COMBINED LAPLACIAN AND OPTIMIZATION-BASED SMOOTHING FOR QUADRATIC MIXED SURFACE MESHES

Zhijian Chen, Joseph R. Tristano, Wa Kwok [1]

[1] *Ansys, inc., 275 Technology Drive, Canonsburg, PA 15317, U.S.A.*
{james.chen,joe.tristano,wa.kwok}@ansys.com

ABSTRACT

Quadratic elements place stringent requirements on a surface mesh smoother. One of the biggest challenges is that a good linear element may become invalid when mid-side nodes are introduced. To help alleviate this problem, a new objective function for optimization-based smoothing is proposed for triangular and quadrilateral elements, linear or quadratic. Unlike the current popular approaches, this objective function makes it possible for a smoothing algorithm to untangle and smooth in a single process. This objective function has higher order continuous derivatives and only one minimum, if any, that make it suitable for optimization techniques. Even though optimization-based smoothing obtains much higher quality results compared to other algorithms, such as constrained Laplacian smoothing, it is also slower than these algorithms. That said, we also present an effective way to limit the number of calls to optimization-based smoothing such that the highest quality mesh is obtained in the least amount of time.

Keywords: optimization-based smoothing, constrained-Laplacian smoothing, smoothing objective function, surface mesh

1. INTRODUCTION

Mesh quality is a key factor in FEM analysis. There are numerous ways to achieve a high quality mesh [1], such as controlling the discretization size, controlling the edge valence of mesh nodes and controlling the distortion of the individual element shapes. Mesh smoothing (relaxation) [2], improves quality by adjusting node locations to reduce the distortion of the element shapes without changing the topology of the mesh. In general, mesh smoothing can be classified into two major groups [3]: local and global. In local smoothing, nodes are moved one by one, while global smoothing changes all the nodal locations in a mesh simultaneously.

The most commonly used smoothing technique is Laplacian smoothing [4], which moves a given node to the geometric center of its incident nodes. Various weighted Laplacian smoothing algorithms have been developed to improve the performance of the original smoothing technique. Laplacian smoothing is computationally inexpensive but does not guarantee improvement in mesh quality. In fact, it is possible to create inverted or invalid elements with this technique. A valid mesh is one whose elements have acceptable quality metrics [2]. Constrained Laplacian smoothing [5] overcomes this problem by placing a node at a new location only when the mesh quality is improved. This method successfully prevents the degradation of mesh quality but does not always improve the quality of the mesh or place nodes at their best locations.

In recent years, optimization-based smoothing algorithms have been drawing the attention of the mesh generation community. Several optimization-based smoothing algorithms have been developed [2,3,4]. These algorithms integrate some mesh quality measures into objective functions. Optimization techniques should, in general yield a better mesh, if the objective function is properly formulated. Optimization-based smoothing varies based on: the type of mesh being smoothed, the optimization method used, and the distortion metric selected to construct the objective function.

One of the keys to the success of an optimization-based smoothing algorithm is to define an appropriate objective function. An inappropriate objective function can waste time in the optimization algorithm along with causing the algorithm to fail to improve the mesh quality. Most efficient optimization algorithms [6] require the objective function be C^1 continuous.

Various measures for element [7] quality have been used in the objective function, such as distortion metrics, aspect ratio, minimum angle, etc…. Recently, the inspiring work of P. Knupp derived an objective function from the condition number of element Jacobian matrix [8]. His work along with the work of L Freitag [3,10], has lead to mesh quality improvement algorithms for 2D and 3D linear elements. S. Paoletti [9] stated that using Interpolation Tensor could be applied to various polyhedral meshes in 2D and 3D. Even though the published works show enormous potential, there are two general limitations in these algorithms:

1. They only apply to linear elements.
2. They require that the initial mesh is valid.

Quite often, the mesh to be smoothed is not valid. Most of the existing objective functions have been designed in such a way that the optimization smoothing schemes mentioned above cannot guarantee a converged solution for an invalid mesh. For this reason, untangling techniques[10] have been proposed to remove invalid elements from the mesh before executing optimization-based smoothing.

Canann [2] combined the use of Laplacian and optimization-based smoothing to speed up the smoothing process along with benefit of better mesh quality from optimization smoothing. For optimization-based smoothing, α for a triangle, as defined by S.H. Lo [11], and β for a quadrilateral[2], are used in the objective function. Based on our experience, reasonably good meshes have been achieved in most cases. However, there are some cases, especially when smoothing nodes near curved boundaries, or smoothing nodes attached to higher order elements, in which the resultant mesh quality around these elements are not always satisfactory (Figure 1). As a modification to Canann's work, we recently improved our smoother to make it suitable for working with quadratic elements by developing a new objective function to be used for optimization-based smoothing [12]

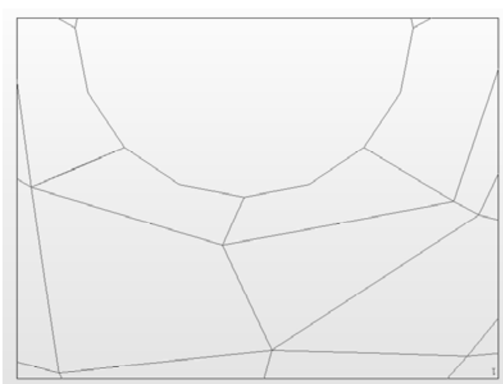


Figure 1. A simple mesh with poor smoothing

In our recent work, we improved our smoother by employing the following:

- introduced a new objective function
- modified Fletcher-Reeves [6] optimization to enhance its performance for our domain
- Judicious use of optimization-based smoothing
- A priority based smoothing order is introduced.

