

MULTIGRID SOLUTION OF AN ELLIPTIC BOUNDARY-VALUE PROBLEM WITH INTEGRAL CONSTRAINTS*

ACHI BRANDT[†], MOSHE ISRAELI[‡], IRAD YAVNEH[‡], AND ANDREW SIEGEL[§]

Abstract. An efficient multigrid approach for solving a discretized elliptic equation whose boundary values are determined in part by integral relations is developed, analyzed, and tested. The algorithm is motivated by a problem that is solved during integration of the 3D quasigeostrophic (QG) equations, which model large-scale rotating stratified flows, where the integral constraints represent mass conservation.

Key words. multigrid, elliptic partial differential equations, integral constraints

AMS subject classifications. 65N22, 65N55

PII. S1064827597331813

1. Introduction. Elliptic partial differential equations (PDEs) are typically posed as boundary-value problems, where the differential equation has to be satisfied inside the domain of solution, while given conditions are imposed on the boundary: usually the value of the solution or its normal derivative or some combination thereof. In some applications, however, some of the boundary conditions are replaced by integral constraints. Such constraints might be related to certain conserved physical properties, such as mass or energy, or they might be artificially imposed relations that lead to a better conditioned problem (see [1, section 5.6] and references therein.)

Multigrid methods are amongst the most effective solvers of discretized elliptic boundary-value problems. Typically, the amount of computational work required to solve such problems to the level of accuracy allowed by the discretization is just a few *work units*, where a work unit is the amount of computational work required for discretizing the problem on the target grid (e.g., calculating the residuals at every fine-grid point). Brandt [1] suggests that similar efficiency can be obtained for problems involving integral constraints with just a few length-scales (typically domain-size), such as one or a few moments of the solution field. The proposed approach is to enforce such constraints only on very coarse grids, while not treating them at all on the fine grids. But in the present work we are concerned with integral constraints that affect many, or even all, scales of the solution.

We consider a second-order elliptic PDE of the form

$$(1) \quad Lu = \nabla \cdot (a(x, y, z)\nabla u) + (g(x, y, z)u_z)_z = f(x, y, z), \quad (x, y, z) \in \Omega,$$

where u is unknown, g and a are uniformly positive (hence, L is elliptic), f is given, $\nabla = (\partial_x, \partial_y)$, and the subscripts denote partial derivatives. We consider cylindrical domains of the form $\Omega = \Omega_{horiz} \times (0, D)$, where Ω_{horiz} is some simply connected, open

*Received by the editors December 23, 1997; accepted for publication (in revised form) January 7, 1999; published electronically February 2, 2000. The research of this paper was supported by the U.S.-Israel Binational Science Foundation under grant 94-00250, the Fund for the Promotion of Research at the Technion, NASA Research grant NAG 1 1823, and the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science.

<http://www.siam.org/journals/sisc/21-4/33181.html>

[†]The Weizmann Institute of Science, Rehovot, Israel (achi@wisdom.weizmann.ac.il).

[‡]Technion—Israel Institute of Technology, Haifa, Israel (israeli@cs.technion.ac.il, irad@cs.technion.ac.il).

[§]University of Colorado at Boulder, Boulder, CO.

domain in the (x, y) plane. The concept of our method, however, is not restricted to such simple geometries. At the “top” and “bottom” boundaries, homogeneous Neumann conditions are imposed:

$$(2) \quad u_z(x, y, 0) = u_z(x, y, D) = 0, \quad (x, y) \in \Omega_{horiz}.$$

The lateral boundary conditions are given by

$$(3) \quad u(x, y, z) = U(z), \quad (x, y, z) \in \partial\Omega_{horiz} \times [0, D],$$

where $U(z)$ is an *unknown* function, which is determined implicitly by the following integral constraint:

$$(4) \quad \langle u \rangle(z) \stackrel{\text{def}}{=} |\Omega_{horiz}|^{-1} \iint_{\Omega_{horiz}} u(x, y, z) \, dx \, dy = F(z),$$

where $F(z)$ is given. Here, $|\cdot|$ denotes area. Equation (4) implies that $\langle u \rangle$ —the horizontal average of u —is imposed for all $z \in [0, D]$.

This completes the formulation of the problem. Our motivation comes from reduced models of large-scale rotating stratified flows, such as the quasigeostrophic (QG) equations [7]. The function g (which only varies with z in QG) then represents the inverse of the stratification parameter, and the variation of the integral relations (4) over time determines the mass that is gained or lost at every height-level. Normally, $F(z) \equiv 0$.

In section 2 we analyze the system (1)–(4) and show that the problem is well-posed under some simplifying assumptions. In section 3 we pose a simplified problem in 2D (for the purpose of analysis) and a finite-difference discretization. In section 4 we present a “naive” algorithm, whose convergence is provable but slow. In section 5 we present a fast algorithm, based on multigrid methods, followed by numerical tests which appear in section 6. In section 7 we modify the basic algorithm in order to improve its robustness, by employing line relaxation and partial coarsening, and demonstrate the efficacy of this algorithm by numerical tests. All the numerical tests are carried out for the 3D problem. Section 8 presents conclusions.

