A Cost/Benefit Analysis of Simplicial Mesh Improvement Techniques as Measured by Solution Efficiency

Lori A. Freitag* and Carl Ollivier-Gooch†

1 Introduction

The quality of finite element and finite volume meshes has long been known to affect both the efficiency and the accuracy of the numerical solution of application problems (e.g., see [6] and [14]). To improve the quality of these meshes, several researchers have devised new algorithms based on local reconnection schemes, node smoothing, and adaptive refinement or coarsening (e.g. [2, 10, 18, 19, 29, 30]). In each case, the edges, vertices, or elements of the mesh are individually evaluated to determine whether performing the local operation improves the mesh. Therefore, these methods typically incur an $\mathcal{O}(N)$ computational cost, where N is the number of vertices in the mesh. This is a significant cost as N increases, and often only anecdotal evidence is given to demonstrate the benefit of these techniques in terms of solution efficiency for a particular application or solver.

In this paper, we provide a deeper analysis of the tradeoffs associated with the cost of mesh improvement in terms of solution efficiency. We consider both finite element and finite volume discretization techniques, a number of different solvers, and a variety of application problems. The issue of solution accuracy will be addressed in a later paper.

We focus initially on problems discretized using the finite element method. Such discretizations lead to large, sparse linear systems, which are often solved by using either conjugate gradient (CG) [16] or GMRES [31] iterative techniques. Several theoretical results that relate the convergence behavior of these algorithms to matrix characteristics such as condition number and spectral distribution. In turn, for simple applications such as Poisson's equation, these matrix characteristics can be theoretically related to the size and quality of the underlying finite element mesh.

Theoretical results are not available for more complicated applications. To obtain insight into the convergence rates of CG and GMRES, we must empirically establish the relationship between mesh size and quality to convergence behavior. We do so by performing a series of experiments in which the parameters of mesh size and quality are varied both individually

*Assistant Scientist, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. freitag@mcs.anl.gov.

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

and together for a number of application problems. So that we can compare our experimental results with theoretical results, we choose Laplace's equation as our simplest test case. Once the relationship between the mesh and the iterative solver convergence rate is established, we perform a cost/benefit analysis of mesh improvement techniques for both simple and complex applications.

The remainder of the paper is organized as follows. In Section 2, we review theoretical results on the convergence of conjugate gradient and GMRES for general matrices and for matrices arising from a specific finite element application. In Section 3 we briefly review the local reconnection and optimization-based smoothing techniques [12, 13] that are used in this paper for mesh improvement. In Sections 4 and 5 we perform a cost/benefit analysis of mesh improvement techniques for both simple and complex applications. We show that the benefits of applying different mesh improvement techniques are cumulative and can be seen even when starting with a fairly good mesh.

2 The Convergence of CG and GMRES

Finite element discretization techniques reduce the problem of solving a partial differential equation to the problem of solving a large, sparse system of linear algebraic equations, traditionally written as $Ax = b$. Because of their size and sparsity, direct methods are generally very inefficient for solving these linear systems, and iterative methods such as conjugate gradient and GMRES are used instead. Several theoretical results relate the convergence of these iterative solvers to the characteristics of the matrix, A, particularly its condition number and spectral distribution. For simple applications these quantities can then be related to the quality and size of the finite element mesh.

Let the eigenvalues of the $N \times N$ matrix A be given by $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{N-1} \leq \lambda_N$. Axelsson [3] showed that, if the eigenvalues are uniformly distributed, the number of iterations required for CG to reduce the error in the energy norm by a factor of ϵ is bounded by

$$I_\epsilon \leq \left\lfloor \sqrt{\frac{\lambda_N}{\lambda_1}} \ln \frac{2}{\epsilon} \right\rfloor + 1, \tag{1}$$

where the ratio $\frac{\lambda_N}{\lambda_1}$ is the condition number of the linear system. For symmetric finite element approximations, λ_1 is bounded below by

$$\lambda_1 \geq c_1 \Delta_{\min} \min_{t \in T} A_t, \tag{2}$$

where c_1 is a mesh independent constant, $\Delta_{\min}$ is the minimum vertex degree in the mesh, T is the set of elements in the mesh, and A_t is the area of element t. The value of λ_N is bounded above by

$$\lambda_N \leq c_2 \Delta_{\max} (\max_{t \in T} A_t + \max_{\theta \in \Theta}, \frac{1}{\sin \theta}) \tag{3}$$

where c_2 is a mesh independent constant, $\Delta_{\max}$ is the maximum vertex degree in the mesh, and Θ is the set of all element angles in the mesh. Axelsson and Barker provide a more complete description of these bounds [4, pp. 232–238]. Two results of interest for this paper naturally follow from these bounds.

1. As the size of the finite element mesh increases, $\min_{t \in T} A_t$ decreases, and λ_1 can decrease proportionally, causing the condition number and number of iterations to increase. For example, Axelsson and Barker showed that for Poisson's equation discretized with the finite element method, the condition number of the stiffness matrix is $\mathcal{O}(N)$ regardless of the vertex ordering or the degree of the polynomial basis functions [4].

2. As the quality of the mesh degrades and small angles are introduced, the bounds on both λ_1 and λ_N are adversely affected and the condition number increases proportionally.

The condition number of a matrix A is not the sole factor that determines the convergence of the CG algorithm; the spectral distribution also plays a critical role. If the spectrum of A is not uniformly distributed—that is, if there are isolated eigenvalues at either end of the spectrum—Equation 1 overestimates the iteration count, and improved bounds can be obtained in both exact arithmetic (e.g., [5]) and in the presence of rounding error (e.g. [24, 17, 33]). Let the eigenvalues of such a system be denoted by $\lambda_1, \lambda_2, ..., \lambda_p, [a, b], \lambda_q, ..., \lambda_N$, where $[a, b]$ represents the continuous part of the spectrum and the remaining eigenvalues are isolated values at either end of the spectrum. The components of error associated with the isolated eigenvalues are eliminated in a few additional steps, and then CG converges as though bounded by the smaller condition number b/a. For small isolated eigenvalues, the number of additional iterations necessary for convergence is proportional to the number of isolated eigenvalues in the spectrum and the value of $\ln(\sum_{i=1}^{p} a/\lambda_i)$. For large isolated eigenvalues, the number of additional steps is proportional to rounding errors, the ratios of $\frac{\lambda_j}{b-a}$, and the quantities $\ln \frac{\lambda_j}{\lambda_i}$, where $i, j = q, ..., N$.

The convergence behavior of GMRES is also determined by the spectrum or pseudospectra of A, depending on whether A is normal, close to normal, or arbitrary [31, 23, 15]. These results use exact arithmetic and do not include the complicating factors of preconditioning or the common practice of restarting GMRES to reduce the computational complexity of the algorithm. Joubert analyzes the impact of the restart parameter on convergence and gives a theoretical analysis for certain classes of matrices [21]. We are unaware of theoretical results for convergence for restarted GMRES applied to arbitrary matrices.