In what follows, section 2 will address our new objective function for optimization-based smoothing, section 3 will discuss our improvement on optimization algorithm, section 4 presents the decision making process for calling optimization-based smoothing. In the last section, we will conclude our discussion and present future work in the area.

2. OBJECTIVE FUNCTION FOR SMOOTHING

A distortion metric is a measure of a mesh's quality. Therefore, an objective function for smoothing is usually constructed based on some distortion metric or combination thereof. A metric is suitable for use in an objective function, if the following criteria are met:

1. Efficiency: Since optimization-based smoothing is computationally expensive, the metric used must be efficient to compute.
2. Continuity: Since derivatives are used during the optimization process, the objective function is expected to be continuous. In general, to have higher rate of convergence, higher order derivatives are used.
3. Monotonically Decreasing: If an objective function has multiple local optimum locations, it will be difficult for the optimization algorithm to find the best solution. If the metric is not monotonically decreasing, the optimized location may vary based on the initial location of the node to be smoothed.
4. Shape Independence: It is favorable for the metric to be defined and normalized in such a way that the metrics for all element shapes can work together

Historically, we at Ansys have used two shape metrics: α for triangular elements and β For quadrilateral elements. Let us now discuss their properties.

2.1 α for triangular elements

The triangular metric, α, [11] is defined as ,

$$\alpha = \pm \frac{2\sqrt{3}\left\|\overrightarrow{\mathbf{BC}} \times \overrightarrow{\mathbf{AC}}\right\|}{l_{AB}^{2} + l_{BC}^{2} + l_{CA}^{2}}, \tag{1}$$

where, l_{AB}, l_{BC}, and l_{CA} are the edge lengths of the triangle ΔABC, and $\overrightarrow{BC}$ and $\overrightarrow{\mathbf{AC}}$ are the edge vectors of the triangle, as shown in Figure 2. The metric is signed to account for a positive valid metric and a negative invalid or inverted element. For a linear triangular element, the numerator of equation (1) is directly related to the area of the triangle, while the denominator is the sum of the squared edge lengths of the triangle. The shape metric α is bounded by $\alpha \in [-1,1]$. A value 1 corresponds to the best triangle, an equilateral triangle, while –1 indicates an inverted equilateral triangle. When all the three points of the triangle are co-linear, the triangle has a zero area which yields a value of α=0.

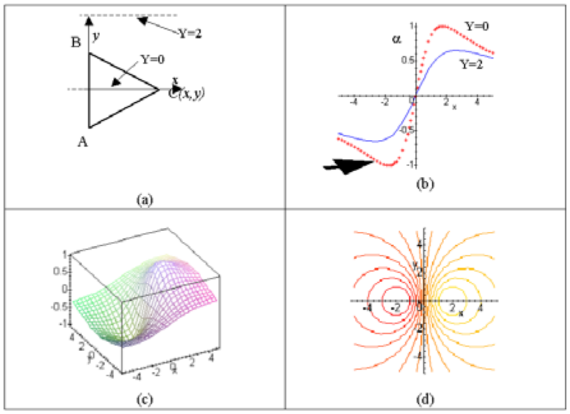


Figure 2. α for a triangle. (a) A triangle with point C perturbed over the x-y plane. (b) The change of α on y=0 and y=2. (c) The 3D plots of α; (d) the contour of α on the x-y plane

It is obvious that if α alone is used in the objective function for optimization-based smoothing, the smoother will have trouble untangling inverted triangles. For example, if an initial point is placed at the location as shown by the arrow in Figure 2 (b), the shape metric α would tell us move the point in the negative x-direction to improve mesh quality.

2.2 β For quadrilateral elements

Similarly, we have used a quality metric β [2] (Figure 3) for quadrilaterals, which is comprised of a combination of the α's of the triangular elements that compose the given quadrilateral. The basic idea is to split a quadrilateral into four different triangles, Δabc, Δdac, Δabd, and Δdbc, Figure 8 (a) . Each of these triangles has a quality metric, α_1, α_2, α_3, and α_4.

$$\beta = \frac{\min(\alpha_1, \alpha_2, \alpha_3, \alpha_4) - n_{neg}}{\alpha_{90}} \qquad (2)$$

where, α_{90} is the α right triangle having unit base and height lengths, and n_{neg} is the number of negative α . α_{90} is used as a normalization factor where n_{neg} is a historical heuristic value that was placed into our code many years ago. n_{neg} was used to help the algorithm, published in [2] , un-invert tangled meshes such that inverted elements would have a very high weight in the objective function.

