair inverted elements; and
- Ability to smooth 2D and 3D meshes comprised of:
    - triangles only [12,18,20,30-32,48,67-69];
    - quadrilaterals only [9-11,19,53,73];
    - mixed triangles and quadrilaterals [18,30,44,49,52];

Section 2 lists the requirements for the smoother. Section 3 lays out the overall mesh smoothing algorithm. Section 4 discusses the details of the constrained Laplacian smoothing implementation. Section 5 describes the basic optimization-based smoothing algorithm. Section 6 presents the distortion metrics used for both the constrained Laplacian smoothing and the optimization-based smoothing. Section 7 shows some speed and quality results, including a few sample meshes using the presented smoothing scheme. A conclusion is then given in Section 8.

2. Smoothing Requirements

The smoothing algorithms described in this paper were designed for use in the ANSYS® software package. Some of the algorithmic requirements for the smoother include that it:

- Work for triangular, quadrilateral, and quad-dominant meshes, created by any of several mesh generation and refinement techniques;
- Handle severely distorted elements;
- That it be efficient and robust; and
- Be tested on a wide range of models and across several types of physics.

ANSYS creates triangular, all-quadrilateral, and quad-dominant meshes and the smoother must be able to successfully smooth these meshes, without giving undue preference to one element shape over another. Smoothing must also be developed independently from any mesh creation technique, so that is can be used by ANSYS' various triangular mesh generators [18,44,71,75], quad-dominant mesh generators [18,51,75], and mesh refinement capabilities [70,76].

Cases arise where it is necessary to improve severely distorted or even mildly inverted elements with the smoother. This distortion may have been caused by the meshing algorithm, mesh refinement, or a mesh cleanup operation. An example of a case where this type of distortion can be routinely created is when performing an element open (a node insertion) in a quadrilateral mesh as a part of a topological cleanup process [15,16,70]. Sometimes the insertion of such an element can create a very geometrically distorted quad, as shown in Figure 1, which the smoother must then fix.

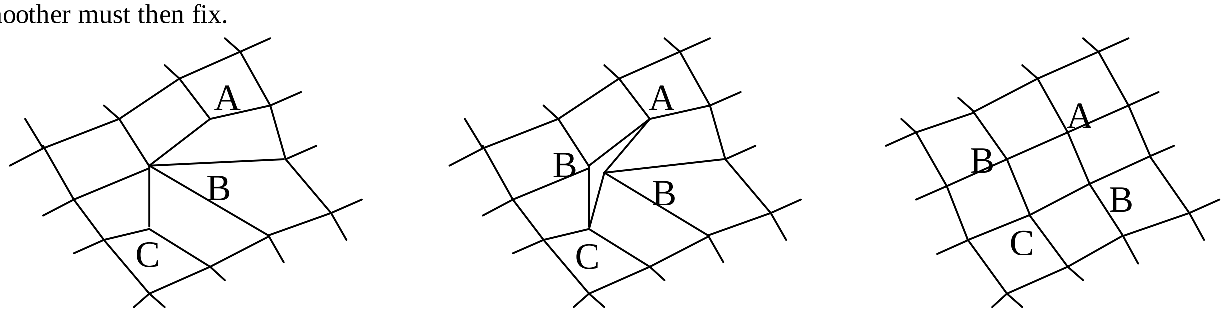

Figure 1a. Before element open   Figure 1b. After element open   Figure 1c. After smooth

3. **Overall Smoothing Algorithm**

In this section, the overall smoothing scheme is presented. Algorithmic details for each of the major steps in this scheme will be presented later in the paper.

- Compute initial distortion metrics for all elements.
- If mesh is a layer mesh (see section 4.3), mark the nodes associated with the layer boundary
- Based on model size, compute distance, δ, for numerical gradient computation in optimization based smoothing (see section 5.1)
- Set number of iterations (*niter)* to 1
- REPEAT (MAIN SMOOTHING LOOP):

    - FOR EACH NODE *v* DO:

        1. If node *v* is not movable or has been deactivated from the smooth, then:
            - Break from loop **--** go on to the next node in the list

        2. If node *v* wasn't moved by optimization-based smoothing in a previous iteration, then do **Constrained Laplacian Smoothing:**
            a) If node *v* is on layer boundary (see section 4.3):
                - Move node *v* using Laplacian smoothing
                - Project to layer boundary

                Else:
                - Perform a constrained Laplacian smooth – see section 4.1 and 4.2

            b) If distance moved is less than the move tolerance:
                - Keep the old position for the node
                - Remove node *v* from the list of active smoothable nodes

                Else allow the move:
                - Put the node's neighbors back in the list of smoothable nodes, if they aren't already there
                - Compare to the largest distance moved by a node and keep track of the largest
                - Update the adjacent elements' distortion metric values

        3. If *niter* $\geq$ 2, then invoke **Optimization-based smoothing**. That is, let the Laplacian smoothing have 2 iterations to improve the mesh, before invoking the more expensive optimization-based smoothing):
            a) Find the minimum distortion metric from all facets connected to the node
            b) If the minimum metric <= a given tolerance (e.g., 0.1)
                - Attempt optimization-based smoothing – See section 5 for details
            c) If optimization-based smoothing moved the node, then:
                - Put the node's neighbors back into the smoothable list, if not already there
                - Compare to the largest distance moved by a node and keep track of the largest
                - Update the adjacent elements' distortion metric values

    - END DO (end of loop through nodes)
    - niter = niter + 1

- UNTIL there are no more nodes that have moved enough to warrant another iteration. That is, if NO nodes have moved OR if the maximum distance moved is less than (1.75 * move tolerance). (END OF MAIN SMOOTHING LOOP)

4. Laplacian Smoothing

Basic Laplacian smoothing iteratively repositions each node to be at the average of each of the nodes connected to it. This works extremely well for convex regions. However, it can invert elements near concavities. In order to avoid this, constraints can be placed on the smoother, so that it doesn't do more damage than good. In this section a description of a constrained Laplacian smoothing implementation is given.