2. Analysis of the separable case. Assume $a = a(x, y)$, and $g = g(z)$. This separability assumption holds in some of the geophysical-flow problems which motivate this work. There then exists a complete orthogonal system of eigenfunctions, $I_k(z)$ (in particular such that satisfy the top and bottom homogeneous Neumann boundary conditions of (1)–(4)), and corresponding negative (real) eigenvalues, $-\alpha_k$, which satisfy

$$(5) \quad \frac{d}{dz} \left(g(z) \frac{d}{dz} I_k(z) \right) = -\alpha_k I_k(z).$$

Furthermore, for any continuous function u with continuous first and second derivatives, which satisfies the top and bottom boundary conditions, the (unique) expansion,

$$(6) \quad u(x, y, z) = \sum_{k=0}^{\infty} u_k(x, y) I_k(z),$$

converges absolutely and uniformly (see discussions of the Sturm–Liouville eigenvalue problem, e.g., in [3, pp. 291–293] and also [4, pp. 107–110]). If we similarly expand U , f , and F , we obtain from (1)–(4),

$$(7) \quad \nabla \cdot (a(x, y) \nabla u_k) - \alpha_k u_k = f_k(x, y), \quad (x, y) \in \Omega_{horiz},$$

$$(8) \quad u_k(x, y) = U_k, \quad (x, y) \in \partial\Omega_{horiz},$$

$$(9) \quad \langle u_k \rangle = F_k,$$

where f_k and F_k are given, but U_k are unknown.

Let u_k^A denote the solution of the boundary-value problem represented by (7), with homogeneous Dirichlet boundary conditions (i.e., $U_k \equiv 0$), and let u_k^B denote the solution to (7) with a zero right-hand side and $U_k \equiv 1$. Then, by (7)–(8), u_k can be uniquely written as the following linear combination of u_k^A and u_k^B :

$$(10) \quad u_k = u_k^A + U_k u_k^B.$$

Now U_k can be determined by taking the horizontal average of (10) and using (9), obtaining

$$(11) \quad U_k = \frac{-\langle u_k^A \rangle + F_k}{\langle u_k^B \rangle}.$$

Here, the denominator, $\langle u_k^B \rangle$, is known to be positive, as u_k^B does not change sign by the maximum principle.

Since u_k^A and u_k^B are the unique solutions of well-posed Dirichlet boundary-value problems, this proves that (7)–(9) also constitute a well-posed problem, whose unique solution is given by (10)–(11). These equations also suggest a direct algorithm for the solution of (1)–(4), which requires performing a 1D transformation in the vertical, and then solving decoupled sets of 2D elliptic equations with Dirichlet boundary conditions to obtain $\langle u_k^A \rangle$ and $\langle u_k^B \rangle$. This method is robust, and we estimate the cost to be comparable to that of the robust method described in section 7. However, this approach is not directly applicable when the separability assumptions are violated, which occurs, e.g., when higher order models (such as Balance models, e.g., [6]) are employed for large-scale flow modeling. Hence, we look for a more general approach.

3. A 2D model problem and its discretization. We simplify the analysis and development of the multigrid solver by first considering a 2D, constant-coefficient (a and g) problem that models (1)–(4), while concentrating on methods that will generalize to the complete problem. With no further loss of generality we assume $a = 1$. The model problem is obtained by eliminating the y coordinate:

$$(12) \quad Lu = u_{xx} + g u_{zz} = f(x, z), \quad (x, z) \in (-W, W) \times (0, D),$$

$$(13) \quad u_z(x, 0) = u_z(x, D) = 0, \quad x \in (-W, W);$$

$$(14) \quad u(-W, z) = u(W, z) = U(z), \quad z \in [0, D],$$

$$(15) \quad \langle u \rangle(z) \stackrel{\text{def}}{=} \frac{1}{2W} \int_{-W}^W u(x, z) \, dx = F(z).$$

3.1. The discrete model. We discretize (12)–(15) by standard second-order central finite differences on a grid of horizontal meshsize $h_x = W/N_x$ and vertical meshsize $h_z = D/N_z$. The discrete variables, $u_{i,k}^h$, $i = -N_x, \dots, N_x$, $k = 1, \dots, N_z$, are located at centers of vertical edges, which is convenient for Neumann boundary conditions at the top and bottom. Thus, $u_{i,k}^h$ is located at $(ih_x, (k - \frac{1}{2})h_z)$. In particular, $u^h(-N_x, k)$ and $u^h(N_x, k)$ are defined on the side boundaries, but the top and bottom boundaries lie one half of a meshsize below $u_{i,1}^h$ and above u_{i,N_z}^h , respectively. Then, “ghost” variables which satisfy the Neumann boundary conditions can be defined at one half of a meshsize below and above the bottom and top boundaries, respectively, as is commonly done.

