REVIEW

Multigrid Methods for Elliptic Problems: A Review

SCOTT R. FULTON, PAUL E. CIESIELSKI AND WAYNE H. SCHUBERT

Department of Atmospheric Science, Colorado State University, Fort Collins, CO 80523

(Manuscript received 14 May 1985, in final form 1 July 1985)

ABSTRACT

Multigrid methods solve a large class of problems very efficiently. They work by approximating a problem on multiple overlapping grids with widely varying mesh sizes and cycling between these approximations, using relaxation to reduce the error on the scale of each grid. Problems solved by multigrid methods include general elliptic partial differential equations, nonlinear and eigenvalue problems, and systems of equations from fluid dynamics. The efficiency is optimal: the computational work is proportional to the number of unknowns.

This paper reviews the basic concepts and techniques of multigrid methods, concentrating on their role as fast solvers for elliptic boundary-value problems. Analysis of simple relaxation schemes for the Poisson problem shows that their slow convergence is due to smooth error components; approximating these components on a coarser grid leads to a simple multigrid Poisson solver. We review the principal elements of multigrid methods for more general problems, including relaxation schemes, grids, grid transfers, and control algorithms, plus techniques for nonlinear problems and boundary conditions. Multigrid applications, current research, and available software are also discussed.

CONTENTS

1. Introduction

Elliptic boundary-value problems arise in many areas of geophysical fluid dynamics. Perhaps the simplest type is the Poisson problem

$$-\nabla^2 u = f, \tag{1.1}$$

which occurs in a variety of contexts. For example, in nondivergent barotropic models ∇^2 is the two dimensional horizontal Laplacian, u the unknown streamfunction, f the known relative vorticity, and (1.1) is solved with Dirichlet boundary conditions. In Boussinesq models of shallow convection ∇^2 is the three dimensional Laplacian, u the unknown nonhydrostatic pressure, and f a function of the known temperature and motion fields, and (1.1) is solved with Neumann boundary conditions (e.g., Ogura and Phillips, 1962). Another simple elliptic problem which often arises is the Helmholtz problem. An example is in the application of semi-implicit time differencing to primitive equation models which use finite differences in space (e.g., Haltiner and Williams, 1980, p. 149). Then u represents the geopotential at the present time level and f is known in terms of the geopotential and wind fields at previous time levels.

Not all the elliptic boundary-value problems which arise in geophysical fluid dynamics have constant coefficients. As a prototype of variable coefficient problems in two dimensions let us consider

$$(au_x)_x + (bu_y)_y = f, \tag{1.2}$$

which can be classified as a problem with weakly or strongly varying coefficients depending on the percentage variation of the known coefficients $a(x, y)$ and $b(x, y)$. In the balanced vortex theory of hurricane development (e.g., Schubert and Hack, 1983) we encounter an equation of the form (1.2) with u representing the streamfunction for the transverse part of the mass circulation; the coefficients involve the po-

tential vorticity and the inertial stability and become strongly varying as the vortex develops. Variable coefficient problems in three dimensions arise in both quasi-geostrophic and semi-geostrophic theory (e.g., Hoskins, 1975; Hoskins and Draghici, 1977). Examples include the omega equations, tendency equations, and the relations between geopotential and potential vorticity.

Occasionally, nonlinear boundary-value problems are also encountered. For example, in the balanced vortex theory of hurricanes there arises a two-dimensional nonlinear problem relating the unknown potential function to the known Ertel potential vorticity (Thorpe, 1985). In more general three-dimensional nonlinear balance models and in certain initialization schemes (e.g., Haltiner and Williams, 1980, pp. 215, 368) one must solve a nonlinear problem which relates the streamfunction for the rotational flow to the known geopotential field.

If the elliptic boundary-value problem needs to be solved many times (e.g., at every time step in evolution problems), an efficient solution procedure is important, especially in three-dimensional problems. The subject of this review paper is the *multigrid method,* which can be regarded as an approach to developing fast solvers for all of the above examples of elliptic boundary-value problems in geophysical fluid dynamics.

Solving an elliptic problem is usually a two-stage procedure. First, one approximates the continuous problem by a discrete analogue, usually based on finite difference approximations over a sufficiently fine grid. Second, one solves the resulting discrete equations, usually by an iterative technique. For the simplest problems, such as the Poisson problem, there exist "fast solvers" which give the solution with optimal efficiency, i.e., computational work proportional to the number of unknowns. However, for more complicated problems iterative schemes tend to be slow, and typically get even slower as the resolution of the grid increases.

In the early 1970s, multigrid methods were introduced as a means of constructing fast solvers for general elliptic problems. While various components of multigrid processes had been known earlier, especially from the work of Fedorenko (1962, 1964), Achi Brandt (1973) was the first to demonstrate the practical efficiency and generality of multigrid methods. The fundamental components were also developed independently by W. Hackbusch and by P. Frederickson at about the same time. Multigrid methods have been analyzed, applied and generalized in many ways since their introduction, and are gradually becoming recognized as a powerful tool in applied mathematics.

Multigrid methods differ from the perhaps more familiar nested grid techniques in that the primary use of multiple grids is not to obtain nonuniform resolution. Instead, the basic multigrid idea is to approximate the same continuous problem on a set of overlapping uniform grids of widely varying mesh sizes, and cycle between these discrete problems to produce the solution on the finest grid with optimal efficiency. Each grid is employed specifically in solving for those components of the solution which require the corresponding resolution; error in these components is reduced efficiently using a simple relaxation scheme. The computational cost of the coarser grids is small since they involve relatively few points.

The primary advantages of multigrid methods are efficiency and generality. Defining a work unit as the number of operations needed simply to express the discrete equations, the typical multigrid efficiency is to solve a problem to the level of truncation error in just a few (4 to 10) work units. This efficiency has been obtained for a wide class of problems, including general elliptic boundary-value problems, singular perturbation and nonelliptic problems, minimization problems, and integral equations. Self-adaptive local mesh refinement can be incorporated in a natural way, and nonlinear problems are solved with the same efficiency as linear problems.

This paper gives a concise review of the basic concepts and techniques of multigrid methods. It is intended both to serve as a simple introduction to these methods and to call attention to their potential for solving problems in geophysical fluid dynamics. The emphasis here is on methods for solving a single elliptic equation such as those discussed above; more detail, plus information on methods for more complicated problems, is available from a number of sources. The first comprehensive treatment of multigrid methods was Brandt (1977a), which remains very useful. The review article of Stüben and Trottenberg (1982) contains detailed analyses of simple model problems, along with some interesting historical notes. Both of the above sources also contain listings of sample multigrid programs. Brandt (1982) is an invaluable practical guide to developing multigrid algorithms for a wide range of problems; an updated version (Brandt, 1984) contains additional introductory material plus detailed information on multigrid methods for systems of equations in fluid dynamics. Comprehensive multigrid bibliographies are given in all of these sources and in Brand (1982).

In the body of this paper we first review several classical methods for solving discretized elliptic boundary-value problems, including Gaussian elimination, point iterative methods, and an ADI method. These are applied to the Poisson problem in section 2 to illustrate the amount of computational work they require. The reason for the slow convergence of relaxation methods is analyzed in section 3, leading naturally to the basic multigrid concept of using corrections computed on coarser grids. A simple two-grid method is described and then generalized to give a workable multigrid method.

Developing a multigrid algorithm for a particular problem involves choosing a relaxation scheme, a set

of grids, grid transfer operators, and a control algorithm. In section 4 we examine each of these principal components in more detail. Also discussed are the Full Approximation Scheme for nonlinear problems and advanced techniques, basic principles governing the treatment of boundary conditions, and some generalizations to finite-element and spectral discretizations and other types of problems. Section 5 surveys recent applications of multigrid methods, areas of current research, and available software. Concluding remarks are given in section 6.

2. Classical methods

As a prototype elliptic problem, consider the Poisson problem

$$Lu \equiv -\nabla^2 u = f \quad \text{in } \Omega \qquad u = g \quad \text{on } \partial\Omega \tag{2.1}$$

where $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$ is the Laplacian operator, $\Omega = [0, 1] \times [0, 1]$ is the unit square with boundary $\partial\Omega$, and f and g are regarded as known. Equation (2.1) may be approximated on the grid

$$\Omega_h = \{(x_j, y_k) = (jh, kh): 0 \leq j, k \leq N, h = 1/N\} \tag{2.2}$$

using second-order centered finite differences as

$$h^{-2}[4u_{jk} - u_{j-1,k} - u_{j+1,k} - u_{j,k-1} - u_{j,k+1}] = f_{jk} \quad (0 < j, k < N), \tag{2.3a}$$

$$u_{jk} = g_{jk} \quad (j = 0, j = N, k = 0, \text{ or } k = N). \tag{2.3b}$$

Here u_{jk} is the discrete approximation to the true solution $u(x, y)$ at the point (x_j, y_k), $f_{jk} = f(x_j, y_k)$ and $g_{jk} = g(x_j, y_k)$. Let us denote the discrete problem (2.3a) by the shorthand notation

$$L^h u^h = f^h, \tag{2.4}$$

where L^h is a linear operator (the discrete approximation to $L = -\nabla^2$) and u^h and f^h are "grid functions" consisting of the values u_{jk} and f_{jk} on the grid Ω_h. Next, we consider several classical methods for solving this problem.

a. Gaussian elimination

Eliminating the boundary values using (2.3b), the discrete problem (2.4) may be expressed in matrix form as

$$\mathbf{Au} = \mathbf{f}. \tag{2.5}$$

Here $\mathbf{u}$ and $\mathbf{f}$ represent the grid functions u^h and f^h with values taken in lexicographic order, e.g., $\mathbf{u} = (u_{11}, u_{21}, \ldots, u_{N-1,1}, u_{12}, u_{22}, \ldots, u_{N-1,2}, \ldots, u_{N-1,N-1})^T$ with T denoting the transpose. The matrix $\mathbf{A}$ has dimension $(N-1)^2$ by $(N-1)^2$ with the structure shown in Fig. 1, which can be viewed as either banded or block-tridiagonal. It is nonsingular and hence the discrete problem (2.4) is uniquely solvable. Standard

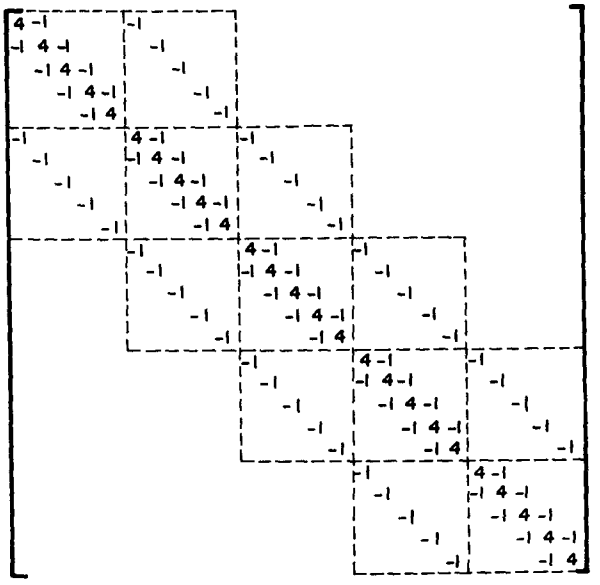

FIG. 1. Structure of the matrix A in (2.5), shown for $N = 6$ after multiplying by h^2.

