ON COMBINING LAPLACIAN AND OPTIMIZATION-BASED MESH SMOOTHING TECHNIQUES

Lori A. Freitag
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, Illinois 60439
freitag@mcs.anl.gov

ABSTRACT

Local mesh smoothing algorithms have been shown to be effective in repairing distorted elements in automatically generated meshes. The simplest such algorithm is Laplacian smoothing, which moves grid points to the geometric center of incident vertices. Unfortunately, this method operates heuristically and can create invalid meshes or elements of worse quality than those contained in the original mesh. In contrast, optimization-based methods are designed to maximize some measure of mesh quality and are very effective at eliminating extremal angles in the mesh. These improvements come at a higher computational cost, however. In this article we propose four smoothing techniques that combine a smart variant of Laplacian smoothing with an optimization-based approach. Several numerical experiments are performed that compare the mesh quality and computational cost for each of the methods in two and three dimensions. We find that the combined approaches are very cost effective and yield high-quality meshes.

INTRODUCTION

The finite element and finite volume solution methods have proven to be effective tools in the numerical solution of many scientific and engineering applications. Both techniques require a spatial decomposition of the computational domain into a union of simple geometric elements such as triangles or quadrilaterals in two dimensions and tetrahedra or hexahedra in three dimensions. If the geometry is complex, automatic mesh generation tools are used to facilitate this decomposition (Lo, 1985, Shephard and Georges, 1991). A problem with these meshes is that the shape of the elements in the mesh can vary significantly. For finite element techniques using simplicial meshes, poorly shaped or distorted elements can result in numerical difficulties during the solution process, particularly as the elements approach the limits of 0° or 180° (Babuska and Aziz, 1976, Fried, 1972). This problem is more severe in three dimensions than in two dimensions because tetrahedral meshes tend to have a larger proportion of severely distored elements than do triangular meshes.

One approach used to repair poor quality meshes is to adjust the grid point locations without changing the mesh topology. This approach is called mesh smoothing, and several research groups have proposed different algorithms to perform grid point adjustment, e.g. Field (1988), Amezua et al. (1995), and Canann et al. (1993). In general, most mesh smoothing algorithms are local techniques and adjust the geometric position of each vertex individually to obtain improvement in a neighborhood around that vertex. Some number of sweeps over the adjustable vertices are performed to achieve overall improvement in the mesh. Because the mesh may contain millions of grid points, it is critical that each individual adjustment be computationally inexpensive.

Perhaps the most commonly used local smoothing technique is Laplacian smoothing (Lo, 1985, Field, 1988). In this method each grid point is moved to the geometric center of the polygon determined by the incident vertices. This method is computationally inexpensive, but operates heuristically and does not guarantee improvement in the element quality. In fact, it is possible to produce an invalid mesh containing elements that are inverted or have negative area. One common variant of Laplacian smoothing allows the grid point movement to take place only if the mesh is improved according to some quality measure. This smart Laplacian smoother guarantees an improved mesh, but the new point location is not optimal.

One class of algorithms that avoids the creation of invalid elements and finds an optimal new location for the mesh vertices uses optimization techniques (Amenta et al., 1997, Parthasarathy and Kodiyalam, 1991, Shephard and Georges, 1991). These techniques offer the advantage of guaranteed mesh improvement and validity. However, this guarantee comes at a much higher computational cost than Laplacian smoothing.

In Freitag et al. (1995) we proposed a local optimization-based mesh smoothing technique for two-dimensional triangular meshes that can serve as the core of an efficient parallel algorithm with a provably fast runtime bound. In Freitag and Ollivier-Gooch (1996) we extended this algorithm to three dimensions and followed a two-pronged approach for improving tetrahedral meshes. We showed that face swapping followed by optimization-based smoothing significantly improved dihedral angles near 0 and 180 degrees for a number of test cases. In contrast, face swapping followed by smart Laplacian smoothers were unable to eliminate extremal angles in the mesh. Neither technique was able to eliminate poor mesh elements that were adjacent to each other or to geometry boundaries.

In two dimensions the optimization-based smoother was approximately five times more computationally expensive than the smart Laplacian smoother and ten times more computationally expensive in three dimensions. In this article, we propose several techniques that combine the low cost Laplacian smoothing with the optimization-based approach used only for the poorest quality elements. Numerical experiments are performed for several test meshes generated using commonly available techniques such as Delaunay point insertion, quadtree mesh generation, and a wavefront algorithm.