Since the β for quadrilaterals is derived from α for triangle, the β has similar problems to α (Figure 3).

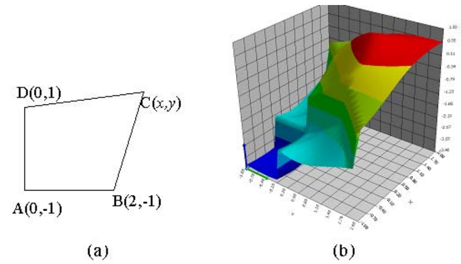


Figure 3. β for a quad. (a) A quadrilateral with point C set to be free , (b) The change of β when point C is perturbed in the x-y plane.

2.3 Problems with higher order elements

It is quite often observed that a linear element is of acceptable quality but becomes unacceptable when it is converted to a higher order element. As demonstrated in Figure 4: The elements in (a) and (c) are of acceptable quality when the elements are linear. However, with the introduction of a mid-side node, the quadratic elements in (b) and (d) show interior angles close to 0 and 180 degrees.

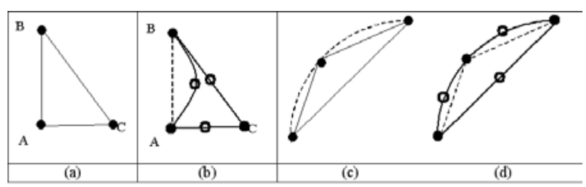


Figure 4 the difference of element quality for linear and quadratic elements

A curved element edge usually happens only when the element edge is on boundary. In the above example, the element (b) might be improved by moving node C, if node C is an interior node. However, element (d) has no node to move to improve its quality. The shape improvement of this class of element is beyond the scope of smoothing.

Smoothing for linear simplex elements (three node triangle for two dimensions and four node tetrahedral for three dimensions) has been studied substantially with fruitful achievements. However, simple straightforward means to extend these studies to quadratic triangular and quadrilateral elements do not seem to exist.

By investigation, the effects of mid-side nodes of higher order elements can be represented by a new term as a function of element interior angles. In the case of quadratic elements, interior angles and vectors at nodes are computed using the tangent vector of the quadratic edge at the given node. The proposed objective function is made up of two parts:

$$f = f(\alpha) + f(\theta) \qquad (3)$$

where, $f(\alpha)$ is based on the shape metric α computed at the node of interest and $f(\theta)$ is a penalty term based on element angles computed at the other two nodes. The purpose of the $f(\theta)$ term is to prevent the element from inverting and smooth the element when it is quadratic.

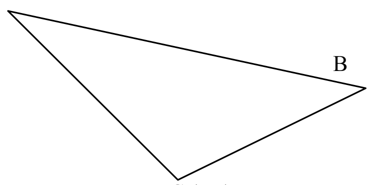


Figure 5. triangle with node C as a moving node

For a node

$$f(\alpha) = \sum_{i=1}^{n_e} f_i(\alpha) \qquad (4)$$

where, n_e is the number of element using the smoothing node, and $f_i(\alpha)$ is the contribution from the *ith* element to the objective function.

$$f_i(\alpha) = (1-\alpha_i)^2 \qquad (5)$$

Where, α_i is the shape metric α *of the* i^{th} *element.* Since the function $f_i(\alpha)$ is derived from α, the properties and problems of $f_i(\alpha)$ are similar to α.

A penalty term, $f(\theta)$, was introduced to give our objective function a monotonically decreasing property with one minima, which we have observed via empirical data. It is because of these penalty terms, that the optimization-smoothing algorithm is able to untangle invalid meshes. Interior angles, at node A and node B, are convenient means to determine the validity of a triangle. When smoothing quadratic elements, the quadratic edge tangent vectors at nodes A and B are used to compute the angles at A and B.

However, angle A and angle B in Figure 5 are not continuous functions on X-Y plane, Figure 6(a). There is a discontinuity on the line, $\{x=0\}$, where the angle jumps between $-\pi$ and π. When point C is on the positive x side of the plane, the angle is π. When the point is at the negative x side of the plane, the angle is $-\pi$. Therefore, the angle on the line of $\{y>1, x=0\}$ is undefined

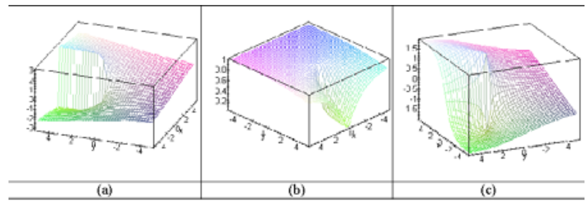


Figure 6. angle is not a continuous function in (x,y) plane. (a) Angle distribution over x-y plane. The angle has a discontinuity along the line (x=0, y>1); (b) weight function introduced to make adjusted angle smooth; (c) the adjusted angle. Adjusted angle is continuous and smooth everywhere except at point A and B.