3 Mesh Improvement Techniques

The theoretical results presented in the preceding section clearly show that the underlying mesh quality can significantly affect the convergence of iterative solvers. Mesh quality can be improved by using a variety of techniques, including point insertion/deletion to refine or coarsen a mesh [28, 30, 20], local reconnection to change mesh topology for a given set of vertices [18, 19], and mesh smoothing to relocate grid points without changing mesh topology [2, 10, 29]. Our approach is based on combining local reconnection schemes and optimization-based mesh smoothing techniques to improve simplicial mesh quality. We now briefly review these procedures; more detailed descriptions can be found in [12, 13].

3.1 Local Mesh Reconfiguration Techniques

The most commonly used local reconnection technique is *face swapping*. In both two and three dimensions, this technique can be used to change the topological connections among the vertices making up the simplices incident on a single face. Lawson [22] showed that there at most two legal connectivities for these vertices, thereby making the task of choosing the best configuration straightforward. In two dimensions, a face is bounded by two triangles; face swapping chooses the best diagonal for the quadrilateral that is the union of these triangles. In three dimensions, a face is bounded by two tetrahedra, but the five vertices in these tetrahedra may form either two or three tetrahedra. Many possible configurations exist, but the only two that can be legally reconnected are shown in Figure 1. One of these configurations switches between two and three tetrahedra; the other, between two two-tetrahedron configurations. The latter is applicable only along mesh boundaries.

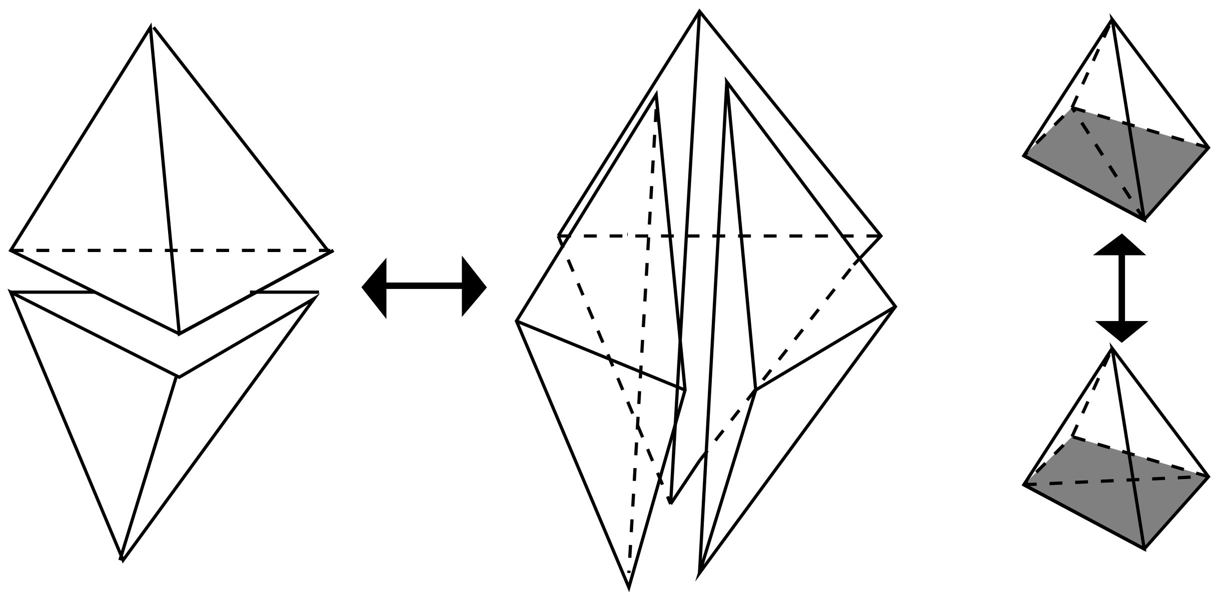

Figure 1: Swappable configurations of five points in three dimensions

A more advanced technique in three dimensions is the reconfiguration of all tetrahedra incident on a single edge. This technique is known as *edge swapping* or edge removal. Our approach, described in detail in [13], operates by viewing the edge to be swapped end-on and choosing the "triangulation" of the vertices not on that edge that maximizes some mesh quality measure. This reconfiguration technique replaces N tetrahedra by $2N-4$ tetrahedra, and the number of possible "triangulations" to be checked grows rapidly with N. At the same time, the fraction of reconfigurations that are successful decreases, in part because of an upper bound on the angle produced as N increases. Consequently, we do not apply this technique for $N > 7$. We employ edge swapping both as a remedy when our face swapping routine is stalled by certain nonconvex configurations of five vertices and as an independent mesh improvement technique to selectively eliminate tetrahedra with bad angles from the mesh.

3.2 Mesh Smoothing

The most frequently used approach for mesh smoothing is some variant on Laplacian smoothing, which in its simplest form moves each vertex to the average of the locations of its neighbors. This approach, while computationally inexpensive, has serious drawbacks in that it does not provide any mechanisms that guarantee improvement in element quality. In fact, the method can produce an invalid mesh containing elements that are inverted or have negative volume.

Optimization-based smoothing techniques offer an alternative to Laplacian smoothing that can be inexpensive, can guarantee valid elements in the final mesh, and can be effective for a wide variety of mesh quality measures. In [12, 13], we described an efficient, local smoothing algorithm based on optimization techniques for piecewise smooth, continuous functions that move the grid points in a manner guaranteed to maintain or improve mesh quality. The vertex is repositioned by using a generalized steepest descent method that requires function and gradient information for a mesh quality measure. Any differentiable mesh quality function can be used, including cell angles and aspect ratios. This technique is robust and effective in improving mesh quality in both two and three dimensions. We note that similar optimization-based smoothing methods have been proposed by a number of researchers [1, 8, 32] for a variety of optimization procedures and mesh quality measures.

The optimization-based approach is quite effective, however its computational cost is high compared with Laplacian smoothing. To address this problem, Freitag developed a family of combined Laplacian and optimization-based smoothing techniques [11]. For each vertex, these techniques apply a single step of a smart Laplacian operator—one that moves the vertex only if the local submesh is improved. If the local submesh has a quality measure below a user-definable threshold, the optimization-based smoother is invoked to further improve the quality of the submesh. This technique was shown to be as effective in mesh improvement as the strictly optimization-based approach, at a fraction of the computational cost.

3.3 Recommendations

We now summarize our recommendations for two-dimensional mesh improvement based on geometric quality measures [13].

- Local reconnection should be followed with mesh smoothing to improve mesh quality. Three to four passes of a combined Laplacian/optimization-based smoother with a floating threshold is the most efficient smoothing technique. To prevent conflict between the behavior of reconnection and smoothing, the same quality criterion should be used for both.

- Element quality criteria designed to eliminate small angles from the mesh are generally more effective in improving overall mesh quality than measures designed to eliminate large angles. We recommend maximizing the minimum sine of element angle (dihedral angle in three dimensions) as a good general-purpose criterion.

- In three dimensions, edge swapping is beneficial and should be used. Also, the use of advanced techniques such as BATR [13] is highly beneficial in improving tetrahedral meshes.

