COMPUTATIONAL STUDY OF THE EFFECT OF UNSTRUCTURED MESH QUALITY ON SOLUTION EFFICIENCY

Michael Batdorf,* Lori A. Freitag,† and Carl Ollivier-Gooch‡

Abstract

It is well known that mesh quality affects both efficiency and accuracy of CFD solutions. Meshes with distorted elements make solutions both more difficult to compute and less accurate. In this article, we review a recently proposed technique for improving mesh quality as measured by element angle (dihedral angle in three dimensions) using a combination of optimization-based smoothing techniques and local reconnection schemes. Typical results that quantify mesh improvement for a number of application meshes are presented. We then examine effects of mesh quality as measured by the maximum angle in the mesh on the convergence rates of the commonly used GMRES and Multigrid solvers. Numerical experiments are performed that quantify the cost and benefit of using mesh optimization schemes for incompressible and weakly compressible flow over a cylinder.

Keywords. Mesh Improvement, Mesh Smoothing, Convergence, Solution Efficiency

1 Introduction

Finite element and finite volume techniques in computational fluid dynamics require that the computational domain be decomposed into simple geometric elements, typically triangles and quadrilaterals in two dimensions and tetrahedra and hexahedra in three dimensions. This decomposition can often be achieved automatically by using available mesh generation tools. Unfortunately, meshes generated in this way can contain poorly shaped or distorted elements, which cause numerical difficulties during the solution process. For example, we know that as element angles become too large, the discretization error in the finite element solution increases;[3] and as angles become too small, the condition number of the element matrix increases.[15] Thus, for meshes with highly distorted elements, the solution is both less accurate and more difficult to compute.

We recently introduced a two-pronged approach for effectively improving the quality of triangular and tetrahedral meshes based on local reconnection schemes and a new optimization-based mesh smoothing technique[14, 12]. In this article we briefly review these mesh optimization procedures and present typical results for a number of application meshes. In the course of those numerical experiments, we identified particular combinations of techniques that resulted in the greatest improvement to mesh quality, and we summarize several recommendations offered in Freitag and Ollivier-Gooch.[13]

The goal of our current research is to quantify the effects of poor mesh quality on solution efficiency for CFD applications. Toward this end, we perform a detailed examination of two test problems: incompressible and weakly compressible flow over a cylinder. For the first case, we analyze a number of numerical experiments that quantify the convergence rate of the solution technique for high-quality meshes, show how this rate is adversely affected by poor element quality, and finally show that the total time required to improve the mesh and solve the problem on the improved mesh is often less than the time required to find an accurate solution on a poor-quality mesh. Our results show that the point at which the total time associated with solution on an improved mesh is less than the solution time on a poor-quality mesh is application and solution technique dependent. We examine in detail three GMRES iterative solvers and find that as the number of grid points increases and/or the mesh quality decreases, mesh optimization techniques become increasingly beneficial. For the second test case, we start with a random mesh for which a multigrid solu-

*Research Intern, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60430. batdorf@mcs.anl.gov

†Assistant Scientist, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60430. freitag@mcs.anl.gov, Associate Member AIAA.

‡Assistant Professor, Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4 Canada. cfog@mech.ubc.ca. Associate Member AIAA.

tion technique does not converge and show that combined swapping and smoothing improve the mesh enough to obtain a convergent solution.

The remainder of the article is organized as follows. In Section 2, we briefly review the mesh smoothing and local reconnection techniques used for mesh improvement. We then present results that show the improvement of several application meshes using a combination of swapping and smoothing, and we review the recommendations for mesh improvement given in Freitag and Ollivier-Gooch.[13] In Section 3, we describe the test cases and solution techniques, followed by several numerical experiments that quantify the effects of mesh quality on convergence behavior.

2 Mesh Improvement Techniques

Much research has been done in the area of improving mesh quality through a variety of techniques, including

1. point insertion/deletion to refine or coarsen a mesh,[22, 24]
2. local reconnection to change mesh topology for a given set of vertices,[16, 18] and
3. mesh smoothing to relocate grid points without changing mesh topology.[2, 9, 23]

We recently introduced a two-pronged approach for effectively improving the quality of tetrahedral meshes based on local reconnection schemes and a new optimization-based mesh smoothing technique.[12] We now briefly review these procedures.

2.1 Mesh Smoothing

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

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

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 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 let 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. In this section we briefly describe several different methods; more details can be found in Freitag et al.,[14] Freitag and Ollivier-Gooch,[12] and Freitag.[11]

"Smart" Laplacian Smoothing. A variant of Laplacian smoothing 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. Computing $\mathbf{x}_{new}$ by this method is quite inexpensive, and the total time required is dominated by the two function evaluations needed to determine the initial quality of the mesh and the resulting quality of the mesh.