The integral in (15) is discretized by the trapezoidal rule as follows:

$$(16) \quad \langle u^h \rangle_k = \frac{1}{4N_x} \sum_{i=-N_x}^{N_x-1} (u_{i,k}^h + u_{i+1,k}^h), \quad k = 1, \dots, N_z.$$

We write the discrete problem as

$$(17) \quad \begin{aligned} L^h u_{i,k}^h &= f_{i,k}^h, & i &= -N_x + 1, \dots, N_x - 1, \\ & & k &= 1, \dots, N_z, \\ u_{i,0}^h &= u_{i,1}^h, \quad u_{i,N_z+1}^h = u_{i,N_z}^h, & i &= -N_x + 1, \dots, N_x - 1, \\ u_{-N_x,k}^h &= u_{N_x,k}^h = U_k^h, & k &= 1, \dots, N_z, \end{aligned}$$

with the integral constraint

$$(18) \quad \langle u^h \rangle_k = F_k^h, \quad k = 1, \dots, N_z.$$

Here, h superscripts denote discrete approximations. The functions f^h and F^h , and the constant g (which appears in L^h), are given, but U^h is unknown.

4. A “naive” algorithm. Assume that, given U^h (constituting Dirichlet side-boundary conditions), there exists an efficient algorithm for solving the model problem, (17), e.g., standard multigrid. Given this, a simple approach to the solution of (17), (18) is to alternately satisfy (17) and (18): the former by fixing the boundary values and using the standard solver and the latter by subtracting the horizontal average from the current approximation for u^h at each height-level k , including the side-boundary values, and adding F_k^h . Indeed, since this is followed by a repetition of the first step, where the interior values are recalculated, it suffices to change only the boundary values in the second step. Thus, the two-stage iteration, which is straightforward enough to apply to more general problems as well, is given by the following.

ALGORITHM 1. Choose an initial guess for the side-boundary values, $U_{(0)}^h$. Then iterate for $m = 0, 1, \dots$, as follows, until a suitable convergence criterion is satisfied.

1. Solve

$$(19) \quad L^h u_{(m)}^h = f^h,$$

with side-boundary conditions given by $U_{(m)}^h$ and, of course, homogeneous Neumann top and bottom boundary conditions.

2. Set

$$(20) \quad U_{(m+1)}^h = U_{(m)}^h + F^h - \langle u_{(m)}^h \rangle,$$

$m \leftarrow m + 1$; return to Step 1.

Convergence analysis. To analyze the convergence properties of Algorithm 1 we first note that we can expand $u^h, U^h, f^h,$ and F^h of (17) and (18) in *cosine* series in $z,$ viz.,

$$(21) \quad u_{i,k}^h = \sum_{\omega=0}^{N_z-1} \hat{u}_i^h(\omega) \cos\left(\frac{\omega k \pi}{N_z}\right), \quad k = 1, \dots, N_z,$$

with analogous expressions for $f^h, F^h,$ and $U^h.$ Accordingly, we substitute these expressions into (17) and (18), with L^h representing a standard second-order finite difference discretization of $L.$ Since the *cosine* functions are orthogonal eigenfunctions (with the standard scalar product) of the discrete vertical-derivative term in $L^h,$ which satisfy the top and bottom boundary conditions, it suffices to consider a single representative $\omega,$ as is usually done in Fourier smoothing analysis of multigrid algorithms. This simplifies the analysis.

Substituting the corresponding eigenvalue for the discrete vertical-derivative term, (17), (18) then produces

$$(22) \quad \frac{\hat{u}_{i+1}^h - 2\hat{u}_i^h + \hat{u}_{i-1}^h}{h_x^2} - \frac{4g \sin^2\left(\frac{\omega\pi}{2N_z}\right) \hat{u}_i^h}{h_z^2} = \hat{f}^h, \quad i = -N_x + 1, \dots, N_x - 1,$$

$$\hat{u}_{-N_x}^h = \hat{u}_{N_x}^h = \hat{U}^h,$$

with the integral constraint

$$(23) \quad \langle \hat{u}^h \rangle = \hat{F}^h.$$

Here, $\hat{u}^h, \hat{f}^h, \hat{F}^h,$ and \hat{U}^h depend on $\omega.$ Similarly, we can expand (19), (20), obtaining analogous equations for $\hat{u}_{(m)}^h$ and $\hat{U}_{(m)}^h.$ Denote the errors in the interior solution and in the boundary values, respectively, after step 1 of the m th iteration of Algorithm 1 by $\hat{e}_{(m)}^h(\omega) = \hat{u}^h - \hat{u}_{(m)}^h$ and $\hat{E}_{(m)}^h(\omega) = \hat{U}^h - \hat{U}_{(m)}^h.$ Subtracting the transformed (19) from (22), we obtain

$$(24) \quad \frac{\hat{e}_{i+1}^h(m) - 2\hat{e}_i^h(m) + \hat{e}_{i-1}^h(m)}{h_x^2} - \frac{4g \sin^2\left(\frac{\omega\pi}{2N_z}\right) \hat{e}_i^h(m)}{h_z^2} = 0, \quad i = -N_x + 1, \dots, N_x - 1,$$

