Electronic Transactions on Numerical Analysis.
Volume 2, pp. 171-182, December 1994.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

DOMAIN DECOMPOSITION AND MULTIGRID ALGORITHMS FOR ELLIPTIC PROBLEMS ON UNSTRUCTURED MESHES *

TONY F. CHAN† AND BARRY F. SMITH‡

Abstract. Multigrid and domain decomposition methods have proven to be versatile methods for the iterative solution of linear and nonlinear systems of equations arising from the discretization of partial differential equations. The efficiency of these methods derives from the use of a grid hierarchy. In some applications to problems on unstructured grids, however, no natural multilevel structure of the grid is available and thus must be generated as part of the solution procedure.

In this paper, we consider the problem of generating a multilevel grid hierarchy when only a fine, unstructured grid is given. We restrict attention to problems in two dimensions. Our techniques generate a sequence of coarser grids by first forming a maximal independent set of the graph of the grid or its dual and then applying a Cavendish type algorithm to form the coarser triangulation. Iterates on the different levels are combined using standard interpolation and restriction operators. Numerical tests indicate that convergence using this approach can be as fast as standard multigrid and domain decomposition methods on a structured mesh.

Key words. domain decomposition, grid refinement, mulitigrid, numerical partial differential equations.

AMS subject classifications. 65N30, 65F10.

1. Introduction.

Recently, unstructured meshes have become quite popular in large scale scientific computing [9], [14]. They have the advantage over structured meshes of the extra flexibility in adapting efficiently to complicated geometries and to rapid changes in the solution. However, this flexibility may come with a price. Traditional solvers that exploit the regularity of the mesh may become less efficient on an unstructured mesh. Moreover, vectorization and parallelization may become more problematic. Thus, there is a need to adapt and modify current solution techniques for structured meshes so that they can run as efficiently on unstructured meshes.

In this paper, we present some domain decomposition (DD) and multigrid (MG) methods for solving elliptic problems on unstructured triangular meshes in two space dimensions. The class of DD methods we consider were introduced in Dryja and Widlund [10] and Dryja [11]. These are among the most efficient algorithms for solving elliptic problems. The application of multigrid methods to unstructured grid problems has received some attention; see, for example, [14] and references therein. There has been relatively little work on domain decomposition methods for unstructured grid problems. Cai and Saad [3] considered overlapping domain decomposition methods for general sparse matrices, which in principle can be applied to the stiffness matrices arising from discretizations of elliptic problems on unstructured grids. However, if a coarse grid is to be used (often necessary for fast convergence), it cannot be deduced

* Receive April 26, 1994. Accepted for publication November 8, 1994. Communicated by O. B. Widlund. This paper will appear, in slightly different form, in the Proceedings of the Seventh Annual International Conference on Domain Decomposition. This research was supported by the National Science Foundation under contract ASC 92-01266, the Army Research Office under contract DAAL03-91-G-0150, and subcontract DAAL03-91-C-0047, and ONR under contract ONR-N00014-92-J-1890.

† Department of Mathematics, University of California, Los Angeles, CA 90024-1555 (chan@math.ucla.edu). Part of this work was performed while the author was visiting the Computer Science Department of the Chinese University of Hong Kong.

‡ Department of Mathematics, University of California, Los Angeles, CA 90024-1555. Current Address: MCS Division, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439-4844. (bsmith@mcs.anl.gov).

from the algebraic structure of the sparse matrix alone and geometric information about the coarse grid and the associated interpolation operators must be supplied.

For multigrid and domain decomposition algorithms, a hierarchy of grids, together with the associated interpolation and restriction operators, is needed. For structured meshes, this grid hierarchy is naturally available and is indeed exploited in these algorithms. For an unstructured mesh, however, the coarser grids may not be given. Thus, a procedure is needed that generates this grid hierarchy, as well as the associated interpolation and restriction operators. One approach is to generate the coarser meshes independently, using a mesh generator, possibly the one that generated the fine mesh in the first place. This approach has been used by Mavriplis [14], who constructed multigrid algorithms for the Navier-Stokes equations on unstructured meshes in two and three space dimensions. Another approach is to generate the grid hierarchy automatically and directly from the given unstructured fine grid. This approach requires less from the user because only the fine grid, on which the solution is sought, is needed.