4 Effect of Mesh Quality on Convergence

We begin our cost/benefit analysis of mesh improvement techniques in terms of solution efficiency by considering two model problems on the unit square. The first is Laplace's equation (diffusion) with a mix of Dirichlet and Neumann boundary conditions:

$$\nabla^2 \phi = 0 \ \in \ (0,1) \times (0,1)$$

$$\phi(0,y) = 0; \quad \phi(1,y) = 1; \quad \frac{\partial \phi(x,0)}{\partial y} = \frac{\partial \phi(x,1)}{\partial y} = 0.$$

Initial, high-quality triangular meshes are created by using GRUMMP [25], and in all cases $\theta_{\min} > 20^o$, $\theta_{\max} < 135^o$, and $\theta_{\text{avg}} \approx 50^o$. We discretize the system with linear, triangular finite elements and solve the resulting symmetric linear system with the PETSc [7] implementation of the conjugate gradient (CG) method with both Jacobi (CG/Jac) and incomplete Cholesky (CG/IC) preconditioners.

The second test case we consider is advection diffusion on the unit square with the same set of test meshes:

$$\nabla^2 \phi + \nabla \phi = 0 \ \in \ (0,1) \times (0,1)$$

$$\phi(0,y) = 0; \quad \phi(1,y) = 1; \quad \frac{\partial \phi(x,0)}{\partial y} = \frac{\partial \phi(x,1)}{\partial y} = 0.$$

In this case, the resulting nonsymmetric linear systems are solved by using the PETSc implementation of GMRES with both Jacobi (GMRES/Jac) and ILU (GMRES/ILU) preconditioners. All GMRES solves are restarted by using the PETSc default value of 30 iterations.

In all test cases we use a convergence test of 10^{-12} on the relative tolerance of the residual. The spectrum of each linear system is computed exactly by using the LAPACK routine `dgeev()`.

We seek to quantify the behavior of these two test problems as the quality and size of the finite element mesh changes and then to perform a simple cost/benefit analysis for mesh smoothing. To accomplish this goal, we perform four experiments.

- *Experiment 1:* We obtain the baseline number of iterations required to solve the problem on a good quality mesh as the mesh size increases. The relationship between mesh size and work to solve these test cases is established.

- *Experiment 2:* We insert one new point into the mesh to create two elements of extremely poor quality. We analyze the spectrum of the resulting linear system and the number of iterations required for convergence.

- *Experiment 3:* We increase the number of poor-quality elements in the mesh and repeat the analysis of the second experiment.

- *Experiment 4:* We analyze the costs and benefits associated with mesh smoothing for the simple test cases. In particular, we compare the time required to solve the problem on a poor-quality mesh with the time required to improve the mesh and solve the problem on the high-quality mesh.

We find that benefits of mesh improvement techniques exceed the costs as mesh size increases or mesh quality degrades. A cost/benefit analysis for more complex examples will be discussed in the next section.

Experiment 1: Convergence of CG and GMRES as N increases

From the results presented in Section 2, we expect the number of iterations required by CG to converge for Laplace's equation to be $\mathcal{O}(N^{\frac{1}{2}})$. In this experiment we empirically establish the multiplicative factor for each of the CG techniques and the relationship between the number of iterations required for convergence and N for each of the GMRES solvers.

So that we can also examine the spectrum of each linear system, we begin by solving our test cases on small meshes of size $N \approx 100$, 200, 300, 400, and 500. The minimum and maximum eigenvalue of the spectrum, λ_1 and λ_N, and the number of iterations required for convergence, I, are given in Table 1. In all cases the spectral distribution is fairly uniform. As predicted theoretically in Equations 2 and 3, λ_N stays roughly constant and λ_1 decreases with increasing N; the condition number and iteration count increase correspondingly. The scaling performed by the Jacobi preconditioner decreases both λ_1 and λ_N by roughly a factor of two and does not affect the spectral distribution. It is not surprising, then, that the number of iterations is only slightly reduced for CG/Jac and GMRES/Jac compared with CG and GMRES. The IC and ILU preconditioners compact the spectral distribution at both ends and, as expected, are the most effective.

Table 1: Minimum and maximum eigenvalues of the linear systems and the number of iterations required to converge for the diffusion and advection diffusion test cases as N increases

Diffusion									
	CG			CG/Jac			CG/IC		
N	λ_1	λ_N	I	λ_1	λ_N	I	λ_1	λ_N	I
100	.0517	3.26	49	.0311	1.73	47	.1411	1.38	21
200	.0248	3.24	73	.0146	1.76	66	.0681	1.42	29
300	.0171	3.09	84	.0199	1.70	79	.0501	1.35	33
400	.0131	3.30	99	.0074	1.69	94	.0360	1.35	39
500	.0100	3.22	110	.0056	1.74	106	.0271	1.42	44
Advection Diffusion									
	GMRES			GMRES/Jac			GMRES/ILU		
N	λ_1	λ_N	I	λ_1	λ_N	I	λ_1	λ_N	I
100	.0606	3.41	67	.0351	1.71	63	.154	1.46	22
200	.0286	3.33	114	.0164	1.75	101	.0733	1.45	29
300	.0196	3.14	149	.0112	1.70	139	.0527	1.46	38
400	.0149	3.35	173	.0083	1.68	164	.0373	1.53	52
500	.0114	3.27	217	.0063	1.74	207	.0284	1.57	61

The computational cost of iteratively solving linear systems is a product of the work per iteration and the number of iterations. For the solvers considered here, each iteration is

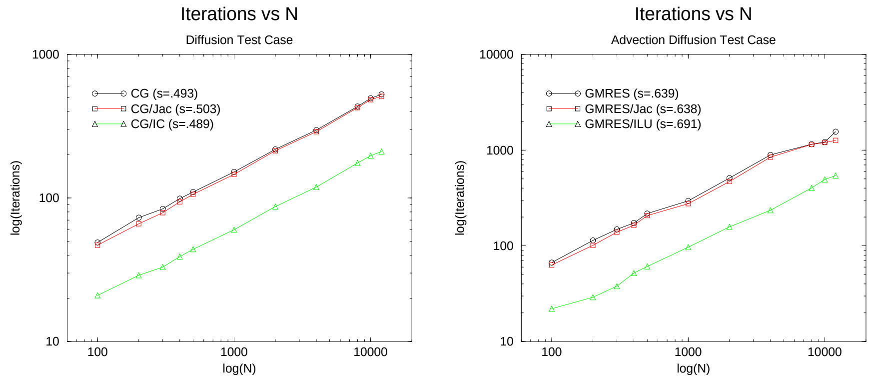

Figure 2: Number of iterations required to converge for the diffusion and advection diffusion test cases as N increases

dominated by a matrix-vector multiplication that is $\mathcal{O}(N)$ operations for the sparse linear systems generated by the finite element technique. To obtain a good estimate the number of iterations required for convergence for our test cases on high-quality meshes, we expanded the test suite to include problems with N = 1,000, 2,000, 4,000, 8,000, 10,000, and 12,000 grid points in the mesh. The number of iterations required for convergence as a function of the size of the mesh is shown in Figure 2 as a log-log plot. The slopes of the lines give the order of convergence, s, for each technique. The results indicate that the number of iterations grows as $\mathcal{O}(N^s)$. Therefore, the total work required to solve the system is $\mathcal{O}(N^{s+1})$ for each of the iterative techniques. To find the value of s, we used linear least-squares analysis and obtained the following relationships between I and N for each solver.

$$\begin{array}{ll} I_{CG} = 3.57N^{.493} & I_{GMRES} = 2.83N^{.639} \\ I_{CG/Jac} = 3.27N^{.503} & I_{GMRES/Jac} = 2.68N^{.638} \\ I_{CG/IC} = 1.79N^{.489} & I_{GMRES/ILU} = 1.12N^{.642}. \end{array}$$

As predicted by Equation 1, the order of convergence for the CG techniques applied to the diffusion equation is .5; the order of convergence for GMRES applied to the advection diffusion equation is approximately .64.