4.1 Constrained Laplacian Smoothing Algorithm

The algorithm used for each node entails:

- Computing the location where Laplacian smoothing would place the node & projected new nodal location to its corresponding curve or area
- Loop (repeat up to the arbitrarily selected value of 20 times):
  - Compute distortion metric for neighboring elements
  - If the new location is "acceptable" (see Section 4.2 below), then break out of the loop
  - If the new location is not "acceptable", then cut the proposed move distance in half and set this as the new location

4.2 Acceptance Criteria for Constrained Laplacian Node Movement

Before determining if a new location is acceptable, the following quantities are computed for each element connected to the node:

- μ (i.e., α or β) – Distortion metrics for the elements connected to the node before and after the proposed node movement. (Note that for efficiency, the elements' distortion metrics from the before the move can be stored with each element).
- N+ – Number of elements whose metric improves
- N- – Number of elements whose metric gets worse
- $\Delta\mu = \sum\{\Delta\mu_i\} / N$ – Average metric change computed as the sum of distortion metrics changes at the connected elements divided by the number of connected elements
- N$\uparrow$ – Number of elements whose metric improves significantly:
  - Metric goes from negative to positive
  - Metric becomes less negative
  - Metric moves from below acceptable to acceptable quality (a value of $\mu_{min} = 0.05$ is reasonable for the minimal acceptable quality limit).
- N$\downarrow$ – Number of elements whose metric worsens significantly:
  - Metric goes from positive to negative
  - Metric becomes more negative
  - Metric drops below acceptable levels (even if it stays positive).
- N$\downarrow\downarrow$ – Number of inverted elements created. This is the same as N$\downarrow$ for triangles. However, a quad can go negative without becoming inverted. That is, the metric at a corner node goes negative as a corner angle goes past 180 degrees, but inversion only occurs when a corner angle drops below 0 degrees.
- θ **--** The largest angle spanned between any two nodes in the element.

Node movement is ruled out if *any* of the following are true:

- (N- = N) – All of the elements get worse; or
- (N$\downarrow\downarrow$ > 0) – Any of the elements go inverted; or
- (N$\downarrow$ > N$\uparrow$) – More elements get significantly worse than get significantly better;
- ($\Delta\mu < -\mu_{min}$) – The average change in the metric is poor; or
- ($\theta > \theta_{max}$) – The element spans too great of an angle (θ_{max} = maximum angle any element can span).

If node movement is not ruled out by the checks about, then the node movement is considered to be acceptable if *any* of the following are true:

- ($N+ = N$) – All of the elements have improved; or
- ($N\uparrow > 0$) & ($N\downarrow = 0$) – No elements get significantly worse and some get significantly better; or
- ($N\uparrow \geq N\downarrow$) & ($\Delta\mu > -\mu_{min}$) – More elements get significantly better than get significantly worse and the average improvement is reasonable.

4.3 Boundary Layer Smoothing

ANSYS provides the ability to create 2D free meshes where the user can request a specified number of rows of very small elements next to a geometric edge and then transition quickly to a much larger element size [53, 75]. This type of mesh is particularly useful for CFD boundary layers or EMAG skin effect studies. See Figure 2 below.

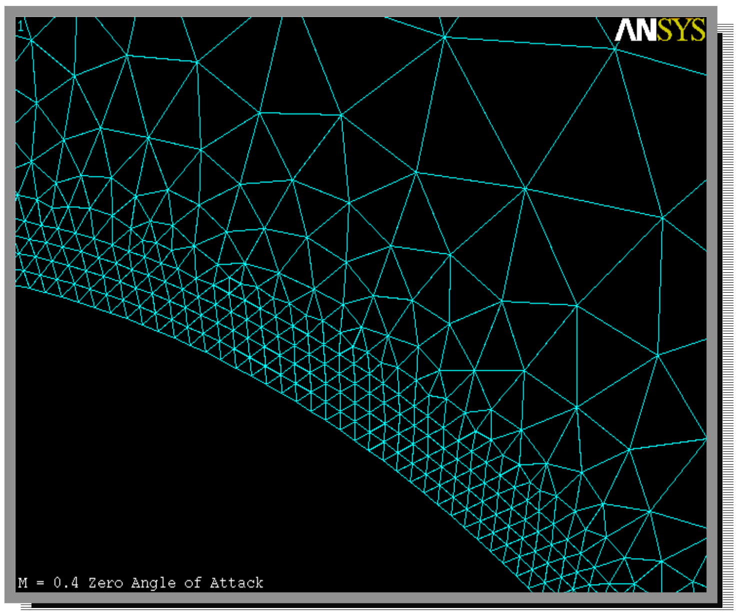

Figure 2. Layer mesh, with four rows of equally sized elements in the boundary layer.

In order to keep the smoother from stretching the elements in the boundary layer out into the rest of the mesh (thereby causing the transition to start at the boundary); several modifications must be added to the smoothing process. Before the smoothing iterations begin, the nodes in the boundary layer that are farthest from the geometric edge are marked. During the smoothing process, these nodes are constrained to remain the same constant boundary layer distance from the edge. This is done by allowing each outside layer node to be smoothed by the Laplacian smoother, projecting it to the geometric edge, and then moving it perpendicularly away from the geometric edge a distance equal to the boundary layer thickness.

5. Optimization-based Smoothing

In some cases – especially near concavities in the model – the constrained Laplacian smoother is unable to improve severely distorted elements. It is often critical that an adequate smoother be invoked to repair these elements, because they will often occur near to the boundary of the mesh – where element shape is typically the most critical.

Optimization-based smoothing directly attacks the element distortion problem. Instead of applying a heuristic-based movement to each node as is done in Laplacian smoothing, optimization-based smoothing seeks to minimize the distortion of the elements connected to each node. Existing optimization-based smoothing techniques [1,5,7,14,21,25,27-29,36,54,55] are relatively new and vary based on:

- The type of mesh being smoothed (structured or unstructured);
- The element shape (triangle, quadrilateral, etc.);
- The optimization and search technique chosen; and
- The distortion metric selected (see Section 6 below).