In this paper, we will follow the second approach. Our techniques generate a sequence of coarser grids by first forming two maximal independent sets of the vertices of the fine grid, one for the interior vertices and the other for the boundary vertices, and then applying Cavendish's algorithm [5] to form the coarser triangulation. With this approach, the coarse mesh vertices form a subset of the fine mesh vertices. We also consider a variant in which this nested property of the vertices does not hold. Iterates on the different levels are combined using standard piecewise linear finite element interpolation and restriction operators. The mesh can be multiply-connected. Numerical tests indicate that convergence using both coarsening approaches can be as fast as standard multigrid and domain decomposition methods on a structured mesh.

Our use of a maximal independent set for the construction of the coarse grid is similar to the approach taken, independently, by Guillard [13]. For a recently developed convergence theory for two level Schwarz domain decomposition methods using nonnested coarse grids, see Cai [4], Chan and Zou [8], and Chan, Smith, and Zou [7].

2. Domain Decomposition and Multigrid Algorithms. For simplicity, we consider the following elliptic problem:

$$-\nabla \cdot \alpha(x,y)\nabla u = f(x,y), \qquad \alpha(x,y) > 0, \tag{2.1}$$

on a two dimensional (not necessarily simply connected) region Ω with appropriate boundary conditions. We assume that Ω is triangulated into a fine grid, which can be unstructured and non-quasi-uniform, and a finite element method is applied resulting in the algebraic system $Au = b$. The DD and MG algorithms we construct are used as preconditioners for A and are used in conjunction with a preconditioned Krylov subspace method.

We first discuss the overlapping Schwarz domain decomposition algorithms of Dryja and Widlund [10], [11] (a recent survey can be found in [6]). The fine grid Ω is decomposed into p overlapping subdomains $\Omega_i, i = 1, \cdots, p$, either as specified by the user or automatically determined by a mesh partitioning algorithm. In this paper, we use exclusively the *recursive spectral bisection* (RSB) method of Pothen, Simon, and Liou [15] to partition the element vertices (and hence the unknowns) into nonoverlapping subsets. The overlap is introduced algebraically by enlarging these subsets to contain all vertices within a fixed number of edges from the original subset. Note that even if the algebraic overlap is zero, the support of the finite element

ETNA
Kent State University
etna@mcs.kent.edu

basis functions in neighboring subdomains does overlap. Associated with each Ω_i are restriction and extension operators R_i and R_i^T ($R_i u$ extracts the components of u corresponding to Ω_i and $R_i^T u_i$ is the zero-extension of an iterate u_i on Ω_i to Ω) and the local stiffness matrix $A_i \equiv R_i A R_i^T$.

To achieve a good convergence rate, we also use a coarse grid consisting of the union of a set of coarse triangles. We denote by Ω_0 the coarse grid region that is the union of these coarse triangles. In this paper, we do not require that Ω_0 be the same as Ω or that the coarse triangles be nested to the fine ones. We construct the associated interpolation operator R_H^T, which maps an iterate u_0 on Ω_0 to Ω, as follows. If a fine grid node lies within a triangle of Ω_0, we use linear interpolation to obtain its value; otherwise we set its value to zero. Once R_H^T is defined, the restriction operator R_H is defined to be its transpose. Finally, we compute the coarse grid stiffness matrix A_H by applying the piecewise linear finite element method to (2.1) on Ω_0. Note that in general $A_H \neq R_H A R_H^T$ due to the non-nestedness of the grids.

With these operators defined, we now define the additive overlapping Schwarz preconditioner (which corresponds to a generalized block Jacobi method) as follows:

$$M_{as}^{-1} = R_H^T A_H^{-1} R_H + \sum_{i=1}^{p} R_i^T A_i^{-1} R_i.$$

Thus, each application of the preconditioner involves restricting the residual vector to each subdomain and performing a subdomain solve. In addition, a weighted restriction of the residual vector is computed on the coarse grid and inverted by a coarse grid solve. These local and coarse solutions are then mapped back onto the fine mesh and added together to obtain the desired result. Multiplicative versions (e.g., Gauss-Seidel) can also be defined analogously given an ordering of the subdomains.

Multilevel preconditioners (including classical multigrid methods) are closely related to domain decomposition methods, and their implementations can be treated in the same framework. A grid hierarchy is needed, and the associated interpolation and restriction operators can be defined analogously. For example, let the fine grid be level 1 and the coarsest grid level l. Let R_i denote the restriction operator from level 1 to level i, and let the transpose R_i^T denote the corresponding interpolation operator. Then an additive multilevel preconditioner can be written in the following form:

$$M_{aml}^{-1} = \sum_{i=1}^{l} R_i^T S_i R_i,$$

where S_i is a "smoother" on level i. For instance, for multi-level diagonal scaling, S_i is simply the inverse of the diagonal of the stiffness matrix on level i. One version of the V-cycle MG method can be viewed as a symmetrized multiplicative versions of the above preconditioner. Note that in practice the actions of R_i and R_i^T are computed via a recursion that use mappings between *adjacent* grid levels.