$$\hat{e}_{-N_x}^h(m) = \hat{e}_{N_x}^h(m) = \hat{E}_{(m)}^h.$$

The solution to (24) can be written as

$$(25) \quad \hat{e}_i^h(m) = \hat{E}_{(m)}^h \frac{\cosh(\alpha i)}{\cosh(\alpha N_x)}, \quad i = -N_x, \dots, N_x,$$

where α is determined (via substitution into (24)) by the following relation:

$$(26) \quad \sinh\left(\frac{\alpha}{2}\right) = \pm\sqrt{C} \sin\left(\frac{\omega\pi}{2N_z}\right),$$

with

$$(27) \quad C = g \frac{h_x^2}{h_z^2}.$$

Next, we calculate the horizontally averaged $\hat{e}_{(m)}^h$ from (16), (25). In the special case where $\alpha = 0$ (and hence $\omega = 0$, in light of the range of ω in (21)), the horizontal average equals the boundary value of the error, $\hat{E}_{(m)}^h$, and Algorithm 1 converges immediately. Therefore, we assume $\alpha \neq 0$ and obtain

$$\begin{aligned} \langle \hat{e}_{(m)}^h \rangle &= \frac{\hat{E}_{(m)}^h}{2N_x} \left[1 + \frac{1}{\cosh(\alpha N_x)} \sum_{i=-N_x+1}^{N_x-1} \cosh(\alpha i) \right] \\ (28) \qquad &= \frac{\hat{E}_{(m)}^h}{2N_x} \left[1 + \frac{1}{\cosh(\alpha N_x)} \frac{\sinh(\alpha(N_x - \frac{1}{2}))}{\sinh(\frac{\alpha}{2})} \right]. \end{aligned}$$

Hence, by (20) and (28), the *convergence factor* of Algorithm 1 is given by

$$(29) \quad \lambda \stackrel{\text{def}}{=} \left| \frac{\hat{E}_{(m+1)}^h}{\hat{E}_{(m)}^h} \right| = 1 - \frac{1}{2N_x} \left[1 + \frac{1}{\cosh(\alpha N_x)} \frac{\sinh(\alpha(N_x - \frac{1}{2}))}{\sinh(\frac{\alpha}{2})} \right].$$

Recall that we are analyzing the convergence of each mode separately. Hence, ω is fixed, and α , which corresponds to it, is determined uniquely by (26). From (28), (29) we immediately obtain the following.

OBSERVATION 1. Algorithm 1 is convergent, with convergence factor $\lambda = 0$ for the mode $\omega = 0$, and

$$0 < \lambda < 1 - \frac{1}{2N_x},$$

otherwise.

Convergence of Algorithm 1 can similarly be proved with $g = g(z)$ as well. In any case, the rate of convergence depends on the smoothness of the error in the boundary values. For a smooth error, $\omega = O(1)$, we obtain, by (26), $\alpha \sim C/N_z \ll 1$ (for typical values of N_z), and λ is then bounded away from 1. But λ grows monotonically with α (hence, with ω). For highly oscillatory boundary errors, $\alpha \sim C$, and (29) evidently implies $\lambda = 1 - O(\frac{1}{N_x})$, and hence $O(N_x)$ iterations are generally required to reduce the error by an order of magnitude. The slow convergence is expected: in an elliptic problem the influence of highly oscillatory boundary data decays very fast away from the boundary. Therefore, such error components contribute very little to the integral of the solution. Thus, step 2 in Algorithm 1 cannot be efficient for such types of errors. The convergence rate can be improved somewhat by judicious overcorrection (similar to SOR) or more sophisticated acceleration methods, but the algorithm would still be quite inefficient.

5. An efficient algorithm. The analysis of Algorithm 1 in section 4 has shown that this approach is inefficient. However, it also points the way to a far better approach, since we encounter the familiar setting that is the basis of multigrid methods: The effect of high-frequency boundary data decays very fast away from the boundary, so it can be treated locally. Low-frequency phenomena must be treated globally; however, they can be approximated on coarser grids.

Again, we consider (18) as a set of equations for the boundary values, U^h , and we treat the integral relations by introducing changes in the boundary data, referred to below as relaxation of the boundary values. But the boundary-data processing algorithm is incorporated within the multigrid algorithm which solves the interior

problem. The two-level algorithm with ν_1 prerelaxation sweeps, ν_2 postrelaxation sweeps, and ν_b boundary-relaxation sweeps per interior sweep is now described.

ALGORITHM 2.

1. Repeat ν_1 times{
 - Relax fine-grid interior problem, (17), to smooth interior error.
 - Repeat ν_b times{
 - Relax boundary equations, (18), e.g., by (30), discussed below.
 - Relax (17) near boundary (all variables that are at most some distance w_b meshsizes from any side boundary) to reduce interior residuals caused by the boundary relaxation.
2. Transfer to the coarse grid the residuals of the interior problem *and* of the integral relations. The latter are defined by

$$R^h = F^h - \langle \tilde{u}^h \rangle,$$

where $\tilde{\cdot}$ denotes current approximation.