The remainder of the article is organized as follows. In the next section we briefly review the optimization-based technique used in our experiments and formulate the proposed combined approaches. We then perform a number of numerical experiments on meshes generated from random point sets in both two and three dimensions. These meshes contain severely distorted elements and give a good indication of smoothing algorithm effectiveness. Following this analysis the suite of test cases is expanded to include several application meshes. Finally, we summarize the performance obtained with the combined approaches and offer concluding remarks.

THE COMBINED SMOOTHING APPROACH

Local mesh smoothing techniques are formulated in terms of the grid point to be adjusted, the *free vertex*, v, and that grid point's adjacent vertices, V. The location of the free vertex is changed according to some rule or heuristic procedure based on information available at the adjacent grid points. Suppose $\mathbf{x}$ is the position of the free vertex; then the general form of the smoothing algorithms is given by

$$\mathbf{x}_{new} = \mathtt{Smooth}(\mathbf{x}, V, |V|, \mathrm{conn}(V)),$$

where $\mathbf{x}_{new}$ is the proposed new position of v, $|V|$ is the number of adjacent vertices, and $\mathrm{conn}(V)$ is the adjacent vertex connectivity information. Ideally, the new location of the free vertex will improve the mesh according to some measure of mesh quality such as dihedral angle or element aspect ratio.

To evaluate the mesh quality for the mesh elements, let $f_i(\mathbf{x})$, $i = 1, \ldots, n$, be the values of mesh quality affected by a change in $\mathbf{x}$. For example, if we use the dihedral angles as a mesh quality measure in a three-dimensional mesh, each tetrahedron would have six function values, one for each edge of the tetrahedron. Thus, the total number of function values affected by a change in $\mathbf{x}$ would be the number of tetrahedra containing the vertex v multiplied by six. Let the minimum of the function values obtained at $\mathbf{x}$ be called the *active value*, and the set of function values that obtain that value, the *active set*, be denoted by $\mathcal{A}(\mathbf{x})$.

The action of the function `Smooth` is determined by the particular algorithm chosen, and in this section six algorithms are described.

"Smart" Laplacian Smoothing

The first algorithm is a variant of Laplacian smoothing that relocates the mesh grid point to the geometric center of the adjacent grid points only if the quality of the local mesh is improved according to some mesh quality measure. In that case, the smoothing operator given above has the following action

Compute $f(\mathbf{x}_0)$ and $\mathcal{A}(\mathbf{x}_0)$
Compute $\hat{\mathbf{x}} = \sum_{i \in V} \mathbf{x}_{V_i}/|V|$
Compute $f(\hat{\mathbf{x}})$ and $\mathcal{A}(\hat{\mathbf{x}})$
If $\mathcal{A}(\hat{\mathbf{x}}) > \mathcal{A}(\mathbf{x}_0)$ set $\mathbf{x}_{new} = \hat{\mathbf{x}}$

where $\mathbf{x}_{V_i}$ is the position of the ith adjacent vertex. Computing $\hat{\mathbf{x}}$ is quite inexpensive, and the total time required by this method is dominated by the two function evaluations, $f(\mathbf{x}_0)$ and $f(\hat{\mathbf{x}})$.

Optimization-based Smoothing

Optimization techniques use function and gradient evaluations to find the minimum (or maximum) value that the function obtains in the solution space. The goal of the optimization approach is to determine the position $\mathbf{x}^*$ that maximizes the composite function

$$\phi(\mathbf{x}) \;=\; \min_{1 \leq i \leq n} \; f_i(\mathbf{x}). \tag{1}$$

For most quality measures of interest, the functions $f_i(\mathbf{x})$ are differentiable. However, the composite function $\phi(\mathbf{x})$ has discontinuous derivatives wherever a change occurs in the active set.

We solve this nonsmooth optimization problem using an analogue of the steepest descent method for smooth functions. The search direction $\mathbf{s}$ at each step is computed by solving a quadratic programming problem that gives the direction of steepest descent from all possible convex linear combinations of the gradients in the active set at $\mathbf{x}$. The line search subproblem along $\mathbf{s}$ is solved by predicting the points at which the set of active functions will change based on the first-order Taylor series approximations of the $f_i(\mathbf{x})$. The distance from the current position to the point at which the active sets are predicted to change gives the initial step length α. Standard step acceptance and termination criteria are used to ensure a robust implementation. The action of the smoothing operator for optimization-based smoothing is given in Figure 1.