As stated previously, discontinuous functions are not suitable for optimization. To overcome the problem a weight, w, is introduced as a multiplier to the angles to generate an adjusted angle, ω.

$$w = 1 - e^{-2(\pi - |\theta|)} \qquad (6)$$

$$\omega = w\theta \qquad (7)$$

The adjusted angle is continuous everywhere but at point A and point B, Figure 7 (c). Our penalty term is introduced as.

$$f(\theta) = \left(k_1 e^{k_2 \omega \sqrt{\frac{a}{c}}} - 1 \right)^2 \qquad (8)$$

Where a is the edge length opposite the node where θ is computed and c is the edge length opposite where α computed. This smooth penalty term increases rapidly when an element angle approaches "0" and the negative angle region. When an element is in the valid region, the penalty is relatively small compared to the near invalid and invalid regions. The constants, k_1 and k_2 are used normalize the metric such that $f(\theta)$ will have little to no effect for an equilateral triangle.

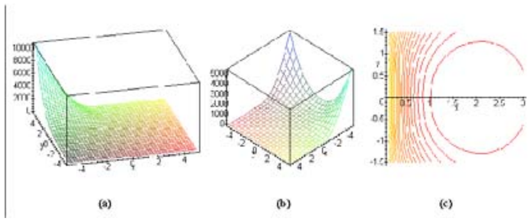


Figure 7. the penalty term and the objective function (a) the penalty term when considering only one side of the triangle, (b) the final shape of the objective function, (c) the contours of the final objective function in the feasible region.

For quadrilateral elements, the objective function, f_q, is formulated as

$$f_q = f_1 + f_2 + f_3 \qquad (9)$$

where, f_1, f_2, and f_3 are the objective functions from triangles ΔADC, ΔADB, and ΔBDC, as shown in Figure 8, and the node D is the moving node. Notice that ΔABC has been omitted from this function since the movement of point D has no effect on this triangle. The triangle is a dead zone to node D.

The issue of how to combine the nodal metrics from a mixed mesh arises when triangular and quadrilateral elements are present. This resolution of this issue is actually quite simple. As shown in equation (9), the objective function for a quadrilateral is made up of three functions using the sub-triangles. Each of these sub-triangles is equivalent to equation (5). Therefore, for a mixed mesh, the contribution of each triangular element to the total objective function should be multiplied by a factor of three in order to evenly weigh triangles and quadrilaterals.

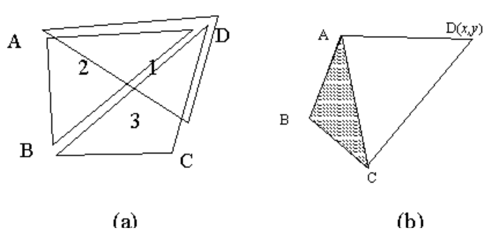
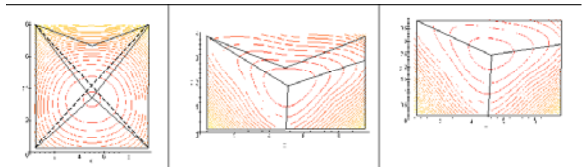


Figure 8. ΔABC is the dead zone for the quadrilateral $ABCD$ during smoothing

As discussed above, each part of the objective function is continuous function with at least up to 2nd order derivatives. This property makes the objective function very suitable for an optimization process. Derivatives such as these enable us to use gradient methods such as the method of conjugate gradients.

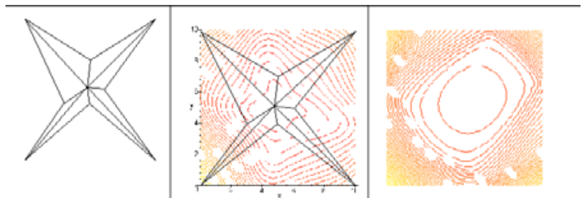
The effectiveness of the above formulation can be demonstrated by the example in Figure 9. As clearly indicated on (a), when the top edge is pushed toward the center node, the best location for the center node is below the intersection of the two dash lines. The other two illustrations show similar results for quadrilateral elements.



(a) a rectangular plate made up of 4 triangular elements with the top edge curved	(b) a rectangular plate made up with 3 quad elements with top edge curved	(c) a rectangular plate made up with 3 quad elements with all straight edges

Figure 9. configuration with one quadratic edge

An example configuration similar to the one presented in [3] is used to show the result of our current work. Figure 10 illustrates that even though the level sets outside of the valid region for the center node is non-convex, the near convexity of the set enables faster convergence of the optimization algorithm..



(a)	(b)	(c)

Figure 10. example with star shaped configuration with 10c illustrating the level sets in the feasible region

3 OPTIMIZATION-BASED SMOOTHING

Optimization-based smoothing is an iterative process. Each node for smoothing is optimized for location in a number of iterations. Generally, as in popular approaches, the optimization process is a constrained optimization process. However, the presented objective function actually combines angle constraints into part of the objective function, which enables the optimization-based smoothing process to be unconstrained.

Let $\boldsymbol{X}$ be nodal location of a node, the optimization process is to find the best location in iterations:

$$\mathbf{x}^q = \mathbf{x}^{(q-1)} + s\mathbf{d}^q \qquad (10)$$

where, q is the iteration number, $\mathbf{d}^q$ is the vector of the search direction and s is the step length to move in this search direction. Optimization is a classical area of study in mathematics and it is not our intent to discuss it in depth here. However, it is valuable to share some of our experiences with it when related to mesh smoothing.

