SIAM J. SCI. STAT. COMPUT.
Vol. 8, No. 6, November 1987

MULTIGRID ON COMPOSITE MESHES*

W. D. HENSHAW† AND G. CHESSHIRE‡

Abstract. The multigrid method is applied to the numerical solution of elliptic equations on general composite overlapping meshes. Computational results show that good convergence rates are obtained.

Key words. composite meshes, overlapping grids, multigrid

AMS(MOS) subject classification. 65N20

1. Introduction. We describe the application of the multigrid method to the solution of elliptic partial differential equations (PDEs) on two-dimensional regions which have been discretized using composite overlapping grids. A general purpose code, CGMG, has been developed (in FORTRAN) which can solve problems on composite meshes created by the grid construction program CMPGRD, Chesshire [5]. With CMPGRD the user may create a composite grid containing any number of component grids at any number of multigrid levels. CGMG can then be used to solve elliptic PDE boundary value problems.

A composite overlapping grid consists of a number of simpler *component* grids. These component grids cover a region and overlap where they meet. Functions defined on the composite mesh are matched by interpolation at the overlapping grid boundaries. The problem of generating grids for regions of complicated geometry can be difficult, especially for those grid generation algorithms which attempt to fit a single global grid. With a composite overlapping grid, however, the component grids can be generated almost independently of each other. Each component grid can be stretched and refined with little effect on the other component grids. The numerical solution of PDEs on such grids has been examined by, among others, Starius [10], [11], Reyna [9], Kreiss [8], Atta and Vadyak [1], Benek et al. [2], Henshaw [7] and Berger [3].

The multigrid method is a fast iterative method for the solution of elliptic problems. Multigrid utilizes a sequence of grids of varying degrees of coarseness to accelerate the convergence of the solution on the finest grid. The basic principle rests on the fact that it is possible to obtain iterative procedures (*smoothers*) for which the high frequency components of the solution converge rapidly. This means that after a few smoothing iterations the part of the solution yet to converge is smooth and hence can be accurately solved for on a coarser grid. Discussions of the multigrid method in general can be found, for example, in Brandt [4], or Stüben and Trottenberg [12]. Multigrid on a model composite mesh has been described in Stüben and Trottenberg [12] and for more general two component meshes in Henshaw [7].

In the next section we outline the implementation of the multigrid algorithm on fairly general composite meshes. This includes a description of the technique we use to discretize PDEs on overlapping meshes. In § 3 numerical results are presented for some test cases. These results show that the good convergence rates expected from multigrid can be obtained on composite meshes.

* Received by the editors March 3, 1986; accepted for publication (in revised form) December 8, 1986. This work was supported by National Science Foundation grant DMS-8312264 and Office of Naval Research grant N00014-83-K-0422.

† Department of Applied Mathematics, California Institute of Technology, Pasadena, California 91125. Present address, IBM, Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

‡ Department of Applied Mathematics, California Institute of Technology, Pasadena, California 91125. Present address, Intel Scientific Computers, Beaverton, Oregon 97006.

2. Description. The multigrid code CGMG was written to solve linear, variable coefficient elliptic PDEs of the form

$$Lu := c_{xx}\frac{\partial^2 u}{\partial x^2} + c_{xy}\frac{\partial^2 u}{\partial x\,\partial y} + c_{yy}\frac{\partial^2 u}{\partial y^2} + c_x\frac{\partial u}{\partial x} + c_y\frac{\partial u}{\partial y} + c_c u = f \quad \text{in } \Omega, \tag{1a}$$

$$Bu := b_x\frac{\partial u}{\partial x} + b_y\frac{\partial u}{\partial y} + b_c u = g \quad \text{on } \partial\Omega. \tag{1b}$$

Periodic boundary conditions are also allowed. Currently a second-order difference approximation has been implemented.