Experiment 2: Effect of One Perturbed Element on Convergence

Using the results of Experiment 1 as a baseline, we now show the effect of a small number of poor-quality elements on the number of iterations required for convergence. To control the quality of the meshes, we start with the high-quality meshes used in the first experiment and insert a single point to create two new poor-quality elements. The new point is inserted a distance ϵ along the bisector of one of the three angles in a randomly chosen element, where ϵ is set to produce the desired minimum angle. An example of this point insertion technique is shown in Figure 3. Using this technique, we created a series of five meshes for each original mesh whose smallest angles are .25°, .5°, 1°, 2.5°, and 5°. For the series of meshes for each N, the same element is perturbed.

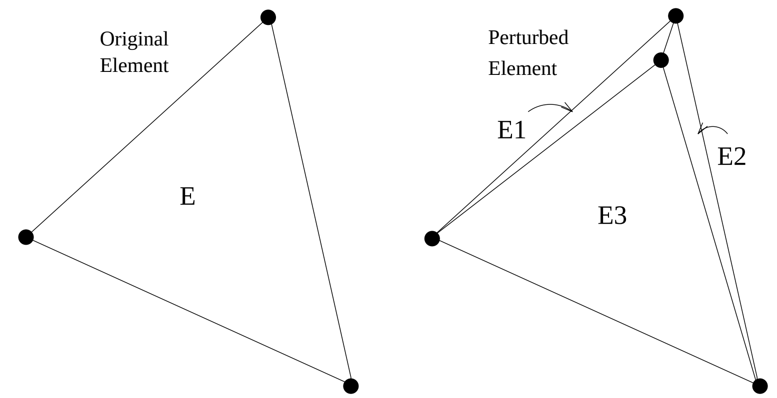

Figure 3: Example of the point insertion technique used to create poor-quality elements

We show typical results for λ_1 and λ_N and the number of iterations, I, required to reach an accuracy comparable to that obtained on the original 200-grid-point mesh in Table 2. The results for $N = 100$, 300, 400 and 500 are qualitatively the same. The theoretical bounds for λ_1 and λ_N given in Equations 2 and 3 for Laplace's equation indicate that both can be adversely affected by poor-quality elements. However, our experiments show that λ_1 is unaffected and that λ_N grows as $1/\sin\theta_{\min}$. This result reflects the fact that the small angles in the mesh enable the discrete solution to capture, at least partially, higher-frequency eigenmodes of the continuous solution of Laplace's equation. For Jacobi preconditioning, λ_N is unaffected by the decreasing angle size and λ_1 decreases as $\ln(\theta_{\min})$. For the CG/IC and GMRES/ILU methods, the condition number and spectrum of the preconditioned system are not significantly changed by the addition of two poor-quality elements.

In most cases, however, the number of iterations required for convergence is only slightly increased compared with the baseline results presented in Experiment 1. For these test cases, the spectrum contains one isolated eigenvalue, and the size of the discontinuity is given in the columns labeled λ_N/λ_{N-1} and λ_2/λ_1 for no preconditioning and Jacobi preconditioning, respectively. For CG, the size of the discontinuity becomes quite pronounced as $\theta_{\min}$ decreases, causing the number of iterations to increase by as much as 15 percent. On the other hand, GMRES is unaffected by the increase in size of the discontinuity, and the number of iterations required for convergence is constant. The Jacobi preconditioner is effective at eliminating the isolated eigenvalue in the spectrum, but the ellipticity of the system is increased. The CG/Jac solver is not significantly affected by this increase, and the number of iterations increases by 10 percent. In contrast, GMRES/Jac is quite sensitive to this increase in ellipticity and shows a 100 percent increase in the number of iterations for $\theta_{\min} = .25^{\circ}$. The convergence of the CG/IC and GMRES/ILU methods is not affected by the poor-quality elements. We conclude that, except for GMRES/Jac, a small number of poor-quality elements do not significantly affect the work required to solve the system.

Experiment 3: Effect of Many Perturbed Elements on Convergence

We now analyze the spectrum and convergence of the iterative solvers for an increasing percentage of poor-quality elements in the mesh. In Table 3 we show $\lambda_{\min}$, $\lambda_{\max}$, and the

Table 2: Minimum and maximum eigenvalues of the linear systems and the number of iterations required to converge for the diffusion and advection diffusion test cases with 200 grid points and two poor quality elements

Diffusion												
	CG				CG/Jac				CG/IC			
θ	λ_1	λ_N	λ_N/λ_{N-1}	I	λ_1	λ_N	λ_2/λ_1	I	λ_1	λ_N	I
Original	.024	3.24	1.03	73	.014	1.76	1.97	66	.066	1.42	29
5°	.024	11.3	3.48	77	.0137	1.96	2.21	69	.065	1.42	29
2.5°	.024	21.5	6.64	79	.0126	1.96	2.12	70	.065	1.42	29
1°	.024	52.4	16.17	80	.0093	1.98	2.53	71	.066	1.42	29
.5°	.024	104.1	32.12	82	.0060	1.99	3.60	72	.066	1.43	29
.25°	.024	219.4	67.59	84	.0032	1.99	6.25	73	.066	1.43	29
Advection Diffusion											
	GMRES				GMRES/Jac				GMRES/ILU		
θ	λ_1	λ_N	λ_N/λ_{N-1}	I	λ_1	λ_N	λ_2/λ_1	I	λ_1	λ_N	I
Original	.028	3.41	1.03	114	.0165	1.75	1.83	101	.0733	1.45	29
5°	.028	11.34	3.40	116	.0154	1.92	1.91	111	.0704	1.46	30
2.5°	.028	21.57	6.47	116	.0147	1.96	1.99	119	.0704	1.45	30
1°	.028	52.51	15.76	116	.0104	1.98	2.42	144	.0704	1.46	30
.5°	.028	104.1	31.23	116	.0065	1.99	3.53	173	.0704	1.46	30
.25°	.028	219.4	65.90	116	.0035	1.99	6.39	204	.0704	1.46	30

percentage increase in the number of iterations required for $p = 2$, 5, 10, and 20 percent perturbed elements. The percentage increase in iterations is computed as follows:

$$\mathcal{P}_I = \frac{I_p - I_N}{I_N} \times 100,$$

where I_p is the number of iterations required for convergence on the poor-quality mesh and I_N is the number of iterations needed to converge on a good-quality mesh of size N. Because we use a point insertion technique to create the poor-quality meshes, the increasing percentage of poor quality elements also increases mesh size. In particular, for the results presented in Table 3, our base mesh has $N = 217$ grid points; the perturbed meshes have $N = 224$, 235, 253, and 290 grid points for $p = 2$, 5, 10, and 20, respectively. From Experiment 1 results, we know that this increase in N contributes to an increase in the iteration count; to account for this, we compute I_N by cN^s where c and s were determined in Experiment 1.