3.1 Search direction

Since the objective function is smooth, the Fletcher-Reeves [6] conjugate gradient method is used.

$$\mathbf{d}^{q} = -\nabla f(\mathbf{x}^{q}) + r\mathbf{d}^{(q-1)} \qquad (11)$$

where,

$$r^{q} = \frac{\left|\nabla f(\mathbf{x}^{q})\right|^{2}}{\left|\nabla f(\mathbf{x}^{(q-1)})\right|^{2}} \qquad (12)$$

In our implementation, the gradient direction is used at the first iteration and conjugate gradient direction is used in the consequent iterations. We have found that after a number of iterations, the convergence speed using the conjugate gradient direction actually slows down. Figure 11. illustrates, the iteration using gradient direction is marked with a letter "g" and the iteration using conjugate gradient directions are marked with a letter "c". At the first iteration, when the gradient direction is used, the proceeding step length is relatively small, while at the second iteration, the conjugate gradient direction is used and the step length is much larger than the first step length. However, two more iterations later, the progress becomes relatively small. In this case, a new gradient direction is used and the process restarts [13].

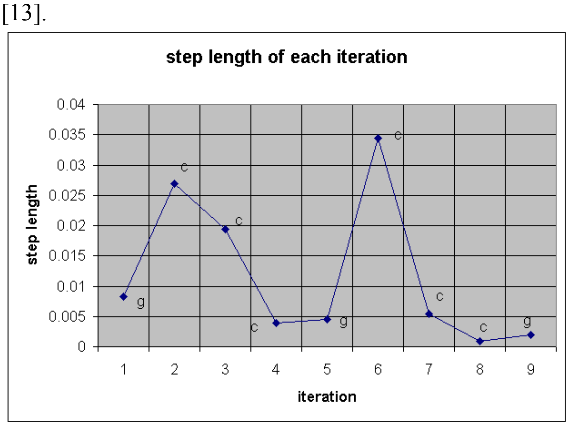


Figure 11. convergence history

Furthermore, we have found that the first few iterations make the most significant contribution to finding the best nodal location, Figure 12. Combining the investigations above, a constant number of three is used as the limit of iterations for each optimization-based smoothing call.

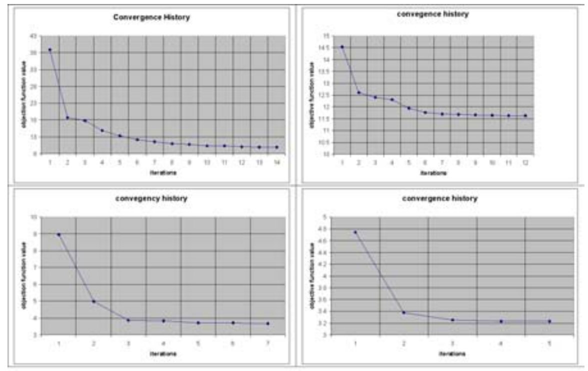


Figure 12. The convergence history for a node using conjugate gradient direction

3.2 Step length for one-dimensional search

A one-dimensional search is conducted to find a local minimum along the search direction. Quadratic interpolation is used in our one-dimensional search. Progressively, three points are found in order to compute a "high-low-high" pattern. For faster speed, an acceleration factor is used. The choice of the acceleration factor is very tricky. In the beginning of the search, because the initial step length is relatively small, a large factor, 8, is used. Other accelerating factors might also be used in the consequent searches. Once a failure to find the "high-low-high" pattern is encountered, the factor is reduced by half. The one-dimensional search fails if a "high-low-high" pattern cannot be found.

4. OVERALL SMOOTHING ALGORITHM

4.1 The use of optimization-based smoothing

Since optimization-based smoothing is much slower than Laplacian smoothing, it is critical to make a correct decision when to use optimization-based smoothing to gain the best cost effectiveness. Here are our rules:

1. a node is connected to a curved boundary node;
2. a node is connected to an invalid element;
3. a node has failed to move from Laplacian smoothing;
4. forced to use optimization-based smoothing by caller.

According to the rules above, an untangling process, for example, is mostly smoothed by optimization smoothing because a tangled mesh contains many nodes that are connected to invalid elements. Once untangled, most interior nodes are actually smoothed by constrained-Laplacian smoothing.

Constrained-Laplacian smoothing has trouble dealing with concave regions. If a new location, determined, by Laplacian smoothing is not acceptable, optimization-based smoothing is called to resolve the problem.

There are cases when the caller detected that a mesh is not acceptable after smoothing. In such cases, one pass of optimization-based smoothing usually gives us satisfactory results.