In our implementation of the domain decomposition algorithms, the coarse grid Ω_0 is obtained by a sequence of recursively applied coarsening steps (see the next section), and hence the grid hierarchy is naturally defined for performing the multigrid iteration as well. Of course, this is not the only way to construct the coarse grid for a DD method. For example, one can use the subdomains to directly construct a coarse grid without going through a grid hierarchy. We do not pursue these other possibilities in this paper.

ETNA
Kent State University
etna@mcs.kent.edu

3. Construction of the Grid Hierarchy. In this section, we describe our techniques for constructing the coarse grid hierarchy, as well as the associated interpolation and restriction operators, directly from the given unstructured fine mesh. It suffices to describe this for one coarse level; the procedure can be recursively applied to obtain all the coarse meshes.

We need the notion of a *maximal independent set* of the vertices of a graph. A subset of vertices V of a graph G is said to be *independent* if no two vertices of V are connected by an edge. V is said to be *maximally independent* if adding any additional vertex to it makes it dependent. Note that maximal independent sets of vertices of a graph are generally not unique.

The procedure has four steps:

(i) Form a maximally independent set of the boundary vertices, and from these construct a set of coarse boundary edges,
(ii) Form a maximally independent set of the interior vertices,
(iii) Apply a Cavendish type algorithm [5] (or any other suitable triangulation algorithm) to triangulate the resulting collection of coarse boundary edges and coarse interior vertices,
(iv) Construct the interpolation and restriction operators.

Step (i) is fairly straightforward. For each disjoint boundary segment, the boundary vertices are ordered, say, in a clockwise direction, starting with a random vertex. Then every other vertex is thrown out and the remaining ones are connected with new coarse boundary edges. This forms a coarse representation of the boundary segment. After several coarsenings, one may find that the boundary is no longer qualitatively similar to the original boundary. This situation may be prevented however, by simply retaining in the coarse grid boundary some of the vertices that would normally be dropped.

Step (ii) uses a greedy wavefront type algorithm. A random interior vertex is selected for inclusion in the maximally independent set. Then every interior vertex connected to it is eliminated from consideration for inclusion in the maximally independent set. Next, one of the interior vertices connected to the newly eliminated vertices is selected for inclusion, and the procedure repeats until all interior vertices have been considered. An algorithm similar to this has been used by Barnard and Simon [1] in designing graph partitioning algorithms. This procedure can be implemented in linear time, i.e. proportional to the total number of interior vertices.

The input to Step (iii) is thus a collection of coarse boundary edges and coarse interior vertices, which can then be triangulated by known triangulation algorithms, such as Delaunay type algorithms. We used a version of Cavendish's algorithm [5], which is an advancing front technique and "grows" new triangles from those already built by selecting an interior vertex to be "mated" to an existing edge. In doing so, it tries to optimize the aspect ratio of the new triangle formed, preferring those that are close to being equilateral. This algorithm may be implemented in linear time, i.e. proportional to the number of interior vertices, but our current implementation is not optimal.

Finally, in Step (iv), the interpolation operator is constructed in the form of a sparse matrix and stored. In order to determine the entries of this interpolation matrix, the coarse triangles are taken in sequence and the entries corresponding to all the fine grid vertices within the coarse triangle are then computed by using a standard piecewise linear interpolation. This procedure can also be implemented in linear time because the fine grid triangles close to the vertices of the coarse triangle (which are also fine grid vertices) can be found by a local search. We emphasize that a simple linear time algorithm is not possible if the coarser grids are generated completely independently. The restriction matrix is then just the transpose of the interpolation matrix. Clearly, higher order finite elements may also be used; then the interpolation is piecewise polynomial and can still be calculated in a local manner and hence in linear time.

In our variant of the above algorithm, only Step (ii) is changed. We consider a candidate coarse vertex at the center of each element. Those adjacent to the selected boundary nodes are then eliminated. We then construct a maximal independent set of the remaining candidate vertices using the same approach as indicated above. This procedure is equivalent to calculating a maximal independent set of the *dual* graph of the mesh. It is important to note that the coarse grid vertices generated in this manner will not lie in the same location as any fine grid nodes. In our experiments on a few sample grids, the number of coarse grid nodes generated using this alternative approach is slightly larger than that generated using the first approach outlined.

4. Numerical Results. All the numerical experiments were performed using the Portable, Extensible Toolkit for Scientific computation (PETSc) of Gropp and Smith [12] running on a Sun SPARC 10.