5.1 Optimization-based Smoothing Algorithm

In this section, the algorithmic details of a new optimization-based smoothing technique are presented. While only 2D elements (triangles and quadrilaterals) are covered in this paper, the techniques described here can be (and have been successfully) extended to 3D volume elements. The approach presented here is similar to the Freitag's approach [25], but adds two advantages over her approach. The first advantage that this approach has over Freitag's is that it can repair meshes with elements that are severely distorted or even inverted. Freitag's approach requires that none of the elements be severely distorted or inverted. In other words, the distortion metrics used are continuous through element distortion and inversion. The other advantage to this approach over Freitag's approach is that her method has only been implemented for simplicial elements (triangles and tetrahedron). The approach presented here works for triangular, quadrilateral, and mixed-meshes.

Let μ be a distortion metric, with values generally varying between -1 and 1. For example, μ could be the triangle metric, α, or the quadrilateral metric, β presented in Section 6 below. A value of 1 for μ corresponds to a perfectly shaped element, while a value of 0 corresponds to a element that is basically unusable[1]. Negative values correspond to extremely distorted elements, including negatively oriented or inverted elements.

Let $\mathbf{x}$ be a node in the mesh, connected to elements $E_1, E_2,\ldots,E_n$, with (positive) metric values $\mu_1, \mu_2,\ldots, \mu_n$, respectively. In an optimization-based smoothing procedure, the goal is to move $\mathbf{x}$ so that $\mu_{min} = \min(\mu_1,\mu_2,\ldots,\mu_n)$ is increased as much as possible, i.e., the quality of the elements incident on $\mathbf{x}$ is optimized. If $\mathbf{x}$ lies on a boundary edge (possibly curved), then the movement of $\mathbf{x}$ is constrained to lie on that edge. Since optimization-based smoothing is more time-consuming than Laplacian smoothing, it will typically only be invoked for nodes whose value of μ_{min} is sufficiently small.

A steepest descent iterative approach is used to move $\mathbf{x}$ in gradient directions so as to increase μ_{min} as much as possible. Usually, the improvement in μ_{min} decreases with each iteration, so two iterations are usually sufficient in practice. The following is done in an iteration of the steepest descent method:

1. Estimate the gradient vector $\mathbf{g}_i = (g_x^i, g_y^i, g_z^i)$ for each element E_i, $1 \le i \le n$, as follows:
   - Perturb the x-coordinate of $\mathbf{x}$ by δ. The size of δ can be based on the size of vertex coordinates, e.g., 10^{-5} times the maximum model dimension.
   - Compute the perturbed metric value μ_i^+ for E_i.
   - Set $g_x^i = (\mu_i^+ - \mu_i)/\delta$. ($g_y^i$ and g_z^i can be computed similarly.)

2. The gradient direction is taken to be $\mathbf{g} = \mathbf{g}_m$, where m is an index such that $\mu_m = \mu_{min}$. If $\mathbf{g} \approx 0$, then let $\mathbf{g} = \mathbf{g}_{m^+}$, where m^+ is an index, such that $\mu_{m^+} = \mu_{min}^+$, where μ_{min}^+ is the next smallest element metric to μ_{min}. Vertex $\mathbf{x}$ is to be moved to $\mathbf{x}^+ = \mathbf{x} + \gamma\mathbf{g}$, where the scalar γ is determined as follows:

   - If μ_i is considered to be a function of $\mathbf{x}$, then the Taylor series expansion yields:

$$\mu_i(\mathbf{x}^+) = \mu_i(\mathbf{x}+\gamma\mathbf{g}) \approx \mu_i(\mathbf{x}) + \gamma\,\mathbf{g}\cdot\mathbf{g}_i \qquad [1]$$

[1] For example, this could be a zero area triangle or a quad with a 180 degree corner angle.

If $\mathbf{g} \cdot \mathbf{g}_i \geq 0$, then μ_i *increases* for a sufficiently small γ. That is, the chosen gradient direction will most likely improve the quality for E_i and we probably won't have to worry about μ_i^+ being less than μ_{min}^+.

- On the other hand, if $\mathbf{g} \cdot \mathbf{g}_i < 0$, then μ_i *decreases* for a sufficiently small γ in the chosen gradient direction. In this case, the value of γ needs to be restricted, so that $\mu_{min}^+ \geq \mu_i^+$ -- or:

$$\mu_{min} + \gamma\, \mathbf{g} \cdot \mathbf{g} \geq \mu_i + \gamma\, \mathbf{g} \cdot \mathbf{g}_i \qquad [2]$$

Therefore, γ is limited to be:

$$\gamma = \min\left\{ \frac{(\mu_i - \mu_{min})}{(\mathbf{g} \cdot \mathbf{g} - \mathbf{g} \cdot \mathbf{g}_i)} \right\}, \text{ over the indices, } i, \text{ for which } \mathbf{g} \cdot \mathbf{g}_i < 0 \qquad [3]$$

3. Once the value for γ is obtained and it is sufficiently large, then a move of $\mathbf{x}$ to $\mathbf{x}^+ = \mathbf{x} + \gamma\mathbf{g}$ is attempted as follows:

- Using the new $\mathbf{x}^+$, the new metric values μ_i^+ are computed for each E_i.

- The new minimum metric value, μ_{min}^+, is computed as:

$$\mu_{min}^+ = \min\,(\mu_1^+, \mu_2^+, \ldots, \mu_n^+) \qquad [4]$$

- The move of $\mathbf{x}$ to $\mathbf{x}^+$ is accepted and μ_i is set to μ_i^+ for each i, if:

$$\mu_{min}^+ \geq \mu_{min} + \text{tol} \qquad [5]$$

Where "tol" is a tolerance, e.g., 10^{-5}.

- Since approximations are used, it is possible that the original γ is too large, so if the move is not accepted, then a smaller movement can be tried. The value γ is decreased by a factor of 2, and the above is repeated for the new $\mathbf{x}^+$. At most a fixed number, say 4, of γ decreases is attempted before giving up on moving $\mathbf{x}$.

