Note

Application of the Multigrid Method to Poisson's Equation in Boundary-Fitted Coordinates

1. Introduction

In the past few years there has been considerable activity in the development of multigrid methods for the numerical solution of partial differential equations governing fluid flow problems. Significant improvement in the convergence rate of iterative finite difference schemes can be achieved by employing a multigrid approach to the following: elliptic equations such as the Poisson and Helmholtz equations [1], the Cauchy–Riemann equations [2], transonic full nonlinear potential flow equations [3], and incompressible Navier–Stokes equations as for the driven square cavity flow [4–6]. Successful applications of the multigrid method to finite element and boundary integral methods can be found in [7].

The multigrid technique is a general strategy for solving partial differential equations by cycling between coarser and finer levels of discretization. The method is based on elimination of error components whose wavelengths on a given grid are comparable to the point spacing in the grid. By cycling between coarse and fine grids, both high- and low-frequency components of the error are reduced efficiently. The multigrid method is highly efficient since most of the computation occurs on coarser grids rather than the finest grid upon which the solution is sought.

The present note reports results showing an order of magnitude reduction in computer time when applying the multigrid method to the solution of the Poisson equation in boundary-fitted coordinates. To the author's knowledge, little or no work on the application of the multigrid method to equations in boundary-fitted coordinates has been reported previously.

2. Poisson Problem Definition

For the solution of fluid flow problems, the use of numerical transformations that map boundary-fitted coordinates (the body being a coordinate line) in physical space onto a Cartesian coordinate system of a computational space (see [8]) has become quite common place (see [9]). The equations often used to generate such a mapping are

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta), \qquad \eta_{xx} + \eta_{yy} = Q(\xi, \eta), \tag{1}$$

0021-9991/83 $3.00

where P and Q are specified functions of ξ and η. In the computational (ξ, η) space these equations become

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0,$$
$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0. \tag{2}$$

Poisson's equation

$$\psi_{xx} + \psi_{yy} = F(x, y) \tag{3}$$

is represented on the computational space as

$$\alpha\psi_{\xi\xi} - 2\beta\psi_{\xi\eta} + \gamma\psi_{\eta\eta} + J^2(P\psi_\xi + Q\psi_\eta) = F \tag{4}$$

with

$$\alpha = x_\eta^2 + y_\eta^2, \qquad \beta = x_\xi x_\eta + y_\xi y_\eta,$$
$$\gamma = x_\xi^2 + y_\xi^2, \qquad J = x_\xi y_\eta - x_\eta y_\xi. \tag{5}$$

In this note the particular mapping of the physical space of Fig. 1a onto the computational space of Fig. 1b will be considered. The physical space coordinate system is the result of solving Eq. (2) subject to Dirichlet data (given (x, y) pairs) at the boundaries of the (ξ, η) computational space which fix the physical location of the boundary grid points. A Cartesian mesh (not shown) overlays the computational space. The wedge-shaped body maps onto the slit in computational space. The functions P and Q are chosen to be

$$P(\xi) = -x_{\xi\xi}/(x_\xi)^3, \qquad Q(\eta) = -y_{\eta\eta}/(y_\eta)^3, \tag{6}$$

with the derivatives evaluated only at the outer boundaries of the computational space. This selection produces nearly orthogonal grids at the outer boundaries of the physical space [10].

The physical coordinate system of Fig. 1a will be used in the numerical simulation of a jet flow from a channel on the left impinging on the wedge-shaped body. This flow problem will not be discussed here. Instead, two test problems for the multigrid method will be considered using the jet flow geometry. Specifically, Eq. (4) will be solved on the computational space of Fig. 1b but excluding the channel (see Fig. 2) and subject to two sets of boundary conditions.

Problem 1:

$$\nabla^2\psi = 0,$$
$$g_1 = g_2 = g_3 = g_4 = 0; \qquad g_5 = 1. \tag{7}$$

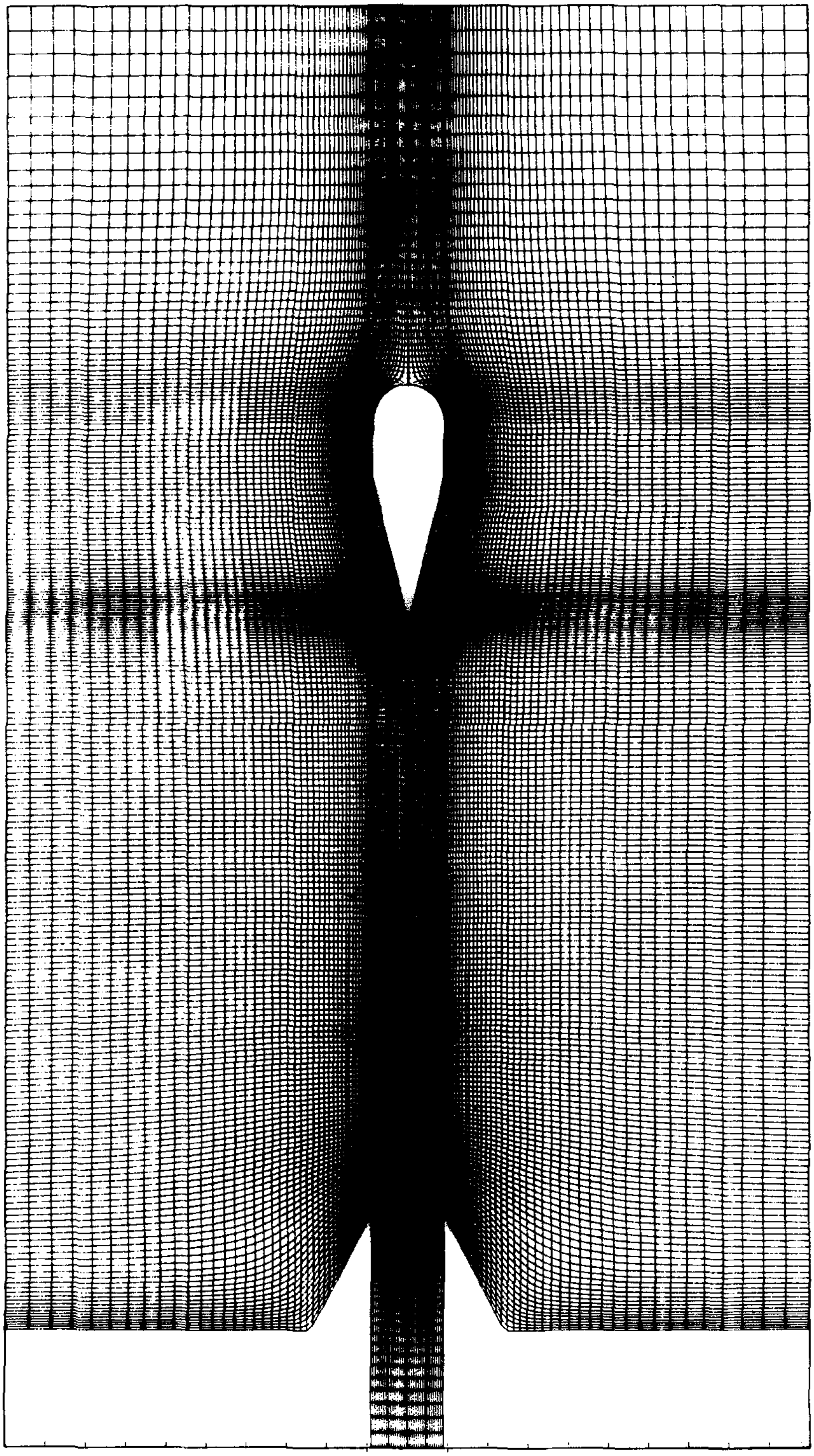

FIG. 1a. Physical coordinate system of mapping (Computer generated and drawn).

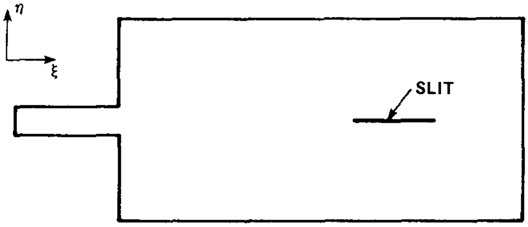

FIG. 1b. Computational space of mapping (Cartesian coordinate system with uniform spacing $\Delta\xi = \Delta\eta = 1$ not shown).