We next report numerical results for solving the Poisson equation on three different unstructured triangular meshes. All of the meshes are enclosed in the unit square, $[0,1]^2$. Two kinds of boundary conditions are used: (1) homogeneous Dirichlet, or (2) a *mixed* condition: for $x > 0.2$, a homogeneous Neumann boundary condition is imposed, with homogeneous Dirichlet imposed for $x \leq 0.2$. We use piecewise linear finite elements for the discretization, and solve the resulting systems of linear equations by either a V-cycle multigrid method (with a pointwise Gauss Seidel smoother, using two pre and two post smoothing sweeps per level) or an overlapping Schwarz domain decomposition method. In all cases, the discrete right hand side is chosen to be a vector of all 1's and the initial iterate set to zero. Both the MG and DD methods are used as preconditioners, with full GMRES [16] as an outer accelerator. The iteration is stopped when the l_2 norm of the residual has been reduced by a factor of 10^{-6}. Our goal is to compare the performance of our versions of domain decomposition and multigrid algorithms on unstructured meshes with that of the same algorithms on similar structured meshes. For this purpose, we also use two structured meshes in our experiments (a uniform mesh on a *square* and on an *annulus*).

Table 4.1 shows the number of MG iterations for the *Eppstein* mesh [2], shown in Figure 4.1, a relatively small quasi-uniform unstructured mesh on the unit square. Figures 4.2 and 4.3 show the coarser meshes using regular and dual graph coarsening. These results should be compared with those for a uniform *square* mesh on the unit square in Table 4.2, because the two meshes are topologically similar. We see that although the performance of our MG algorithm on the unstructured mesh is slightly worse than that on the structured square mesh, its performance is quite satisfactory for both types of boundary conditions.

Next, we look at a more realistic mesh, the *airfoil* mesh (from T. Barth and D. Jesperson of NASA Ames), shown in Figure 4.4. The coarse meshes are shown in Figures 4.5, 4.6 and 4.7. One may note that several poorly shaped triangles are generated on the coarsest grid. These do not seem to seriously affect the convergence rate. In theory, these bad elements could be removed during a "cleanup" pass over the mesh, after the Cavendish algorithm was applied. We see that the performance for the Dirichlet boundary condition cases, given in Table 4.3, are quite comparable with

ETNA
Kent State University
etna@mcs.kent.edu

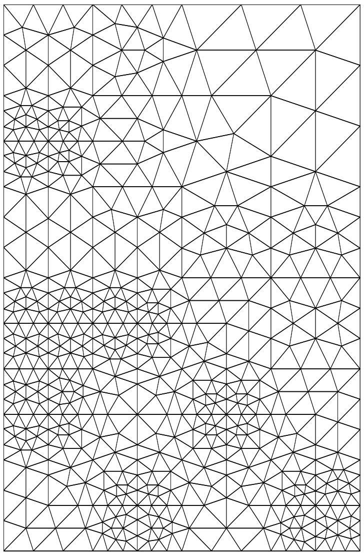

Fig. 4.1. *The Eppstein mesh: 547 nodes*

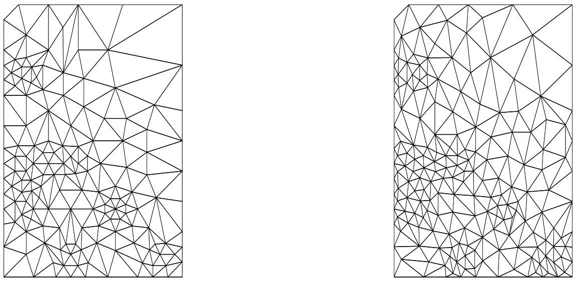

Fig. 4.2. *The Eppstein mesh: level 2. regular coarsening (left) and dual graph coarsening (right)*

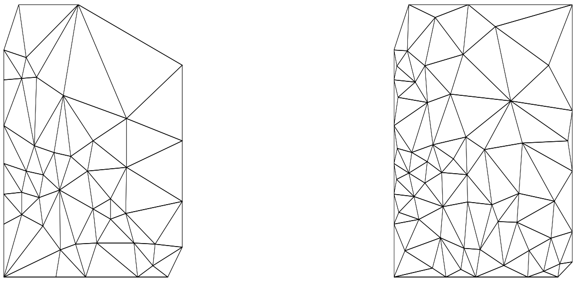

Fig. 4.3. *The Eppstein mesh: level 3. regular coarsening (left) and dual graph coarsening (right)*