For the nonpreconditioned systems, an increasing number of perturbed elements resulted in an increasing value of λ_N and a one-to-one increase in the subspace size associated with the extremal eigenvalues. As in Experiment 2, the Jacobi and IC/ILU preconditioners are effective in eliminating the extremal eigenvalue subspaces. For the Jacobi preconditioning, as the number of perturbed elements increases, the condition number of the system increases: λ_1 decreases by as much as a factor of seven and λ_N is unchanged. For the IC and ILU preconditioned systems, both λ_1 and λ_N are adversely affected by an increase in the number of perturbed elements. The results in the columns labeled $\mathcal{P}_I$ show that as we increase the number of bad elements in the mesh and worsen their quality, the number of additional iterations required to converge increases. The GMRES techniques are more sensitive to large numbers of perturbed elements in the mesh than their CG counterparts. For the GMRES/ILU solver, the work estimate obtained in Experiment 1 is not accurate for small mesh sizes and overestimates the number of iterations, yielding negative percentage increases for $p = 2$ and $p = 5$. For a good quality mesh, the number of iterations required for GMRES/ILU on the 200 grid point mesh is 29. For $p = 2$ and $p = 5$, the number of iterations is 34, 33, 33 and 53, 51, 48 for $\theta_{min} = .5,\ 1$, and 2.5, respectively.

To show general trends as N increases and p remains fixed, we plot the number of iterations required for convergence for each of the solvers under consideration for meshes of size $N =$ 2,000, 4,000, 8,000 and 12,000 and fixed percentage $p = 10$. We again consider increasingly poor-quality elements whose minimum angles range from $.25^o$ to 17^o. In a number of advection diffusion cases, the GMRES and GMRES/Jac algorithms fail to converge prior to reaching the maximum allowed number of iterations (10,000). For $\theta_{\min} < 2.5^o$, the poor quality of the mesh significantly affects the number of iterations required for convergence and the curve is exponential in nature. However, for $\theta_{\min} > 2.5^o$ the number of iterations remains roughly constant and is determined primarily by the number of grid points in the mesh. The curves shown in Figure 4 are typical results; as the percentage of poor-quality elements decreases, the general trends remain the same, but the number of iterations decreases.

We now summarize the results of the first three experiments presented in this section.

- The number of iterations required to converge increases as N increases, and the work to solve the linear system grows as $\mathcal{O}(N^{s+1})$, where s is approximately .5 for the CG techniques and .63 for the GMRES techniques.

- In most cases, the number of iterations required for convergence is not significantly affected by a small number of poor-quality elements. In the case of no preconditioning, the iteration count is not affected because the resulting spectrum contains isolated eigenvalues. The preconditioners considered are effective at eliminating the isolated eigenvalue at the cost of a slight increase in the ellipticity of the linear system.

- As the percentage of poor-quality elements increases, the subspace size associated with the extremal eigenvalues increases proportionally, and the number of iterations required for convergence increases. The preconditioners are effective at eliminating the subspace associated with extremal eigenvalues, but the condition number of the linear system is adversely affected.

- For N=200, p=10 percent, and $\theta_{\min} > 2.5^o$, the number of iterations required for convergence is primarily determined by the size of the mesh. As $\theta_{\min}$ decreases, the additional iterations associated with the poor-quality elements start to dominate convergence.

In light of these observations, the following question naturally arises:

For what values of N, minimum angle, and percentage of poor-quality elements in the mesh is the total cost of improving the mesh and solving the problem on the improved mesh less than the cost of obtaining an accurate solution on a poor-quality mesh?

Table 3: Minimum and maximum eigenvalues of the preconditioned linear system and the number of iterations required to converge using various iterative solvers for the two test cases with 200 grid points and an increasing number of perturbed elements

Diffusion												
	2 Percent			5 Percent			10 Percent			20 Percent		
θ	λ_1	λ_N	$\mathcal{P}_I$	λ_1	λ_N	$\mathcal{P}_I$	λ_1	λ_N	$\mathcal{P}_I$	λ_1	λ_N	$\mathcal{P}_I$
No Preconditioning												
	$I_N = 52$			$I_N = 53$			$I_N = 55$			$I_N = 59$		
.5°	.024	104	90.3	.023	131	128	.021	151	293	.017	153	366
1°	.024	52	82.7	.023	65	115	.021	75	249	.017	76	313
2.5°	.024	21.5	71.1	.023	26	94.3	.021	30	187	.017	31	238
Jacobi Preconditioning												
	$I_N = 50$			$I_N = 51$			$I_N = 53$			$I_N = 57$		
.5°	.0036	2.01	68.0	.0017	2.01	102	.0012	2.02	270	.0006	2.03	319
1°	.0061	2.03	62.0	.0032	2.03	92.1	.0022	2.04	219	.0011	2.06	265
2.5°	.0098	2.05	54.0	.0063	2.05	78.4	.0045	2.07	164	.0025	2.11	200
IC Preconditioning												
	$I_N = 26$			$I_N = 26$			$I_N = 27$			$I_N = 29$		
.5°	.064	1.48	11.5	.015	1.96	38.4	.009	2.44	70.3	.009	2.33	93.1
1°	.064	1.47	11.5	.025	1.93	38.4	.017	2.29	66.6	.017	2.28	86.2
2.5°	.064	1.46	11.5	.041	1.83	34.6	.035	1.99	51.2	.028	2.01	72.3
Advection Diffusion												
	2 Percent			5 Percent			10 Percent			20 Percent		
θ	λ_1	λ_N	$\mathcal{P}_I$	λ_1	λ_N	$\mathcal{P}_I$	λ_1	λ_N	$\mathcal{P}_I$	λ_1	λ_N	$\mathcal{P}_I$
No Preconditioning												
	$I_N = 90$			$I_N = 93$			$I_N = 98$			$I_N = 106$		
.5°	.027	104.1	52.2	.026	131	91.4	.024	150	580	.020	153.9	881
1°	.027	52.5	52.2	.026	65.3	89.2	.024	75.3	423	.020	76.9	789
2.5°	.027	21.57	51.1	.026	25.9	83.9	.024	30.3	306	.020	30.7	461
Jacobi Preconditioning												
	$I_N = 85$			$I_N = 88$			$I_N = 92$			$I_N = 100$		
.5°	.0040	2.01	164	.0019	2.01	198	.0013	2.02	471	.00065	2.02	540
1°	.0068	2.02	83.5	.0036	2.02	123	.0025	2.04	304	.00127	2.05	390
2.5°	.0111	2.05	65.9	.0072	2.05	80.7	.0052	2.07	130	.00288	2.11	244
ILU Preconditioning												
	$I_N = 37$			$I_N = 38$			$I_N = 40$			$I_N = 43$		
.5°	.067	1.53	-16	.016	1.96	-10	.0097	1.99	32.5	.0051	2.01	432
1°	.067	1.53	-16	.027	1.93	-10	.018	1.97	27.5	.0094	2.02	274
2.5°	.067	1.51	-16	.045	1.83	-10	.034	1.92	20.0	.017	2.01	174