6. Distortion Metrics

As explained in the previous sections, distortion metrics can be used to do more than just determine the final quality of a mesh. They can also be used to guide smoothing operations by constraining the movement during constrained Laplacian smoothing and to drive the optimization-based smoothing. The selection of a suitable metric is critical. Some of the criteria by which a metric can be judged include:

- Efficiency – it will be called many times
- Suitability for use by an optimization-based smoothing algorithm, i.e.:
  - Continuous, since derivatives are needed to determine the direction which minimizes the metric.
  - Precise in its description of what element distortion is. Nothing can find a flaw in a metric faster than optimization-based smoothing.
  - Preferably, it will remain defined through inversion of the element, so that mildly inverted elements can be repaired.
- Application to mixed meshes. It is advantageous for the metric to be defined and normalized in such a way that the metrics for each element type can be used together. That is, the quad and tri distortion

metrics should be compatible, so that smoothing of quad-dominant meshes does not over-emphasize the quality of one element shape over another at a given node.

The following sections will describe the distortion metrics that have been developed for quantifying triangle and quadrilateral element distortion.

6.1 Triangular Distortion Metric

The distortion metric used for triangle elements is an extension of the one presented by Lee and Lo in [44]:

$$\alpha(ABC) = (I)2\sqrt{3}\frac{\|CA \times CB\|}{\left[\|CA\|^2 + \|AB\|^2 + \|BC\|^2\right]}$$

$$where: \qquad [6]$$

$$I = \begin{Bmatrix} 1, (CA \times CB)\cdot N_x > 0 \\ -1, (CA \times CB)\cdot N_s < 0 \end{Bmatrix}$$

Where N is the surface normal evaluated at the center of the triangle; and AB, CB, and CA are the edge lengths between those nodes. Note that the metric is signed (with "I") to capture inversion of the element. This distortion metric is basically the area of the triangle divided by the sums of the squares of the lengths of the sides. The $2\sqrt{3}$ term is a normalizing factor so that equilateral triangles will have a maximum α value of 1.

The only change between this metric and Lee's metric, is the inclusion of the "*I*" term, which assigns a negative distortion value to inverted elements. This is important for helping to determine if the proposed Laplacian smoothing node movement will indeed improve the quality of the local mesh. Optimization-based smoothing also gains form a distortion metric that continuously quantifies element distortion as an element goes inverted. This modification makes it possible for the smoother to repair mildly inverted elements.

6.2 Quadrilateral Distortion Metric

The metric used for the quadrilaterals is also based on the metric presented by Lee and Lo in [44], again with a few modifications. The distortion metric, β, is computed by dividing the quadrilateral into triangles along each of its diagonals. The distortion metric, α, (described above) is computed for each of the four resulting triangles and is then sorted in descending order of magnitude. That is:

$$\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \alpha_4 \qquad [7]$$

Lee [44] uses the following as his metric:

$$\beta_{Lee} = \left(\frac{\alpha_3\alpha_4}{\alpha_1\alpha_2}\right) \qquad [8]$$

Instead of Lee's metric, the distortion metric used here is:

$$\beta = \{\min(\alpha_1, \alpha_2, \alpha_3, \alpha_4)\} - negval \qquad [9]$$

Where *negval* = 1 if any of the angles corner angles of the quad are less than 6 degrees, any two of the nodes are coincident within a tolerance, or two of the triangles are inverted (i.e., their α_i's are negative). In addition *negval* is set to 2 or 3, if 3 or 4 α_i's are negative, respectively. This constant allows for a subjective qualification of whether a node movement really improves the quality of the mesh. The use of *negval* here is certainly heuristic, but it has been developed over several years of experiences in smoothing thousands of area meshes.

While this is not a continuous function, it has proven to be effective for both types of smoothing. For optimization-based smoothing, **g** is correct, but the correct value for γ may require additional iterations, when $\mu_{min} < 0$.

This metric is used, instead of equation 8 above, because negative metric values are now allowed and if two or four α's are negative (inverted), then the β value will be incorrectly signed. The new metric also emphasizes improving minimum metric values over average metric values and analyses are more affected by minimum metrics than by low average metric values. In addition, Lee's metric does not detect aspect ratio problems. His metric will give perfect distortion metric values to long, stretched rectangles.

7. Results and Examples

7.1 Smoothing improvement statistics

The mesh smoothing scheme presented in this paper has been exhaustively tested and has proven to be sufficiently robust and efficient for use in a commercial product. The following distortion metric improvement statistics are based on meshing 23 models at several different sizes, such that the smoother was called a total 124 times. The set is comprised of triangle, quadrilateral, and mixed meshes.

Table 1. Improvements to distortion metric values from smoothing

	Average value before smoothing	Average value after smoothing	Average increase
Average Metric	0.82	0.86	0.04
Minimum Metric	0.58	0.70	0.12
When minimum metric was	< 0.1	~0.55	0.48

In Table 1, the rows are listed in increasing order of importance. The minimum metric is the most important number to improve – especially when it drops below 0.1 (our arbitrary minimum quality allowed). One of the tests started with a minimum metric as low as -0.49, but ended up with a minimum metric of 0.12.

Of the 124 times that the smoother was called, the smoother was called the minimum metric got slightly worse 8 times (by a maximum of 0.06). There is no cause for alarm in these cases, however, because the worst minimum metric that any of these was only 0.42 (nowhere near the poor quality cutoff of 0.1) and the maximum that any got worse was 0.06. The reason for these minor variations in minimum metric are due to the way in which the acceptance criteria for the Laplacian smooths are set up.

7.2 Speed

The average amount of time spent in the smoother is less 30% of the total meshing time. Over 160 models were selected and meshed at several mesh sizes, resulting in over 850 calls to the smoother, so as to extract the following timing statistics.

Table 2. Timing statistics

1. Percent of total meshing time spent in the smoother	29
2. Percent of smoothing time spent computing distortion metrics	38
3. Percent of smoothing time performing Laplacian constraint	55
4. Percent of nodes requiring optimization-based smoothing	< 1.0

Note that in Table 2, the time spent on performing the Laplacian constraint is largely a superset of the time spent computing the distortion metrics. Therefore, the actual Laplacian constraint checks take less than 20% of the total smoothing time.