Table 4.1
MG iterations for the Eppstein mesh, 547 nodes

	Regular Coarsening		Dual Graph Coarsening	
MG Levels	Dir. B.C.	Mixed B.C.	Dir. B.C.	Mixed B.C.
2	4	4	3	4
3	4	5	4	6

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.2
MG iterations for the uniform square mesh, 4225 nodes

MG Levels	Nodes	Dir. B.C.	Mixed B.C.
2	1089	3	3
3	289	3	3
4	81	3	3

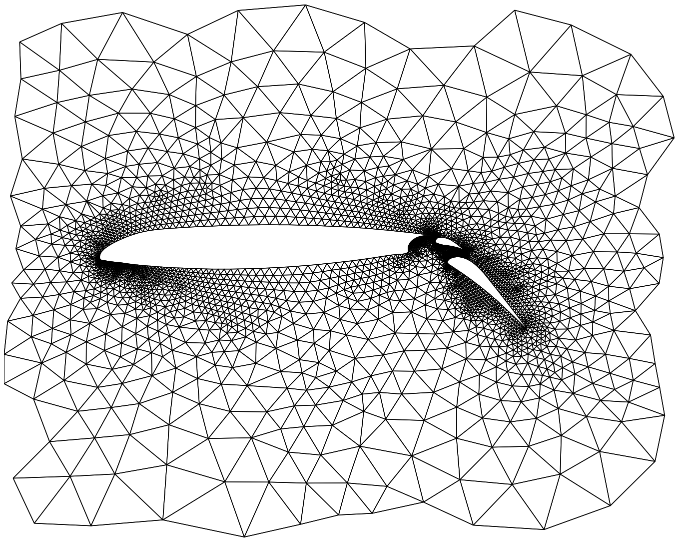

FIG. 4.4. *The airfoil mesh: 4253 nodes*

that for the *Eppstein* mesh, but the performance is noticeably worse for the mixed boundary conditions. Note there is essentially no difference in the performance for both types of coarsening. The deterioration in performance for the mixed boundary condition case is due to the fact that the coarser grid domains do not completely cover the portion of the fine grid boundary on which Neumann boundary conditions are applied. In Chan, Smith, and Zou [7], it is shown that in order to obtain an optimal convergence rate the coarser grid must completely cover the Neumann part of the fine grid boundary. We also compare the performance for the two types of boundary conditions with that on a quasi-uniform mesh on an annulus region; see Figure 4 and Table 4.4. Due to our treatment of the curved boundary, the fine grid boundary is again not covered by the coarser grids and hence we obtain poor convergence for the case of mixed boundary conditions. Overall, the performance of our MG algorithm on the unstructured *airfoil* mesh is comparable with or better than that on the structured *annulus* mesh.

In Table 4.5, we show results for a larger unstructured mesh around an airfoil, namely the *Barth* mesh (from T. Barth of NASA Ames), shown in Figure 4.9. The levels 2 and 4 coarse meshes are shown in Figure 4.10 (not all the grid points are shown). We include only the Dirichlet boundary condition results. Observe that the MG performance is quite comparable with that of the other unstructured meshes (i.e. *airfoil, Eppstein*) and the structured meshes (i.e., *square, annulus*). Again, both coarsening strategies work equally well.

Finally, we show in Table 4.6 the results for the multiplicative version of the over-

ETNA
Kent State University
etna@mcs.kent.edu

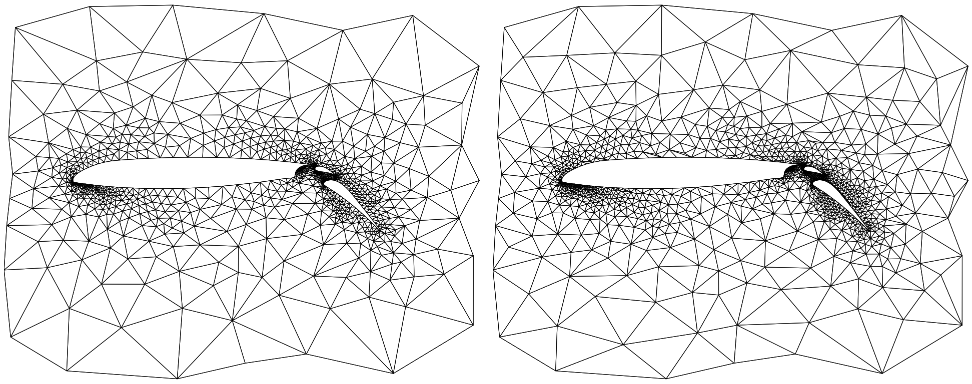

FIG. 4.5. *The airfoil mesh: Level 2. Regular (left) and dual graph coarsening (right).*