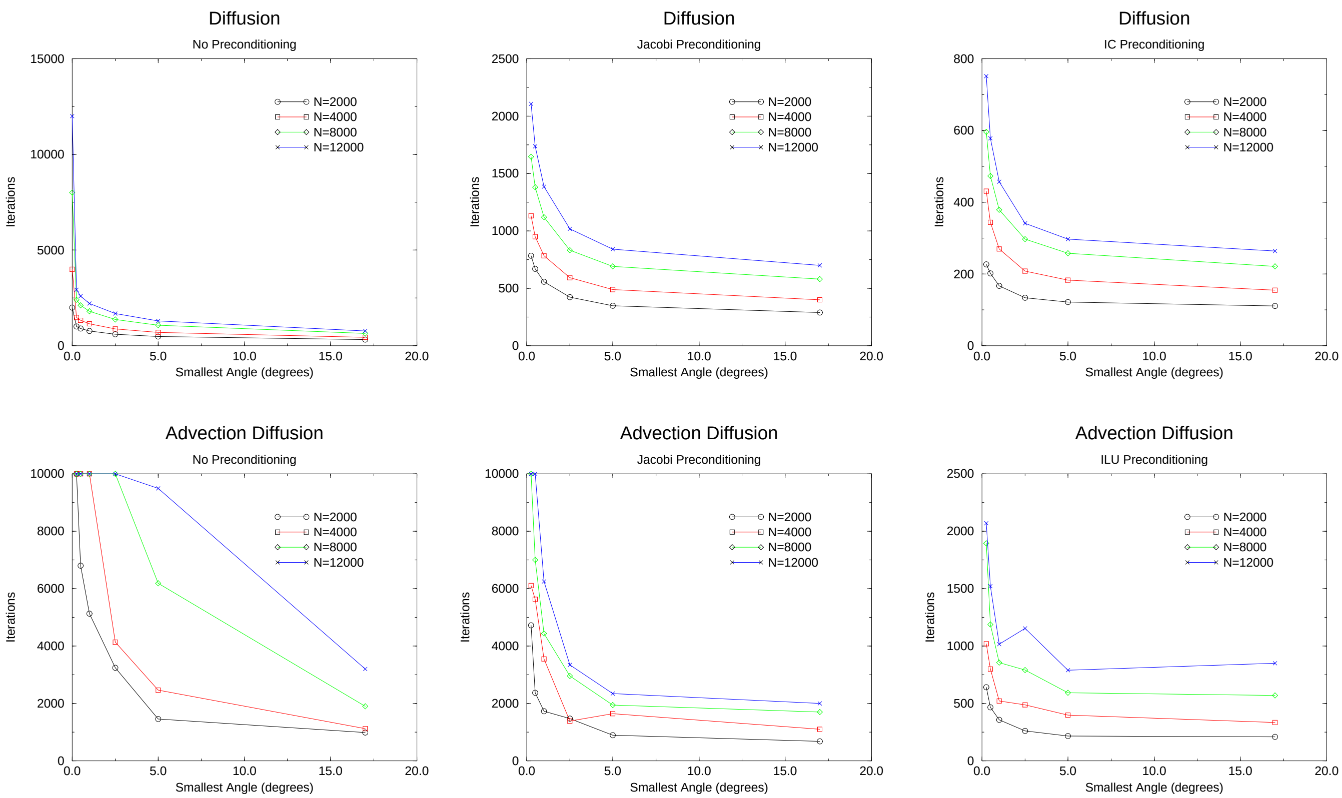

Figure 4: Effect of decreasing the minimum angle on the convergence rate of the iterative solvers for the diffusion and advection diffusion test case when 10 percent of the elements are perturbed and as N increases

Experiment 4: Cost/Benefit Analysis for Mesh Improvement Techniques

To quantify the benefits of the mesh improvement algorithms in terms of solution efficiency, we compare the difference in solution times on the poor-quality mesh and on an improved mesh (including the time to improve the mesh). From our first experiment, we know that the work required to solve the problem grows as $\mathcal{O}(N^{s+1})$ for a good-quality mesh. For poor-quality meshes, we showed in Experiment 3 that this amount of work increases proportionally with the number of and "badness" of the poor-quality elements. In all cases, the work required to improve the mesh is $\mathcal{O}(N)$, and we therefore expect that the benefits of smoothing will be more pronounced as N increases and quality of the mesh degrades.

We present results for poor-quality meshes similar to those used in Experiment 3 for which the number of perturbed elements was 10 percent. In each case we improve the poor-quality mesh with three passes of the combined smoothing approach described in Section 2. Element quality typically improves to greater than 15° for the minimum angles and to less than 140° for the maximum angle. The amount of time required for mesh smoothing is approximately 3.7, 6.6, 14.3, and 21.6 seconds for $N = 2000$, 4000, 8000, and 12000, respectively.

In Figure 5, we plot the difference in the time required to reach convergence on a poor-quality mesh and the total time to reach convergence on an improved mesh including the time for mesh smoothing. The horizontal line in each plot is zero and represents the breakeven point between the increased cost of smoothing the mesh and the time required to solve the problem on a poor-quality mesh. Therefore, data points located above the breakeven line are cases in which a benefit is obtained by using mesh smoothing; those below the breakeven line indicate that the total solution time was worsened by mesh smoothing.

In all cases, as the minimum angle in the mesh decreases, the cost of solving the problem on the poor-quality mesh exceeds the cost of mesh improvement. For each iterative solver as N increases, the minimum angle for which there is a breakeven point increases. For CG and GMRES and the largest mesh size, there is significant advantage to smoothing the mesh for all values of $\theta_{\min}$. For CG/Jac, and GMRES/Jac there are significant benefits for $\theta_{\min} \leq 3°$. Finally, for CG/IC and GMRES/ILU, and the values of N considered here, smoothing benefits are obtained only when $\theta_{\min} < 1.5°$.

5 Cost/Benefit Analysis for Complex Applications

From the experiments presented in the preceding section, one is tempted to conclude that unless the mesh contains a large number of very poor quality elements, mesh improvement techniques offer little to no benefit in terms of solution efficiency. However, the test problems chosen in that analysis were exceptionally simple; they are useful in determining trends in finite element applications and for obtaining comparisons with theoretical results, but do not adequately illustrate benefits of mesh improvement techniques.

In this section we show that the efficiency benefits of mesh improvement techniques such as swapping and smoothing are cumulative and effective for complex applications even when the initial mesh is of reasonable quality. We focus on two applications in this section: a linear elasticity application solved using finite elements and a compressible Euler flow problem solved using the finite volume method.