It has be shown that this technique is equivalent to generalized linear programming techniques by Amenta et al. (1997), and thus the convex level set criterion can be used to determine whether there is a unique solution $\mathbf{x}^*$. Amenta et. al.

$i = 0$
Compute $f(\mathbf{x}_0)$ and $\mathcal{A}(\mathbf{x}_0)$
While (($\mathbf{x}_i \neq \mathbf{x}^*$) and ($\alpha >$ MIN_STEP)
and ($i <$ MAX_ITER) and
($|\mathcal{A}(\mathbf{x}_i) - \mathcal{A}(\mathbf{x}_{i-1})| >$ MIN_IMP))
Compute the gradients $\mathbf{g}_i$
Compute search direction $\mathbf{s}_i$
Compute α
While (STEP_NOT_ACCEPTED)
and ($\alpha >$ MIN_STEP)
Compute $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{s}_i$
Compute $f(\mathbf{x}_{i+1})$ and $\mathcal{A}(\mathbf{x}_{i+1})$
Test for step acceptance
$\alpha = \alpha/2$
Endwhile
$i = i + 1$
Endwhile

Figure 1: The optimization-based smoothing algorithm

(1997) describe the level sets for several mesh quality criteria and found that many of them meet the convexity requirement for unique solutions. We note that similar local optimization-based smoothing methods have been proposed for a variety of optimization procedures and mesh quality measures, e.g. Bank and Smith, Shephard and Georges (1991).

Experimental results demonstrating the effectiveness of the optimization-based method compared with Laplacian smoothing for two- and three-dimensional simplicial meshes are given in Freitag, et al. (1995) and Freitag and Ollivier-Gooch (1996). The optimization-based method was extremely effective at eliminating extremal angles from the mesh in both two and three dimensions, whereas the Laplacian smoother was often unable to significantly improve the most severely distorted elements. These results also showed that more than three sweeps of the mesh offer minimal improvements.

The Combined Approaches

Preliminary experiments in Freitag and Ollivier-Gooch (1996) showed that the most effective and efficient smoothing approach combined the smart Laplacian smoother with the optimization-based algorithm. In this approach, the smart Laplacian smoother was used to adjust every grid point and was followed by the optimization-based algorithm in only the poorest-quality elements. This technique was only twice as computationally expensive as the smart Laplacian smoother used alone and achieved meshes comparable in quality to those obtained when the optimization-based smoother was used for all grid points. In this article we will more fully investigate the combined approach initially presented in Freitag and Ollivier-Gooch (1996), along with three related approaches.

Combined Approach 1. In this technique, the active value of the initial mesh is compared with a user-defined threshold value. If the threshold value is exceeded, the smart variant of Laplacian smoothing is used; otherwise, optimization-based smoothing is performed.

Combined Approach 2. In this technique, smart Laplacian smoothing is used as the first step for every grid point. The active value in the local mesh after this step is compared with a user-defined threshold value. If the active value exceeds the threshold value, the algorithm terminates; otherwise, optimization-based smoothing is performed.

Combined Approach 3. In this technique, the active value of the initial mesh is compared with a user-defined threshold value. If the threshold value is exceeded, no smoothing is performed; otherwise, Laplacian smoothing is used. If the active value still does not exceed the threshold value following Laplacian smoothing, the optimization-based smoother is used.

Combined Approach 4. This technique is similar to the second combined approach except that we use a floating threshold rather than a fixed one. After each smoothing pass, the threshold value for the next pass is set equal to the global minimum active value in the mesh plus a constant. Experiments showed that a constant of five degrees works well in practice for both two and three dimensions.