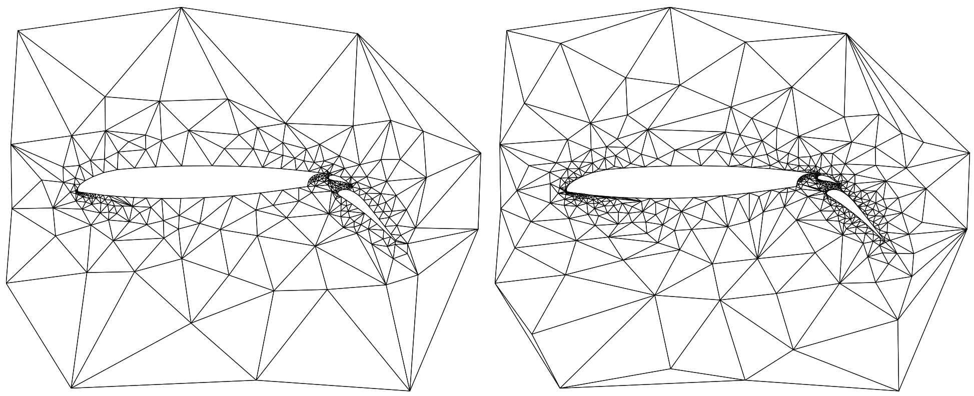

FIG. 4.6. *The airfoil mesh: Level 3. Regular (left) and dual graph coarsening (right).*

lapping Schwarz domain decomposition algorithm on the *airfoil* mesh. The column labeled *Overlap* refers to the algebraic overlap into the interior of its neighbors. Thus, an overlap of 0 means there is no overlap at all in the nodes. That is, the sets of nodes in each subdomain are disjoint. There is, however, overlap in the support of the finite element functions in neighboring subdomains. The column labeled *Level of Coarse Grid* refers to that level of the grid hierarchy that is used as the coarse grid in the DD algorithm. The 16 subdomains computed by the recursive spectral bisection method are shown in Figure 4.11. We can make several observations from these numerical results. First, the use of a coarse grid (even a relatively coarse one, e.g., level 4 in Table 4.6) reduces the number of iterations significantly. Second, the use of some overlap is very cost effective, but the number of iterations levels off quickly as the overlap increases. This behavior is well known from numerical studies of the overlapping Schwarz methods on structured grids. Finally, there is little difference in performance between the two coarsening strategies. These results for the overlapping Schwarz methods are also very similar to those obtained for structured grids.

5. Summary. We have constructed domain decomposition and multigrid algorithms for solving elliptic problems on general unstructured meshes, which in our limited experience perform nearly as well as these algorithms would perform on similar structured meshes. Only the fine mesh is needed; all auxiliary components of

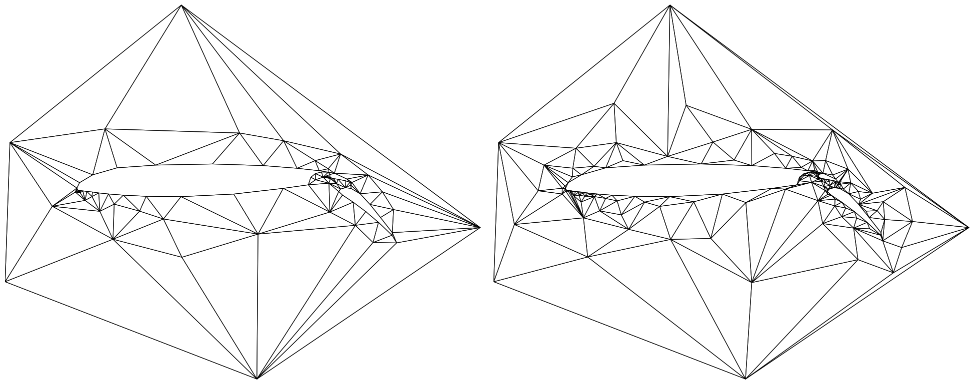

Fig. 4.7. *The airfoil mesh: Level 4. Regular (left) and dual graph coarsening (right).*

Table 4.3
MG iterations for the airfoil mesh, 4253 nodes

	Regular Coarsening			Dual Graph Coarsening		
MG Levels	Nodes	Dir. B.C.	Mixed B.C.	Nodes	Dir. B.C.	Mixed B.C.
2	1180	4	8	1507	4	8
3	518	4	9	328	4	9
4	89	4	10	171	5	10