Let us first describe how the above PDE boundary value problem is discretized on a composite mesh, without any references to multigrid. The grid construction program CMPGRD can be used to generate a composite overlapping mesh for the region Ω. This mesh is composed of one or more component meshes. See, for example, the composite meshes of Figs. 1 and 2. Each component mesh is logically rectangular,

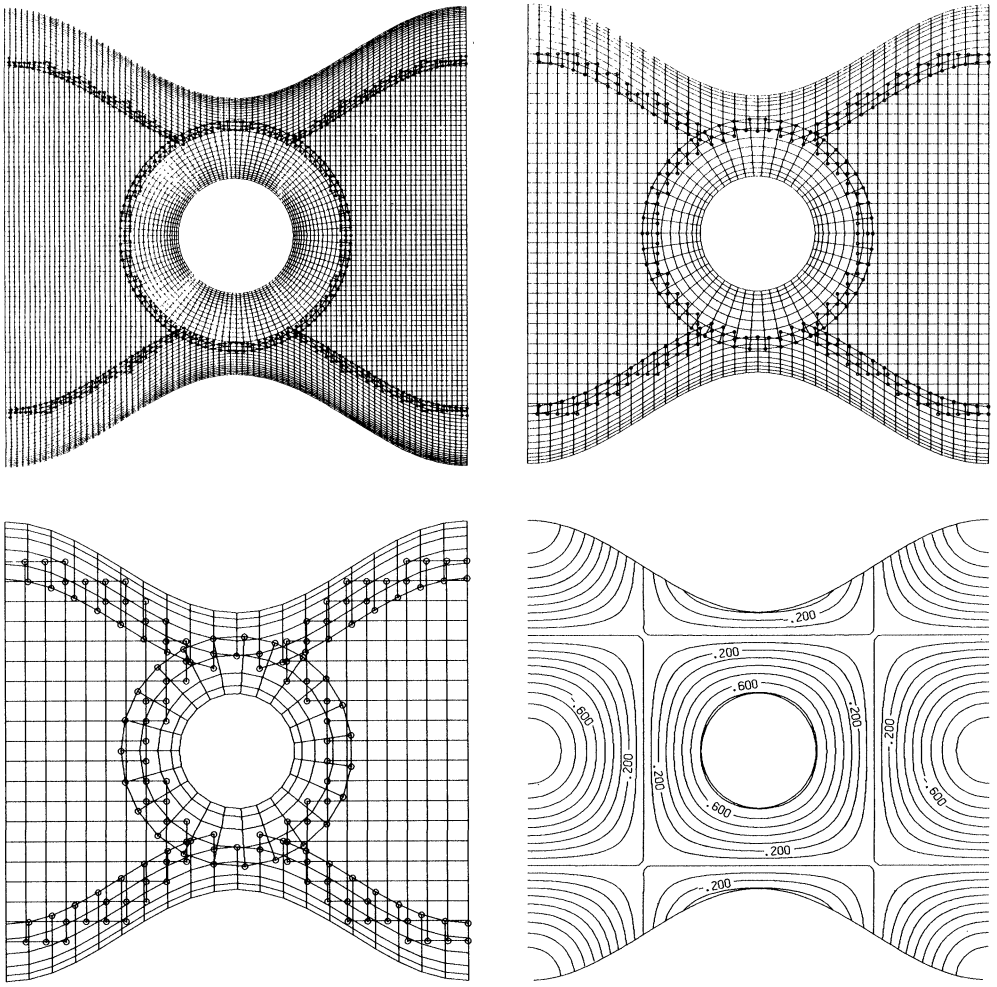

FIG. 1. *Grid* 1: *Composite meshes for multigrid levels* 1, 2 *and* 3 *and a contour plot of the calculated solution.*