Gaussian elimination can be used to solve (2.5); this is a direct method in that (with exact arithmetic) it yields the exact (discrete) solution in a finite number of operations. Treating $\mathbf{A}$ as a band matrix of bandwidth $2N-1$, this method requires $O(N^4)$ operations [$O(N^4)$ for the decomposition and $O(N^3)$ for the forward and backward substitution] and $O(N^3)$ storage. However, the true sparsity of $\mathbf{A}$ (the fact that $\mathbf{A}$ has only five nonzero bands) is not utilized. Gaussian elimination based on the block-tridiagonal point of view, e.g., the algorithm of Lindzen and Kuo (1969), requires the same number of operations and storage, i.e., $O(N^4)$ and $O(N^3)$, respectively.

b. Point iterative methods

In contrast to direct methods, indirect or iterative methods do not generate the true (discrete) solution itself, but rather a sequence of approximations which converge to it. In this paper we will consistently use a tilde to denote such approximations. In a point iterative method the values $\tilde{u}_{jk}$ of an approximation $\tilde{u}^h$ are modified point by point to obtain a new approximation $\tilde{u}^h_{new}$. For example, the Jacobi method (also known as simultaneous relaxation) defines the new value $\tilde{u}^{new}_{jk}$ at a point (x_j, y_k) by requiring that the discrete equation (2.3a) hold at that point using the old values at the surrounding points, so that

$$h^{-2}[4\tilde{u}^{new}_{jk} - \tilde{u}_{j-1,k} - \tilde{u}_{j+1,k} - \tilde{u}_{j,k-1} - \tilde{u}_{j,k+1}] = f_{jk}. \tag{2.6}$$

A Jacobi "sweep" consists of computing $\tilde{u}^{new}_{jk}$ at all (interior) points via (2.6) and then replacing the old approximation $\tilde{u}^h$ by $\tilde{u}^h_{new}$. This method requires $O(N^2)$ operations per sweep and $O(N^2)$ storage.

The Gauss-Seidel method (also known as successive relaxation) is similar to the Jacobi method except that each value is updated immediately after it is calculated. In contrast to the Jacobi method, the order in which the points are taken is significant. Using lexicographic ordering [i.e., taking the points (x_j, y_k) in the order $(j, k) = (1, 1), (2, 1), \ldots, (N-1, 1), (2, 1), \ldots, (N-1, N-1)$] the Gauss-Seidel relaxation equation is

$$h^{-2}[4\tilde{u}_{jk}^{\text{new}} - \tilde{u}_{j-1,k}^{\text{new}} - \tilde{u}_{j+1,k} - \tilde{u}_{j,k-1}^{\text{new}} - \tilde{u}_{j,k+1}] = f_{jk}. \quad (2.7)$$

This equation gives $\tilde{u}_{jk}^{\text{new}}$ explicitly since $\tilde{u}_{j-1,k}^{\text{new}}$ and $\tilde{u}_{j,k-1}^{\text{new}}$ have been computed before reaching the point (x_j, y_k). As before, $O(N^2)$ operations per sweep and $O(N^2)$ storage are required; however, the storage is actually less than for the Jacobi method since the new values may be written over the old values immediately.

The Gauss-Seidel method can be viewed as adding a correction $\tilde{u}_{jk}^{\text{new}} - \tilde{u}_{jk}$ to each value $\tilde{u}_{jk}$ successively. The rate of convergence may be accelerated by scaling this correction by a relaxation parameter ω before adding it, leading to the SOR (successive over-relaxation) method defined by

$$\tilde{u}_{jk}^{\text{new}} = (1-\omega)\tilde{u}_{jk} + \frac{\omega}{4}[h^2 f_{jk} + \tilde{u}_{j-1,k}^{\text{new}} + \tilde{u}_{j+1,k} + \tilde{u}_{j,k-1}^{\text{new}} + \tilde{u}_{j,k+1}]. \quad (2.8)$$

Again, $O(N^2)$ operations per sweep and $O(N^2)$ storage are required, with the SOR method reducing to the Gauss-Seidel method if $\omega = 1$.

The convergence of the above methods can be described by writing them as fixed-point iterations so that the errors $v^h = u^h - \tilde{u}^h$ satisfy

$$v_{\text{new}}^h = T^h v^h, \quad (2.9)$$

where T^h is an iteration operator. From (2.9) the iteration converges (for any initial approximation) if and only if the spectral radius $\rho(T^h)$ is less than one, with $\rho(T^h)$ providing an estimate of the factor by which the error will be reduced by one relaxation sweep. For the discrete Poisson problem considered above it can be shown (e.g., Stoer and Bulirsch, 1980) that the optimal rate for the SOR method is obtained with the relaxation parameter $\omega = 2/[1 + \sin(\pi h)]$, and that

$$\rho(T^h) = \begin{cases} \cos(\pi h) = 1 - \frac{1}{2}\pi^2 h^2 + O(h^4), \\ \qquad\qquad \text{Jacobi} \\ \cos^2(\pi h) = 1 - \pi^2 h^2 + O(h^4), \\ \qquad\qquad \text{Gauss-Seidel} \\ \dfrac{1-\sin(\pi h)}{1+\sin(\pi h)} = 1 - 2\pi h + O(h^2), \\ \qquad\qquad \text{SOR (optimal).} \end{cases} \quad (2.10)$$

Thus the rate of convergence of each of these methods depends on the resolution, with higher resolution (smaller h) resulting in slower convergence. Since each sweep essentially reduces the error by the factor $\rho(T^h)$, reducing the error by a specified factor ϵ requires s sweeps, where $[\rho(T^h)]^s = \epsilon$. Solving for s and using (2.10) gives

$$s = (-\ln\epsilon)x \begin{cases} [\tfrac{1}{2}\pi^2 h^2 + O(h^4)]^{-1} = O(N^2), \\ \qquad\qquad \text{Jacobi} \\ [\pi^2 h^2 + O(h^4)]^{-1} = O(N^2), \\ \qquad\qquad \text{Gauss-Seidel} \\ [2\pi h + O(h^2)]^{-1} = O(N), \\ \qquad\qquad \text{SOR (optimal).} \end{cases} \quad (2.11)$$

Since each method requires $O(N^2)$ operations per sweep, the total work required to reduce the error by a specified factor ϵ is $O(N^4)$ operations for the Jacobi and Gauss-Seidel methods and $O(N^3)$ operations for the optimal SOR method.

c. ADI method

In contrast to the point iterative methods described above, block iterative methods produce a new approximation $\tilde{u}_{\text{new}}^h$ from a previous one $\tilde{u}^h$ by simultaneously modifying the values of a "block" or set of points. As an example, consider the following ADI (alternating direction implicit) method, in which the "blocks" are lines of points in the x and y directions. Each ADI sweep consists of two steps, each based on the discrete Eq. (2.3a) in the form

$$(-u_{j-1,k} + 2u_{jk} - u_{j+1,k}) + (-u_{j,k-1} + 2u_{jk} - u_{j,k+1}) = h^2 f_{jk}. \quad (2.12)$$

The first step treats the x-dependence implicitly to produce an intermediate approximation $\tilde{u}_{\text{int}}^h$ via

$$-\tilde{u}_{j-1,k}^{\text{int}} + (2+r)\tilde{u}_{jk}^{\text{int}} - \tilde{u}_{j+1,k}^{\text{int}} = h^2 f_{jk} - [-\tilde{u}_{j,k-1} + (2-r)\tilde{u}_{jk} - \tilde{u}_{j,k+1}], \quad (2.13a)$$

where r is an iteration parameter, and the second step treats the y-dependence implicitly to obtain the new approximation $\tilde{u}_{\text{new}}^h$ via

$$-\tilde{u}_{j,k-1}^{\text{new}} + (2+r)\tilde{u}_{jk}^{\text{new}} - \tilde{u}_{j,k+1}^{\text{new}} = h^2 f_{jk} - [\tilde{u}_{j-1,k}^{\text{int}} + (2-r)\tilde{u}_{jk}^{\text{int}} - \tilde{u}_{j+1,k}^{\text{int}}]. \quad (2.13b)$$

Each step involves solving N tridiagonal linear systems of order N; thus the ADI method requires $O(N^2)$ operations per sweep and $O(N^2)$ storage.

For the Poisson problem the above ADI method can be shown to converge if a fixed positive iteration parameter r is used for all sweeps. The optimal constant r is $2\sin(\pi h)$, for which the method converges at the

same rate as the optimal SOR method. Faster convergence can be obtained by allowing r to vary from sweep to sweep. A simple algorithm for determining the optimal set of n parameters, with n a power of 2, is given in Stoer and Bulirsch (1980). One cycle of n ADI sweeps using these parameters reduces the error by a factor of approximately $1 - 8(\frac{1}{4}\pi h)^{1/n}$, so that

$$s \approx -\frac{n}{8}(\tfrac{1}{4}\pi h)^{-1/n} \ln \epsilon \text{ ADI}$$

sweeps are required to reduce the error by a specified factor ϵ. This number is minimized by choosing $n \approx \ln(4/\pi h)$, yielding $s = O(\ln N)$. Since $O(N^2)$ operations are required per sweep, the ADI method requires $O(N^2 \ln N)$ operations to reduce the error by a specified factor.

d. Discussion

With low resolution (small N), direct solution of discretized elliptic problems by Gaussian elimination may be practical. With higher resolution (larger N), iterative methods become more attractive, in part because they require less storage [$O(N^2)$ vs $O(N^4)$] and in part because they are less sensitive to round-off error (in Gaussian elimination round-off error tends to accumulate—albeit slowly—while in iterative methods it is automatically reduced by the iteration). The simplest iterative schemes (e.g., Jacobi and Gauss-Seidel relaxation) require $O(N^4)$ operations, as does Gaussian elimination. The SOR method reduces this operation count to $O(N^3)$ by introducing a relaxation parameter; the ADI method lowers it to $O(N^2 \ln N)$ by using several iteration parameters and a more complicated iteration scheme. For more general elliptic problems with variable coefficients, suitable values for these parameters must be determined by either eigenvalue analysis or numerical experimentation.

A number of algorithms for the efficient direct solution of the discretized Poisson problem have been introduced (e.g., Buzbee et al., 1970; Sweet, 1977; Swarztrauber, 1977). These "fast Poisson solvers" are based on cyclic reduction, matrix decomposition, or a combination of the two; many make use of the Fast Fourier Transform (FFT) algorithm. While highly efficient for the Poisson problem, these techniques usually cannot be extended to more complicated problems such as (1.2). As we shall see, multigrid methods solve the Poisson problem as efficiently as these fast Poisson solvers, giving the solution in $O(N^2)$ operations, but also generalize to solve a wide range of more complicated problems with essentially the same efficiency.