Optimization-based Smoothing. In Freitag et al.[14] and Freitag and Ollivier-Gooch,[12] a low-cost, optimization-based alternative to Laplacian smoothing was proposed. This optimization technique uses function and gradient evaluations to find the minimum (or maximum) value that a mesh quality measure obtains in the solution space. The goal of the optimization approach is to determine the position $\mathbf{x}^*$ that maximizes the composite function

$$\phi(\mathbf{x}) = \min_{1 \le i \le n} f_i(\mathbf{x}). \qquad (1)$$

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.

Amenta et al. have shown that this technique is equivalent to generalized linear programming techniques.[1] Thus, the convex level set criterion can be used to determine whether there is a unique solution $\mathbf{x}^*$. Amenta et al. describe the level sets for several mesh quality criteria and show that many of them meet the convexity requirement for unique solutions.

Other optimization-based smoothing techniques have been developed by researchers in the mesh generation and computational geometry communities. These methods differ primarily in the optimization procedure used or in the quantity optimized. For example, Bank[6] and Shephard and Georges[25] propose similar techniques for triangles and tetrahedra, respectively. In these methods, an element shape quality measure, $q(t)$, is defined based on a ratio of element area (volume) to side lengths (face areas). In each case, $q(t)$ is equal to one for equilateral elements and is small for distorted elements. The free vertex is moved along the line that connects its current position to the position that makes $q(t)$ equal to one for the worst element in the local submesh. The line search in this direction is terminated when two elements have equal shape measure. We note that this does not necessarily guarantee that the optimal local solution has been found.

All the techniques mentioned above optimize the mesh according to element geometry. In contrast, Bank and Smith[5] propose two smoothing techniques to minimize the error in finite element solutions computed with triangular elements with linear basis functions. Both methods use a damped Newton's method to minimize the interpolation error or the a posteriori error estimates for an elliptic partial differential equation. The quantity minimized in both of these cases requires the computation of approximate second derivatives for the finite element solution as well as the shape function $q(t)$ for triangular elements mentioned above.

Combined Approaches. In Freitag[11] and Freitag and Ollivier-Gooch[13] experiments showed that the most effective and efficient smoothing approach combined the smart Laplacian smoother with the optimization-based algorithm. Four related combination approaches, which used Laplacian smoothing as a first step followed by optimization-based smoothing for the worst quality elements, were compared with results obtained with smart Laplacian and optimization-based smoothing used alone. Test meshes for several application geometries in both two and three dimensions were obtained by using a variety of meshing techniques. In all cases, the mesh quality function used to determine the active value was the minimum sine of the angles (dihedral angles in three dimensions) 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 was measured by examining the global minimum and maximum angles/dihedral angles in two/three dimensions.

In those experiments we found that the optimization-based method yielded a greater increase in the minimum angle than the Laplacian smoother did. In fact, the Laplacian smoother often failed to eliminate extremal angles in the mesh. The Laplacian smoother yielded a greater number of near equilateral triangles and tetrahedra due to the averaging effect of the operator. The increase in computational cost associated with the optimization-based smoother compared with the Laplacian smoother was approximately a factor of four in two dimensions and a factor of ten in three dimensions. For all but one case, the combined approaches were able to obtain the same minimum angle as optimization-based smoothing used alone at a fraction of the cost. In addition, the combined approaches created more equilateral elements than optimization-based smoothing used alone. We concluded that the combined techniques generally generate higher-quality meshes than either Laplacian or optimization-based smoothing used alone. The cost the combined approaches varied depending on the number of optimization steps performed. We note that more than three sweeps of the mesh offer minimal improvements for the meshes and methods tested.

All the mesh smoothing results presented in this article use three to five passes of a combined approach in which smart Laplacian smoothing is used a first step to improve all elements, followed by optimization-based smoothing for the worst-quality elements (those with angles less than 30 degrees in 2D and 15 degrees in 3D). This combined approach has a computational cost of roughly two times the cost of smart Laplacian smoothing. The quality criterion used is maxmin sine.

2.2 Local Mesh Reconfiguration Techniques

Local mesh reconfiguration techniques change the connectivity of part of a simplicial mesh to improve mesh quality. For triangles, these techniques are based on edge swapping, and for tetrahedra, these techniques can be divided into two classes: face swapping and edge swapping.

Face swapping reconnects the tetrahedra separated by a single interior face. Each interior face in a tetrahedral mesh separates two tetrahedra made up of a total of five points. A large number of non-overlapping tetrahedral configurations are possible with these five points, but only two can be legally reconnected. These two cases are shown in Figure 1. On the left is a case in which either two or three tetrahedra can be used to fill the convex hull of a set of five points. Switching from two to three tetrahedra requires the addition of an edge interior to the convex hull. On the right of the figure is a configuration in which two tetrahedra can be exchanged for two different ones. The shaded faces in the figure are coplanar, and swapping exchanges the diagonal of the coplanar quadrilateral. The two coplanar faces must either be boundary faces or be backed by another pair of tetrahedra that can be swapped two for two. Otherwise, the new edge created by the two-for-two swap will not be conformal.