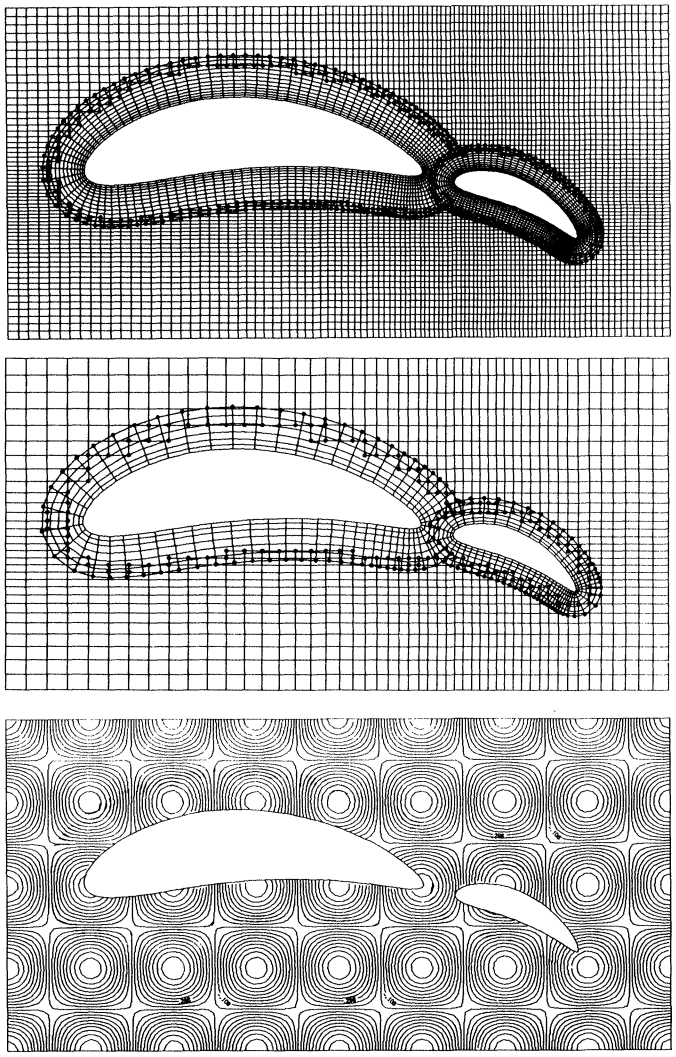

FIG. 2. *Grid* 2: *Composite meshes for multigrid levels* 1 *and* 2 *and a contour plot of the calculated solution.*

although some points in overlap regions are discarded. A point on a component grid will be one of the following four types:

(i) An *interior* point where the PDE (1a) should be applied.

(ii) A *boundary* point, which corresponds to $\partial\Omega$, where the boundary condition (1b) holds.

(iii) A point where the solution is matched, by interpolation to the solution on other component grids. We sometimes call the set of all such interpolation points the *interpolation boundary.*

(iv) A point in a region of overlap which is not used.

There are a number of possible ways to generate the discrete approximations to the PDE (1a), the boundary conditions (1b) and the interpolation equations. The approach we take is a *mapping* method and proceeds as follows. Each component grid is simple enough so that it can be mapped smoothly to a unit square (coordinates (r, s)). The

PDE can be written in these (r, s) coordinates.

$$Lu = c_{rr}\frac{\partial^2 u}{\partial r^2} + c_{rs}\frac{\partial^2 u}{\partial r\,\partial s} + c_{ss}\frac{\partial^2 u}{\partial s^2} + c_r\frac{\partial u}{\partial r} + c_s\frac{\partial u}{\partial s} + c_c u = f,$$

$$Bu = b_r\frac{\partial u}{\partial r} + b_s\frac{\partial u}{\partial s} + b_c u = g.$$

The coefficients in these equations depend on derivatives of the mapping. These derivatives are supplied as output from the composite grid construction program CMPGRD. The solution of a PDE on a composite mesh can then be considered as the solution of PDEs on a sequence of unit squares. The solutions on the squares are coupled to each other through the interpolation boundaries. Let the discrete solution at point (i, j) on component grid k be denoted by $v_k(i, j)$. Then for each k we obtain discrete approximations to the elliptic equation and the boundary conditions of the form

$$L_k v_k = f_k, \tag{2a}$$

$$B_k v_k = g_k. \tag{2b}$$

A point (i, j, k) on an interpolation boundary is interpolated from some other component grid k'. CMPGRD supplies the position (r', s') of the point (i, j, k) on grid k'. Hence standard interpolation formulae for rectangular grids can be applied:

$$v_k(i, j) = \sum_{i', j'} \alpha_{kk'}(i, j, i', j')\, v_{k'}(i', j'). \tag{2c}$$

These equations (2a), (2b) and (2c) will be written as a single linear system:

$$\mathbf{A}\mathbf{v} = \mathbf{f}. \tag{3}$$

The vector $\mathbf{v}$ of all unknowns will be called a composite mesh function. $\mathbf{A}$ is an example of a composite mesh operator, mapping one composite mesh function to another composite mesh function.