NUMERICAL EXPERIMENTS

We now compare the effectiveness and computational cost of the Laplacian smoother, the optimization-based smoother, and the four combined approaches given in the preceding section. For each of the combined approaches, four different threshold values, $\theta_T = 5°, 10°, 15°$, and $30°$, were investigated. In all cases, the mesh quality function used to determine the active value is the minimum sine of the angles in the incident elements. Because the sine function is small near the angles of $0°$ and $180°$, this mesh quality measure has the effect of eliminating both large and small angles in the mesh. Effectiveness of the smoothing technique is measured by examining the global minimum and maximum angles/dihedral angles in two/three dimensions. For selected cases, we examine the distribution of mesh angles to analyze the overall mesh improvement. For all cases we report the number of cells smoothed with the Laplacian technique and the number of cells smoothed with the optimization-based technique. Computational cost is measured by the average time required to smooth each vertex in the mesh. All computational experiments were performed on a 110 MHz SPARC 5 workstation.

Meshes Generated from Random Point Sets

The first suite of tests was performed on meshes generated

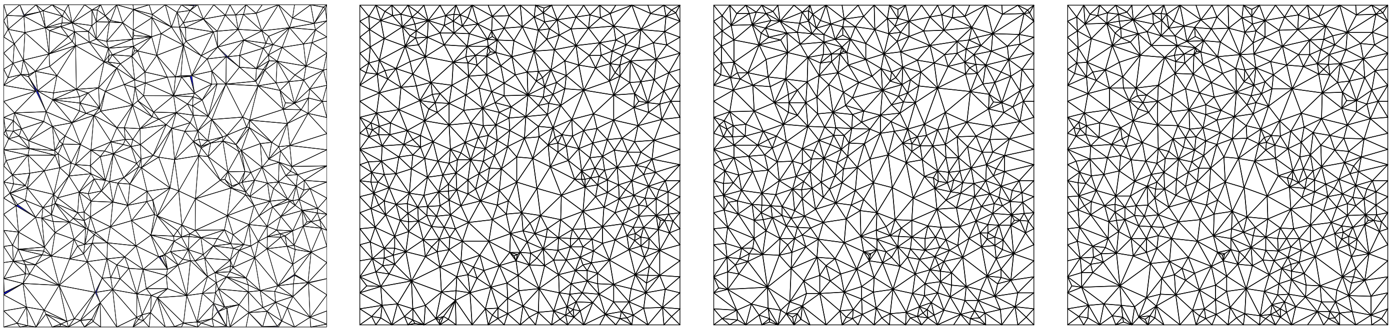

Figure 2: The original two-dimensional Delaunay mesh generated from 500 random points and the same mesh after three passes of Laplacian smoothing, optimization-based smoothing, and the second combined approach with a threshold of $30°$

from random point sets. For the two-dimensional test case, we considered a Delaunay triangulation of 500 random points in the unit square. The current version of the smoothing code adjusts only interior vertices; and to eliminate the problem of poor aspect ratio triangles on the boundary, we placed 20 equally spaced grid points on each side of the square. This mesh, `rand2D`, was generated by using the Carnegie Mellon University Triangle code (Shewchuk, 1996) and has a total of 580 points and 1078 triangles.

In three dimensions, we use the unit cube geometry with points incrementally inserted at random in the interior. Each point is connected to the vertices of the tetrahedron containing it, with points near an existing face or edge in the tetrahedralization projected onto that face or edge. This mesh has 1086 points approximately equally distributed through the domain and 5104 tetrahedra. This test case was also considered in Freitag and Ollivier-Gooch (1996), which focused on the evaluation of swapping and smoothing mesh improvement techniques used separately and together. It was found that swapping the faces of the original mesh before applying the smoothing technique is critical in obtaining good-quality meshes. Therefore, we swap the faces of the initial mesh as follows using an in-sphere criterion followed by a maxmin dihedral angle criterion and use this mesh as the initial mesh, `rand3D`, for the numerical experiments in this section.

Figure 2 shows the original two-dimensional mesh, rand2D, and the same mesh after three passes of Laplacian smoothing, optimization-based smoothing, and the second combined approach with a 30° threshold value. Each smoothing algorithm considerably improves the mesh, and in Table 1 we summarize the smoothing results.