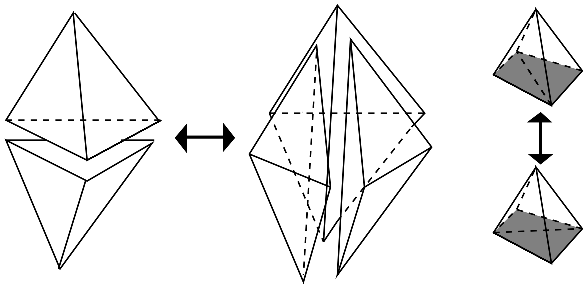

Figure 1: Swappable configurations of five points in three dimensions

Because each reconfigurable case has only two valid configurations, a quick comparison to find the one with the higher quality is possible. If the higher-quality configuration is not already present, reconnection is performed to obtain it. In the case of configurations of equal quality, we select the two-tet configuration when choosing between two- and three-tet configurations, and we choose not to swap in the two-for-two reconfiguration case.

Edge swapping reconfigures N tetrahedra incident on an edge of the mesh by removing that edge and replacing the original N tetrahedra by $2N-4$ tetrahedra. The reconfiguration is performed only if every new tetrahedron has better quality than the worst of the N original tetrahedra. In principal, edge swapping could be used to replace, for example, 12 tetrahedra with 20, but in practice we have found that the number of transformations that improve the mesh declines dramatically with increasing N. In particular, for practical cases 7-for-10 transformations are rare, and consequently we have not investigated these techniques for $N > 7$. Edge swapping is used in two ways: first, as a supplement to face swapping, and second as a separate procedure specifically designed to remove poor-quality tetrahedra. More details can be found in Freitag and Olliver-Gooch.[13]

We use two geometric quality measures to determine whether to locally reconnect a tetrahedral mesh: the minmax angle (or sine of the angle) criterion and the in-sphere criterion. The minmax angle (sine) criterion chooses the configuration that minimizes the maximum dihedral angle (sine of) of the tetrahedra formed by the five points in the two tets incident on a face. The in-sphere criterion selects the configuration in which no tetrahedron formed by four of the five points contains the other point in its circumsphere. This leads to a locally Delaunay tetrahedralization in the sense that there is no face in the mesh with incident cells violating the in-sphere criterion that are reconfigurable. For either criterion, however, the optimum reached by this face-swapping algorithm will probably be local rather than global. Recent work by Joe[18] describes a more advanced technique for improving mesh quality by local transformations. This approach notwithstanding, it is not known whether the global optimum can be reached by *any* series of local transformations.

2.3 Mesh Improvement Results

In Freitag and Ollivier-Gooch,[12] we presented results for mesh improvement using in-sphere and minmax dihedral angle face swapping and Laplacian and optimization-based smoothing techniques for three-dimensional tetrahedral meshes. For two random meshes and three application meshes, we showed that neither swapping nor smoothing was able to make significant improvements in mesh quality when used alone. The face-swapping techniques fail to remove very small and very large angles, and the smoothing techniques fail to improve the overall distribution of angles because they cannot change local mesh connectivity. However, we showed for these test cases that the cumulative improvement obtained when combining in-sphere and minmax swapping

Table 1: Mesh improvement for three application meshes

Case	Min Dihed	Max Dihed	% Dihedral Angles <			% Dihedral Angles >		
			6°	12°	18°	162°	168°	174°
Tire incinerator before	0.66°	178.88°	0.11	0.54	1.27	0.074	0.035	0.0075
Tire incinerator after	13.67°	159.82°	0	0	0.038	0	0	0
T-fire boiler before	0.048°	179.86°	0.16	0.50	0.99	0.24	0.12	0.037
T-fire boiler after	5.61°	174.15°	0.0013	0.029	0.11	0.019	0.0071	0.00045
ONERA M6 wing before	0.0066°	179.984°	0.78	1.63	2.85	0.57	0.41	0.23
ONERA M6 wing after	0.098°	179.76°	0.16	0.66	1.46	0.17	0.076	0.018

followed by the combined smoothing technique results in very high quality meshes.