3. Solve coarse-grid problem, including the coarse-grid residual integral-constraint problem (see below).
4. Prolong and add coarse-grid correction to the fine-grid field, including the boundary values.
5. Repeat step 1, substituting ν_2 for ν_1 .

Algorithm 2 is generalized to many levels (grids) in the usual V-cycle manner, by applying it recursively to solve the coarse-grid problem (step 3). Below, we shall also refer to the multilevel version as Algorithm 2.

Remarks on Algorithm 2. There are several choices to make, and we also remark on some important considerations. However, we first expound upon the main thought behind Algorithm 2. We distinguish between errors associated with nonzero residuals in the interior equations (corresponding to the differential equation in the continuous problem) and errors associated with nonsatisfaction of the integral relations. In both cases, “errors” refer to the difference between the current approximation and the exact solution of the discrete system. In the usual spirit of multigrid analysis (rigorous in special cases and heuristic otherwise), we refer to smooth and nonsmooth error (Fourier) components. It is well known and understood that nonsmooth errors of the interior problem (i.e., those that vary appreciably over just a few mesh-intervals) may be reduced by local processes, such as Gauss–Seidel relaxation. However, smooth components, by definition, cannot even be detected locally, so local processes are inefficient for such error components. However, smooth error components can be approximated well on a coarser grid. Such is the underlying mechanism of the usual multigrid algorithm, and we employ the same idea in treating the integral relations in Algorithm 2. The idea is to eliminate nonsmooth components of the errors in the integral relations (which is a 1D vector) by performing local processing at the lateral boundaries. There are various effective ways for doing this, as long as the remaining error in the integral relations can be represented accurately on the coarser grid. Note that there is no need to *eliminate* the error in the integral relations on a given grid—only to reduce its nonsmooth part—as the remaining error can be corrected on a coarser grid. Indeed, a smooth error of amplitude E^h in the boundary data produces, by (28), a boundary residual of similar size, whose elimination requires, by (16), an $O(N_x E^h)$ change in the boundary values to eliminate. This produces a large

relative change in the residuals of the interior problem near the boundary. Thus, the simple but somewhat heavy-handed approach of changing boundary values so as to eliminate the residuals in the integral relations might not be robust, e.g., in problems with geometrically complicated boundaries. Nevertheless, in all our numerical tests, which are carried out in simple domains, we found that similar efficiency is obtained if we eliminate the residuals in the boundary equations altogether in the boundary-relaxation process, rather than just smooth them. This is the simplest approach, and we use it in our tests below. By (16), the boundary relaxation is thus defined by

$$(30) \quad \tilde{U}_{new}^h = \tilde{U}^h + 2N_x(F^h - \langle \tilde{u}^h \rangle).$$

Compare (30) to (20)—the second step in Algorithm 1. Here, we introduce a change in the boundary values that is $2N_x$ times as large. If we were to do so in Algorithm 1, we would get very fast *divergence* (due to overcorrected smooth error modes in the boundary equations). However, in Algorithm 2 we do not follow this step with a solution of the interior problem, but rather just local relaxation. This serves to eliminate the interior errors due to high-frequency boundary errors and also to allow the coarser grids to correct the interior errors due to smooth boundary errors. Furthermore, we found that a smoother convergence history was obtained if, instead of transferring the residuals of the boundary equations, we eliminated them just before going to the coarse grid by repeating (30) (not followed by additional interior relaxation) at no extra cost (because now the boundary residuals need not be transferred). We use this form in the numerical examples reported below.

Following are some further details relating to Algorithm 2.

1. For ν_1 and ν_2 the typical values can be used, e.g., $0 \leq \nu_1, \nu_2 \leq 2$, $2 \leq \nu_1 + \nu_2 \leq 4$.
2. Normally, we found $\nu_b = 2$ with $w_b = 1$ to suffice in order to regain the typical convergence rate for the basic problem with Dirichlet boundary conditions.
3. We used Gauss–Seidel in red-black ordering (RBGS), also with overrelaxation (RBSOR), for relaxing the interior problem whenever g was moderate (between 0.2 and 10 in our examples). More extreme values of g are discussed later.
4. Recalling that the coarse grid variables are vertically staggered with respect to the fine grid variables, we have a choice of elementary vertical prolongation (coarse-to-fine) and restriction (fine-to-coarse) operators. In the horizontal direction we used linear interpolation (prolongation) and its suitably scaled adjoint full-weighting restriction operator. In the vertical we used nearest-neighbor averaging for restriction. For prolongation we tested both the adjoint of this operator and linear interpolation, because the so-called high-frequency order of the nearest-neighbor operator is one (compared to three of the linear vertical interpolation), which is marginal for this problem (see, e.g., [1, 2, 5, 11]). The operators are shown in Figure 1 in terms of the weights associated with the contributions that the coarse-grid variables give to (get from) the neighboring fine-grid variables.
5. Let u^H denote the coarse-grid correction field. Then the coarse-grid integral-constraint equation is

$$\langle u^H \rangle = F^H \stackrel{\text{def}}{=} I_h^H R^h,$$

where I_h^H denotes the restriction to the coarse grid from the fine grid (with meshsizes H and h , respectively.)