3. Basic multigrid concepts

In this section we motivate multigrid methods by analyzing the convergence of Gauss-Seidel relaxation. This analysis suggests a simple two-grid method, which naturally generalizes to an efficient multigrid method.

a. Analysis of relaxation

Consider a single sweep of Gauss-Seidel relaxation, and denote the approximations to the true (discrete) solution u^h of (2.4) before and after the sweep by $\tilde{u}^h$ and $\tilde{u}^h_{\text{new}}$, respectively. Subtracting the relaxation equation (2.7) from the discrete equation (2.3a) shows that the corresponding errors $v^h = u^h - \tilde{u}^h$ and $v^h_{\text{new}} = u^h - \tilde{u}^h_{\text{new}}$ satisfy

$$v^{\text{new}}_{jk} = \frac{1}{4}(v^{\text{new}}_{j-1,k} + v_{j+1,k} + v^{\text{new}}_{j,k-1} + v_{j,k+1}). \tag{3.1}$$

Thus the new error produced at a point (x_j, y_k) by relaxation is simply the average of the current errors at the four surrounding points. This indicates that error components which oscillate on the scale of the grid will be reduced substantially by relaxation, due to cancellation on the right-hand side of (3.1), while smooth error components will be reduced only slightly. Thus relaxation is efficient as a smoother, but inefficient as a solution method.

To quantify this result without the complexity of a rigorous eigenvalue analysis one can use local mode analysis (Brandt, 1977a). The discrete Fourier modes are

$$E_\theta(x_j, y_k) = \exp[i(j\theta_1 + k\theta_2)], \tag{3.2}$$

where $\theta = (\theta_1, \theta_2)$ is the discrete vector wavenumber; on a grid with mesh spacing h the components θ_1 and θ_2 are integral multiples of $2\pi h$ between $-\pi$ and $+\pi$. For periodic problems, expansions in terms of the modes (3.2) are appropriate, but even in nonperiodic problems they provide useful and quantitatively correct information about *local* processes such as relaxation.

Suppose that the error v^h before the relaxation sweep has a component $A_\theta E_\theta$, where A_θ is the amplitude. Then from (3.1) and (3.2) the error $v_{\text{new}}{}^h$ after the sweep has the component $A_\theta{}^{\text{new}} E_\theta$, with A_θ and $A_\theta{}^{\text{new}}$ related by

$$A_\theta{}^{\text{new}} = \frac{1}{4}(e^{-i\theta_1} A_\theta{}^{\text{new}} + e^{i\theta_1} A_\theta + e^{-i\theta_2} A_\theta{}^{\text{new}} + e^{i\theta_2} A_\theta). \tag{3.3}$$

Thus, one relaxation sweep reduces the amplitude of the error component E_θ by the convergence factor

$$\mu(\theta) = \left|\frac{A_\theta{}^{\text{new}}}{A_\theta}\right| = \left|\frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}\right|. \tag{3.4}$$

Figure 2 shows μ as a function of θ_1 and θ_2; as argued above, μ is small for high wavenumbers and large for low wavenumbers. In particular, from (3.4) we have $\mu = \frac{1}{3}$ when $|\theta_1| = |\theta_2| = \pi$, but $\mu = 1 - O(h^2)$ when both θ_1 and θ_2 are $O(h)$. Thus the modes responsible for the slow convergence (2.10) of Gauss-Seidel relaxation are in fact the *lowest* wavenumbers; these modes could be adequately approximated on a coarser grid with much less work. This observation is the key to the multigrid method.

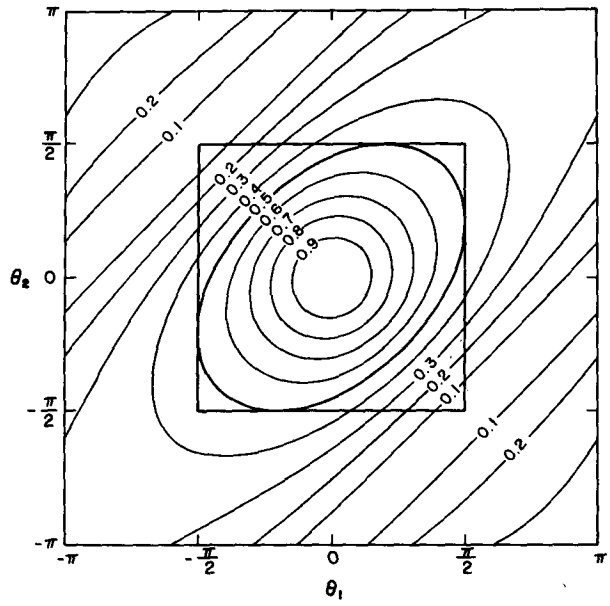

FIG. 2. Convergence factor $\mu(\theta)$ for the Poisson problem (2.3) using Gauss-Seidel relaxation with lexicographic ordering.

b. A two-grid method

The use of a coarser grid to approximate low-wavenumber error components can be illustrated as follows. Suppose that several relaxation sweeps have been carried out for the problem (2.4) on the grid Ω_h, yielding an approximate solution $\tilde{u}^h$. From the discussion in the previous section, the associated error v^h is smooth, i.e., the high-wavenumber modes have been reduced substantially, and further relaxation will reduce the remaining low-wavenumber modes very slowly. Now the error satisfies the residual equation

$$L^h v^h = r^h, \tag{3.5}$$

where

$$r^h = f^h - L^h \tilde{u}^h \tag{3.6}$$

is the residual; since v^h is smooth, (3.5) can be approximated on the coarser grid Ω_H with mesh spacing $H > h$ by

$$L^H v^H = I_h^H r^h. \tag{3.7}$$

Here L^H is the same discrete operator as L^h except for grid spacing H, and v^H is the grid H approximation to v^h. Usually one takes $H = 2h$ as shown in Fig. 3. The right-hand side of (3.7) is obtained by computing the residual r^h on the fine grid Ω_h using (3.6) and transferring it to the coarse grid Ω_H, e.g., by injection (simply copying the values of r^h at points common to both grids); the operator I_h^H represents the fine-to-coarse grid transfer. Assuming that (3.7) can be solved, the result v^H is an approximation to the error v^h and hence can be transferred back to the fine grid Ω_h, e.g., by bilinear interpolation, and added to the previous approximate solution $\tilde{u}^h$ to obtain a new approximation

$$\tilde{u}^h_{\text{new}} = \tilde{u}^h + I_H^h v^H, \tag{3.8}$$

where I_H^h represents the coarse-to-fine grid transfer.

In the simple two-grid cycle described above, relaxation is used on the fine grid to efficiently reduce high-wavenumber error components, and the coarse-grid correction cycle (3.6)–(3.8) is used to eliminate the low-wavenumber errors. Here the "high-wavenumber" modes are the modes *not* representable on the coarse grid Ω_H; these satisfy $\hat{\rho}\pi < |\theta| \leq \pi$, where $\hat{\rho} = h/H$ and $|\theta| = \max(|\theta_1|, |\theta_2|)$, corresponding to the region between the two boxes in Fig. 2 (assuming $\hat{\rho} = 1/2$). We define the *smoothing factor* $\bar{\mu}$ as the maximum convergence factor $\mu(\theta)$ for the high wavenumbers, i.e.,

$$\bar{\mu} = \max_{\hat{\rho}\pi \leq |\theta| \leq \pi} \mu(\theta), \tag{3.9}$$

where $|\theta| = \hat{\rho}\pi$ has been included for convenience. For Gauss-Seidel relaxation with $\hat{\rho} = 1/2$ one finds from (3.4) or Fig. 2 that $\bar{\mu} = 1/2$; thus each relaxation sweep reduces the high-wavenumber error by at least a factor of 2. Assuming that the coarse-grid equation (3.7) is solved exactly and neglecting any errors introduced by the grid transfers, a two-grid cycle consisting of ν relaxation sweeps on the fine grid followed by the coarse-grid correction (3.6)–(3.8) reduces the error in *all* wavenumbers by the factor $\bar{\mu}^\nu$. Since the grid transfers in fact usually introduce a small amount of error, the accuracy cannot be increased without limit by simply increasing ν. However, with small ν the above estimate holds, and

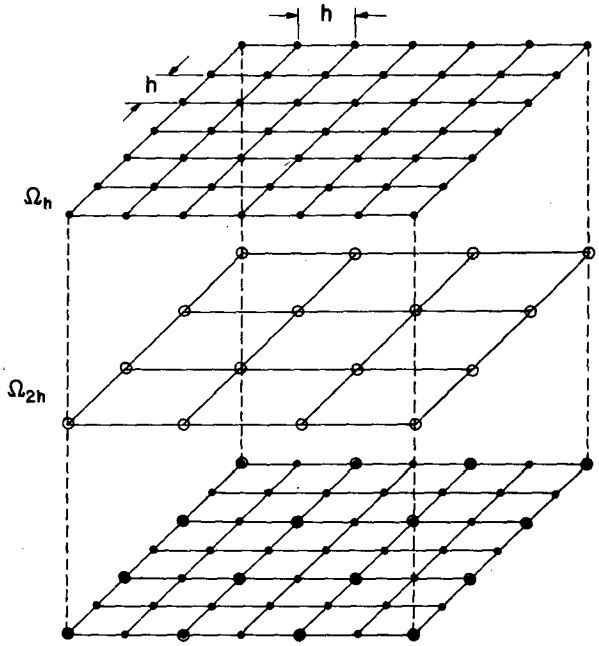

FIG. 3. The fine grid Ω_h and coarse grid $\Omega_H = \Omega_{2h}$ for the two-grid method and their superposition.

the two-grid cycle can be repeated as needed to attain the desired accuracy. With $\nu = 3$ the error is reduced by almost an order of magnitude per cycle, independent of h.

c. A multigrid method

The method used to solve the coarse-grid equation (3.7) was left unspecified in the two-grid cycle described above. However, since this equation has the same form as the original equation (3.1), it can be solved by the same method; i.e., make several relaxation sweeps on the coarse grid Ω_H and then apply a correction computed using a still coarser grid Ω_{2H}. Continuing this process recursively leads directly to a simple multigrid cycle.

More precisely, we define a set of grids $G_l = \Omega_{h_l}$ ($l = 1, \ldots, M$) with mesh sizes h_l satisfying $h_{l-1} = 2h_l$. The finest grid G_M has the desired resolution $h_M = h$, while the coarsest grid G_1 has only a few points in each direction; the number of grids M is typically about five or six. Changing notation slightly, for each grid G_l (often referred to as "level" l) we form a discrete problem

$$L^l u^l = f^l, \tag{3.10}$$

where L^l is the finite difference approximation to the operator $L = -\nabla^2$ for mesh size h_l. For the finest grid $l = M$, (3.10) is simply the original discrete problem (2.4) with $u^M = u^h$ and $f^M = f^h$. However, for each coarser grid (3.10) is an approximation to the *residual* problem on the next finer grid, so that for $1 < l \leq M$, u^{l-1} and f^{l-1} are approximations to the error $v^l = u^l - \tilde{u}^l$ and residual $r^l = f^l - L^l\tilde{u}^l$, respectively (where $\tilde{u}^l$ denotes the current approximation to the exact discrete solution u^l). Thus v^H and $I_h^H r^h$ in the two-grid method have been replaced by u^{l-1} and f^{l-1} for uniformity of notation.