Two of the application meshes were generated in the interior of a tangentially fired (t-fired) industrial boiler and a tire incinerator, respectively. The third application mesh was generated around the ONERA M6 wing attached to a flat wall. Each mesh was generated by using point insertion techniques combined with face swapping.[12]

The improvement in mesh quality achieved for each of the three application meshes is shown in Table 1. For each case we show the minimum and maximum angle in the mesh before and after mesh improvement, as well as the percentage of angles in the lower and upper three 6° bins. For all three cases, mesh quality is improved significantly. The final mesh quality differs dramatically among the three cases because of the initial topology and point distribution of the meshes. For example, the M6 wing mesh began with a very large number of poor dihedral angles in adjacent tetrahedra. Clustering of bad tetrahedra was fairly common in our initial meshes, with the worst cells often sharing vertices, edges, or even faces. While smoothing improved many tetrahedra, some could not be improved without making a neighboring cell worse, and so no improvement was made.

These experiments led to several general recommendations for the improvement of tetrahedral meshes:

- Never use the in-sphere criterion during the final pass of face swapping. It performs poorly in practice with respect to extremal angles. Edge swapping is a beneficial supplement to face swapping and should be used.

- Meshes whose connectivity has not been improved during generation should be reconnected using in-sphere face swapping, followed by face and edge swapping by using the maxmin sine of dihedral angle criterion. For meshes that have initially reasonable connectivity, only the second pass need be performed.

- The local reconnection schemes should be followed by two passes of a combined Laplacian/optimization-based smoothing technique, followed by an edge-swapping procedure to remove the worst tetrahedra from the mesh, and finishing with two more passes of smoothing. Quality criteria that tend to eliminate small angles in the mesh are more effective than criteria that tend to eliminate large angles.

3 Numerical Experiments

We now examine the effect of mesh quality on the convergence rates of commonly used solution techniques for incompressible and weakly compressible, two-dimensional flow applications. We consider two test cases; the first is incompressible flow in a channel around a cylinder, and the second is weakly compressible flow over a cylinder at Mach 0.3. The solution for the first test case is obtained using by linear finite element techniques with exact integration. The linear systems are solved by using the GMRES solvers in the PETSc toolkit for scientific computation[4] with a relative convergence tolerance of 10^{-12}. The solution for the second test case is computed by using an edge-based, vertex-centered finite volume solver for which second-order accuracy is attained through least-squares reconstruction. Results for both cases show that element quality has a significant effect on the convergence rate of the solution procedure and that the total cost of solving the problem an improved mesh, which included the mesh optimization costs, is less than the cost to obtain comparable accuracy on poor-quality meshes.

3.1 Case Study 1: Incompressible Flow over a Cylinder

Our first case study is incompressible flow over a cylinder centered in a channel. The computational domain is four cylinder diameters long and two wide, with a symmetry condition imposed on the upper and lower surfaces. For our test problem we choose the radius of the cylinder to be .25 and the uniform flow to be $U = 1$ in the x-direction. The computational domain is triangulated with the Delaunay mesh generation package Triangle.[26] The geometry and a typical mesh with $N = 800$ grid points are shown in Figure 2.

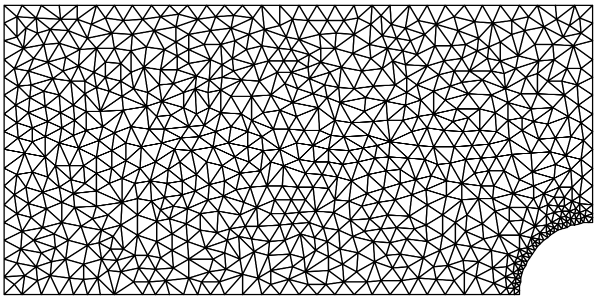

Figure 2: Delaunay mesh for flow over a cylinder (N=800)

Experiment 1: *Convergence of GMRES.* Our first experiment examines the effect of the number of grid points N on the convergence rate of the GMRES solvers. It is well known that the number of iterations for the Incomplete Cholesky conjugate gradient (ICCG) algorithm increases as the number of grid points increases. In particular, for elliptic operators on the unit square, several authors have shown that the number of iterations required by ICCG is proportional to $N^{\frac{1}{2}}$ when a second order central finite difference method is used with the natural ordering of grid points.[10] Although the potential flow problem can be solved using ICCG, we are interested in developing a more general code for the solution of CFD problems and in understanding the convergence behavior of the more commonly used GMRES iterative techniques.

To empirically obtain the convergence behavior of GMRES on the domain shown in Figure 2, we ran a series of numerical experiments with $N = 200, 400, 800, 1600, 3200, 6400,$ and 12800 grid points in the mesh. Each of these meshes has a minimum angle of approximately 30° and a maximum angle between 110° and 120°. We consider three different solvers: GMRES with no preconditioning (GMRES), GMRES with Jacobi preconditioning (GMRES/Jac), and GMRES with no-fill ILU preconditioning (GMRES/ILU). All are restarted by using the PETSc default value of 30 iterations. Table 2 gives the number of grid points used and the number of iterations required for each of the three solution techniques.