Problem 2:

$$\nabla^2\psi = 0,$$

$$g_1 = -\tfrac{1}{2};\qquad g_3 = \tfrac{1}{2};\qquad g_4 = 0;\qquad g_5 = 0;$$

$$g_2 = -6y(\tfrac{1}{4} - y^2/3) \tag{8}$$

Discretization of Problems 1 and 2 in this paper is accomplished by replacing all derivatives by central second-order differences (i.e., in Eq. (4) $\psi_\varepsilon = (\psi(\xi + \Delta\xi, \eta) - \psi(\xi - \Delta\xi, \eta))/(2\Delta\xi) + O((\Delta\xi)^2))$. Uniform mesh spacing is used with $\Delta\xi = \Delta\eta = 1$.

These two problems are motivated by the desire to solve the Navier–Stokes equations in stream function-vorticity form, where the stream function is represented by ψ and the vorticity by F in Eq. (4). Two test problems arise because the stream function is an unknown constant on the body in the jet flow problem. Solutions to Problems 1 and 2 must be superposed such that the pressure remains single-valued in the doubly connected region, thus determining the unknown value of the stream function constant [11]. In Problem 2, g_2 represents Couette flow in the channel. For the purposes of this paper the vorticity is set to zero to yield potential flow. The solutions of Problems 1 and 2 are initial conditions for a time-dependent viscous jet flow calculation.

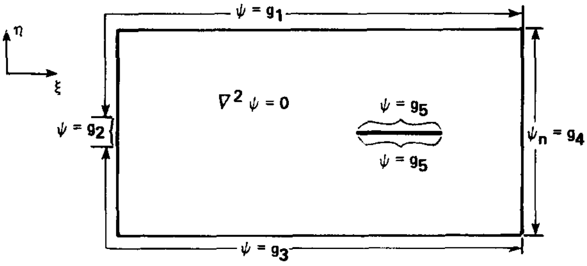

FIG. 2. Computational space for Problems 1 and 2.

3. Use of the Multigrid Method

The Cycle C algorithm of [1] is used to solve Problems 1 and 2. This algorithm starts on the finest grid with several relaxation sweeps (usually 2 or 3) for the system of simultaneous linear equations that represents the finite difference approximations to Problems 1 or 2. At first, convergence is rapid, with short wave length components of the error essentially being eliminated. When the convergence rate slows, the longer wave length components of the error remain. The equation for the error at each grid point (the residual equation) is then transferred to a coarser subgrid which has twice the spacing of the fine grid. On this coarser subgrid the longer wavelength error components are shorter in terms of grid interval representation. A few relaxations essentially eliminate these shorter components on this coarser subgrid. Upon convergence, the error (called the correction) is added to the approximate fine grid solution at each point and a few more relaxation sweeps on the fine grid commence, starting with this improved approximate solution. If convergence occurs to within a specified tolerance, the process is complete. If it does not, an updated error equation at each point is transferred to the coarser subgrid. If the solution to the error equation at each point on the coarser subgrid has not converged after a few relaxation sweeps, the error equation of the error equation at each grid point is transferred to a still coarser subgrid, etc. Every error (residual) equation at each point has its own error (residual) equation on a still coarser subgrid. Each residual equation is treated in the same fashion as the original problem on the finest grid. Thus a multiple sequence of grids can be used for a rapid solution procedure.

This algorithm contains three key parts: (1) the type of relaxation sweeps, (2) transfer of the error equation at each grid point to a coarser subgrid, and (3) interpolation of the coarse subgrid error (correction) for addition to an approximate solution at each grid point of the next finer grid.

In this note linear interpolation is used which according to Brandt [1] is sufficient for linear second-order equations.

The transfer of the error (residual) equation at each grid point to a coarser subgrid is accomplished in the following way: At each point the residual equation for the error (which is the difference between an approximate and exact discrete solution) has the same differential operator (left-hand side) as Eq. (4) and thus is finite differenced at each point on the coarser subgrid in the usual manner. The right-hand side is the residual (the amount by which an approximate discrete solution fails to exactly satisfy the discrete equation at a grid point). A local weighted average is taken of these residuals at fine grid points about a fine grid point coincident with the coarser subgrid point. This weighted average is then taken to be the residual at this coarser subgrid point. These weighted averages can be viewed as a smoothing process in which residuals are made further representable on a coarser subgrid. Weighting of residuals is essential for linear equations with rapidly varying coefficients (which is the case here due to the stretching of the physical coordinate system (Figure 1a) and for nonlinear equations [1]. The stencil used here for the weighted average at a fine grid point is shown in Figure 3a.