4.2 Smoothing by priority order

It is found that node smoothing in order of "worst one first" is very helpful. As shown in Figure 13, when priority is used, smoothing takes 4 iterations for the tangled model to be untangled, while 7 passes are needed if nodes are smoothed in a order of "first come first serve". The priority is simply computed based on the shape metrics of each node. For the node with the worst quality, the highest priority is assigned. The other priorities are computed by linearly dividing the range of shape metric value into 5 bins. The priority is then computed for each smoothing iteration. An inner priority loop counter sets the current priority during a smoothing iteration. If the current node's priority is less than that of the current priority, it will not be smoothed in the inner priority loop

5. EXAMPLES AND RESULTS

5.1 Smoothing improvement statistics

The improved smoothing algorithm has been fully tested under the ANSYS/Classic and ANSYS/Workbench regression test sets along with many customer problems to verify that it is sufficiently robust and efficient as a commercial product.

Table 1. mesh quality without new smoothing

	NQUAD	NTRI	MAX	MIN	AVG	STDEV
MIN			0.76579	0.02914	0.37137	0.08026
AVG	133.549	0.457263	0.940831	0.307522	0.689708	0.160157
MAX						0.32282

Table 2. mesh quality with new smoothing

	NQUAD	NTRI	MAX	MIN	AVG	STDEV
MIN			0.70207	0.06631	0.52954	0.07245
AVG	137.708	0.415837	0.936968	0.415502	0.7136	0.128519
MAX						0.32282

The above two tables are statistical results from meshing 183 complex and planar surfaces randomly picked from regression test sets. Some of the surfaces are meshed with just several elements where some are meshed with thousands of elements.

Table 1 is the result before the new smoothing was implemented where Table 2 is the result with new smoothing. A detailed explanation of Table 1 and Table 2 follows:

1. With new smoother, the number of Quadrilateral elements (NQUAD) generated is increased and consequently, the number of Tri elements (NTRI) generated is decreased. The reason the number of elements varies between the two smoothing algorithms is because the smoother is integrated into the mesh generation process. This phenomenon illustrates how different mesh generation algorithms are sensitive to node placement.
2. As indicated in the fourth column (MAX), the best quality elements generated with new smoothing is actually not as good as it used to be, however, this is not statistically significant (sig = .49 > .05).
3. The worst elements have improved substantially, column 5 (MIN).
4. The average, column 6 (AVG), of the element shape metrics has also increased.
5. The last column (STDEV) of the tables illustrates the standard deviation that measures the variance of the samples. It is clearly indicated that, the standard deviations of the minimum and average have decreased in Table 2. However, the max standard deviation remains the same.

Table 3. t-test for shape metrics

Metric	Significance (2-tailed)
Min	0.00
Max	0.00
Avg	0.49
Standard Deviation	0.00

Table 3 condenses the information found in Tables 1 and 2 with a paired samples t-test, a statistical test that compares means (the details of the test are outside the scope of this paper. For more information, consult almost any basic statistics textbook), for minimum, maximum, average, and standard deviation of shape metrics. The improvement minimum, average, and standard deviation of the shape metrics are significant while the decrease in the maximum is not statistically significant

.

5.2 Examples meshes

In this section, we will present a number of example meshes. Figure 13 is an example for the untangling of the "plate with a hole" model. The tangled mesh (Figure 13a) is created by perturbing the nodal locations of each interior node (a node not on surface boundaries) randomly. The tangled mesh has many nodes outside the surface domain. Figure 13b is the mesh after smoothing. It takes 3 iterations for this model to be untangled(Figure 14 and Figure 15).

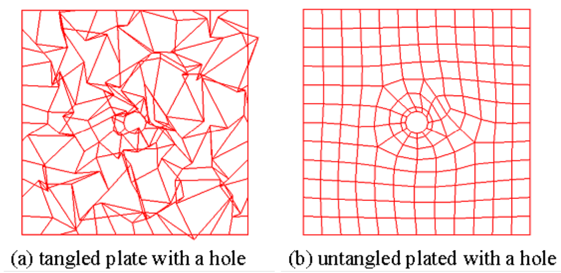


Figure 13. untangling example

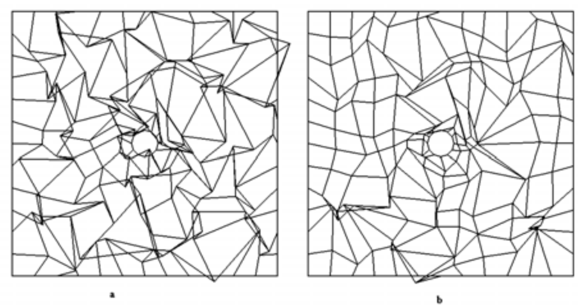


Figure 14. untangling, initial mesh and first iteration mesh

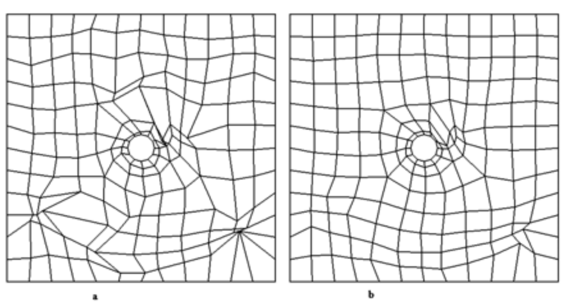


Figure 15. untangling, second iteration mesh and third iteration mesh

Figure 16 is a unit test case where a node is connected to a curved boundary sharing two quadratic boundary elements. This is a difficult case for most smoothing algorithms that deal only with linear elements. Figure 16a is the result of smoothing with only constrained Laplacian smoothing where Figure 16b is the result of smoothing using our presented optimization-based smoothing.