The multigrid cycle, shown schematically in Fig. 4, starts on the finest grid G_M with an approximation $\tilde{u}^M$ to the desired solution u^M. For each level $l = M, M-1, \ldots, 2$ in turn, the corresponding problem (3.10) is relaxed ν_1 times to smooth the error on the scale of grid G_l, and the resulting residual $r^l = f^l - L^l\tilde{u}^l$ is transferred to the next coarser grid G_{l-1} via

$$f^{l-1} = I_l^{l-1} r^l = I_l^{l-1}(f^l - L^l\tilde{u}^l) \tag{3.11}$$

to form the right-hand side of the level $l-1$ problem (3.10). In this procedure the initial approximations $\tilde{u}^l$ on the coarser grids G_l ($1 \leq l < M$) are simply taken to be zero. Upon reaching the coarsest grid G_1 the resulting problem $L^1u^1 = f^1$ is simply solved, either directly (by Gaussian elimination) or by many relaxation sweeps; either approach is efficient since the coarsest grid contains very few points. Then for each level $l = 1, \ldots, M-1$ in turn, the current approximate solution $\tilde{u}^l$, being an approximation to the error $u^{l+1} - \tilde{u}^{l+1}$ on the next finer grid, is interpolated to that grid and added to the previous approximation via

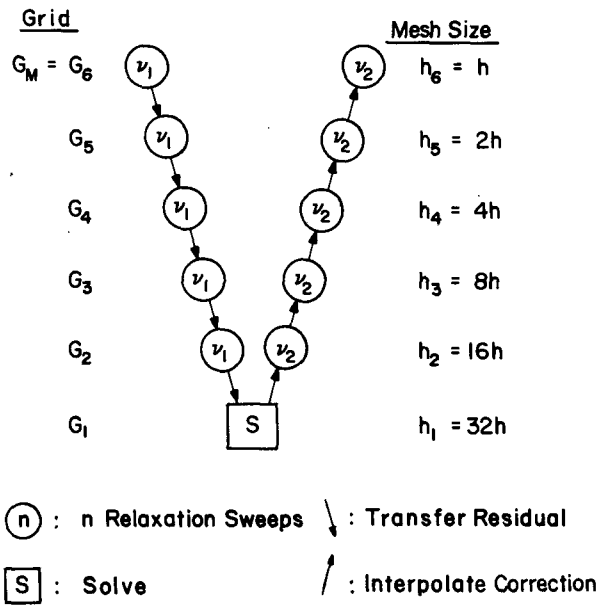

FIG. 4. A multigrid V-cycle for $M = 6$ grids.

$$\tilde{u}^{l+1}_{\text{new}} = \tilde{u}^{l+1} + I_l^{l+1}\tilde{u}^l, \tag{3.12}$$

and the level $l+1$ problem is relaxed ν_2 more times to smooth any remaining errors on the scale of that grid. The overall cycle from the finest grid to the coarsest and back is called a multigrid *V-cycle.*

To estimate the effect of a multigrid V-cycle, note that each Fourier mode θ on the original grid Ω_h is in fact a high-wavenumber mode on *one* of the grids used. Thus the error in each mode is reduced by at least the factor $\bar{\mu}^\nu$ by $\nu = \nu_1 + \nu_2$ relaxation sweeps on one of the grids during one V-cycle. If ν is not large (typically one might choose $\nu_1 = 2$ and $\nu_2 = 1$), the errors introduced by the grid transfers are negligible, so the overall error (i.e., the error in *all* modes) is reduced by at least the factor $\bar{\mu}^\nu$ per V-cycle. This convergence factor is *independent of* h, unlike the factors (2.10) for classical iterative methods which deteriorate as the grid resolution increases. Therefore, reducing the overall error by a specified factor ϵ requires $s = \ln\epsilon/(\nu \ln\bar{\mu})$ V-cycles, independent of h. Furthermore, since each grid contains only one-fourth as many points as the next finer grid, the total number of grid points (and hence storage) involved is only

$$1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{2^{M-1}} < \frac{4}{3}$$

times that required for the original problem (3.1) on the finest grid. Therefore, the work required per V-cycle (ignoring the overhead of residual transfers and interpolation, which is usually small) is less than $4/3$ times the work required for ν relaxation sweeps on the finest grid alone. Thus with $O(N^2)$ points on the finest grid the work per V-cycle is $O(N^2)$ operations; since the number s of V-cycles required is independent of h

(and hence N), the multigrid method reduces the error by a specified factor in $O(N^2)$ operations.

4. Survey of multigrid techniques

In this section we discuss in more detail the various elements of a multigrid method for a problem of the general form

$$Lu = f, \tag{4.1}$$

where L is an elliptic partial differential operator (typically in two or three space dimensions), u is the unknown, and f is specified. These elements include the relaxation scheme, grids, grid transfers, and control algorithm. In most of the discussion we assume for simplicity that L is linear; the Full Approximation Scheme, which can be used to treat nonlinear problems, is also described. Since the boundary conditions do not strongly influence the essential features or efficiency of a multigrid method, their treatment is discussed separately. More detailed information on all of the topics reviewed here can be found in Brandt (1984).

a. Relaxation schemes

The heart of a multigrid method is the relaxation scheme used on each grid to smooth the error, i.e., to reduce the high-wavenumber error components. The relaxation scheme is the most problem-dependent part of a multigrid method, and has the most impact on the overall efficiency. Since smoothing high-wavenumber errors is essentially a *local* process, it may be analyzed using local mode analysis, and constant (frozen) coefficients may be assumed in the analysis for problems with variable coefficients (unless they vary drastically on the scale of the grid). The primary measure of the effectiveness of relaxation is the smoothing factor $\bar{\mu}$ defined by (3.9); small $\bar{\mu}$ is desired for fast smoothing. In addition, a relaxation scheme should be simple (relatively few operations per grid point for efficiency) and robust (relatively insensitive to changes in parameters or coefficients).

The simplest relaxation schemes are the successive (Gauss-Seidel) schemes, in which the new values of unknowns replace the old values immediately after being calculated. These schemes are appropriate for most elliptic problems. For example, ordinary (point) Gauss-Seidel relaxation with lexicographic ordering gives $\bar{\mu} = 0.5$ for the Poisson problem, as shown above. This scheme is simple, involving only four additions and one multiplication per grid point. The smoothing factor $\bar{\mu} = 0.5$ is not improved by introducing a relaxation parameter; in contrast, when relaxation is used as a *solver*, faster convergence is obtained by overrelaxation (i.e. the SOR method).

For anisotropic problems, point Gauss-Seidel relaxation may not be as effective. For example, discretizing the equation (1.2) using centered differences on a grid with mesh spacing h yields

$$h^{-2}[a_{j+1/2,k}(u_{j+1,k} - u_{jk}) - a_{j-1/2,k}(u_{jk} - u_{j-1,k}) + b_{j,k+1/2}(u_{j,k+1} - u_{jk}) - b_{j,k-1/2}(u_{jk} - u_{j,k-1})] = f_{jk}, \tag{4.2}$$

where $a_{j+1/2,k} = a((j + 1/2)h, kh)$ and so forth. Point Gauss-Seidel relaxation for (4.2) with lexicographic ordering, defined by

$$h^{-2}[a_{j-1/2,k}\tilde{u}^{\text{new}}_{j-1,k} + a_{j+1/2,k}\tilde{u}_{j+1,k} + b_{j,k-1/2}\tilde{u}^{\text{new}}_{j,k-1} + b_{j,k+1/2}\tilde{u}_{j,k+1} - (a_{j-1/2,k} + a_{j+1/2,k} + b_{j,k-1/2} + b_{j,k+1/2})\tilde{u}^{\text{new}}_{jk}] = f_{jk}, \tag{4.3}$$

yields the convergence factor

$$\mu(\theta) = \left| \frac{ae^{i\theta_1} + be^{i\theta_2}}{2(a + b) - ae^{-i\theta_1} - be^{-i\theta_2}} \right| \tag{4.4}$$

using local mode analysis with a and b frozen. When $a = b$ (i.e., the problem is isotropic), (4.4) reduces to (3.4) so that $\bar{\mu} = 0.5$ as for the Poisson problem, but when a and b differ, $\bar{\mu}$ can be considerably larger. Figure 5 shows $\mu(\theta)$ for the case $b/a = 10$, for which $\bar{\mu} \doteq 0.83$; as $b/a \to \infty$ (or $a/b \to \infty$), $\bar{\mu} \to 1$.

The difficulty here is that point relaxation smooths the error only in directions of strong local coupling. This can be seen by noting from (4.2) and (4.3) that the errors v_{jk} and v^{new}_{jk} satisfy

$$v^{\text{new}}_{jk} = \frac{a(v^{\text{new}}_{j-1,k} + v_{j+1,k}) + b(v^{\text{new}}_{j,k-1} + v_{j,k+1})}{2(a + b)}. \tag{4.5}$$

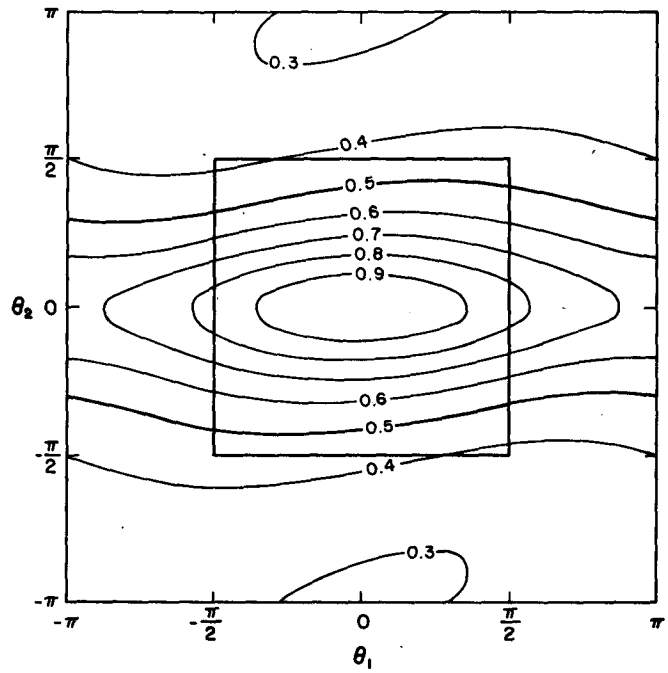

FIG. 5. Convergence factor $\mu(\theta)$ for problem (4.3) using Gauss-Seidel relaxation with lexicographic ordering and $b/a = 10$.