In each case the initial angle of 0.714° is improved to at least 10° and in some cases is increased to more than 18° after three passes of mesh smoothing. For this mesh, the optimization-based technique is approximately five times more expensive than the smart Laplacian smoother. In general increasing the threshold value in the combined approaches corresponds to an increase in both the global minimum angle and the computational cost. We note that obtaining a minimum angle of 18° with the first combined approach costs nearly as much as using optimization alone, because so many elements have minimum angles less than 30°. The second and third approaches are much more cost effective because using Laplacian smoothing eliminated the need to do optimization for more than half of the grid points compared to the first combined approach. No tested threshold value for C4 yielded the highest quality mesh. However, the $\theta_T = 5°$, $10°$, and $15°$ cases yield comparable or better results than their counterparts in the other combined approaches.

Table 1: Mesh quality improvement for rand2D

Case	θ_T	Min. Angle	Max. Angle	Num. Lap.	Num. Opt.	Time (ms)
Orig	—	.714°	175.92°	—	—	—
Lap	—	10.50°	156.2°	1500	0	.440
Opt	—	18.20°	143.5°	1500	0	2.08
C1	5°	10.37°	156.3°	1457	43	.503
C1	10°	12.24°	149.1°	1337	163	.678
C1	15°	15.95°	148.0°	1107	393	.984
C1	30°	18.20°	143.5°	258	1242	1.97
C2	5°	10.47°	156.3°	1500	5	.458
C2	10°	11.52°	154.8°	1500	23	.487
C2	15°	16.28°	145.8°	1500	54	.542
C2	30°	18.55°	140.5°	1500	593	1.17
C3	5°	5.26°	166.0°	23	5	.264
C3	10°	10.01°	155.1°	65	24	.304
C3	15°	15.00°	148.8°	132	58	.366
C3	30°	18.55°	140.5°	749	609	1.10
C4	5°	16.74°	143.9°	1500	11	.479
C4	10°	12.22°	153.4°	1500	27	.509
C4	15°	16.28°	145.8°	1500	56	.611
C4	30°	15.97°	145.5°	1500	358	.936

The results for `rand3D` are given in Table 2. In this case the Laplacian smoother used alone improved the minimum angle only to 0.471°, whereas the optimization-based smoother and combined approaches increase the minimum angle to between 5° and 9.8° In three dimensions, this increase in quality using the optimization-based smoother alone is obtained at approximately ten times the cost of the smart Laplacian smoother used alone. For each of the combined approaches, a minimum angle of

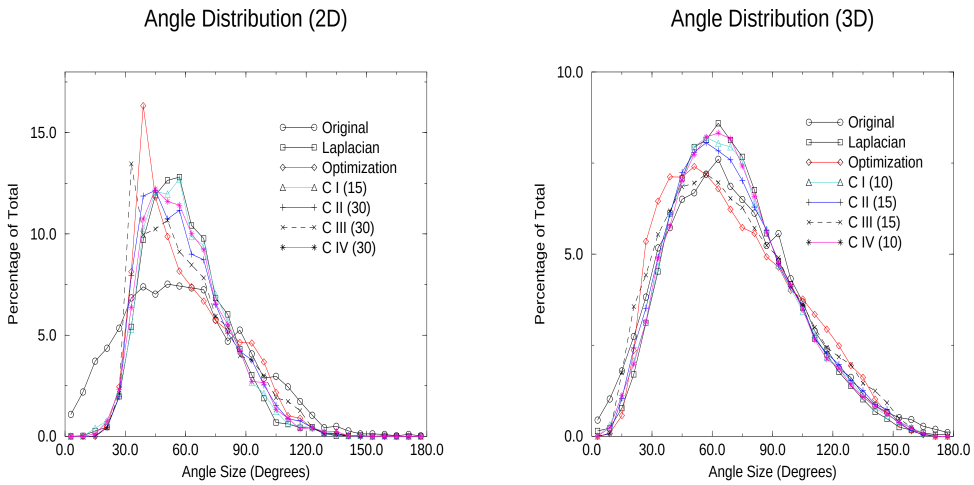

Figure 3: The distribution of angles in the original mesh and the same mesh after three passes of the indicated smoothing techniques