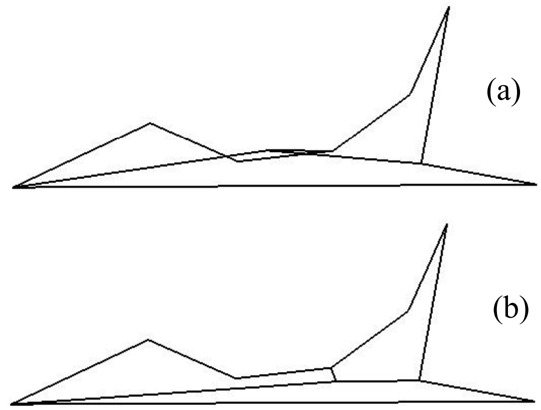


Figure 16. a node with three edges close to a curved boundary and two quadratic boundary edges

Figure 17 illustrates another unit test example. Similar to the above example, a node is connected to some curved quadratic boundary elements. The optimization-based smoothing result (Figure 17a) yields higher quality than the same configuration smoothed with constrained Laplacian smoothing, (Figure 17b).

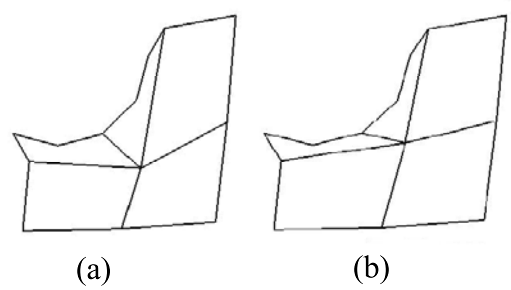


Figure 17. quadratic curved boundary quads

Figure 18 shows an example of high quality mesh generated using the new smoother.

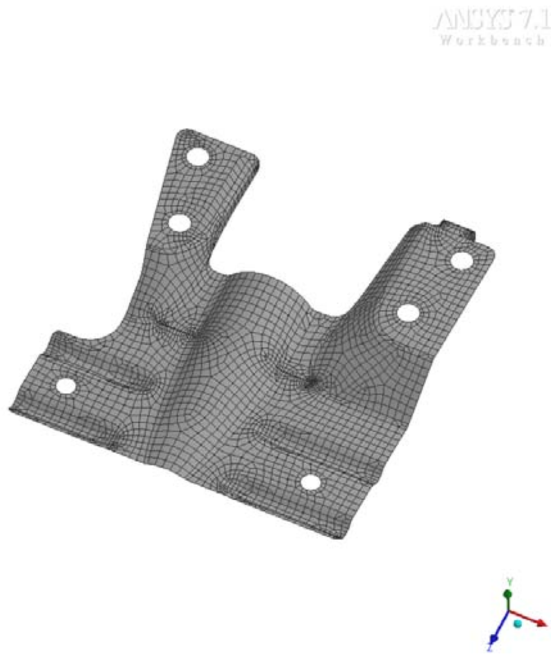


Figure 18. example surface mesh

Figure 19 illustrates a mesh of a human head using the new smoothing algorithm.

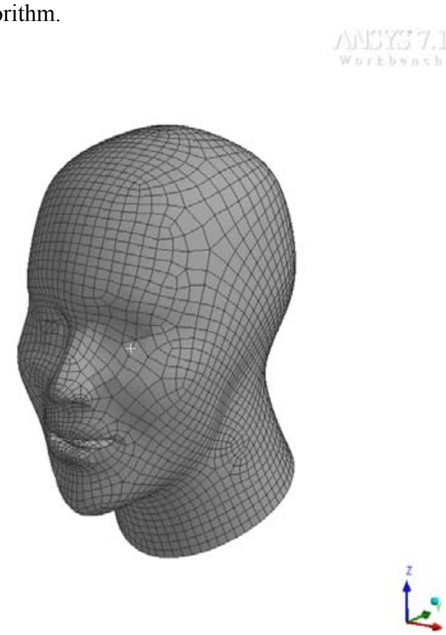


Figure 19 Head

5.3 Objective function comparisons

Our previous method of constructing the objective function-based on the maximizing the minimum shape metric posed serious convergence problems for any optimization method. Figure 20 illustrates the differences between the old objective (b) function and the new objective function (c) for a given configuration of quadrilaterals.