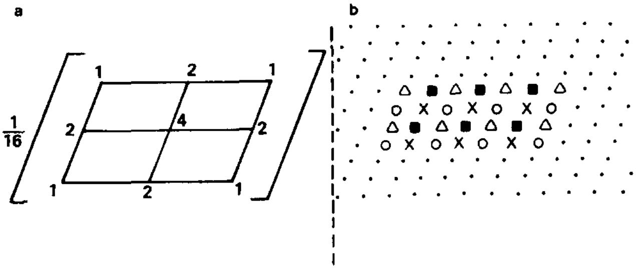

FIG. 3. (a) Stencil for weighting of residuals; (b) schematic drawing for "four-color" relaxation scheme.

A very successful relaxation scheme used in this note was a "four-color" scheme in which points of like symbol (Figure 3b) were relaxed simultaneously. These four groupings were relaxed in succession. This scheme performed far better than "simultaneous point relaxation" (Jacobi point iteration) as a "smoother" in eliminating high-frequency error components. The "four-color" scheme and the analogous "red-black" (two color) relaxation scheme along with Jacobi point iteration are vectorizable on most parallel processor computers. This vectorization results in an additional order of magnitude savings in computer time. Successive point relaxation is not vectorizable and therefore was not considered. Although the "four-color" scheme is vectorizable on the Texas Instruments Advanced Scientific Computer TI–ASC on which the computations were performed, this vectorization was not done due to additional computer storage requirements. The "red-black" scheme is not vectorizable on the TI–ASC.

Three parameters that govern the algorithm are now defined. The numerical solution to Problem 1 or 2 on the finest grid G^M had to satisfy the following specified convergence criterion:

$$\|\mathbf{E}\|_{L_2}^M \leqslant \varepsilon^M.$$

For all cases considered the fixed parameter ε^M was input as 0.001, which is the estimated order of the truncation error. Here $\|\mathbf{E}\|_{L_2}^M$ is the L_2 error norm of $\mathbf{E}$ (the vector of dynamic (calculated while sweeping) residuals). That is, $\|\mathbf{E}\|_{L_2}^M$ is the length of $\mathbf{E}$ on the finest grid G^M. (Superscripts are not exponents.) A fixed parameter λ $(0 < \lambda < 1)$ was specified for all grids such that, if $(\|\mathbf{E}\|_{L_2}^k)^{i+1}/(\|\mathbf{E}\|_{L_2}^k)^i < \lambda$, convergence was considered rapid enough to continue relaxation sweeps. (Here k refers to a kth grid and i is the number of the iteration sweep). When transferring to a coarser subgrid G^{k-1}, the convergence criterion to be satisfied on the coarser subgrid G^{k-1} became $\|\mathbf{E}\|_{L_2}^{k-1} \leqslant \delta(\|\mathbf{E}\|_{L_2}^k)^{i_l}$, where the fixed specified parameter δ satisfies $0 < \delta < 1$ and i_l is the last iteration sweep on grid k before transfer.

The Cycle C algorithm can be used efficiently when a reasonable starting guess for the solution on the finest grid is available. Since such a guess exists for the test problems, namely zero everywhere except for specified boundary values, Cycle C was used.

4. Results

Results are presented in Tables I–III. The grid size for the computational space (Fig. 2) was 113 × 217, with 113 points in the η-direction and 217 points in ξ. The multigrid method used four grids (including the finest). Here WU refers to a work unit, which is equivalent to the work done during a relaxation sweep on the finest grid. (Overhead, such as coarse grid transfers of residual equations and interpolations, is not included in the number of work units). The number of arithmetic operations in a work unit is $51n$, where n is the number of grid points (here $n = 113 \times 217$). At the expense of storing computer arays for the coefficients α, β, γ, J (which can be preprocessed) of Eq. (4) in computer memory, the arithmetic operation count for a work unit can be reduced to $31n$, but this was not done here. By CPU we refer to total central processor computer time in seconds; RF is the relaxation factor. The value of $\|E\|_{L_2}^M$ has been rounded off to four significant digits in the tables.

The results in Tables I and II show an order of magnitude reduction in computer time when the multigrid method uses the "four-color" scheme and a somewhat smaller reduction when using point Jacobi iteration compared to the continuous use of the corresponding relaxation schemes on the finest grid only. The results are even more favorable for the multigrid method than indicated by the CPU times since a maximum of 700 WU's were permitted. Because of this maximum, two of the three nonmultigrid cases were terminated before convergence.