Since the vast majority of nodes do not require optimization-based smoothing, nearly all of the expense incurred for optimization-based smoothing is in checking to see if it is required. For most models, less than 1% of the nodes end up requiring optimization-based smoothing. Moreover, since the same metric is used for both types of smoothing, it does not have to be recomputed for the optimization-based smoothing checks.

7.3 Examples

Figures 3a and 3b below show a triangular mesh on a doubly curved surface both before and after smoothing. Note how the curvature is nicely maintained.

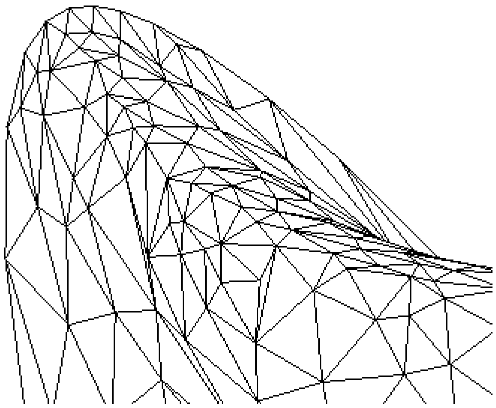

Figure 3a. Tri mesh before smoothing

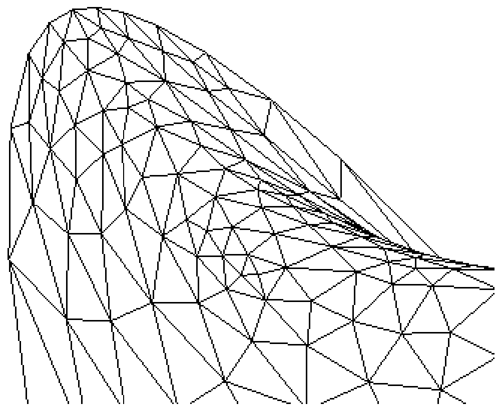

Figure 3b. Tri mesh after smoothing

Figures 4a and 4b below show another triangular mesh with and without smoothing. This tri mesh will be used as to create the quad mesh in Figure 5.

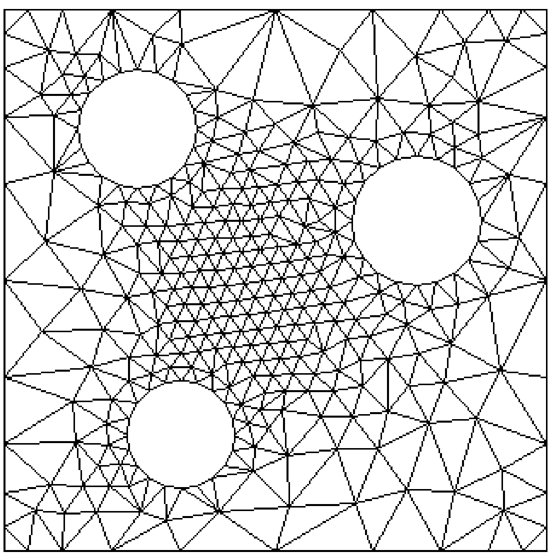

Figure 4a. Tri mesh before smoothing

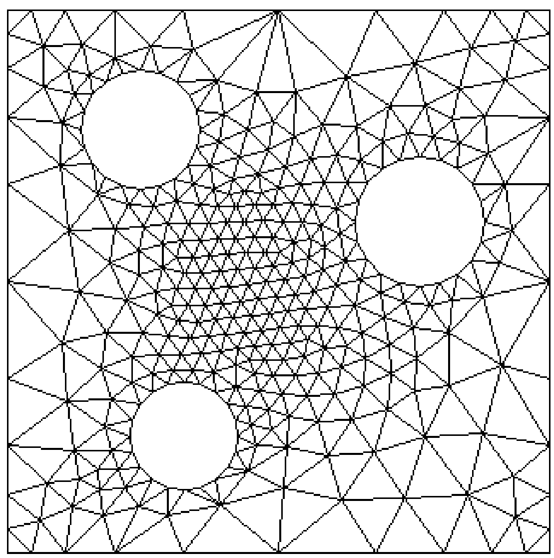

Figure 4b. Tri mesh after smoothing

Figures 5a and 5b show a quad mesh before and after smoothing. Note how severely distorted some of the quads are that must be handled by the smoother.

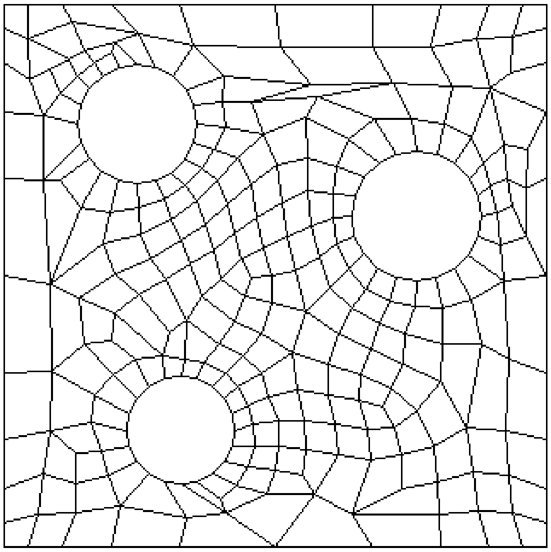

Figure 5a. Quad mesh before smoothing

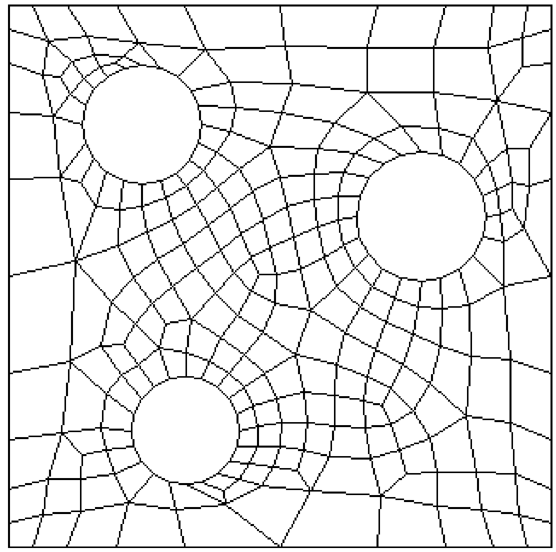

Figure 5b. Quad mesh after smoothing

Figures 6a and 6b show a simple quad mesh before and after smoothing.