Table 2: Mesh quality improvement for rand3D

Case	θ_T	Min. Dihed.	Max. Dihed.	Num. Lap.	Num. Opt.	Time (ms)
Orig	—	10^{-6}°	180.0°	—	—	—
Lap	—	.471°	179.21°	2808	0	2.03
Opt	—	7.88°	169.4°	0	2808	21.3
C1	5°	5.06°	172.6°	2722	86	2.69
C1	10°	8.21°	169.6°	2395	413	5.08
C1	15°	9.83°	168.0°	1883	925	9.69
C1	30°	7.88°	169.4°	47	2761	21.7
C2	5°	5.07°	172.6°	2808	38	2.38
C2	10°	8.35°	169.3°	2808	191	3.67
C2	15°	8.61°	167.6°	2808	531	6.38
C2	30°	8.79°	168.3°	2808	2704	20.7
C3	5°	5.02°	174.7°	55	39	1.39
C3	10°	7.63°	169.5°	296	222	3.00
C3	15°	8.73°	169.3°	717	579	5.91
C3	30°	8.79°	168.3°	2757	2704	20.6
C4	5°	8.12°	168.2°	2808	104	2.93
C4	10°	8.31°	170.0°	2808	191	3.75
C4	15°	9.78°	168.8°	2808	345	5.10
C4	30°	8.64°	168.8°	2808	953	9.71

approximately 8° is obtained for a threshold value of 5° or 10°. The third and fourth combined approaches with threshold values of ten and five degrees, respectively, are the most cost effective strategies for obtaining a high quality mesh. Fewer than one tenth the grid points require optimization-based smoothing, and in these cases the cost is roughly 3 ms per smoothing call (compared with 2.2 ms for Laplacian smoothing and 20.5 ms for optimization-based smoothing).

Complete angle distributions for the random meshes are shown in Figure 3. In each graph we show the percentage of angles in each six degree bin for the original mesh and the same mesh after three passes of the Laplacian and optimization-based smoothing techniques. In addition we include the angle distributions for each combined technique at the threshold value that corresponds to roughly two times the cost of Laplacian smoothing. It is interesting to note that although the optimization-based smoothing techniques more effectively eliminate extremal angles in the meshes in both two and three dimensions, the averaging technique used in Laplacian smoothing produces more elements that tend toward equilateral (note the peaks near 60° in 2D and 72° in 3D). As expected, the angle distributions for the first and second combined approaches fall between the Laplacian and optimization-based smoothing techniques. The third combined approach effectively eliminates the extremal angles, but the overall distribution is not significantly improved because only selected grid points are adjusted. In fact, in two dimensions there is a peak at the threshold value of 30° for the C3 curve.

Application Meshes

We now analyze the performance of each of the smoothing algorithms for various two- and three-dimensional application meshes. We compare cost and effectiveness of the the smart Laplacian smoother, the optimization-based smoother and the combined approaches. For each combined technique we include the case with the smallest threshold value that gives results comparable to the optimization-based approach.

The application meshes are generated for six different geometries meshed with different techniques. For two dimensions, we use the following test cases:

1. a four element airfoil meshed with the Carnegie Mellon Triangle Delaunay mesh generation code (Shewchuk, 1996)

2. a single element airfoil meshed with the QMG quadtree mesh generator (Vavasis and Mitchell, 1996),

3. a branched channel geometry meshed with an anisotropic wavefront algorithm used by Tim Barth at NASA Ames (Barth, 1994). For this case we smooth only the subset of grid points that are not incident to anisotropic el-

ements. In this way, transition regions between anisotropic and isotropic elements can be significantly improved without affecting boundary layer elements.

For three dimensions, we have

1. a tire incinerator meshed with the point insertion GRUMMP software Ollivier-Gooch (1996),
2. a polygonal approximation to a teapot provided with the GEOMPACK Delaunay mesh generation software (Joe, 1991), and
3. and tube geometry provided with the QMG octree mesh generation software (Vavasis and Mitchell, 1996).

A summary of the initial mesh sizes and qualities is given for the application meshes in Table 3.

Table 3: Application meshes

Geom/Mesher	Dim.	Num. Vert.	Num. Elem.	Min. Angle	Max. Angle
Air-4/Triangle	2	5506	10068	15.04°	144.9°
Chan./Barth	2	8547	16414	.983°	173.7°
Air-1/QMG	2	2644	4790	3.81°	162.2°
Incin./Grummp	3	2570	11098	10^{-6}°	180.0°
Pot/G-Pack	3	2660	10765	15.84°	139.1°
Tube/QMG	3	2846	11540	1.71°	177.7°