When $b/a \gg 1$ the solution is coupled more strongly in y than in x; cancellation between errors at $(j-1, k)$ and $(j+1, k)$ will have little effect in (4.5) so that high wavenumbers in x will be changed only slightly by relaxation. Since point relaxation is slow to converge in this case, *locally strongly coupled blocks of points should be relaxed simultaneously.* For the case $b/a \gg 1$, y-line Gauss-Seidel relaxation is appropriate; this scheme is defined the same as (4.3) except with $\tilde{u}_{j,k+1}$ replaced by $\tilde{u}^{\text{new}}_{j,k+1}$, so that all points along a y-line (constant x) are relaxed simultaneously. Relaxing each y-line in turn is easy and efficient, since the linear systems involved are tridiagonal and diagonally dominant. The resulting convergence factor

$$\mu(\theta) = \left| \frac{ae^{i\theta_1}}{2(a + b - b\cos\theta_2) - ae^{-i\theta_1}} \right| \qquad (4.6)$$

is shown in Fig. 6a for the case $b/a = 10$. It can be shown from (4.6) that

$$\bar{\mu} = \max\left(5^{-1/2}, \frac{a}{a+2b}\right) \qquad (4.7)$$

so that $\bar{\mu} \approx 0.45$ for *any* $b/a \geq 1$. However, (4.7) shows that $\bar{\mu} \to 1$ as $b/a \to 0$, as can be seen in Fig. 6b which shows $\mu(\theta)$ for the case $b/a = 1/10$. This case requires x-line relaxation by the same reasoning. When b/a is large in some regions and small in others, one can use alternating direction line relaxation, consisting of one sweep of y-line relaxation followed by one sweep of x-line relaxation. Such a double sweep gives $\bar{\mu} \leq 5^{-1/2} \approx 0.45$ for *any* ratio b/a. This method is similar to the ADI solution method, but much simpler since there are no iteration parameters to be chosen.

The Gauss-Seidel schemes discussed above are not directly vectorizable and hence do not exploit the full potential of vector processors such as the CRAY-1 or CYBER 205. Simultaneous (Jacobi) relaxation schemes, in which all of the new values in a single sweep are computed before overwriting any of the old values, would seem preferrable since they do vectorize. However, in addition to using more storage, Jacobi schemes usually require underrelaxation (i.e., using a relaxation parameter $\omega < 1$) and thus more work per grid point, and have poorer smoothing factors (e.g., $\bar{\mu} = 0.6$ with $\omega = 4/5$ for the Poisson problem). A better approach is to use Gauss-Seidel schemes with the grid points or lines taken in a different order. For isotropic problems one can use point Gauss-Seidel relaxation with *red-black* ordering: first relax at all "red" points ($j + k$ even) and then at all "black" points ($j + k$ odd), where the colors refer to the pattern of a checkerboard. Similarly, when line relaxation is required one can use *zebra* relaxation: first relax along all "white" (even) lines and then along all "black" (odd) lines. Each of these schemes is fully vectorizable. Moreover, they have better smoothing factors than the corresponding schemes with standard ordering. (Their mode analysis is more complicated, however, since they couple several modes.) For example, point Gauss-Seidel relaxation gives $\bar{\mu} = 0.25$ with red-black ordering, as opposed to $\bar{\mu} = 0.5$ with lexicographic ordering; this excellent smoothing factor makes the red-black scheme the relaxation method of choice for the Poisson problem. For anisotropic problems with variable coefficients, alternating direction zebra relaxation is robust and efficient.

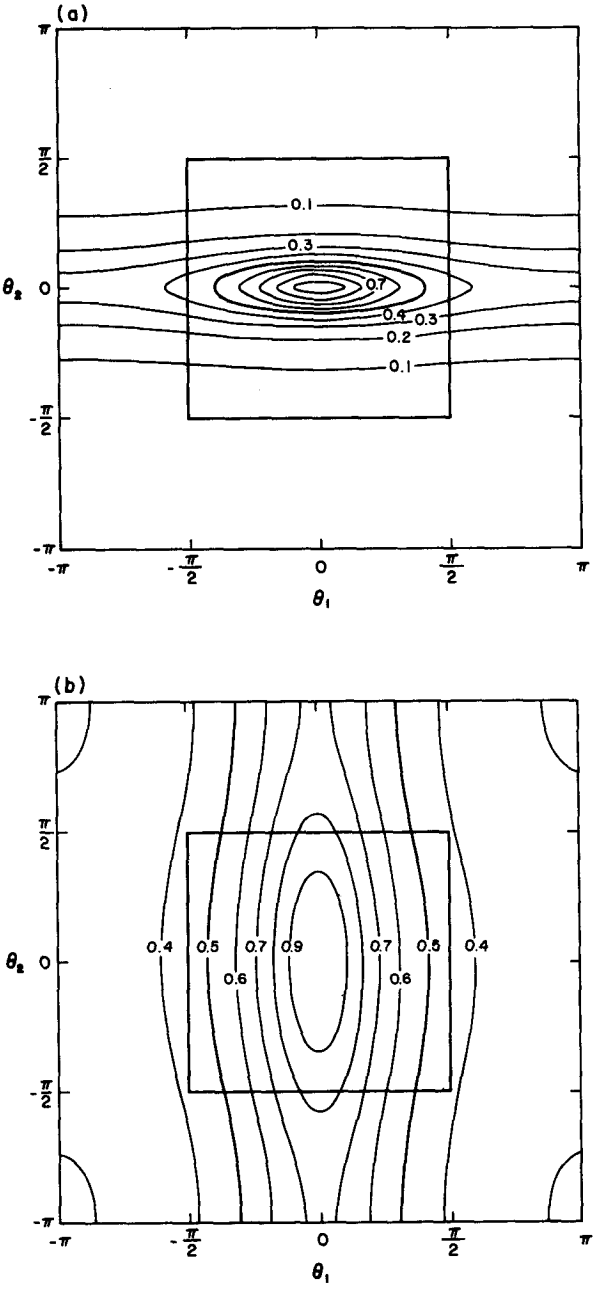

FIG. 6. Convergence factor $\mu(\theta)$ for problem (4.3) using y-line Gauss-Seidel relaxation for (a) $b/a = 10$ and (b) $b/a = 1/10$.

b. Grids

A multigrid method for (4.1) typically uses a grid structure much like that described in section 3c: a set of grids $G_1, G_2, \ldots, G_M$ with mesh sizes $h_1 > h_2 > \cdots > h_M$, each grid covering the whole computational domain. For convenience of notation it is assumed that on each grid the same mesh size is used in each coordinate direction, although this need not be the case. The mesh size ratio h_{l+1}/h_l is usually the same constant $\hat{\rho}$ between all grids; this ratio affects the efficiency in two ways. First, $\hat{\rho}$ must be relatively small so that the coarse grids $G_1, \ldots, G_{M-1}$ involve relatively few points compared to the finest grid G_M, and thus do not contribute substantially to the total work or storage. Second, if $\hat{\rho}$ is too small then relaxation becomes inefficient as a smoother, since more modes must be considered as high wavenumbers in (3.9). For these reasons, the computationally convenient ratio $\hat{\rho} = 1/2$ is usually optimal, although $\hat{\rho} = 1/3$ or $\hat{\rho} = 1/\sqrt{2}$ (with rotatable operators) may be useful in some cases.

The basic grid structure described above may be modified in several ways. First, a given grid G_l may be coarser than the next finer grid G_{l+1} in only some of the coordinate directions; this *semi-coarsening* can provide an alternative to block relaxation. For example, for the three-dimensional problem $Lu = au_{xx} + bu_{yy} + cu_{zz} = f$ with $0 < a \ll b \ll c$, the strong coupling in y and z requires *y-z plane* relaxation, which is rather awkward. An alternative approach is to coarsen the grids in y and z only, and use z-line relaxation point by point in x and y. This approach is economical in three dimensions since the coarse grids still contain relatively few points.

Second, the grids do not all need to cover the whole problem domain. (This requires the Full Approximation Scheme described below.) In this way nonuniform resolution can be obtained, e.g., more resolution near singularities and sharp gradients and less resolution in outer regions to simulate unbounded domains. Programming is relatively easy since all operations are still performed on *uniform* grids. Local refinements can also be made adaptively, i.e, as required by the numerical solution as it evolves, leading to what Brandt refers to as Multi-Level Adaptive Techniques (MLAT).

c. Grid transfers

Associated with each grid G_l is a discrete problem

$$L^l u^l = f^l, \tag{4.8}$$

where L^l is a discrete (finite difference) approximation to L for mesh size h_l. For linear problems with coextensive grids, these discrete problems are interpreted as described in section 3c. That is, on the finest grid ($l = M$), (4.8) approximates the original problem (4.1), while on each coarser grid ($l < M$), (4.8) approximates the *residual* problem for the next finer grid. This interpretation is called the *Correction Scheme* (CS), since u^{l-1} is a correction to u^l. Two types of grid transfers are required: the fine-to-coarse transfer of the residual $r^l = f^l - L^l\tilde{u}^l$ via (3.11), represented by the operator I_l^{l-1}, and the coarse-to-fine interpolation of the correction via (3.12), represented by I_l^{l+1}. The proper choice of grid transfers depends in part on the order of the derivatives in L; precise rules are given in Brandt (1984, section 4.3). For most second-order problems one can use linear interpolation in each coordinate direction for the coarse-to-fine transfers. The choice of appropriate residual transfers, however, is somewhat more involved.

When the residual is smooth after relaxation, it may usually be transferred to the coarse grid by *injection*, i.e., simply copying the values to the coarse grid at the points common to both grids. In some cases, however, the residual is not smooth after relaxation, even though the error is. For example, with red-black or zebra ordering of relaxation, the residual is zero at the set of points most recently relaxed and nonzero elsewhere; nonsmooth residuals also can occur when L^l has nonsmooth coefficients. In such cases injection does not give an adequate approximation to the residual, and *full weighting* should be used instead. In this procedure the residual value transferred to a coarse-grid point is a weighted average of residual values at surrounding fine-grid points, with the weights chosen to preserve the integral of the residual over the problem domain. For example, 9-point full weighting in two dimensions is given by

$$I_l^{l-1} r^l = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} r^l, \tag{4.9}$$

where the weights are applied in the pattern indicated. As in all residual transfers, if the discrete problem has been multiplied by a power of h_l this must be taken into account: the residual to be transferred must be based on the divided form of L^l, i.e., the form which directly approximates L.

d. Control algorithms

A multigrid method requires an algorithm which determines when to switch from one grid to another. Such algorithms fall into the two basic classes discussed in this subsection.

1) CYCLING ALGORITHMS

A multigrid cycle for a discrete problem on a given grid G_l starts with an initial approximation to the solution on that grid and ends with an improved approximation on the same grid. A cycle uses relaxation for the G_l problem and corrections computed by transferring the G_l residual problem to the next coarser grid

G_{l-1}; the G_{l-1} problem in turn is also solved approximately by one or more multigrid cycles (unless G_{l-1} is the coarsest grid, in which case it is solved either directly or by many relaxation sweeps). Multigrid cycles can be used as a solution method by starting on the finest grid G_M with an initial approximation to the solution and making cycles repeatedly until the desired accuracy is obtained.

Multigrid cycles can be either *fixed* or *accommodative*. That is, the decisions of when to switch grids and which direction to go (i.e., to a coarser or finer grid) can be either prescribed in advance or made internally as the numerical solution develops. The *V*-cycle described in section 3c is a simple example of a fixed multigrid cycle. Other examples include the *W*- and *F*-cycles described in Brandt (1984); these solve the coarse-grid problems more accurately than *V*-cycles, and hence are more robust but often less efficient. Such fixed multigrid cycles are especially desirable during the development of a multigrid code, since accommodative algorithms can obscure conceptual or programming errors by their more complex internal interactions. On the other hand, accommodative algorithms are more robust.