The mesh equations (3) can be solved in any number of ways; using multigrid is but one possibility. If there are not too many equations the system can be solved directly. Sparse matrix routines [6] have been used for this purpose. For large systems iterative methods become attractive. Many standard iterative methods are applicable. The matrix is not symmetric, however, so that some schemes do not apply. In general it is best to try and solve all the equations simultaneously, rather than iterating for too long on one component grid. The work of J. Linden, as reported in Stüben and Trottenberg [12], indicates that an iteration based on the Schwartz alternating procedure (where the equations on each component grid are solved exactly before updating interpolation boundaries) is slower and sensitive to the amount of overlap. In contrast it was found that by using the smoothers presented below, the multigrid convergence rate was fairly insensitive to the amount of overlap. The general principle to follow seems to be to iterate in such a way that, at any time, all equations have converged to about the same degree.

2.1. Multigrid. The multigrid algorithm can be applied to the solution of the mesh equations (3). Once a composite mesh has been constructed using CMPGRD it is a simple matter to have CMPGRD generate the sequence of coarser composite meshes which are used for the multigrid algorithm. Figures 1 and 2 show some composite meshes which have been generated for multigrid. Denote the finest composite mesh, level 1, by $\mathbf{M}^1$ and successively coarser meshes by $\mathbf{M}^l$, $l = 2, 3, \cdots$. Note that for simplicity the composite meshes at the different levels all have the same number of component meshes. The elliptic PDE boundary value problem can be discretized on each of the composite meshes $\mathbf{M}^l$. Let $\mathbf{v}^l$ denote the composite mesh function for level

l. Then at each level l there will be mesh equations of the form (3):

$$\mathbf{A}^l\mathbf{v}^l = \mathbf{f}^l. \tag{4}$$

Now let us outline the multigrid algorithm as it applies to composite meshes. Let $\mathbf{v}^1(p)$ be the pth iterate in the solution of the mesh equations on the finest mesh $\mathbf{M}^1$.

MULTIGRID ALGORITHM.
while not converged **do**
  smooth ν_1 times
    $\mathbf{v}^1(*) \leftarrow (\mathbf{S}^1)^{\nu_1}\mathbf{v}^1(p)$
  compute the defect and transfer to the coarser grid
    $\mathbf{f}^2 \leftarrow \mathbf{R}^{1\to 2}(\mathbf{f}^1 - \mathbf{A}^1\mathbf{v}^1(*))$
  "solve" the defect equation on the coarser grid
    $\mathbf{v}^2 \leftarrow (\mathbf{A}^2)^{-1}\mathbf{f}^2$
  correct the fine grid solution from coarse grid solution
    $\mathbf{v}^1(**) \leftarrow \mathbf{v}^1(*) + \mathbf{P}^{2\to 1}\mathbf{v}^2$
  smooth ν_2 times
    $\mathbf{v}^1(p+1) \leftarrow (\mathbf{S}^1)^{\nu_2}\mathbf{v}^1(**)$
end while

This is the basic defect correction scheme. The smoothing operator $\mathbf{S}^1$, the restriction operator $\mathbf{R}^{1\to 2}$ and the prolongation operator $\mathbf{P}^{2\to 1}$ will be described in the context of composite grids. The defect equation need only be "solved" approximately. This approximate solution can be obtained by multigrid, in which case the algorithm becomes recursive. At the coarsest level the equations are usually solved directly.

Smoothers. The smoothing operator $\mathbf{S}$ for composite grids consists of smoothing each component grid and updating the interpolation boundaries. A component grid may be smoothed with any of the standard smoothers that exist. The program CGMG allows the user to choose from a number of possibilities including Gauss-Seidel, Red-Black-Gauss-Seidel, Zebra line smoothers and alternating Zebra line smoothers. Each component grid can have a different smoother; the smoother can be tailored to the grid. For example, a particular component grid might be stretched to resolve a boundary layer, in which case one can use a line smoother in the appropriate direction.

There is some freedom as to the order of smoothing and interpolation. One possibility is to smooth all component grids before updating the interpolation boundaries:

$$\mathbf{S}^l = \begin{cases} \text{Smooth first component grid,} \\ \text{Smooth second component grid,} \\ \quad\vdots \\ \text{Smooth last component grid,} \\ \text{Interpolate.} \end{cases}$$

From experience, however, it seems that a good procedure involves interpolating after each component grid is smoothed:

$$\mathbf{S}^l = \begin{cases} \text{Smooth first component grid,} \\ \text{Interpolate.} \\ \text{Smooth second component grid,} \\ \text{Interpolate.} \\ \quad\vdots \end{cases}$$

The latter composite smoother requires more interpolations but this extra work is usually small compared to the smoothing operations.