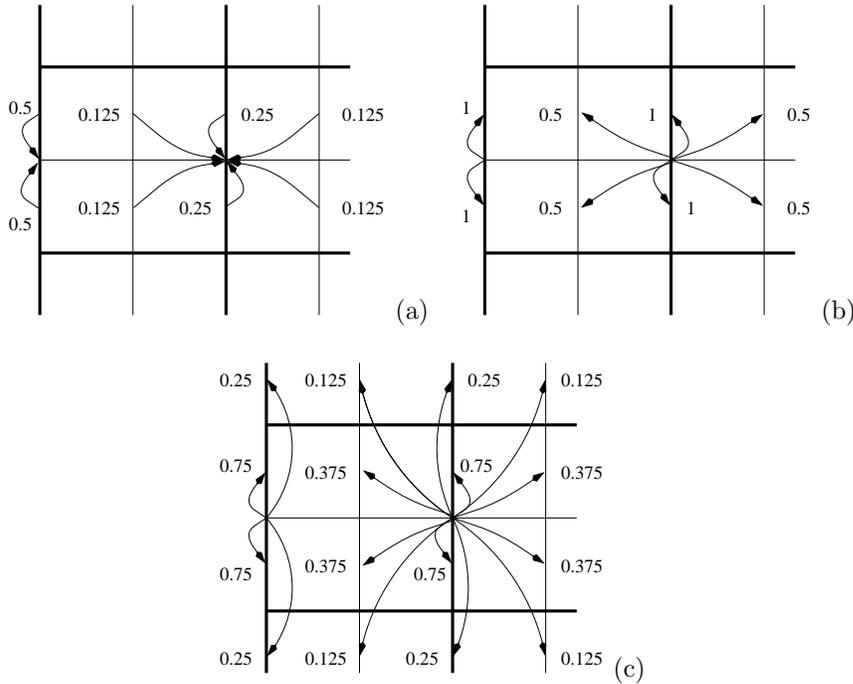


FIG. 1. 2D stencils are shown of the fine-to-coarse restriction operator (a) and the coarse-to-fine prolongation operators employing nearest-neighbor values (b) and linear interpolation (c) in the vertical, both for the domain interior and for the (left-hand) boundary. For the restriction operator arrow-sources mark fine-grid variable locations, while arrow-heads point at the coarse-grid variable (and vice versa for the prolongation operators). The numbers indicate weights associated with the values that the coarse-grid variable contributes to (or obtains from) the fine-grid variables.

6. The computational cost of each boundary relaxation sweep is dominated by the cost of calculating the residuals in the integral relations, since these require $O(N_x N_z)$ operations, while the remaining parts of the boundary process cost only $O(N_z)$ operations. Furthermore, even if several consecutive sweeps are performed, we need only to calculate these residuals once, and then monitor only the changes brought about by the relaxation at and near the boundaries. Thus, the added cost involved in the boundary process is negligible provided that only a moderate number of boundary sweeps is performed.

6. Numerical tests in three dimensions. We test Algorithm 2 for the 3D problem. In all our tests the domain of the solution is a cube and the mesh is uniform on the finest grid. Hence, $C = g$ in (27). We compare the convergence factor obtained for the 3D *model problem*, which is the generalization of (17), (18) to three dimensions, to that of the 3D *basic problem*, which is the 3D generalization of just (17) with U^h given (i.e., Dirichlet side-boundary conditions). In solving the basic problem we also applied two additional relaxation sweeps at points adjacent to the lateral boundaries after each internal relaxation sweep. This only improved the convergence rates by at most a few percent in all the cases tested, but it allowed a “fair” comparison. (There is a very slight loss of performance if the extra relaxation is not performed, presumably due to the weak singularity where the lateral Dirichlet boundaries meet the top and bottom Neumann boundaries.)

TABLE 1

Comparison of asymptotic convergence factors per cycle between the 3D basic problem (with Dirichlet side-boundary conditions) and the 3D model problem (with integral constraints). The grid is uniform in a cube, and $C = 1$. Smoothing is done by RBGS relaxation.