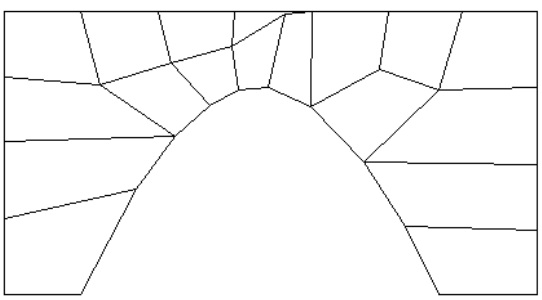

Figure 6a. Quad mesh before smoothing

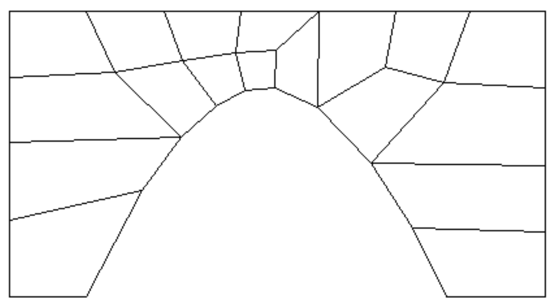

Figure 6b. Quad mesh after smoothing

8. Conclusion

An efficient and robust mesh smoothing scheme for 2D and 3D surface meshes is presented which works for tri, quad, and mixed meshes. This technique includes the combination of both constrained Laplacian and optimization-based smoothing. A distortion metric is presented that is well suited to both types of smoothing. The technique has been tested extensively and is in production use in ANSYS. The technique can (and has) been extended to 3D elements.

Acknowledgements

The authors wish to express gratitude to Steve Owen and Barry Joe for their help in the development and implementation of several of the ideas presented in this paper. We would also like to thank ANSYS, Inc. for its support of the research, development and publication of this work.

References

[1] Amenta, N., M. Bern, and D. Eppstein, “Optimal point placement for mesh smoothing,” In *8th ACM-SIAM Symposium on Discrete Algorithms*,” New Orleans, p. 528-537, 1997.

[2] Amezua, E., M.V. Hormaza, A. Hernandez, and M. B. G. Ajuria, “A method of the improvement of 3D solid finite element meshes,” *Advances in Engineering Software*, Vol. 22, p. 45-53, 1995.

[3] Babuska, I., and A.K. Aziz, “On the angle condition in the finite element method, *SIAM Journal on Numerical Analysis*, Vol. 13, p. 214-227, 1976.

[4] Bank, R.E., A.H. Sherman, and A. Weiser, “Refinement algorithms and data structures for regular local mesh refinement,” In R. Stepleman et al., editor, *Scientific Computing*, p. 3-17, IMACS/North-Holland Publishing Company, Amsterdam, 1983.

[5] Bank, R.E., and R.K. Smith, “Mesh smoothing using a posteriori error estimates,” *SIAM Journal on Numerical Analysis*, Vol. 34, No. 3, p. 979-997, 1997.

[6] Barlow, J., “More on optimal stress points-reduced integration, element distortions and error estimation,” *International Journal for Numerical Methods in Engineering*, Vol. 28, No. 7, p. 1487-1504, 1989.

[7] Batdorf, M., Freitag, L.A., and C. Ollivier-Gooch, “Computational study of the effect of unstructured mesh quality on solution efficiency,” *Presented at the 13th Annual Computational Fluid Dynamics Meeting*, Snowmass Village, CO, 1997.

[8] Berzins, M., “Solution-based mesh quality for triangular and tetrahedral meshes,” *Proceedings of the Sixth International Meshing Roundtable*, Sandia National Laboratories, p. 427-436, 1997.

[9] Blacker, T.D., M.B. Stephenson, and S.A. Canann, “Analysis automation with paving: a new quadrilateral meshing technique, *Proceedings of Design Productivity Institute and International Conference*, University of Missouri-Rolla, Honolulu, Feb. 1991.

[10] Blacker, T.D., and M.B. Stephenson, “Paving: A new approach to automated quadrilateral mesh generation,” *Sandia Report No. SAND90-0249*, Sandia National Laboratories, Albuquerque, N. M., 1990.

[11] Blacker, T.D., and M.B. Stephenson, “Paving: A new approach to automated quadrilateral mesh generation,” *International Journal for Numerical Methods in Engineering*, Vol. 32, p. 811-847, 1991.

[12] Bossen, F.J., and P.S. Heckbert, “A pliant method for anisotropic mesh generation,” *Proceedings of 5th International Meshing Roundtable*, Sandia National Laboratories, p.63-74, October 1996.

[13] Briere de L’Isle, E. and P.L. George, “Optimization of tetrahedral meshes,” *IMA Volumes in Mathematics and its Applications*, I. Babuska, W. D. Henshaw, J. E. Oliger, J. E. Flaherty, J. E. Hopcroft, and T. Tezduyar (Eds.), Vol. 75, p. 97-128, 1995.

[14] Canann, S.A., M.B. Stephenson, and T.D. Blacker, “Optismoothing: An optimization-driven approach to mesh smoothing,” *Finite Elements in Analysis and Design*, Vol. 13, p. 185-190, 1993.

[15] Canann, S.A., S.N. Muthukrishnan, and R. Phillips, “Topological improvement procedures for triangular and quadrilateral finite element meshes,” *In Proceedings of 3rd International Meshing Roundtable*, Sandia National Laboratories, October 1994.