The number of iterations for each of the solution techniques is also given graphically in Figure 3 as a log-log plot. It is clear that in each case the number of iterations is growing as a function of N and that GMRES/ILU requires considerably fewer iterations than the other two techniques. Linear least-squares analysis gives the slopes of these curves to be $s = .907$, $.866$, and $.792$ for GMRES, GMRES/Jac, and GMRES/ILU, respectively. Thus, as N in creases, the number of iterations grows as N^s. We further note that the work required for each iteration is dominated by a matrix-vector multiplication that is $\mathcal{O}(N)$ operations for the sparse linear systems generated by the finite element technique. Therefore, the total work required to solve the sys-

Table 2: Number of iterations as a function the number of grid points in the mesh

	Number of Iterations		
N	GMRES	GMRES/Jac	GMRES/ILU
200	199	172	41
400	316	217	79
800	670	567	143
1600	1176	949	218
3200	2093	1571	303
6400	4205	3128	690
12800	8571	5605	1263

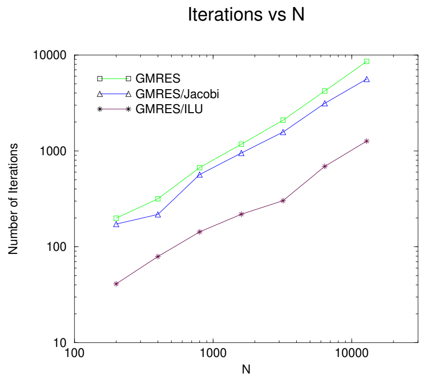

Figure 3: The number of iterations as a function of the number of grid points in the mesh

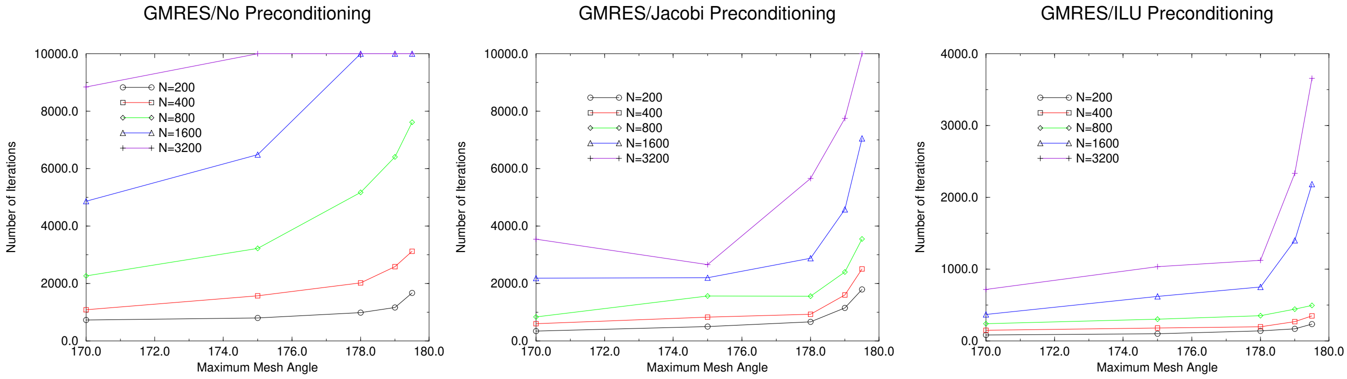

Figure 5: The effect of element quality on the convergence rate of the three solution techniques elements

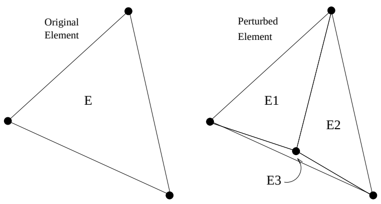

Figure 4: An example of perturbing an element to create poor quality elements

tem is approximately $\mathcal{O}(N^{s+1})$ for each of the iterative techniques.

Experiment 2: *Effect of element quality on convergence.* We use the iteration counts given in Table 2 as a baseline to examine the effect on the convergence rates of element quality as measured by the maximum angle. To control the maximum angle in the mesh, we start with the original meshes created with Triangle and insert a new grid point a distance ϵ from and perpendicular to the midpoint of an edge of 10 percent of the elements. The distance ϵ is chosen to result in an element with the desired maximum angle. An example of this point insertion technique is shown in Figure 4. For each point inserted, two new elements are created so that the final mesh has $N + .1N$ grid points and $2N + .2N$ elements. Using this technique, we created a series of meshes for each original mesh with $N = 200$, 400, 800, 1600, and 3200 grid points. Each series consists of five meshes with poor quality elements whose maximum angles are 170°, 175°, 178°, 179°, and 179.5°, respectively.