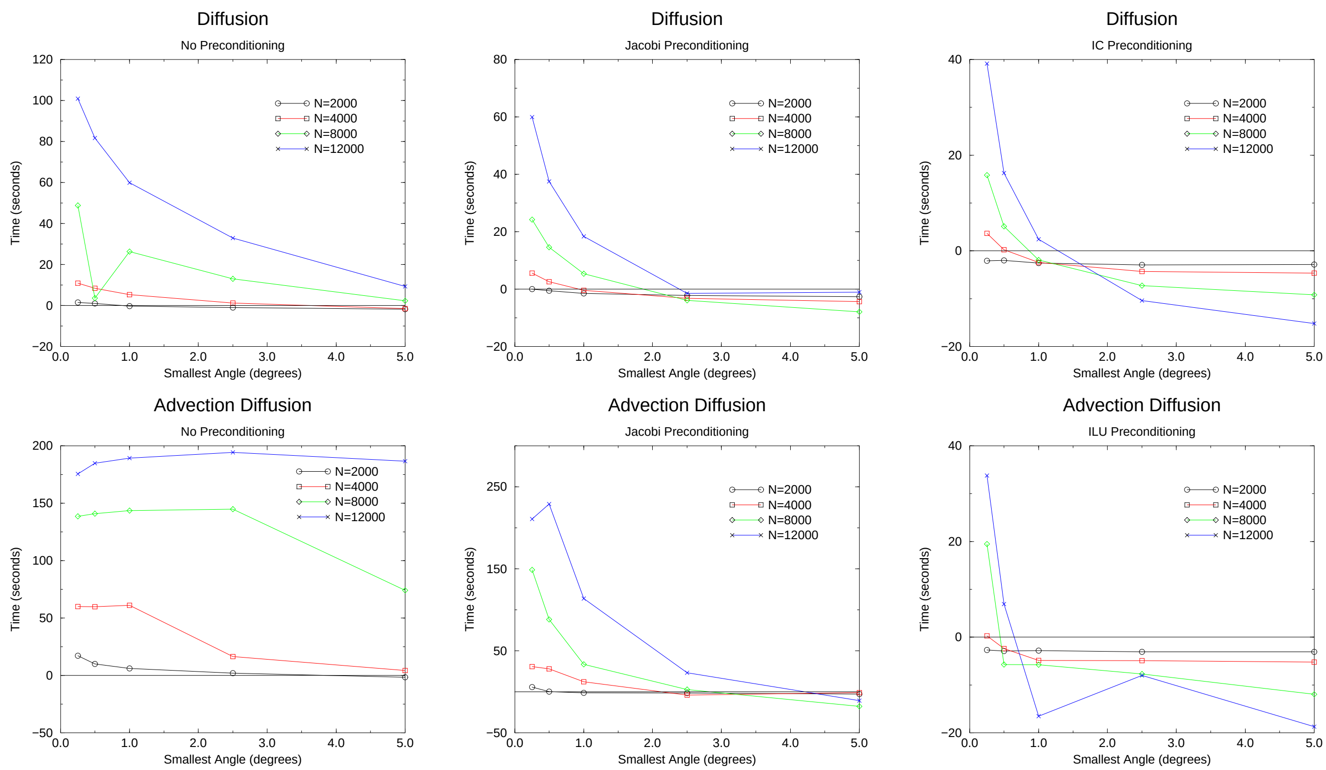

Figure 5: Difference in time to solve the system on a poor quality mesh and the total time required to solve the system on the improved mesh, including the mesh improvement time, for the diffusion and advection diffusion test cases

5.1 Linear Elasticity

The first problem that we consider is linear elasticity for the plane stress problem, which is given as

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= \frac{1+\nu}{2}\left(\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x \partial y}\right), \\
\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} &= \frac{1+\nu}{2}\left(\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 u}{\partial x \partial y}\right),
\end{aligned}$$

where u and v are the x and y displacements, respectively. These equations are solved on a rectangular region with a central hole. One side of the region is constrained to have zero displacement, and a constant traction is applied to the opposite side. For this example the initial mesh is created by recursively refining a coarse grid approximation to the domain using bisection techniques and an analytic description of the boundary. The upper right quadrant of an initial mesh for the elasticity problem is shown on the left of Figure 6; the bisection lines from the original coarse grid are clearly evident.

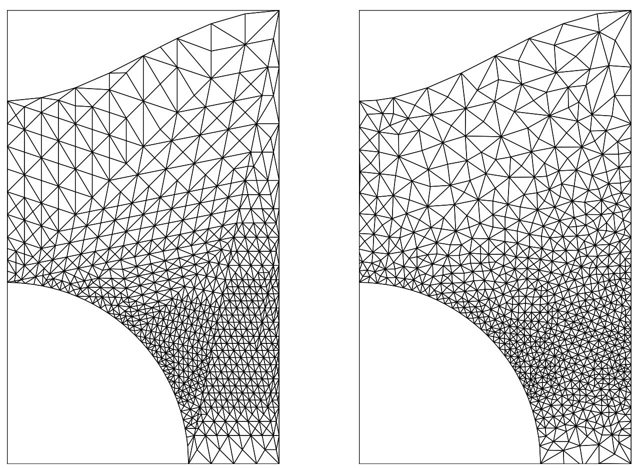

Figure 6: Upper right quadrant of a rectangular region with a circular hole. The figure on the left is the mesh resulting from bisection refinement without smoothing. The figure on the right is the same mesh with smoothing incorporated.

In Table 4 we give the minimum, maximum, and average angle measures for the original elasticity mesh, the same mesh improved by swapping, and the same mesh improved by swapping and smoothing for three different problem sizes. We also give the time required to perform the mesh improvement operation in seconds. The minimum angle in the original mesh is typically about 6°, the maximum angle is 168°, and the average angle is 38.53° Face swapping is effective in improving the average angle in the mesh, but does not affect the extremal angles. When mesh swapping and mesh smoothing are used together, θ_{min} and θ_{max} are both significantly improved over using swapping alone, and the average angle increases slightly over swapping used alone. In the current implementation, face swapping and node smoothing incur approximately the same computational cost and are clearly linear functions of N.

Table 4: Mesh characteristics and improvement times for three different problem sizes

	Original			Swapped				Swapped and Smoothed			
N	θ_{min}	θ_{max}	θ_{avg}	θ_{min}	θ_{max}	θ_{avg}	Time_{swap}	θ_{min}	θ_{max}	θ_{avg}	Time_{smooth}
20976	6.11	167.7	38.25	6.11	167.7	45.22	2.03	22.81	134.3	47.22	2.31
41216	6.09	167.7	36.28	6.11	167.7	41.59	4.93	16.23	146.51	42.43	3.90
82192	5.90	168.14	38.52	5.90	168.14	45.56	8.50	20.54	138.8	46.7	8.26

The elasticity equations are approximated with linear finite elements, and the resulting symmetric linear system is solved by using the CG/IC solver. The number of iterations required for convergence of the relative residual to less than 10^{-9} and the total time to solve the problem, including mesh improvement times when appropriate, are given in Table 5 for the meshes described in Table 4. Swapping used alone improved the total solution time by a small amount for the two largest problem sizes, but actually increased the total time to solution for the smallest problem considered. Using a combination of swapping and smoothing improved the solution efficiency in all three cases and, for the largest two problems, saved approximately 10 percent of the total solution time.

In the columns labeled E, we give the global H_1 error indicator values associated with the discrete approximation of the linear elasticity equations on that particular mesh. As the resolution increases, the value of E decreases. In addition, better-quality meshes also result in better accuracy; for the largest problem size the error indicator function was reduced an additional 13 percent for the mesh improved by both swapping and smoothing. We intend to examine the effect of mesh quality on solution accuracy in greater depth in future work.