[16] Canann, S.A., S.N. Muthukrishnan, and R. Phillips, “Topological refinement procedures for quadrilateral finite element meshes,” to appear in *Engineering with Computers*, submitted and accepted, 1994.

[17] Canann, S.A., S.N. Muthukrishnan, and R. Phillips, “Topological refinement procedures for triangular finite element meshes,” *Engineering with Computers*, Vol. 12, Nos. 3-4 combined, p. 243-255, 1996.

[18] Canann, S.A., Y.C. Liu, A.V. Mobley, “Automated 3D surface meshing to address today’s industrial needs,” *Finite Elements in Analysis and Design*, Vol. 25, Nos. 1-2 combined, p. 185-198, Mar 1997.

[19] Cass, R.J., S.E. Benzley, R.J. Meyers, and T.D. Blacker, “Generalized 3-D paving: an automated quadrilateral mesh generation algorithm,” *International Journal for Numerical Methods in Engineering*, Vol. 39, p. 1475-1489, 1996.

[20] Cuilliere, J.C., “An adaptive method for the automatic triangulation of 3D parametric surfaces,” *Computer-Aided Design*, Vol. 30, p. 139-149, 1998.

[21] de Cougny, H.L., M.S. Shephard, and M.K. Georges, “Explicit node point smoothing within the octree mesh generator,” *SCOREC Report #10-1990*, 1990.

[22] Edelsbrunner H., and N. Shah, “Incremental topological flipping works for regular triangulations. In *Proceedings of the 8th ACM Symposium on Computational Geometry*, p. 43-52, 1992.

[23] Field, D., “Laplacian smoothing and Delaunay triangulations,” *Communications in Numerical Methods in Engineering*, Vol. 4, p. 709-712, 1988.

[24] Field, D., "Give me a good mesh", *Proceedings of 5th International Meshing Roundtable*, Sandia National Laboratories, p.3, October 1996.

[25] Freitag, L., M. Jones, and P. Plassmann, “An efficient parallel algorithm for mesh smoothing,” In *Proceedings of the Fourth International Meshing Roundtable*, Sandia National Laboratories, p. 47-58, 1995.

[26] Freitag, L., and C. Ollivier-Gooch, “A comparison of tetrahedral mesh improvement techniques.” In *Proceedings of the Fifth International Meshing Roundtable*, Sandia National Laboratories, p. 87-100, 1996.

[27] Freitag, L., “On combining Laplacian and optimization-based mesh smoothing techniques.” *AMD Trends in Unstructured Mesh Generation*, ASME, Vol. 220, p.37-43, July 1997.

[28] Freitag, L., and C. Ollivier-Gooch, “The effect of mesh quality on solution efficiency,” In *Proceedings of the Sixth International Meshing Roundtable*, Sandia National Laboratories, p. 249, 1997.

[29] Freitag, L., “Tetrahedral mesh improvement using swapping and smoothing,” to appear in *International Journal for Numerical Methods in Engineering*, submitted, 1997.

[30] George, P.L., Automatic Mesh Generation, Paris, Masson, p. 234-236, 1991.

[31] George, P.L., and H. Borouchaki, Triangulation de Delaunay et Maillage Applications aux Elements Finis, Paris, Editions Hermes, 1997.

[32] George, P.L., and H. Borouchaki, Delaunay Triangulation and Meshing, Application to Finite Elements, Paris, France, Editions Hermes, p. 230-234, 1998.

[33] Gutierrez, M.A., and R. de Borst, “An algorithm for quadrilateral mesh rezoning,” *Presented at Finite Elements in Fluids; New Trends and Applications*, Vol. 2, Eds. M. Morandi Cecchi et al, Univ. of Padova, Padova, 1995.

[34] Hansbo, P., “Generalized Laplacian smoothing of unstructured grids,” *Communications in Numerical Methods in Engineering*, Vol. 11, p. 455-464, 1995.

[35] Herrmann, L.R., “Laplacian-isoparametric grid generation scheme,” *Journal of Engineering Mechanics*, EM5, p. 749-756, Oct. 1976.

[36] Jacquotte, O.P., and G. Coussement, “Structured mesh adaption: space accuracy and interpolation methods,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 101, p. 397-432, 1992.

[37] Joe, B., “Three-dimensional triangulations from local transformations,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, p. 718-741, 1989.

[38] Joe B., “Construction of three-dimensional improved quality triangulations using local transformations,” *SIAM Journal on Scientific Computing*, Vol. 16, p. 1292-1307, 1995.

[39] Jones, N.L., and S.G. Wright, “Algorithm for smoothing triangulated surfaces,” *J. Comput. Civil Eng.*, ASCE 1, p. 85-102, 1991.

[40] Jones, R.E., “QMESH: A self-organizing mesh generation program,” *Sandia Report SLA-73-1088*, Sandia National Laboratories, p. 23, 1974.

[41] Khamayseh, A., and A. Kuprat, “Anisotropic smoothing and solution adaption for unstructured grids,” *International Journal for Numerical Methods in Engineering*, Vol. 39, p. 3163-3174, 1996.

[42] Kohli, H.S., and G.F. Carey, “Shape optimization using adaptive shape refinement,” *International Journal for Numerical Methods in Engineering*, Vol. 36, p. 2435-2451, 1993.

[43] Lautersztajn-S, N., and A. Samuelsson, “Distortion measures and inverse mapping for isoparametric 8-node plane finite elements with curved boundaries,” *Communications in Numerical Methods in Engineering*, Vol. 14, p. 87-101, 1998.

[44] Lee, C.K, and S.H. Lo, “A new scheme for the generation of a graded quadrilateral mesh,” *Computers and Structures*, Vol. 52, No. 5, p. 847-857, 1994.

[45] Lewis, R.W., Y. Zheng, and D.T. Gethin, “Three-dimensional unstructured mesh generation: Part 3. Volume meshes,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 134, p. 285-310, 1996.

[46] Liu, A. and B. Joe, “On the shape of tetrahedra from bisection,” *Mathematics of Computations*, Vol. 63, p. 141-154, 1994.