In Figure 5 we show the number of iterations required to reach an accuracy comparable to that obtained on the original meshes versus the maximum angle in the mesh for each of the three solution techniques. For each iterative technique, large maximum angles significantly affect the convergence rate, particularly if the maximum angle is 178° or greater. In fact, for the largest values of N and for maximum angles greater than 175° and 179°, GMRES and GMRES/Jac failed to converge to the desired tolerance in less than the maximum number of iterations allowed (10,000). GMRES/ILU performs significantly better and has severely degraded performance only when maximum angles are greater than 178°.

For each N and a maximum angle of 170°, the number of iterations for GMRES is roughly tripled compared with GMRES on the original mesh, more than doubled for GMRES/Jac, and almost doubled for GMRES/ILU. We note that the amount of work required to smooth each mesh is an $\mathcal{O}(N)$ operation, and the following question arises: For what values of N and maximum angle in the mesh is the total cost of smoothing the mesh and solving the problem on the improved mesh less than the cost of obtaining an accurate solution on a poor-quality mesh?

Experiment 3: *Determination of smoothing benefits on solution time.* We address the question given above by comparing the difference in solution times on the poor-quality mesh and on an improved mesh (including the time to improve the mesh). For this experiment, we improve each mesh with three passes of the combined smoothing approach described in Section 2.1. Element quality typically improves to greater than 15° for the minimum angles and less than 140° for the maximum angle. We include the original meshes and their smoothed counterparts so that the maximum angles considered in this experiment range from 110° to 179.5°.

From our first and second experiments, we expect that the benefits of smoothing will be more pronounced as both N and the maximum angle increase. In Figure 6, we plot the difference in the time required to reach convergence on a poor quality mesh and the total time to reach convergence on

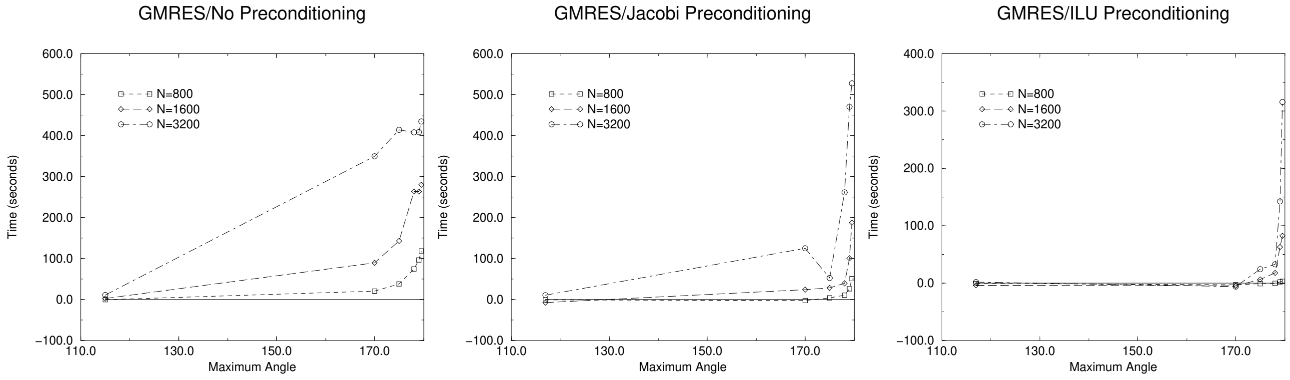

Figure 6: The solution times for convergence of the three iterative techniques for poor quality meshes and the same meshes after smoothing

an improved mesh which includes the time for mesh smoothing for the three values of N for each iterative technique. The amount of time required for mesh smoothing is approximately 3.8, 8, and 15.7 seconds for $N = 800$, 1600 and 3200, respectively. Clearly, as N and the maximum angle in the mesh increase, the amount of time saved by smoothing the mesh also increases. It is also clear that as the iterative technique improves, the value of N for which smoothing is beneficial increases.

3.2 Case Study 2: Compressible Flow over a Cylinder

Our second case study examines the effect of mesh quality on convergence behavior for weakly compressible flow over a cylinder at Mach 0.3. The computational domain is nine cylinder diameters long and three diameters wide, with a symmetry condition imposed on the upper surface. For this experiment, we generated three meshes each beginning with the same random point set with point density falling exponentially with distance from the surface. This distribution corresponds to a constant stretching ratio for structured meshes. The point set contains 2500 interior points and 190 boundary points, which are evenly spaced on the cylinder, inflow, outflow, and upper symmetry plane and exponentially stretched along the lower symmetry plane. The first mesh, the left mesh in Figure 7, was generated by simply inserting the random points into the mesh and swapping using the Delaunay criterion. The smallest angle in this mesh is 0.56° and the largest is 178.86°. The middle mesh in Figure 7 was obtained by performing five passes of optimization-based smoothing on the vertices of the first mesh; this procedure improves the extremal angles to 12.3° and 145.6°. The rightmost mesh in Figure 7 was obtained from the first mesh by performing five passes of smoothing alternating with passes of swapping using the Delaunay criterion; this mesh has extremal angles of 23.2° and 131.9°. Figure 8 compares the overall angle distribution for the three meshes. Clearly, smoothing alone improves the angle distribution, dramatically reducing the number of both small and large angles. When combined with swapping, the improvement is even greater.