ν_1	ν_2	Vertical Prolongation	3D Basic Problem	3D Model Problem
1	1	Nearest	0.558	0.480
2	0	Nearest	0.415	0.367
2	1	Nearest	0.166	0.153
2	2	Nearest	0.098	0.091
1	1	Linear	0.218	0.212
2	0	Linear	0.239	0.227
2	1	Linear	0.131	0.131
2	2	Linear	0.102	0.102

We are interested in worst-case asymptotic convergence rates. Hence, in order to avoid roundoff errors, we set $f^h = F^h = 0$ (so that the exact solution is zero and the errors are therefore never very small compared to the solution), and begin with a random guess for U^h . For testing purposes we also performed calculations with random f^h and F^H and found essentially the same performance until double-precision roundoff errors were encountered. We then run 25 cycles, ignoring the first 10 and defining the asymptotic convergence factor (per cycle) as the geometric average of the residual-reduction factor of the last 15 cycles. Several values of pre- and postrelaxation sweeps, ν_1 and ν_2 , respectively, are tested.

Isotropic problem. We begin with the isotropic problem, $C = 1$, where C is defined in (27). The generalization of the discretization and the algorithm to three dimensions is completely straightforward. We use six levels, coarsening down to two by two by two mesh-intervals. In three dimensions the best relaxation is RBSOR. Following [10], the optimal overrelaxation parameter (assuming for simplicity $h_x = h_y$) is, to a good approximation, given by

$$(31) \quad \eta = \frac{2}{1 + \sqrt{1 - s^2}},$$

where $s = \max(2, 1 + C)/(2 + C)$. In the isotropic case this yields $\eta = 1.146$ (compared to $\eta = 1$ of RBGS). We compare the convergence factor of Algorithm 2 to that of the multigrid solution to the basic problem with and without overrelaxation in Tables 1 and 2, respectively. We use $\nu_b = 2$ and $w_b = 1$. Evidently, the efficiency of Algorithm 2 is maintained in the isotropic case, even when SOR is used. There are a few exceptions, but $\nu_b = 3$ always sufficed. We find that linear vertical interpolation generally performs better than nearest-neighbor interpolation, and we use this interpolation in the remaining tests.

Moderately anisotropic problem. We next test our solver on moderately anisotropic problems, with C equal to 10 and 0.2. The results appear in Table 3. We find that for relatively large C the performance for the model problem with boundary constraints fully matches that of the basic problem. When SOR smoothing is used, this performance is quite adequate. However, for relatively small C the method fails for the model problem. The reason for this is that when C is decreased, highly oscillatory boundary data influence the interior solution deeper into the domain. One way of seeing this is by considering the reduction in C as a reduction in the horizontal meshsize, with h_z and g fixed. Given a smaller horizontal meshsize, a given function requires

TABLE 2

Same as Table 1, but smoothing is done by RBSOR, with overrelaxation parameter calculated from (31). The values in parentheses are obtained with $\nu_b = 3$.

ν_1	ν_2	Vertical Prolongation	3D Basic Problem	3D Model Problem
1	1	Nearest	0.187	0.185
2	0	Nearest	0.206	0.184
2	1	Nearest	0.091	0.117 (0.077)
2	2	Nearest	0.063	0.097 (0.059)
1	1	Linear	0.111	0.111
2	0	Linear	0.173	0.176
2	1	Linear	0.078	0.114 (0.075)
2	2	Linear	0.060	0.097 (0.059)

TABLE 3

Comparison of asymptotic convergence factors per cycle between the 3D basic problem (with Dirichlet side-boundary conditions) and the 3D model problem (with integral constraints). The grid is uniform in a cube, and linear vertical interpolation is used in the prolongation. $\nu_b = 2$ and $w_b = 1$, except the values in parentheses, for which $\nu_b = w_b = 3$. DIV means that the algorithm failed to converge.

ν_1	ν_2	Smoothing	C	3D Basic Problem	3D Model Problem
1	1	RBSOR	10	0.671	0.671
2	0	RBSOR	10	0.671	0.671
2	1	RBSOR	10	0.564	0.564
2	2	RBSOR	10	0.474	0.474
1	1	RBSOR	10	0.393	0.392
2	0	RBSOR	10	0.391	0.391
2	1	RBSOR	10	0.211	0.210
2	2	RBSOR	10	0.112	0.109
1	1	RBSOR	0.2	0.648	0.647
2	0	RBSOR	0.2	0.648	0.646
2	1	RBSOR	0.2	0.535	0.533
2	2	RBSOR	0.2	0.443	0.440
1	1	RBSOR	0.2	0.355	0.510 (0.331)
2	0	RBSOR	0.2	0.364	0.754 (0.344)
2	1	RBSOR	0.2	0.179	DIV (0.145)
2	2	RBSOR	0.2	0.085	0.931 (0.236)

more mesh-intervals to decay away from the boundary. Alternatively, note that α in (26) in the 2D case (which is straightforward to generalize to 3D) decreases as C is decreased, resulting, by (25), in an error which decays more slowly away from the boundary. Indeed, if we increase w_b —the width of the strip near the boundary where additional processing is performed—we regain good behavior (see values in parentheses in Table 3). Of course, the width of this strip must increase even more as C is decreased, so this approach is not recommended for anisotropic problems. Instead, we prefer not to coarsen in the vertical, employing so-called semicoarsening. Then, the coarse grids resolve all components of the fine-grid boundary data, so each grid need only converge boundary-errors whose influence decays on a scale of just a few horizontal mesh-intervals. This approach is described next.