Restriction operators (fine to coarse grid transfer). The defect computed on a given level is transferred to the next coarsest level by the restriction operator $\mathbf{R}^{1\to 2}$. (The superscript $1\to 2$ indicates that this operator maps mesh functions on $\mathbf{M}^1$ to mesh functions on $\mathbf{M}^2$.) A typical restriction operator determines the value at points on the coarse grid as some weighted average of the surrounding points on the fine grid. We use the so-called full weighting restriction. The defects in the boundary equations are averaged separately from the defects in the interior equations; boundary defects are averaged along the boundary line. Since the final stage of the smoothing operation involves an interpolation, the defects in the interpolation equations are all zero. Hence no defect need be transferred at these points.

Prolongation operators (coarse to fine grid transfer). The prolongation operator $\mathbf{P}^{2\to 1}$ maps the coarse grid solution to the fine grid. This mapping usually takes the form of an interpolation. We use second order interpolation. Interpolation boundary values could be corrected as well. However, once all other values have been corrected the interpolation equations can be solved to update the interpolation boundary. It turns out that less overlap is needed on finer meshes for the latter approach.

Choice of parameters and cycle. An important part of the multigrid algorithm is the choice of the parameters ν_1, ν_2, etc. and the choice of cycle. *Cycle* is the term used to denote the sequence in which the different levels of grids are traversed. The program dynamically determines the type of cycle and the values for ν_1 and ν_2 in a manner similar to that described by Brandt [4]. There are two basic principles:

(i) Perform smoothing iterations until the smoothing rate (the reduction in residual per iteration) becomes larger than some value η, where $\eta \approx .6$.

(ii) Return to a finer grid once the residual at this level has been reduced by a certain factor δ with $\delta \approx .1$.

We have found it very helpful to monitor the smoothing rates and component grid residuals as the iteration proceeds. This information can be used to determine what changes can be made to improve the convergence rate. If, for example, one component grid is converging slowly one can choose a better component smoother for that grid. Ideally the program should automatically choose the parameters η and δ and the types of component smoothers. A program with this level of sophistication has not yet been developed.

Remarks on grid construction. For practical reasons the amount of overlap between component grids is kept fairly small. This reduces the number of computational points. With sufficient overlap, points on an interpolation boundary can be explicitly interpolated from noninterpolation points on other component grids. However, as the amount of overlap decreases the interpolation equations may couple interpolation points from different component grids. In this case a system of equations must be solved to obtain values on the interpolation boundaries in terms of other values. Of course, as the overlap goes to zero these equations may become singular. The behaviour of the numerical solution to a model elliptic problem, as a function of the amount of overlap, was considered in Henshaw [7]. Suppose that the amount of overlap goes to zero as the grid is refined. To maintain accuracy, the order of accuracy of the interpolation formulae must be greater than the order of accuracy of the interior formulae. In the results presented here we use second order accurate approximations to the elliptic equations and the amount of overlap is required to be greater than one half a grid line. In this case the interpolation formulae should be third order accurate.

To make the restriction and prolongation operators simple the composite meshes at different multigrid levels are strongly related. For example, interior points on a

coarse grid coincide with interior points on finer grids. In Henshaw [7] the coarse grid was generated first and all finer grids were generated by simply doubling the number of grid lines. This gave reasonable results for composite grids with 2 component grids. However, this method does not immediately generalize to more component grids and it tends to lead to more overlap on the finer grids than is really necessary. Hence the grid construction program CMPGRD was designed explicitly to meet the requirements of the multigrid routine [5].

Remarks on programming. A computer code for solving problems on composite meshes is somewhat more complicated than a code written for a single grid. One of the major difficulties arises in the handling of all the data which describes the composite grid. We have worked out a scheme for storing the composite grid data in an efficient and flexible manner. Physically, all the data is stored on a single array. Logically, the data is stored in a "directory-tree" fashion. Variables can be stored or accessed by name using utility routines. Arrays can be stored with a minimum of wasted space. With this storage structure the composite grid data can be easily passed to subroutines.