The smoothing results are given in Table 4. In each case the optimization-based method yields a greater increase in the mimimum angle than the Laplacian smoother does. As was the case with the random meshes, the corresponding increase in computational cost is approximately a factor of four in two dimensions and a factor of ten in three dimensions. For all but one case (C4 applied to the tire incinerator), the combined approaches are able to obtain the same minimum angle as optimization-based smoothing used alone at a fraction of the cost. For the fixed threshold methods, the threshold value required for the QMG, GRUMMP, and Barth meshes was less than 30° for all of the combined approaches. The Triangle and GEOMPACK initial meshes had a minimum angle that was greater than 15° and therefore required a 30° threshold to match the optimization-based improvements. One advantage that the C4 approach has it that a low threshold value of 5 or 10 degrees can be used initially, and it will automatically be increased or decreased as necessary to target only the worst angles in the mesh.

In two dimensions, the cost of the combined approaches in decreasing order is C1, C2, C4, and C3, which corresponds to a decreasing total number of function evaluations. In fact, for these test cases, the third combined approach required less time than the Laplacian smoother because so few grid points required smoothing. In three dimensions, the ordering for cost effectiveness of the methods is mixed, but the C4 approach is always obtains a good mesh at a low computational cost. We note that the cost of the third approach can be further reduced by evaluating only grid points that changed location or are adjacent to grid points that changed location in the previous smoothing pass.

Table 4: Mesh quality improvement for the application meshes

2D						
Geom	Tech.	Min. Angle	Max. Angle	Num. Lap.	Num. Opt.	Time (ms)
Air-4	Orig	15.04°	144.9°	—	—	—
	Lap	15.06°	138.3°	13668	—	.460
	Opt	19.07°	132.7°	—	13668	1.68
	C1/30	19.06°	132.7°	12403	1265	.650
	C2/30	19.10°	132.7°	13668	1098	.615
	C3/30	19.11°	132.7°	1265	1105	.434
	C4/5	19.11°	133.0°	13668	201	.550
Chan	Orig	.983°	173.7°	—	—	—
	Lap	2.21°	169.9°	15090	—	.474
	Opt	2.94°	169.7°	—	15090	1.83
	C1/5	2.98°	172.5°	14988	102	.476
	C2/5	2.98°	172.1°	15090	83	.476
	C3/5	2.95°	170.5°	96	82	.274
	C4/5	2.98°	169.6°	15090	131	.459
Air-1	Orig	3.81°	162.2°	—	—	—
	Lap	8.10°	155.1°	6438	—	.463
	Opt	12.37°	139.1°	—	6438	1.60
	C1/15	12.46°	144.1°	6245	193	.535
	C2/15	12.66°	144.0°	6438	125	.518
	C3/15	12.18°	149.6°	183	128	.321
	C4/10	12.03°	147.7°	6438	88	.484

3D						
Geom	Tech	Min. Angle	Max. Angle	Num. Lap.	Num. Opt.	Time (ms)
Incin	Orig	10^{-6}°	180.0°	—	—	—
	Lap	.657°	178.8°	3966	—	2.36
	Opt	9.56°	163.6°	—	3966	22.5
	C1/10	9.36°	167.2°	3852	114	3.27
	C2/10	9.11°	168.4°	3966	49	2.83
	C3/15	9.75°	165.1°	288	164	2.70
	C4/10	7.98°	166.9°	3966	62	2.76
Pot	Orig	15.84°	139.1°	—	—	—
	Lap	16.76°	138.2°	3375	—	2.60
	Opt	19.64°	148.5°	—	3375	19.8
	C1/30	19.64°	154.4°	3168	207	4.04
	C2/30	19.67°	143.3°	3375	159	3.73
	C3/30	19.66°	143.1°	212	171	2.69
	C4/5	19.66°	142.5°	3375	13	2.64
Tube	Orig	1.71°	177.7°	—	—	—
	Lap	1.17°	177.7°	3654	—	2.26
	Opt	4.77°	174.9°	—	3654	21.4
	C1/10	4.36°	175.5°	3080	574	6.24
	C2/15	4.64°	175.3°	3654	1225	9.93
	C3/15	4.37°	175.5°	1382	1234	9.39
	C4/5	4.69°	175.3°	3654	209	3.63