The *cycle-C* algorithm of Fig. 7 is a generally applicable, fully accommodative cycling algorithm. The flow can be traced starting with the box labeled $\tilde{u}^l \leftarrow$ Relax $[L^l \cdot = f^l]\tilde{u}^l$, a shorthand notation for "make one relaxation sweep on the level l problem $L^l u^l = f^l$ to improve the current approximation $\tilde{u}^l$". After a relaxation sweep, $\tilde{u}^l$ is tested to see whether it has converged. If it has, the level l problem has been solved, so if $l < M$ its solution is interpolated to the next finer level $l + 1$ and used as a correction to the current approximation $\tilde{u}^{l+1}$ there. On the other hand, if $\tilde{u}^l$ has not converged, the speed of convergence is checked. Fast convergence indicates that the error is not smooth yet (i.e., substantial high-wavenumber errors still exist); relaxation is still effective, so another sweep is made. Slow convergence indicates that the error is smooth (i.e., high-wavenumber errors have been reduced substantially by relaxation); further relaxation would be ineffective, so the residual is transferred to the next coarser level $l - 1$ to serve as the right-hand side of the discrete problem there, and the initial approximate solution $\tilde{u}^{l-1}$ is set to zero. On the coarsest grid $l = 1$, relaxation continues until the solution converges. The cycle starts on the finest grid $l = M$ with an initial approximation $\tilde{u}^M$ to u^M, and ends on that grid when $\tilde{u}^M$ has converged.

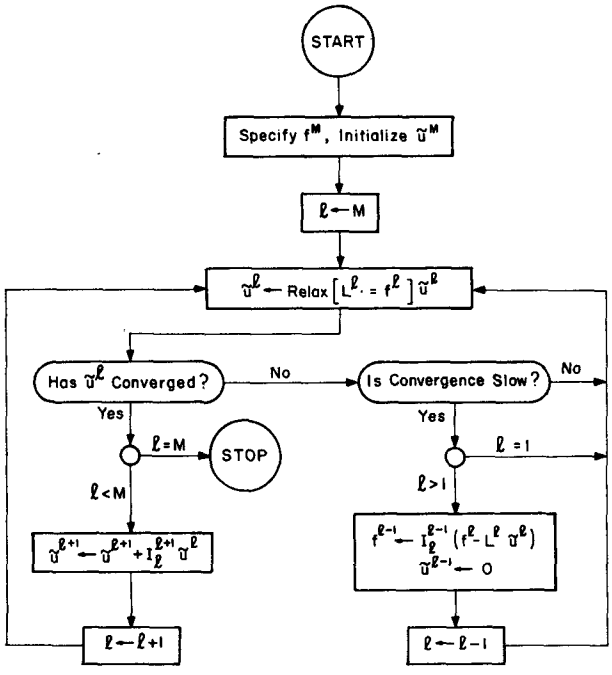

FIG. 7. The multigrid cycle-*C* algorithm (after Brandt, 1977a).

In an accommodative algorithm the tests of convergence are generally made in practice using *dynamic residuals*, i.e., residuals computed during the relaxation process. For example, point Gauss-Seidel relaxation with lexicographic ordering (2.7) for the Poisson problem can be written as

$$\tilde{u}_{jk}^{\text{new}} = \tilde{u}_{jk} + \frac{h^2}{4}\bar{r}_{jk}, \tag{4.10}$$

where

$$\bar{r}_{jk} = f_{jk} - h^{-2}[4\tilde{u}_{jk} - \tilde{u}_{j-1,k}^{\text{new}} - \tilde{u}_{j+1,k} - \tilde{u}_{j,k-1}^{\text{new}} - \tilde{u}_{j,k+1}] \tag{4.11}$$

is the dynamic residual at the point (j, k). The speed of convergence can be tested by computing the ratio of some norm (e.g., rms or max norm) of the dynamic residual associated with the current relaxation sweep to the corresponding residual norm for the previous sweep. If this ratio is close to the smoothing factor $\bar{\mu}$ (or smaller), the convergence is fast and relaxation is still effective; if it is significantly larger than $\bar{\mu}$, the convergence is slow and relaxation is no longer effective. Similarly, convergence is obtained on a given grid G_l when the residual norm is less than a specified factor η times the latest residual norm on the next finer grid, i.e., the accuracy of the G_{l+1} solution (for which it is a correction). The performance of the algorithm is not very sensitive to the choice of the parameter η; Brandt (1984) suggests the choice $\eta = 2^{-d}$, where d is the number of dimensions of the problem.

On the finest grid $l = M$ one usually solves "to the level of truncation error", i.e., so the norm of the residual $r^M = f^M - L^M\tilde{u}^M$ (or the dynamic residual) is comparable to that of the truncation error

$$\tau^M = L^M(\hat{I}^M u) - f^M, \tag{4.12}$$

where u is the solution of the continuous problem (4.1) and $\hat{I}^M$ represents a continuum-to-grid M transfer of the solution. Further algebraic convergence of $\tilde{u}^M$ to u^M is usually not needed, since it would not necessarily

make $\tilde{u}^M$ any closer to u. An estimate of the norm of the truncation error can be specified in advance (using Taylor expansions and typical values of derivatives of u), or can be generated internally during the multigrid process if the Full Approximation Scheme (FAS) is used.

2) FULL MULTIGRID (FMG) ALGORITHMS

When using multigrid cycles as a solution method one starts with an initial approximation $\tilde{u}^M$ on the finest grid G_M. One possible choice is simply $\tilde{u}^M = 0$; however, if a better initial approximation can be made then fewer cycles will be required to solve the problem. To get a better first approximation one can first solve the discrete approximation to (4.1)—not a residual problem—on the next coarser grid G_{M-1}, and interpolate that solution to grid G_M. Extending this idea recursively back through coarser and coarser grids leads to a *Full Multigrid (FMG) algorithm.*

Figure 8 illustrates a simple FMG algorithm for a problem with $M = 4$ grid levels. The process starts on the coarsest grid $l = 1$ where the discrete approximation to (4.1) is simply solved. The resulting solution is interpolated to the next finer grid $l = 2$ and used as the first approximation to the solution of the corresponding G_2 problem, which is then solved by a multigrid V-cycle. This process is repeated, interpolating the solution from one level to the next as a first approximation and solving by a V-cycle, until the final solution is obtained on the finest level $l = M$. Thus the FMG algorithm is, in fact, a nested iteration method, i.e., a "bootstrap" method which uses an iterative solver on each grid G_1 through G_M in turn; the iterative solver used for each grid is a multigrid cycle. Although Fig. 8 shows simple V-cycles, any multigrid cycle, fixed or accommodative, can be used. The initial interpolation to a finer grid (denoted by a double arrow in Fig. 8) may be different than that used in the multigrid cycles; in particular, it generally should have a higher order of accuracy, since the function being interpolated is relatively smooth (Brandt, 1984, sec. 7.1).

FMG algorithms are very efficient. In going from one level l to the next finer level $l + 1$, the truncation error decreases by roughly the factor $\hat{\rho}^p$, where $\hat{\rho} = h_{l+1}/h_l$ is the mesh ratio and p is the order of the difference approximations. Since a single multigrid cycle typically reduces the error by at least this factor, an FMG algorithm which uses just a single cycle per grid level will give the final solution on each level (denoted by the double circles in Fig. 8) to at least the level of the respective truncation error. The computational work expended is usually expressed in terms of the multigrid *work unit,* defined as the amount of computational work required per relaxation sweep on the finest grid. For the algorithm of Fig. 8 the total work required (assuming a two-dimensional problem with $\hat{\rho} = \frac{1}{2}$ and ignoring the work of grid transfers) is

$$\nu\left(1 + \frac{2}{4} + \frac{3}{4^2} + \cdots + \frac{M}{4^{M-1}}\right) < \frac{16}{9}\nu \quad (4.13)$$

work units, where $\nu = \nu_1 + \nu_2$. Typically $\nu = 2$ or $\nu = 3$ is sufficient, so only 3.6–5.3 work units are required. This efficiency is typical: optimal multigrid methods for most problems yield the solution to the level of truncation error in only a few (4–10) work units.

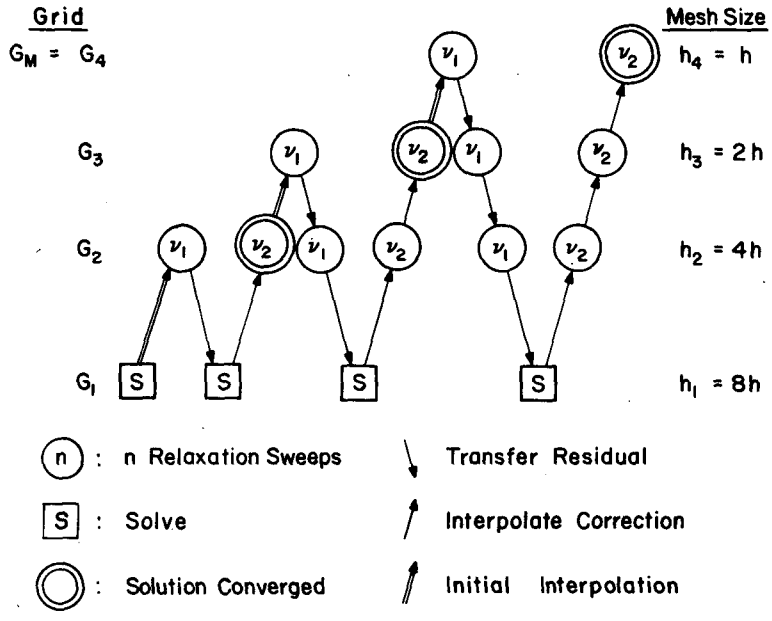

FIG. 8. A Full Multigrid (FMG) algorithm for a problem with $M = 4$ grids.

e. The Full Approximation Scheme (FAS)

The multigrid methods described so far use the correction scheme: each coarse-grid equation is an approximation to the residual problem on the next finer grid, so the coarse-grid solution is an approximate correction to the fine-grid solution. This scheme is valid for linear problems only, since it depends on the linearity of L to write the residual problem for the correction $v^h = u^h - \tilde{u}^h$ as $L^h v^h = r^h = f^h - L^h \tilde{u}^h$. Similarly, the correction scheme requires coextensive grids, since the coarse grid cannot provide a correction to the fine-grid solution where the fine grid does not exist. An alternative scheme which circumvents these restrictions is described below.

Without assuming linearity, the fine-grid residual equation

$$L^h(\tilde{u}^h + v^h) - L^h\tilde{u}^h = r^h = f^h - L^h\tilde{u}^h \quad (4.14)$$

can be approximated by the coarse-grid equation

$$L^H(\hat{I}_h^H\tilde{u}^h + v^H) - L^H(\hat{I}_h^H\tilde{u}^h) = I_h^H r^h. \quad (4.15)$$

Here, v^H approximates the true correction v^h to $\tilde{u}^h$, and the solution transfer operator $\hat{I}_h^H$ may differ from the residual transfer operator I_h^H. Putting the known quantities on the right reduces (4.15) to

$$L^H\hat{u}^H = \hat{f}^H, \quad (4.16)$$

where

$$\hat{u}^H = \hat{I}_h^H\tilde{u}^h + v^H \quad (4.17)$$

and

$$\hat{f}^H = L^H(\hat{I}_h^H\tilde{u}^h) + I_h^H r^h. \quad (4.18)$$

This is known as the *Full Approximation Scheme* (FAS), since from (4.17) the coarse-grid variable $\hat{u}^H$ approximates the full solution u^h, not just the correction v^h. After solving (4.16), the fine-grid solution is updated via

$$\tilde{u}^h_{\text{new}} = \tilde{u}^h + I_H^h(\hat{u}^H - \hat{I}_h^H\tilde{u}^h), \quad (4.19)$$

which represents interpolating and adding the approximate correction v^H. Note that (4.19) cannot be simplified to $\tilde{u}^h_{\text{new}} = \hat{I}_H^h\hat{u}^H$ without losing all high-wavenumber information in $\tilde{u}^h$.