3. Numerical results. In this section we present results from the multigrid solver CGMG. The composite grids used in the examples are shown in Figs. 1 and 2. Grid 1 (Fig. 1) has 3 levels and Grid 2 (Fig. 2) has 2 levels. The interpolation points are marked with small circles. Notice that the finer meshes are not simply the coarser meshes with double the number of lines. There is less overlap between component grids on the finer meshes. One restriction associated with composite overlapping grids is illustrated by these figures; for complicated regions the most coarse mesh that can be constructed will still have a substantial number of grid points.

In the two test examples which follow we solve

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=f, \qquad f=-2\pi^2 \cos \pi x \cos \pi y. \tag{5}$$

The boundary conditions are chosen so that the true solution is

$$u_{\text{true}}=\cos \pi x \cos \pi y.$$

One reason for solving this rather simple problem is to allow the reader to compare the results presented here to other well known results for Poisson's equation. In addition it is shown that similar convergence rates are obtained for solving the same problem on two widely different composite meshes.

Test 1. The Poisson equation (4) is solved on Grid 1. This grid might be used to study flow around an obstacle in a varying channel. Periodic boundary conditions are applied at the left and right ends of the channnel. Dirichlet boundary conditions, $u=u_{\text{true}}$, are given at all other boundaries. For smoothers we chose Red-Black on the rectangular grid and line-Zebra smoothers for the 3 curvilinear grids. The Zebra smoothers are on lines normal to the boundaries since this is the direction in which the grids are stretched. Smoothing was performed until the smoothing rate became greater than $\eta=.6$. The residual at level 2 had to be decreased by a factor of $\delta=.01$ before the program would return to level 1. These parameters resulted in a "W" cycle, that is the levels were traversed in the order (1, 2, 3, 2, 3, 2, 1). A value of $\delta=.1$ resulted in a "V" cycle, (1, 2, 3, 2, 1), which gave slightly inferior results.

Test 2. Equation (4) is solved on Grid 2 with Dirichlet boundary conditions at all boundaries. Grid 2 is meant to represent an airfoil with a flap. Alternating Zebra

smoothers were used on all component grids since they all are stretched in both the r and s directions. A value of $\eta = .36 = (.6)^2$ was chosen since the alternating smoothers comprise two sweeps.

Convergence results for the two tests are summarized in Table 1. $r(p)$ is the residual on the finest composite mesh after the pth multigrid iteration,

$$r(p) = \|\mathbf{f}^1 - \mathbf{A}^1\mathbf{v}^1(p)\|.$$

Define $WU(p)$ to be the number of work units used for the pth iteration. A work unit is the amount of work (number of multiplications) to perform one iteration of SOR on the composite mesh. The effective convergence rate, ECR, is defined as

$$\text{ECR}(p) = \left(\frac{r(p)}{r(p-1)}\right)^{\rho}, \qquad \rho = (WU(p))^{-1}.$$

Theoretically the ECR for multigrid should be independent of the grid spacing h as $h \to 0$. In contrast the (effective) convergence rate for many other standard iterative schemes deteriorates as $h \to 0$. For optimal SOR, $\text{ECR} \approx 1 - c_1 h$, and for Gauss-Seidel, $\text{ECR} \approx 1 - c_2 h^2$. In Table 1 the ECR increases on the fourth iteration for Grid 1 since the solution has almost converged to single precision accuracy.

In Tables 2 and 3 the errors in the solution to the elliptic problem are given. The problem was solved on the fine mesh (level 1) and also on each of the coarser meshes. The error on component grid k, multigrid level l, $e(k, l)$, is the maximum difference

TABLE 1
Convergence rates.