Flow around the cylinder was computed using an edge-based, vertex-centered finite volume solver. Second-order accuracy was attained using least-squares reconstruction[8, 20, 21]. Following reconstruction, fluxes were computed by using Roe's flux formula and integrated for each control volume. Time advance was performed using an explicit multistage scheme with multigrid convergence acceleration.[22] In each case, the same three coarse meshes were used to eliminate the effects of coarse mesh convergence behavior on the results. Figure 9 shows the convergence rates for each of the fine meshes. The random mesh fails to converge, falling into a limit cycle with large variations in flow parameters. The smoothed mesh and the smoothed and swapped mesh cases both converge, with the asymptotic rate being about 25% faster for the latter case. In both cases, the mesh optimization procedures required less time than a single cycle of the multigrid solver. In both cases, convergence is limited by behavior near the rear separation point on the cylinder, where local time step is limited by acoustic modes while the solution is changing due to convective modes with a propagation speed of $M = 0.01$ or less.

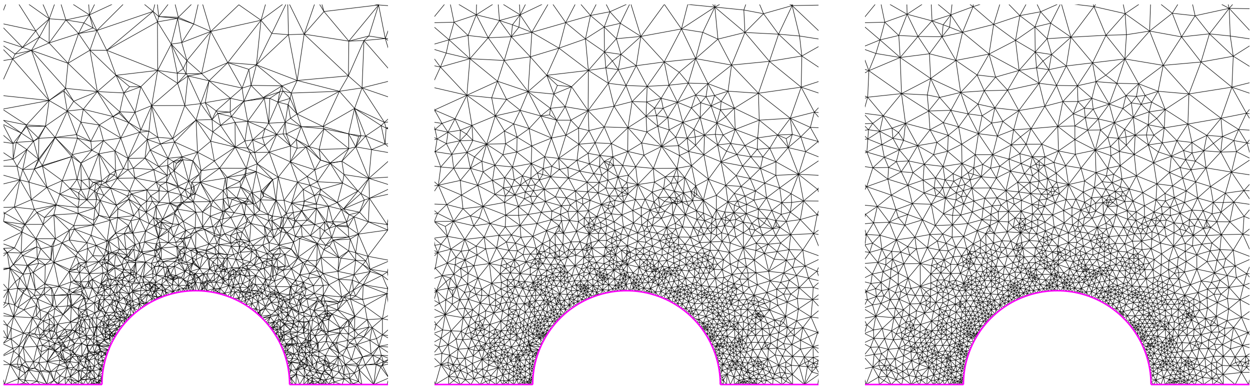

Figure 7: From left to right: the random mesh, smoothed mesh, and smoothed and swapped mesh