Table 5: Mesh improvement results for the linear elasticity problem

	Original			Swapped				Swapped and Smoothed			
N	I	E	Total Time	I	E	Total Time	Time Saved	I	E	Total Time	Time Saved
20976	629	1.53E-4	39.3	639	1.49E-4	43.09	-3.79	532	1.39E-4	38.60	.69
41216	904	7.71E-5	112.8	814	7.42E-5	110.5	2.3	734	7.83E-5	104.6	8.2
82192	1289	4.17E-5	321.0	1209	4.09E-5	317.8	3.16	1085	3.62E-5	278.57	25.62

5.2 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

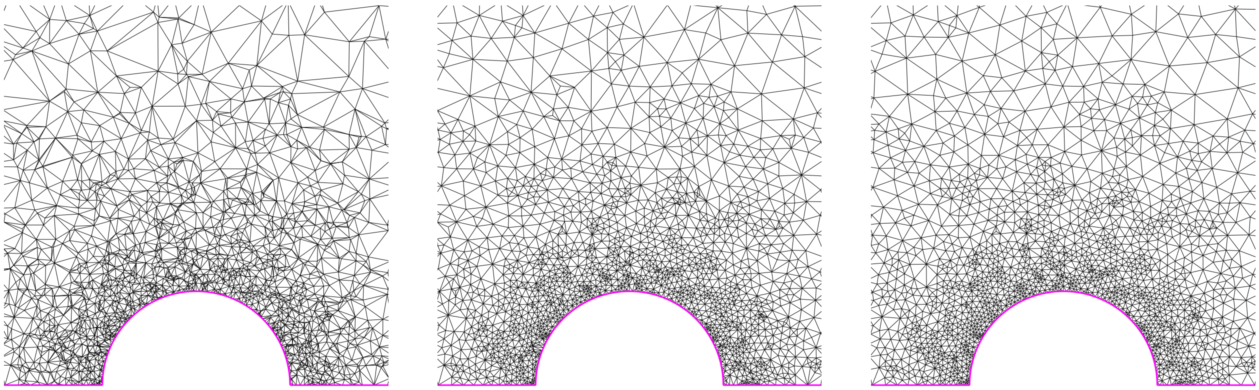

Figure 7: From left to right: random mesh, smoothed mesh, and smoothed and swapped mesh

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 by using the Delaunay criterion. The smallest angle in this mesh is 0.56°, the largest 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 by 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 by using an edge-based, vertex-centered finite volume solver. Second-order accuracy was attained by using least-squares reconstruction [9, 26, 27]. 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 [28]. In each case, the same three coarse meshes were used to eliminate the effects of coarse mesh convergence behavior on the results. Figure 8 also 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 each case, the mesh optimization procedures required less time than a single cycle of the multigrid solver, and the convergence rate is limited by behavior near the rear separation point on the cylinder, where the local time step is limited by acoustic modes while the solution is changing because of convective modes with a propagation speed of M = 0.01 or less.

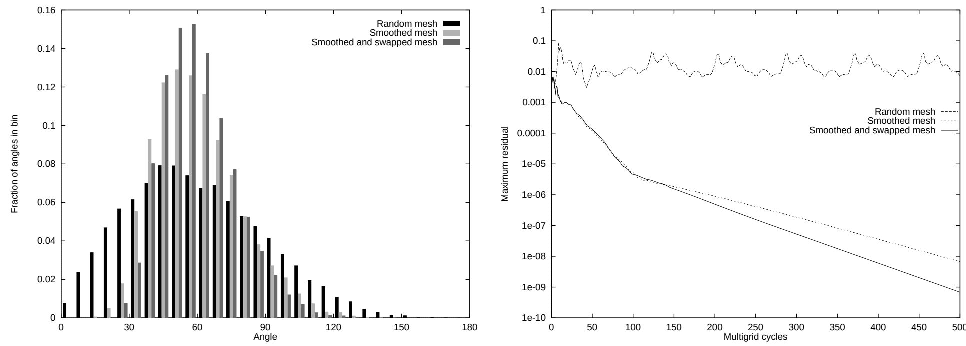

Figure 8: Angle distribution for cylinder meshes and the convergence histories for subsonic flow around a cylinder

6 Summary

In this paper we have presented a series of results that quantify the effect of mesh quality on both CG and GMRES iterative solvers, with and without preconditioning, for finite element and finite volume discretizations. We summarize our results as follows.

- In most cases, a few poor-quality elements do not significantly affect the convergence of the iterative solvers. The exception is GMRES with Jacobi preconditioning, which is sensitive to an increase in the ellipticity of the linear system.
- As the problem size or number of poor-quality elements increased, or as the element quality in the mesh degraded, the work to solve the problem increases. As the preconditioning of the iterative solver improves, the effect of poor mesh quality on solution time is less pronounced.
- For two very simple applications, a cost benefit analysis shows that for all the iterative solvers, the breakeven point in terms of mesh quality for which mesh smoothing was beneficial moves toward higher-quality meshes as N increases.
- For more complex applications, the benefits of mesh improvement are pronounced with savings of 10 percent or more in total solution time even when the initial meshes are of reasonable quality.

Future work includes an analysis of the benefits of mesh improvement in terms of solution accuracy for both finite element and finite volume discretization techniques. Our goal is to quantify the relationship between solution error and local mesh quality. We intend to begin by examining the error in model problem solutions and use Green's function techniques to determine the local discretization errors that induce the observed solution error.

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*, pages 528–537, 1997.

[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] O. Axelsson. A class of iterative methods for finite element equations. *Computational Methods and Applications in Mechanical Engineering*, 9:123–137, 1976.

[4] O. Axelsson and V. Barker. *Finite Element Solution of Boundary Value Problems*. Academic Press, 1984.

[5] O. Axelsson and G. Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numerische Mathematik*, 48:499–523, 1986.

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

[7] Satish Balay, William Gropp, Lois Curfman McInnes, and Barry Smith. PETSc 2.0 users manual. Technical Report ANL-95/11 – Revision 2.0.21, Argonne National Laboratory, 1997.

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

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

[10] 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.

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

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

[15] Anne Greenbaum and Lloyd Trefethen. GMRES/CR and Arnoldi/Lanczos at matrix approximation problems. *SIAM Journal on Scientific Computing*, 15(2):359–368, 1994.

[16] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[17] A. Jennings. Influence of the eigenvalue spectrum on the convergence rate of the conjugate gradient method. *Journal of the Institute of Mathematics and Its Applications*, 20:61–72, 1977.

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

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

[20] Mark T. Jones and Paul E. Plassmann. Parallel algorithms for adaptive mesh refinement. *SIAM Journal on Scientific Computing*, 18(3):686–708, 1997.

[21] Wayne Joubert. On the convergence behavior of the restarted gmres algorithm for solving nonsymmetric linear systems. *Numerical Linear Algebra with Applications*, 1(5):277–286, 1994.

[22] C. L. Lawson. Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3:231–246, 1986.

[23] N. M. Nachtical, S. C. Reddy, and L. N. Trefethen. How fast are nonsymmetric matrix iterations? *SIAM Journal of Matrix Analysis and Applications*, 13:778–795, 1992.

[24] Yvan Notay. On the convergence rate of the conjugate gradients in the presence of rounding errors. *Numerische Mathematik*, pages 301–317, 1993.

[25] Carl Ollivier-Gooch. Beta Release of the GRUMMP software, http://tetra.mech.ubc.ca/GRUMMP/, 1998.

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

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

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

[29] 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.

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

[31] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[32] 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.

[33] Z. Strakos. On the real convergence rate of the conjugate gradient method. *Linear Algebra and its Applications*, 154-156:535–549, 1991.