[47] Liu, A. and B. Joe, “Relationships between tetrahedron shape measures,” *BIT*, Vol. 34, p. 268-287, 1994.

[48] Lo, S.H., “A new mesh generation scheme for arbitrary planar domains,” *International Journal for Numerical Methods in Engineering*, Vol. 21, p. 1403-1426, 1985.

[49] Lo, S.H., “Generating quadrilateral elements on plane and over curved surfaces,” *Computers and Structures*, Vol. 31, No. 3, p. 421-426, 1989.

[50] Lohner, R., K. Morgan, and O.C. Zienkiewicz, “Adaptive grid refinement for the compressible Euler equations,” in I. Babuska, O.C. Zienkiewicz, J. Gago, and E.R. de A. Oliviera (Eds.), Accuracy Estimates and Adaptive Refinements in Finite Element Computations, Wiley, p. 281-297, 1986.

[51] Oddy, A., J. Goldak, M. McDill, M. Bibby, "A distortion metric for isoparametric finite elements," *Trans. CSME*, No. 38-CSME-32, Accession No. 2161, 1988.
[52] Owen, S.J., M.L. Staten, S.A. Canann, and S. Saigal, "Q-MORPH: An indirect approach to advancing front quad meshing," submitted to *International Journal for Numerical Methods in Engineering*, 1998.
[53] Owen, S.J., and S. Saigal, "Neighborhood-based element sizing control for finite element surface meshing," *Proceedings of 6th International Meshing Roundtable*, Sandia National Laboratories, p. 143-154, Oct. 1997.
[54] Parthasarathy, V., and S. Kodiyalam, "A constrained optimization approach to finite element mesh smoothing," *Finite Elements in Analysis and Design*, Vol. 9, p. 309-320, 1991.
[55] Riccius, J., K. Schweizerhof, and M. Baumann, "Combination of adaptivity and mesh smoothing for the finite element analysis of shells with intersections," *International Journal for Numerical Methods in Engineering*, Vol. 40, p. 2459-2474, 1997.
[56] Rivara, M.C., "New mathematical tools and techniques for the refinement and/or improvement of unstructured triangulations," *Proceedings of 5th International Meshing Roundtable*, Sandia National Laboratories, p. 77-86, October 1996.
[57] Rivara, M.C., "Non-obtuse boundary Delaunay triangulations," In *Proceedings of 6th International Meshing Roundtable*, Sandia National Laboratories, p. 391, October 1997.
[58] Robinson, J., and G. Haggenmacher, "Element Warning Diagnostics," *Finite Element News*, Jun/Aug, 1982.
[59] Robinson, J., "Some new distortion measures for quadrilaterals," *Finite Element Analysis*, Vol. 3, p.183-197, 1987.
[60] Robinson, J., "Distortion measures for quadrilaterals with curved boundaries," *Finite Element Analysis*, Vol. 4, p. 115-131, 1988.
[61] Roddeman, D.G., and Jansen, L.F., "An a priori geometry check for a single isoparametric finite element," *Computers and Structures*, 47, p. 69-72, 1993.
[62] Salem, A. Z. I., S.A. Canann, and S. Saigal, "Robust distortion metric for quadratic triangular 2D finite elements," *ASME Applied Mechanics Division*-Vol. 220, p.73-80, 1997.
[63] Salem, A.Z.I., S.A. Canann, and S. Saigal, "Mid-node admissible space for quadratic triangular 2D finite elements with one curved edge," *submitted to International Journal for Numerical Methods in Engineering*, 1998.
[64] Salem, A.Z.I., S.A. Canann, and S. Saigal, "Mid-node admissible space for quadratic triangular 2D finite elements," *submitted to International Journal for Numerical Methods in Engineering*, 1998.
[65] Salem, A.Z.I., S.A. Canann, and S. Saigal, "Mid-node admissible space for 3D quadratic tetrahedral finite elements," *submitted to International Journal for Numerical Methods in Engineering*, 1998.
[66] Shephard, M.S., and M.K. Georges, " Automatic three-dimensional mesh generation by the finite octree technique," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 4, p. 709-749, 1991.
[67] Shimada, K., "Physically-based mesh generation: Automated triangulation of surfaces and volumes via bubble packing, Ph.D. thesis, ME Dept., Massachusetts Institute of Technology, Cambridge, MA, 1993.
[68] Shimada, K., and D.C. Gossard, "Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing." *In ACM Third Symposium on Solid Modeling and Applications*, p. 409-419, May 1995.
[69] Shimada, K., "Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles," *Proceedings of the Sixth International Meshing Roundtable*, Sandia National Laboratories, p. 375-390, 1997.
[70] Staten, M.L., and S.A. Canann, "Post refinement element shape improvement for quadrilateral meshes," AMD-Vol. 220 Trends in Unstructured Mesh Generation, ASME, p. 9-16, July 1997.
[71] Tristano, J.R., S.J. Owen, and S.A. Canann, "Advancing front surface mesh generation in parametric space using a Reimannian surface definition," submitted to the 7th International Meshing Roundtable, Dearborn, MI, Oct 1998.
[72] Winslow, A.M., "Equipotential zoning of two-dimensional meshes, Lawrence Livermore Laboratory, University of California, Livermore, California, *Report No. UCRL-7312*, 1963.
[73] Zhu, J.Z., O.C. Zienkiewicz, E. Hinton, and J. Wu, "A new approach to the development of automatic quadrilateral mesh generation," *International Journal for Numerical Methods in Engineering*, Vol. 32, p. 849-866, 1991.
[74] Zienkiewicz, O.C., The Finite Element Method, London, McGraw Hill, 1971.
[75] Meshing your solid model, *ANSYS Modeling and Meshing Guide*, Release 5.4, 000862, 2nd Edition, SAS IP, Inc., Chapter 7, September 1997.
[76] Revising your solid model, *ANSYS Modeling and Meshing Guide*, Release 5.4, 000862, 2nd Edition, SAS IP, Inc., Chapter 8, September 1997.