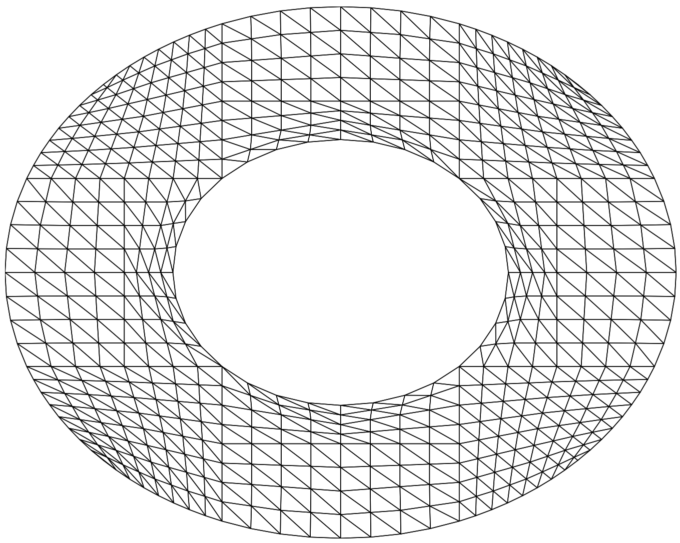

Fig. 4.8. *The annulus mesh: 576 nodes*

Table 4.4
MG iterations for the annulus mesh, 2176 nodes

MG Levels	Nodes	Dir. B.C.	Mixed B.C.
2	576	4	18
3	160	5	18
4	48	5	18

ETNA
Kent State University
etna@mcs.kent.edu

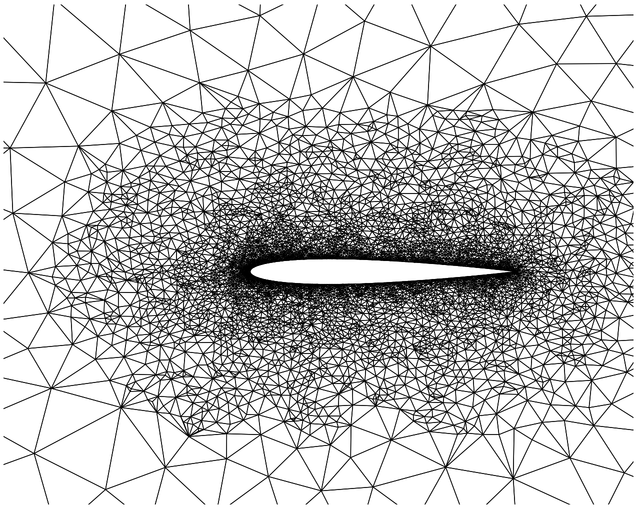

FIG. 4.9. *The Barth mesh: 6691 nodes*

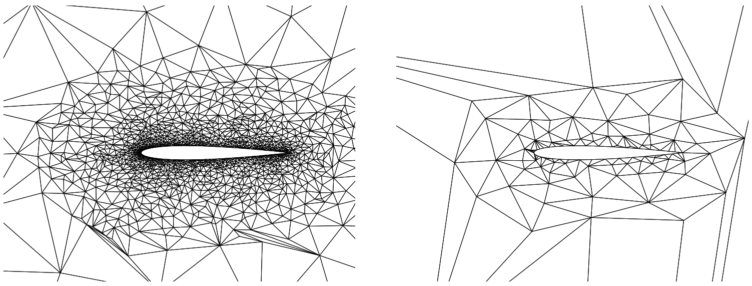

FIG. 4.10. *The Barth mesh: Level 2, 1614 nodes (left) Level 4, 112 nodes (right)*

the algorithms, such as the coarse grid hierarchy, the interpolation operators, and the domain partitioning, are computed automatically. The algorithms can, in principle, be extended to three space dimensions and to indefinite, non-self-adjoint and higher order problems.

Acknowledgments: We thank Mr. Nip Chun-kit of the Chinese University of Hong Kong for providing us with a Matlab implementation of the Cavendish algorithm, which helped us in developing the C version. We also thank Horst Simon and John Gilbert for providing several of our test meshes. The first author also thanks Prof. Ji-achang Sun of the Computing Center of Academia Sinica for many helpful discussions on the subject.