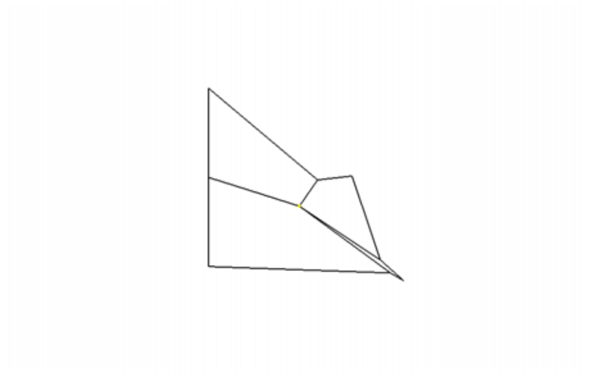


Figure 20 a Test quadrilateral configuration with the center node being smoothed

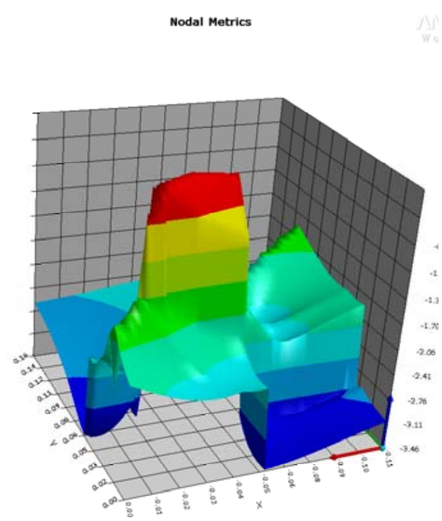


Figure 20 b Maximizing Minimum Objective Function

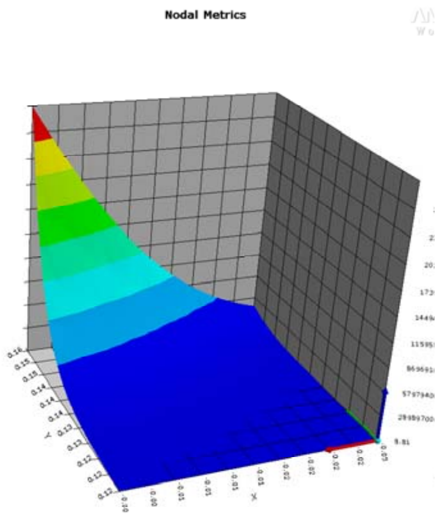


Figure 20 c The newly presented objective function

6. CONCLUSIONS AND FUTURE WORK

A new objective function for optimization-based smoothing is proposed for both triangular and quadrilateral elements. Unlike the current popular approaches, the new objective function makes it possible to untangle and smooth in a single process. The objective function has higher order continuous derivatives and only one minimum, if any, that make it quite suitable for optimization techniques. Because optimization-based smoothing is much slower than other algorithms, such as constrained Laplacian smoothing, an effective way to limit the number of calls to optimization-based smoothing is critical in order to obtain the best result in terms of quality and speed.

Future work in this area may include:

- Speed improvement on metric calculation so we can use optimization smoothing more often
- Mathematically prove properties of the objective function
- Extend the algorithm to solid elements

REFERENCES

[1] Steven J. Owen, "A Survey of Unstructured Mesh Generation Technology", *Proceedings, 7th International Meshing Roundtable*, Sandia National Lab, pp.239-267, October 1998

[2] Scott A. Canann, Joseph R. Tristano, Matthew L. Staten, "An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes", *7th International Meshing Roundtable*, Sandia National Labs, pp.479-494, October 1998

[3] Lori A**.** Freitag, "Local Optimization-Based Untangling Algorithms for Quadrilateral Meshes", *Proceedings, 10th International Meshing Roundtable*, Sandia National Laboratories, pp.397-406, October 7-10 2001

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

[5] Lori A. Freitag, "*On combining Laplacian and optimization-based smoothing techniques*", Trends in Unstructured Mesh Generation, ASME Applied Mechanics Division, AMD-Vol 220, 37-44, 1997

[6] R. Fletcher,, and Reeves, C.M.: *Function Minimization by conjugate gradients*, Comput. J. 7, 149 (1964)

[7] Wa Kwok and Zhijian Chen, "A Simple and Effective Mesh Quality Metric for Hexahedral and Wedge Elements**",** p57-66*, Proceedings ot the 9th International Meshing Roundtable* '2000, New Orleans, Louisiana, October 325-333 2000.

[8] Patrick M Knupp, "Matrix Norms & The Condition Number: A General Framework to Improve Mesh Quality Via Node-Movement", *Proceedings, 8th International Meshing Roundtable*, South Lake Tahoe, CA, U.S.A., pp.13-22, October 1999

[9] Stefano Paoletti, "Polyhedral Mesh Optimization Using the Interpolation Tensor", p19-28, *Proceedings ot the 9th International Meshing Roundtable* '2000, New Orleans, Louisiana, October 325-333 2000

[10] Lori A. Freitag, and Paul E. Plassmann, "Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number", *Proceedings, 8th International Meshing Roundtable*, South Lake Tahoe, CA, U.S.A., pp.247-258, October 1999

[11] C.K Lee, 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.

[12] Zhijian Chen, Joseph Tristano and Wa Kwok**,** Construction of an Objective Function for Optimization Based Smoothing for Surface Meshing, The 7th US National Congress on Computational Mechanics.

[13] ANSYS, Inc., "ANSYS Theory Manual", 20.8. First Order Optimization Method