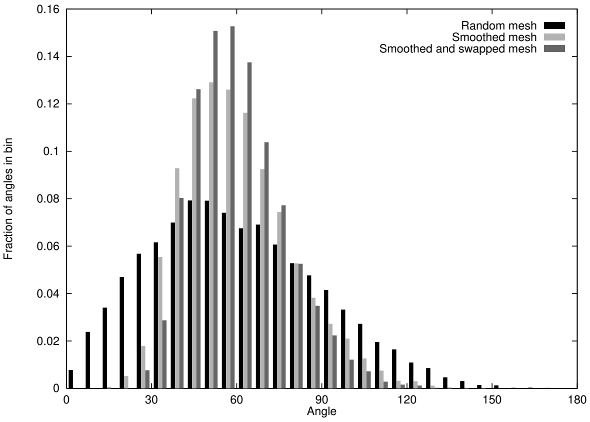

Figure 8: Angle distribution for cylinder meshes

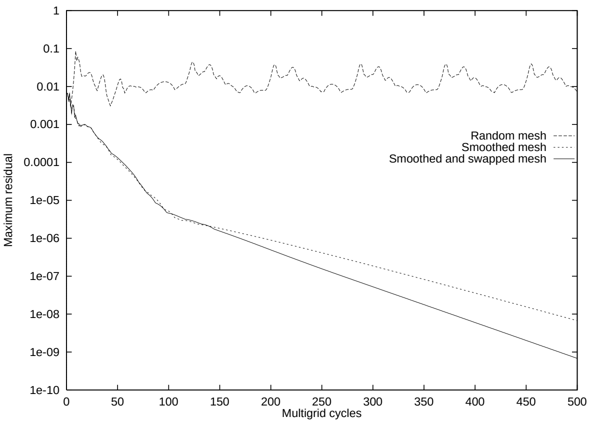

Figure 9: Convergence histories for subsonic cylinder flow

3.3 Summary and Conclusions

In this paper we examined the benefits of using mesh optimization procedures to improve mesh quality for computational fluid dynamics applications. We briefly reviewed several mesh improvement techniques and strategies for triangular and tetrahedral meshes and presented typical results for the improvement of application meshes. We then examined two CFD applications involving flow over a cylinder solved with finite element and finite volume solution techniques. In both cases, we showed that mesh improvement is critical to the efficient solution of the application and that the total cost of the mesh improvement procedures and solution time on a high-quality mesh is often less than the solution time on a poor-quality mesh.

In future work we will extend our study of the ramifications of mesh quality on solution techniques to include more difficult three-dimensional CFD applications and an in-depth analysis of solution accuracy.

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

[1] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. In *8th ACM-SIAM Symp. on Discrete Algorithms*, New Orleans, to appear.

[2] E. Amezua, 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*, 22:45–53, 1995.

[3] I. Babuska and A. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13:214–226, 1976.

[4] Satish Balay, William Gropp, Lois Curfman McInnes, and Barry Smith. PETSc 2.0 users manual. Technical Report ANL-95/11 – Revision 2.0.17, Argonne National Laboratory, October 1996.

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

[6] Randy Bank. *PLTMG: A Software Package for Solving Ellipitc Parital Differential Equations, Users' Guide 7.0*, volume 15 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1994.

[7] Tim Barth. Aspects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations, 1994. von Karman Institute for Fluid Dynamics Lecture Series 1994-05.

[8] Timothy J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. *AIAA paper*, 93-0668, January, 1993.

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

[10] Tony F. Chan and Howard C. Elman. Fourier analysis of iterative methods for elliptic boundary value problems. *SIAM Review*, 31:20–49, March 1989.

[11] Lori Freitag. On combining laplacian and optimization-based smoothing techniques. In *Proceedings of the 1997 Joint Summer Meeting of American Society of Mechanical Engineers (ASME) American Society of Civil Engineers (ASCE) and Society of Engineering Science (SES)*, to appear.

[12] Lori Freitag and Carl Ollivier-Gooch. A comparison of tetrahedral mesh improvement techniques. In *Proceedings of the Fifth International Meshing Roundtable*, pages 87–100. Sandia National Laboratories, 1996.

[13] Lori Freitag and Carl Ollivier-Gooch. Tetrahedral mesh improvement using swapping and smoothing. *International Journal of Numerical Methods in Engineering*, 40, to appear.

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

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

[16] Barry Joe. Three-dimensional triangulations from local transformations. *SIAM Journal on Scientific and Statistical Computing*, 10:718–741, 1989.

[17] Barry Joe. Geompack - a software package for the generation of meshes using geometric algorithms. *Advanced Engineering Software*, 56(13):325–331, 1991.

[18] Barry Joe. Construction of three-dimensional improved quality triangulations using local transformations. *SIAM Journal on Scientific Computing*, 16:1292–1307, 1995.

[19] Carl Ollivier-Gooch, 1996. Alpha Release of the GRUMMP software, http://www.mcs.anl.gov/Projects/SUMAA/index.html.

[20] Carl F. Ollivier-Gooch. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, to appear.

[21] Carl F. Ollivier-Gooch. High-order ENO schemes for unstructured meshes based on least-squares reconstruction. *AIAA paper 97-0540*, January 1997.

[22] Carl F. Ollivier-Gooch. Multigrid acceleration of an upwind Euler solver on unstructured meshes. *AIAA Journal*, 33(10):1822–1827, October, 1995.

[23] V. N. Parthasarathy and Srinivas Kodiyalam. A constrained optimization approach to finite element mesh smoothing. *Finite Elements in Analysis and Design*, 9:309–320, 1991.

[24] M. Rivara. Mesh refinement processes based on the generalized bisection of simplices. *SIAM Journal on Numerical Analysis*, 21:604–613, 1984.

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

[26] Jonathan Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Proceedings of the First Workshop on Applied Computational Geometry*, pages 124–133, Philadelphia, Pennsylvania, May 1996. ACM.

[27] Steve Vavasis and Scott Mitchell. Quality mesh generation in higher dimensions. Technical Report ANL/MCS-P633-1296, Argonne National Laboratory, 1996.