Table III shows that the pair (λ, δ) was not "robust" in the sense that a change in choice of (λ, δ) caused a significant change in convergence rate. The choice of RF also affected the convergence rate. Similar results concerning the dependence of the convergence rate on (λ, δ) and RF were found for Problem 1 and point Jacobi iteration. The choices of parameters $(\lambda, \delta) = (0.3, 0.15)$ in Tables I and II were the best of several tried in the unit square (RF = 1.6 is approximately optimal).

TABLE I

Problem 1

	Using multigrid method	Using relaxations only
Relaxation scheme	"Four-color"	"Four-color"
$\|E\|_{L_2}^M$	0.001	0.0022
(λ, δ)	(0.3, 0.15)	—
RF	1.0	1.6
WU	29.23	700.00
CPU	27.64[a]	471.89[a]

[a] Indicates full vectorization requiring additional computer storage was not used. With full vectorization, CPU times would be approximately one-third those quoted.

TABLE II

Problem 2

	Using multigrid method	Using relaxations only	Using multigrid method	Using relaxations only
Relaxation scheme	"four-color"	"four-color"	Point Jacobi	Point Jacobi
$\|\mathbf{E}\|_{L_2}^M$	0.0008	0.001	0.001	0.0014
(λ, δ)	(0.3, 0.15)	—	(0.3, 0.15)	—
RF	1.6	1.6	—	—
WU	47.56	359.00	172.40	700.00
CPU	40.73[a]	244.41[a]	50.95	132.91

[a] See Table I for explanation.

TABLE III

Problem 2–Multigrid Method with "Four-Color" Scheme Only

$\|\mathbf{E}\|_{L_2}^M$	0.0008	0.001	0.001	0.001
(λ, δ)	(0.3, 0.15)	(0.6, 0.3)	(0.3, 0.15)	(0.6, 0.3)
RF	1.6	1.6	1.0	1.0
WU	47.56	259.00	119.0	366.5
CPU	40.73[a]	198.04[a]	94.27[a]	279.46[a]

[a] See Table I for explanation.

Acknowledgment

This work was performed under NAVSEA Mathematical Sciences Program, "Numerical Methods for Naval Vehicles," Program Element 61153N, Task Area SR0140301, Task 15321, DTNSRDC Work Unit 1808–010.

References

1. A. Brandt, *Math. Comput.* **31** (1977), 333.
2. A. Brandt, AIAA Paper 79–1455, *in* "Proceedings of 4th AIM Computational Fluid Dynamics Conference," Williamsburg, Va., July 23–24, 1979.
3. A. Jameson, AIAA Paper 79–1458, *in* "Proceedings of 4th AIM Computational Fluid Dynamics Conference," Williamsburg, Va., July 23–24, 1979.
4. P. Wesseling and P. Sonneveld, *in* "Proceedings, Symp. on Approximation Methods for Navier–Stokes Problems, Paderborn 1979" (R. Rautmann, Ed.), Lecture Notes in Mathematics No. 771, pp. 543–562, Springer-Verlag, Berlin, 1980.
5. U. Ghia et al., "Solution of Incompressible Navier–Stokes Equations by Coupled Strongly-Implicit Multi-Grid Method," Symposium on Multigrid Methods, Oct. 21–22, 1981, NASA Ames Research Center, Moffet Field, Calif. (published separately from [7]).

6. R. K. Agarwal, MDRL 81–35, McDonnell Douglas Research Labs, St. Louis, Mo., 1981.
7. "Multigrid Methods, A Symposium Held at NASA Ames Research Center, Moffett Field, California, Oct. 21–22, 1981," NASA publication 2202.
8. J. F. Thompson, F. C. Thames, and C. W. Mastin, *J. Comput. Phys.* **15** (1974), 299.
9. "Proceedings of the AIAA 5th Computational Fluid Dynamics Conference," Palo Alto, Calif., June 22–23, 1981.
10. T. J. Plant, Technical Report AFFDL–TR–77–116, Air Force Flight Dynamics Laboratory, Wright–Patterson Air Force Base, Ohio, August 1977.
11. D. R. Sood and H. G. Elrod, *AIAA J.* **12** (1974), 636.

Received: May 12, 1982

Samuel Ohring

David W. Taylor Naval Ship Research
and Development Center,
Bethesda, Maryland 20084