ETNA
Kent State University
etna@mcs.kent.edu

Table 4.5
MG iterations for the Barth mesh, 6691 nodes

	Regular Coarsening		Dual Graph Coarsening	
MG Levels	Nodes	Dir. B.C.	Nodes	Dir. B.C.
2	1614	5	1810	5
3	405	6	574	6
4	112	7	189	6

Table 4.6
Multiplicative DD iterations for the airfoil mesh. 16 Subdomains

Overlap (no. elements)	Level of coarse grid	Regular Coarsening	Dual Graph Coarsening
0	None	56	56
0	4	21	22
0	3	15	12
1	None	16	16
1	4	10	10
1	3	7	7
2	None	14	14
2	4	8	8
2	3	5	5

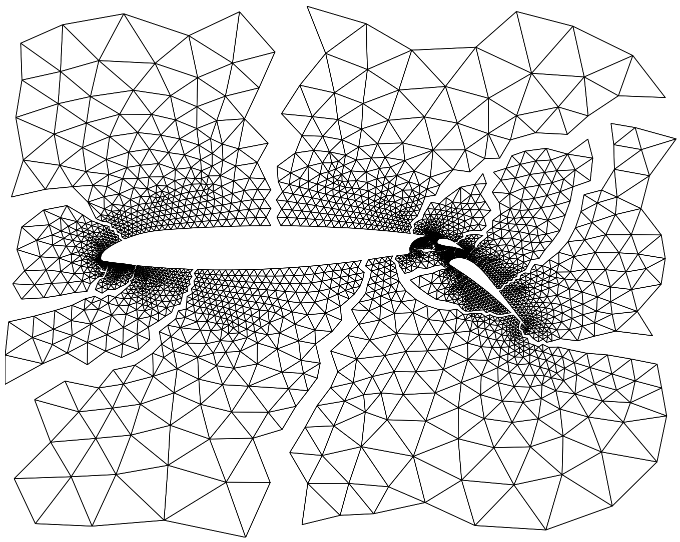

Fig. 4.11. *The airfoil mesh: 16 subdomains computed by RSB*

REFERENCES

[1] S. T. Barnard and H. D. Simon, *A fast multilevel implementation of recursive spectral bisection*, Proc. 6th SIAM Conf. Parallel Proc. for Sci. Comp. 1993, pp. 711–718.

[2] M. Bern, D. Eppstein, and J. Gilbert, *Provably good mesh generation*, 31st Annual Symp. on Foundations of Computer Science, IEEE, 1990, pp. 231-241.

[3] X.-C. Cai and Y. Saad, *Overlapping domain decomposition algorithms for general sparse matrices*, Technical Report 93-027, Univ. of Minn., AHPCRC, March 1993.

[4] X.-C. Cai, *The use of pointwise interpolation in domain decomposition methods with non-nested coarse meshes*, to appear, SIAM J. Sci. Comp., 16, (1995).

[5] J. C. Cavendish, *Automatic triangulation of arbitrary planar domains for the finite element method*, Internat. J. Numer. Methods Engrg., 8 (1974) pp. 679–696.

[6] T. F. Chan and T. Mathew, *Domain decomposition algorithms*, Acta Numerica, 1994, pp. 1–83.

[7] T. F. Chan, B. F. Smith, and J. Zou, *Overlapping Schwarz methods on unstructured grids using non-matching coarse grids*, Technical Report: CAM 94-08, UCLA Department of Mathematics, February 1994.

[8] T. F. Chan and J. Zou, *Additive Schwarz domain decomposition methods for elliptic problems on unstructured meshes*, Technical report CAM 93-40, Dept. of Mathematics, Univ. of Calif. at Los Angeles, December, 1993, to appear in Numer. Algorithms.

[9] H. Deconinck and T. J. Barth, *Special course on unstructured grid methods for advection dominated flows*, Lecture Series, Von Karman Inst., Belgium, AGARD-R-787, March 1992.

[10] M. Dryja and O. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Technical Report 339, Department of Computer Science, Courant Institute, 1987.

[11] M. Dryja, *An additive Schwarz algorithm for two- and three-Dimensional finite element elliptic problems*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1989.

[12] W. D. Gropp and Barry F. Smith, *Portable, Extensible Toolkit for Scientific Computation*, (PETSc), Available via anonymous ftp at `info.mcs.anl.gov` in the directory `pub/pdetools.`

[13] H. Guillard, *Node-nested multigrid method with Delaunay coarsening*, Technical report, INRIA, Sophia Antipolis, France, 1993.

[14] D. J. Mavriplis, *Unstructured mesh algorithms for aerodynamic calculations*, Technical Report 92-35, ICASE, NASA Langley, Virginia, July 1992.

[15] A. Pothen, H. Simon, and K. P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990) pp. 430–452.

[16] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986) pp. 856–869.