Using the FAS, multigrid methods can solve nonlinear problems with the same efficiency as the corresponding linearized problems. No global linearization is needed: a single Newton iteration often suffices to solve the nonlinear relaxation equation at a single point, but even this simple local linearization is usually unnecessary. In addition, FAS allows local mesh refinement for obtaining nonuniform resolution organized by simple uniform grids, as mentioned in section 4b. Where the fine grid exists, the coarse-grid variable represents $\hat{u}^H$ as defined by (4.17); where the fine grid does not exist, the coarse-grid variable simply represents the coarse-grid solution.

The FAS also provides a useful analogue of the local truncation error as follows. From (4.12), the truncation error τ^H on the coarse grid satisfies

$$L^H(\hat{I}^H u) = I^H f + \tau^H \quad (4.20)$$

and thus

$$\tau^H = L^H(\hat{I}^H u) - I^H(Lu), \quad (4.21)$$

where $\hat{I}^H$ and I^H represent continuum-to-grid H transfers of the solution and right-hand side, respectively. Thus τ^H is the quantity which must be added to the coarse-grid equation so that its solution coincides (on the coarse grid) with the continuous solution. Similarly, substituting from (4.18) the FAS coarse-grid equation (4.16) becomes

$$L^H\hat{u}^H = I_h^H f^h + \tau_h^H \quad (4.22)$$

where

$$\tau_h^H = L^H(\hat{I}_h^H\tilde{u}^h) - I_h^H(L^h\tilde{u}^h). \quad (4.23)$$

At convergence, $\hat{u}^H = \hat{I}_h^H u^h$, so τ_h^H is the *relative local truncation error*, i.e., the quantity which must be added to the coarse-grid equation so that its solution coincides (on the coarse grid) with the fine-grid solution.

The computable analogue τ_h^H to the truncation error makes possible several useful techniques. First, τ_h^H can be used to raise the order of approximation on the coarse grid; this *τ-extrapolation* is similar to Richardson extrapolation but more general, since it extrapolates the equation, not the solution. Second, τ_h^H can be used to estimate the truncation error τ^H (or τ^h), thus providing criteria for stopping multigrid cycling or for adaptive local mesh refinement. These and other applications of τ_h^H are discussed in Brandt (1984, sections 8 and 9).

f. Boundary conditions

Ideally, boundary conditions should not affect the overall efficiency of a multigrid method (assuming, of course, that the problem is well-posed). The boundary typically contains far fewer grid points than the interior of the problem domain; if extra work per grid point must be invested near the boundary it will not increase the total computational work by much. However, with complicated boundary conditions one must design each element of the multigrid method carefully in order to attain this ideal.

Dirichlet conditions (i.e., u specified along the boundary) are easily incorporated into multigrid methods. Each time an initial approximation is formed on a grid, the known boundary values are inserted directly. In correction schemes, this means using the specified values on the boundaries of the finest grid and zeros on the boundaries of the coarser grids. Only the interior equations are relaxed, so the boundary conditions are satisfied on all grids at all times.

More general conditions, such as Neumann conditions (i.e., the normal derivative of u specified along the boundary) can be implemented, keeping in mind the following two basic principles. First, boundary

conditions are equations and hence must be relaxed, just as the interior equations are relaxed; their residuals after relaxation must be transferred to the coarse grid along with the interior residuals. These residuals must be based on the divided form of the equations, i.e., with the finite difference approximations to derivatives not multiplied through by a power of h. Second, the role of boundary relaxation is not to impose the boundary conditions but to smooth the error along the boundary (similar to the role of interior relaxation). Thus, relaxation should change a boundary value not so that the corresponding boundary condition is satisfied exactly but so that the resulting error is the average of the errors at neighboring boundary points.

g. Generalizations

The basic multigrid idea of cycling between different levels of discretization to reduce the error efficiently on all scales can be applied to discretizations other than finite differences. For example, series approximation methods (finite element and spectral) represent the unknown as a truncated series of known basis functions. Multigrid (or, more properly, multilevel) techniques for such discretizations use a set of approximation subspaces $S_1, S_2, \ldots, S_M$ in place of the grids $G_1, G_2, \ldots, G_M$. The transfer operations between levels are usually very natural; for example, since S_{l-1} is usually a subset of S_l the coarse-to-fine transfer I_{l-1}^{l} is simply the identity operator. With finite-element discretizations (i.e., local basis functions) most aspects of the multigrid process are similar to the finite difference case (e.g., Brandt, 1977a, sections 7.3 and A.5; Nicolaides, 1977; Brandt, 1980; Bank, 1981). With spectral or pseudospectral discretizations (i.e., global basis functions) the relaxation scheme must be designed carefully to take advantage of fast transform techniques (Zang et al., 1982, 1984).

The multigrid idea can also be used in solving discrete problems which have no underlying geometric interpretation. *Algebraic multigrid* (AMG) methods solve matrix problems of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is positive definite, using only the information in the matrix $\mathbf{A}$ to construct an appropriate set of coarser "levels", each consisting of a subset of the original variables in $\mathbf{x}$ (Stüben, 1983). Such methods exhibit the usual multigrid efficiency, i.e., solution in just a few work units; however, they require considerable overhead in terms of storage and set-up time. They are useful when the construction of conventional (geometric) multigrid methods would be impossible (e.g., no underlying geometry) or very difficult (e.g., nonuniform finite-element discretizations).

Multigrid methods can be generalized in many other ways, e.g., to nonelliptic problems, eigenvalue problems, and systems of equations; some of these generalizations are mentioned briefly in section 5. However, a discussion of the techniques involved is outside the scope of this review, and the interested reader is referred to the multigrid literature cited, especially Brandt (1984), for details.

5. Applications

Multigrid methods have been applied successfully to a wide range of problems. Much of this work is quite recent; some methods are still being refined, and new applications are currently being demonstrated. This section briefly surveys some of the more prominent applications to date, and points out areas of current research.

Multigrid methods were originally developed as fast solvers for discretized elliptic boundary-value problems. The multigrid treatment of the Poisson problem was discussed in detail by Stüben and Trottenberg (1982). Gary et al. (1983) compared multigrid, SOR, and conjugate gradient methods for the diffusion equation

$$-\nabla \cdot (D\nabla u) = f, \tag{5.1}$$

where D is a diffusion coefficient (scalar or tensor), treating the three-dimensional case with D assumed to vary slowly. In some applications, such as neutron diffusion, D may have strong discontinuities (i.e., jumps of several orders of magnitude); this more difficult case has been treated in two dimensions by Alcouffe et al. (1981) and in three dimensions by Behie and Forsyth (1983). Kettler (1982) studied multigrid methods for the convection-diffusion equation

$$-\nabla \cdot (D\nabla u) + \mathbf{v} \cdot \nabla u = f \tag{5.2}$$

in two dimensions, giving detailed analyses of smoothing properties for many different relaxation schemes. Singular-perturbation problems have also been treated (Brandt, 1979; Hemker, 1982); efficient multigrid methods for such problems rely heavily on appropriate local mesh refinements.

Various types of nonlinear problems have also been solved by multigrid methods. Among mathematicians, a favorite nonlinear test problem is

$$\lambda e^{u} - \nabla^2 u = f, \tag{5.3}$$

which arises in chemical reaction theory. Multigrid methods for (5.3) have been developed by Meis et al. (1982); current research includes continuation techniques for this and other bifurcation problems (Mittlemann and Weber, 1985). Ordinary (linear) eigenproblems can also be regarded as nonlinear, since the unknown eigenvalue multiplies the unknown eigenfunction. An FAS-FMG method obtains each eigenfunction with the usual multigrid efficiency (Hackbusch, 1979; Brandt et al., 1983).

Much research has been done on applying multigrid methods to more complicated nonlinear problems in fluid dynamics, including systems of equations. The equation governing transonic potential flow (e.g., air over an airfoil) is not only nonlinear, but also elliptic over parts of the domain and hyperbolic elsewhere. It has been solved efficiently by multigrid methods in two dimensions (Boerstoel, 1982), three dimensions (Caughey, 1983), with mesh refinement (McCarthy and Swanson, 1983), and with finite-element (Deconinck and Hirsch, 1982) and spectral (Streett et al., 1985) discretizations. Multigrid methods have also been developed for the Euler equations (e.g., Jespersen, 1983) and the Navier-Stokes equations (e.g., Ghia et al., 1982); a detailed summary of techniques for these systems appears in Brandt (1984). Most applications in fluid dynamics so far have been for steady state solutions; multigrid methods for time-dependent flows have been discussed by Brandt et al. (1980), Lee and Myers (1980), Jameson (1983), and Johnson (1983).

Research in basic multigrid techniques also continues. One area of interest is higher-order techniques. Schaffer (1984) investigated higher-order difference schemes, defect correction methods, and τ-extrapolation as means of obtaining high accuracy with multigrid methods. Brandt et al. (1985) developed improved relaxation schemes for Fourier-spectral multigrid methods for isotropic and anisotropic problems. Mesh generation and refinement techniques have also seen much attention (e.g., Van Rosendale, 1983; Bai and Brandt, 1984). With the increasing importance of vector and parallel processors such as the CRAY-1, CRAY X-MP and CYBER 205, research on vectorizable and parallelizable multigrid algorithms has expanded. Brandt (1981) gives a comprehensive discussion of such algorithms, and Barkai and Brandt (1983) discuss the details of implementing a multigrid Poisson solver on the CYBER 205.

The relative complexity of multigrid methods may have been a factor in their relatively slow acceptance up until now. Recognizing this, many researchers have attempted to develop "black box" solvers, i.e., software which can be used as a tool by other researchers without having to deal with the multigrid details. Dendy (1982, 1983) has developed such routines for various types of problems; similar work has been done by Wesseling (1982) and Hemker et al. (1983). Foerster and Witsch (1982) describe a package of routines (MG00) for solving second-order elliptic boundary value problems on rectangular domains; listings of some routines from this package appear in Stüben and Trottenberg (1982), and further details appear in Stüben et al. (1984). The latter paper also describes a code (AMG01) based on perhaps the ultimate "black box" approach, namely, the algebraic multigrid concept mentioned in subsection 4g. Many of the solvers described above are available on a tape of multigrid software (MUGTAPE) from the Department of Applied Mathematics at the Weizmann Institute of Science, Rehovot 76100, Israel. This tape also contains code for simple model problems from Brandt (1977a) and Brandt (1984), plus a package of general grid manipulation routines (GRIDPACK) described by Brandt (1977b) and Brandt and Ophir (1984). In addition, the code MG00 has been incorporated into the ELLPACK system of software for solving elliptic problems (Rice, 1984).