	Grid 1		Grid 2	
Iteration	$r(p)/r(p-1)$	ECR (p)	$r(p)/r(p-1)$	ECR (p)
$p=1$	.019	.71	.091	.73
$p=2$	.065	.72	.033	.63
$p=3$	.055	.71	.11	.75
$p=4$	.385	.89	.14	.77

TABLE 2
Errors in solution for Grid 1.

l	k	n_r	n_s	$e(k, l)$	$e(k, l)/e(k, 1)$
1	1	93	77	1.5×10^{-3}	1
	2	81	21	1.1×10^{-3}	1
	3	81	21	1.1×10^{-3}	1
	4	93	17	1.7×10^{-3}	1
2	1	47	39	7.4×10^{-3}	5
	2	41	11	5.4×10^{-3}	4.8
	3	41	11	5.4×10^{-3}	4.8
	4	47	9	7.4×10^{-3}	4.4
3	1	24	20	3.7×10^{-2}	25
	2	21	6	2.6×10^{-2}	23
	3	21	6	2.6×10^{-2}	23
	4	24	5	3.3×10^{-2}	20

TABLE 3
Errors in solution for Grid 2.

l	k	n_r	n_s	$e(k,l)$	$e(k,l)/e(k,1)$
1	1	101	61	1.8×10^{-2}	1
	2	121	13	2.0×10^{-2}	1
	3	111	13	5.7×10^{-2}	1
2	1	51	31	9.8×10^{-2}	5.4
	2	61	7	8.7×10^{-2}	4.4
	3	56	7	2.7×10^{-2}	4.7

between the calculated and true solution. The number of points on the component grids in the r and s directions is given by n_r and n_s, respectively. Not all the points are used since there is overlap between the grids. Level 1 of Grid 1 has 8431 computational points while there are 7431 computational points on the finest level of Grid 2. Contour plots of the calculated solutions are shown in Figs. 1 and 2.

4. Summary. We have given a brief description of a computer code, CGMG, which solves elliptic PDEs on general composite overlapping meshes using the multigrid algorithm. Multigrid meshes can be created using the composite grid construction program CMPGRD. Care has been taken in the grid construction to keep the amount of overlap to a minimum. The multigrid algorithm for composite meshes involves only minor modifications to its standard form. These changes are necessary in order to handle the interpolation points of the overlapping grids. We have shown the results of solving Poisson's equation with Dirichlet boundary conditions on two nontrivial composite meshes. The numerical solutions were shown to be accurate. Convergence rates were obtained which compare favourably with those rates which might be obtained on a more simple grid.

Acknowledgments. Computations were performed on the Fluid Dynamics VAX 11/750 at the California Institute of Technology. The authors would like to thank Professor Heinz-Otto Kreiss for his encouragement and support.

REFERENCES

[1] E. H. ATTA AND J. VADYAK, *A grid overlapping scheme for flowfield computations about multicomponent configurations*, AIAA J., 21 (1983), pp. 1271–1277.

[2] J. A. BENEK, J. L. STEGER AND F. C. DOUGHERTY, *A flexible grid embedding technique with application to the Euler equations*, AIAA Paper No. 83-1944, Danvers, MA, 1983.

[3] M. BERGER, *On conservation at grid interfaces*, ICASE report no. 84-43, NASA Langley Research Center, Hampton, VA, 1984.

[4] A. BRANDT, *Guide to multigrid development*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1982, pp. 220–312.

[5] G. S. CHESSHIRE, *Composite grid construction and applications*, Ph.D. thesis, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1986.

[6] S. C. EISENSTAT, M. C. GURSKY, M. H. SCHULTZ AND A. H. SHERMAN, *Yale sparse matrix package*, Research Rpts. 112, 114, Yale University, Dept. Comput. Sci., New Haven, CT, May 1977.

[7] W. D. HENSHAW, *Part* I: *The numerical solution of hyperbolic systems of conservation laws*; *Part* II: *Composite overlapping grid techniques*, Ph.D. thesis, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1985.

[8] B. KREISS, *Construction of a curvilinear grid*, this Journal, 4 (1983), pp. 270–279.

[9] L. G. M. REYNA, I. *Stability of Tchebyshev collocation*; II. *Interpolation for surfaces with* 1-D *discontinuities*; III. *On composite meshes*, Ph.D. thesis, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1982.

[10] G. STARIUS, *Composite mesh difference methods for elliptic boundary value problems*, Numer. Math., 28 (1977), pp. 243-258.

[11] ———, *On composite mesh difference methods for hyperbolic difference equations*, Numer. Math., 35 (1980), pp. 241-255.

[12] K. STÜBEN AND U. TROTTENBERG, *Multigrid methods: fundamental algorithms, model problem analysis and applications*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1982, pp. 1-176.