CONCLUDING REMARKS

In this article we presented four smoothing techniques that combined the low computational cost of the Laplacian smoothing technique with the effectiveness of an optimization-based approach. Numerical experiments in both two and three dimensions show that each approach is as effective at eliminating extremal angles in the mesh as optimization-based smoothing used alone at a fraction of the cost. In addition, the C1, C2, and C4 approaches created more equilateral elements than optimization-based smoothing in both two and three dimensions. We conclude that these techniques may generate higher-quality meshes than either Laplacian or optimization-based smoothing used alone. In contrast, the C3 approach had a relatively poor angle distribution because only a small number of the grid points were relocated. The computational costs of the C3 technique were correspondingly small, and we found that, in general, this approach was the least expensive combined approach. In fact, it was often computationally cheaper than Laplacian smoothing used on every grid point.

Several enhancements are being incorporated into the mesh improvement software to increase its effectiveness and efficiency. The current software uses mesh smoothing to improve the quality of the interior mesh once the surface mesh has been generated. We plan to add surface mesh-smoothing capabilities to the optimization-based algorithm by incorporating additional constraints to bind the free vertex to the boundary surfaces. We are also interested in examining optimization-based smoothing with other measures including aspect ratio and solid angles and in developing smoothing measures appropriate for use on anisotropic meshes.

ACKNOWLEDGMENTS

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

REFERENCES

Amenta, N., Bern, M., and Eppstein D., 1997, "Optimal point placement for mesh smoothing," To appear in *8th ACM-SIAM Symp. on Discrete Algorithms*, New Orleans, LA.

Amezua, E., Hormaza, M., Hernandez, A., and Ajuria, M., 1995, "A method of the improvement of 3D solid finite-element meshes," *Advances in Engineering Software*, Vol 22, pp 45–53.

Babuska, I. and Aziz, A., 1976, "On the angle condition in the finite element method," *SIAM Journal on Numerical Analysis*, Vol 13,pp 214–226.

Bank, R. and Smith, R., "Mesh smoothing using a posteriori error estimates," to appear *SIAM Journal on Numerical Analysis.*

Barth, T., 1994, "Aspects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations,", Lecture Series, *von Karman Institute for Fluid Dynamics.*

Canann, S., Stephenson, M., and Blacker, T., 1993, "Optismoothing: An optimization-driven approach to mesh smoothing," *Finite Elements in Analysis and Design*, Vol 13, pp 185–190.

Field, D., 1988, "Laplacian smoothing and Delaunay triangulations," *Communications and Applied Numerical Methods*, Vol 4, pp 709–712.

Freitag, L. and Ollivier-Gooch, C., 1996, "A comparison of tetrahedral mesh improvement techniques," In *Proceedings of the Fifth International Meshing Roundtable*, Sandia National Laboratories, pp 87–100.

Freitag, L., Jones, M., and Plassmann, P., 1995, "An efficient parallel algorithm for mesh smoothing," In *Proceedings of the Fourth International Meshing Roundtable*, Sandia National Laboratories, pp 47–58.

Fried, I., 1972, "Condition of finite element matrices generated from nonuniform meshes," *AIAA Journal*, Vol 10, pp 219–221.

Joe, B., 1991, "GEOMPACK - a software package for the generation of meshes using geometric algorithms," *Advanced Engineering Software*, Vol 56, Num 13, pp 325–331.

Lo, S., 1985, "A new mesh generation scheme for arbitrary planar domains," *International Journal for Numerical Methods in Engineering*, Vol 21, pp 1403–1426.

Ollivier-Gooch, C., 1996, Alpha Release of the GRUMMP software, http://www.mcs.anl.gov/Projects/SUMAA/

Parthasarathy, V. and Kodiyalam, S., 1991, "A constrained optimization approach to finite element mesh smoothing," *Journal of Finite Elements in Analysis and Design*, Vol 9, pp 309–320.

Shephard, M. and Georges, M., 1991, "Automatic three-dimensional mesh generation by the finite octree technique," *International Journal for Numerical Methods in Engineering*, Vol 32, pp 709–749.

Shewchuk, J., 1996, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," In *Proceedings of the First Workshop on Applied Computational Geometry*, ACM, Philadelphia, Pennsylvania, pp 124–133.

Vavasis, S. and Mitchell, S., 1996, "Quality mesh generation in higher dimensions," Technical Report ANL/MCS-P633-1296, Argonne National Laboratory, Argonne, IL.