6. Concluding remarks

The multigrid approach is a powerful, flexible way to develop fast solvers for discretized elliptic boundary-value problems. This power comes at the price of some complexity, but some general software is available, in addition to detailed information on developing specific multigrid codes. As discussed in section 1, many elliptic problems arise in geophysical fluid dynamics, often involving variable coefficients or nonlinearities. Even though high efficiency is often crucial, these problems are currently being solved by methods which are less than optimal. We hope this review will encourage others to try multigrid methods in such cases. In addition, multigrid methods can be combined with adaptive discretization in so-called Multi-Level Adaptive Techniques (MLAT); while still in their infancy (especially for hyperbolic problems), these techniques hold considerable promise for the efficient and accurate solution of geophysical problems involving localized phenomena and widely differing scales of motion.

Acknowledgments. The authors gratefully acknowledge the help and guidance provided by Achi Brandt, especially during his sabbatical year at the Institute for Computational Studies at Colorado State University. We have also benefitted from discussions with Klaus Stüben, Gerald Taylor, and Thomas Zang, and thank Odilia Panella for her help in preparing the manuscript.

This work was funded by the Naval Environmental Prediction Research Facility, Monterey, California, under program element 61153N "Multigrid Elliptic Solver Development."

REFERENCES

Alcouffe, R. E., A. Brandt, J. E. Dendy, Jr. and J. W. Painter, 1981: The multi-grid method for the difusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput.*, **2**, 430–454.

Bai, D., and A. Brandt, 1984: Local mesh refinement multilevel techniques. 45 pp. [Available from Department of Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel.]

Bank, R. E., 1981: A multi-level iterative method for nonlinear elliptic equations. *Elliptic Problem Solvers.* M. Schultz, Ed., Academic Press, 1–16.

Barkai, D., and A. Brandt, 1983: Vectorized multigrid Poisson solver for the CDC CYBER 205. *Appl. Math. Comp.*, **13**, 215–227.

Behie, A., and P. A. Forsyth, Jr., 1983: Multi-grid solution of three-dimensional problems with discontinuous coefficients. *Appl. Math. Comp.*, **13**, 229–240.

Boerstoel, J. W., 1982: A multigrid algorithm for steady transonic potential flows around aerofoils using Newton iteration. *J. Comput. Phys.*, **48**, 314–343.

Brand, K., 1982: Multigrid bibliography. *Multigrid Methods.* W. Hackbusch, U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 631–650. [Continued in *Multigrid Newsletter,* available from S. McCormick, Department of Mathematics, Box 170, University of Colorado of Denver, 1100 14th Street, Denver, CO 80202.]

Brandt, A., 1973: Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. Proc. Third Int. Conf. Numerical Methods in Fluid Mechanics, Paris, 1972, *Lecture Notes in Physics,* H. Cabannes and R. Temam, Eds., Springer-Verlag, **18**, 82–89.

——, 1977a: Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, **31**, 333–390.

——, 1977b: Multi-level adaptive techniques (MLAT) for partial differential equations: Ideas and software. *Mathematical Software III.* J. R. Rice, Ed., Academic Press, 277–318.

——, 1979: Multi-level adaptive techniques (MLAT) for singular-perturbation problems. *Numerical Analysis of Singular Perturbation Problems.* P. W. Hemker and J. J. H. Miller, Eds., Academic Press, 53–142.

——, 1980: Multi-level adaptive finite-element methods: I. Variational problems. *Special Topics of Applied Mathematics.* J. Frehse, D. Pallaschke and U. Trottenberg, Eds., North-Holland, 91–128.

——, 1981: Multigrid solvers on parallel computers. *Elliptic Problem Solvers.* M. Schultz, Ed., Academic Press, 39–83.

——, 1982: Guide to Multigrid Development. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 220–312.

——, 1984: *Multigrid Techniques: 1984 Guide with applications to fluid dynamics,* 176 pp. [Available from GMD, Postfach 1240, D-5205 St. Augustin, 1, F. R. G.]

——, and D. Ophir, 1984: GRIDPACK: Toward unification of general grid programming. *PDE Software: Modules, Interfaces and Systems.* B. Engquist and T. Smedsaas, Eds., Elsevier, 269–288.

——, J. E. Dendy, Jr. and H. Ruppel, 1980: The multigrid method for semi-implicit hydrodynamics codes. *J. Comput. Phys.*, **34**, 348–370.

——, S. McCormick and J. Ruge, 1983: Multigrid methods for differential eigenproblems. *SIAM J. Sci. Stat. Comput.*, **4**, 244–260.

——, S. R. Fulton and G. D. Taylor, 1985: Improved spectral multigrid methods for periodic elliptic problems. *J. Comput. Phys.*, **58**, 96–112.

Buzbee, B. L., G. H. Golub and C. W. Nielson, 1970: On direct methods for solving Poisson's equations. *SIAM J. Numer. Anal.*, **7**, 627–656.

Caughey, D. A., 1983: Multi-grid calculation of three-dimensional transonic potential flows. *Appl. Math. Comp.*, **13**, 241–260.

Deconinck, H., and C. Hirsch, 1982: A multigrid finite element method for the transonic potential equation. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 387–409.

Dendy, J. E., Jr., 1982: Black box multigrid. *J. Comput. Phys.*, **48**, 366–386.

——, 1983: Black box multigrid for nonsymmetric problems. *Appl. Math. Comp.*, **13**, 261–283.

Fedorenko, R. P., 1962: A relaxation method for solving elliptic difference equations. *U.S.S.R. Comput. Math. Math. Phys.*, **1**, 1092–1096.

——, 1964: The speed of convergence of an iterative process. *U.S.S.R. Comput. Math. Math. Phys.*, **4**, 227–235.

Foerster, H., and K. Witsch, 1982: Multigrid software for the solution of elliptic problems on rectangular domains: MG00 (Release 1). *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 427–460.

Gary, J., S. McCormick and R. Sweet, 1983: Successive overrelaxation, multigrid, and preconditioned conjugate gradients algorithms for solving a diffusion problem on a vector computer. *Appl. Math. Comp.*, **13**, 285–309.

Ghia, U., K. N. Ghia and C. T. Shin, 1982: High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, **48**, 387–411.

Hackbusch, W., 1979: On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method. *SIAM J. Numer. Anal.*, **16**, 201–215.

Haltiner, G. J., and R. T. Williams, 1980: *Numerical Prediction and Dynamic Meteorology,* second edition. J. Wiley and Sons, 477 pp.

Hemker, P. W., 1982: Mixed defect correction iteration for the accurate solution of the convection diffusion equation. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 485–501.

——, R. Kettler, P. Wesseling and P. M. de Zeeuw, 1983: Multigrid methods: Development of fast solvers. *Appl. Math. Comp.*, **13**, 311–326.

Hoskins, B. J., 1975: The geostrophic momentum approximation and the semi-geostrophic equations. *J. Atmos. Sci.*, **32**, 233–242.

——, and I. Draghici, 1977: The forcing of ageostrophic motion according to the semi-geostrophic equations and in an isentropic coordinate model. *J. Atmos. Sci.*, **34**, 1859–1867.

Jameson, A., 1983: Solution of the Euler equations for two dimensional transonic flow by multigrid method. *Appl. Math. Comp.*, **13**, 327–355.

Jespersen, D., 1983: Design and implementation of a multigrid code for the Euler equations. *Appl. Math. Comp.*, **13**, 357–374.

Johnson, G. M., 1983: Multiple-grid convergence acceleration of viscous and inviscid flow computations. *Appl. Math. Comp.*, **13**, 375–398.

Kettler, R., 1982: Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 502–534.

Lee, H. N., and R. E. Meyers, 1980: On time dependent multi-grid numerical technique. *Comp. Math. Appl.*, **6**, 61–65.

Lindzen, R. S., and H.-L. Kuo, 1969: A reliable method for the numerical integration of a large class of ordinary and partial differential equations. *Mon. Wea. Rev.*, **97**, 732–734.

McCarthy, D. R., and R. C. Swanson, 1983: Embedded mesh multigrid treatment of two-dimensional transonic flows. *Appl. Math. Comp.*, **13**, 399–417.

Meis, Th., H. Lehmann and H. Michael, 1982: Application of the multigrid method to a nonlinear indefinite problem. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 545–557.

Mittlemann, H. D., and H. Weber, 1985: Multi-grid solution of bifurcation problems. *SIAM J. Sci. Stat. Comp.*, **6**, 49–60.

Nicolaides, R. A., 1977: On the l^2 convergence of an algorithm for solving finite element equations. *Math. Comp.*, **31**, 892–906.

Ogura, Y., and N. A. Phillips, 1962: Scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.*, **19**, 173–179.

Rice, J. R., 1984: Building elliptic problem solvers with ELLPACK. *Elliptic Problem Solvers II.* G. Birkhoff and A. Schoenstadt, Eds., Academic Press, 3–22.

Schaffer, S., 1984: Higher order multi-grid methods. *Math. Comp.*, **43**, 89–115.

Schubert, W. H., and J. J. Hack, 1983: Transformed Eliassen balanced vortex model. *J. Atmos. Sci.*, **40**, 1571–1583.

Stoer, J., and R. Bulirsch, 1980: *Introduction to Numerical Analysis.* Springer-Verlag, 609 pp.

Streett, C. L., T. A. Zang and M. Y. Hussaini, 1985: Spectral multigrid methods with applications to transonic potential flow. *J. Comput. Phys.*, **57**, 43–76.

Stüben, K., 1983: Algebraic multigrid (AMG): Experiences and comparisons. *Appl. Math. Comp.*, **13**, 419–452.

——, and U. Trottenberg, 1982: Multigrid methods: Fundamental algorithms, model problem analysis and applications. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 1–176.

——, U. Trottenberg and K. Witsch, 1984: Software development based on multigrid techniques. *PDE Software: Modules, Interfaces and Systems.* B. Engquist and T. Smedsaas, Eds., Elsevier, 241–263.

Swarztrauber, P. N., 1977: The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle. *SIAM Review,* **19**, 490–501.

Sweet, R. A., 1977: A cyclic reduction algorithm for solving block tridiagonal systems of arbitrary dimension. *SIAM J. Numer. Anal.*, **14**, 706–720.

Thorpe, A. J., 1985: Diagnosis of balanced vortex structure using potential vorticity. *J. Atmos. Sci.*, **42**, 397–406.

Van Rosendale, J., 1983: Algorithms and data structures for adaptive multigrid elliptic solvers. *Appl. Math. Comp.*, **13**, 453–470.

Wesseling, P., 1982: A robust and efficient multigrid method. *Multigrid Methods.* W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics,* Vol. 960, Springer-Verlag, 614–630.

Zang, T. A., Y. S. Wong and M. Y. Hussaini, 1982: Spectral multigrid methods for elliptic equations. *J. Comput. Phys.*, **48**, 485–501.

——, Y. S. Wong and M. Y. Hussaini, 1984: Spectral multigrid methods for elliptic equations II. *J. Comput. Phys.*, **54**, 489–507.