7. A robust algorithm. We have seen that the basic form of Algorithm 2 is not robust due to the fact that for small C , as defined in (27), the width of the interior region near the boundary which requires extra processing grows. Also, it is well known

TABLE 4

Same as Table 1, employing zebra relaxation in the vertical and coarsening only in the horizontal.

ν_1	ν_2	C	3D Basic Problem	3D Model Problem
1	1	100	0.104	0.086
2	1	100	0.077	0.079
1	1	1	0.102	0.088
2	1	1	0.077	0.079
1	1	0.01	0.100	0.092
2	1	0.01	0.077	0.079
1	1	$50 + 49.99 \sin(2\pi z/D)$	0.103	0.088
2	1	$50 + 49.99 \sin(2\pi z/D)$	0.077	0.079

that point relaxation is not an effective smoother for very anisotropic problems (see, e.g., [8] or [1], and also [9, 10] for a quantitative assessment). We treat both these difficulties by employing red-black (zebra) line relaxation in the vertical, with (semi) coarsening only in the horizontal directions. For the basic problem, we then expect the convergence rates to be nearly independent of C and to be similar to those obtained for the basic 2D Poisson problem; (this is shown by Fourier smoothing analysis in [9]). But this also resolves the lack of robustness associated with the integral-constraint equations. Since we do not coarsen in the vertical, all boundary-error frequencies are resolved (vertically) on all the grids. Those frequencies whose influence on the interior solution decays over a small number of meshsizes are eliminated by the boundary relaxation, followed by interior relaxation along a thin strip near the side boundaries (again we find that this strip need only include the interior points adjacent to the side boundaries, i.e., $w_b = 1$.) The remaining error frequencies are converged on coarser grids.

In Table 4 we show the results of calculations on a uniform cubic grid with values of C ranging from 0.01 to 100, including a case where a variable C spans this range. The performance is, as expected, comparable to the 2D basic problem, virtually independent of C , and the performance of the model problem (with integral constraints) is as good as that of the basic problem, using as before $\nu_b = 2$, $w_b = 1$.

The additional cost due to employing semicoarsening is just a small fraction. Line relaxation is, however, somewhat more expensive than point relaxation. Hence, for nearly isotropic problems point relaxation and full coarsening are more efficient.

8. Conclusions. An efficient multigrid approach is presented for the solution of a discretized elliptic problem with boundary conditions determined in part by integral relations. The approach is to treat the integral relations as equations for the boundary values. These are relaxed after each relaxation sweep of the interior problem, followed by additional relaxation of the equations at interior points adjacent to the boundary. This boundary process reduces errors in the integral relations that are associated with high-frequency boundary data. The remaining errors in the integral relations are resolved on coarser grids. For anisotropic problems an efficient and robust approach is to apply the same method while coarsening only in the horizontal and using line relaxation in the vertical for the interior problem. The additional cost involved in these solvers, compared to similar solvers for the basic problem with standard boundary conditions, is essentially just the cost of calculating the integral relations. Furthermore, the convergence rates are just as good, consistent with the general theory of [2], which claims that the “interior efficiency” can be obtained for

many types of boundary conditions by performing extra work that is asymptotically negligible near boundaries. Hence, this approach can be considered optimal.

We believe that our approach will be equally effective for more general boundary shapes, though the boundary relaxation may require greater subtlety. We shall investigate this matter in future work when we apply this method to models of large-scale ocean flows in realistic domains.

REFERENCES

- [1] A. BRANDT, *Multigrid Techniques: 1984 with Applications to Fluid Dynamics*, GMD-Studien 85, Gesellschaft für Mathematik und Datenverarbeitung m.b.H., Bonn, St. Augustin, West Germany, 1984.
- [2] A. BRANDT, *Rigorous quantitative analysis of multigrid, I: Constant coefficients two-level cycle with L_2 -norm*, SIAM J. Numer. Anal., 31 (1994), pp. 1695–1730.
- [3] R. COURANT AND D. HILBERT, *Methods of mathematical physics*, Vol. I, Interscience Publishers, New York, 1953.
- [4] E. HAIRER, S. P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I. Non-stiff Problems*, 2nd ed., Springer-Verlag, Berlin, Heidelberg, 1993.
- [5] P. W. HEMKER, *On the order of prolongations and restrictions in multigrid procedures*, J. Comput. Appl. Math., 32 (1990), pp. 423–429.
- [6] J. C. MCWILLIAMS, N. J. NORTON, P. R. GENT, AND D. B. HAIDVOGEL, *A linear balance model of wind-driven, midlatitude ocean circulation*, J. Phys. Oceanogr., 20 (1990), pp. 1349–1378.
- [7] J. PEDLOSKY, *Geophysical Fluid Dynamics*, Springer, New York, 1979.
- [8] K. STÜBEN AND U. TROTTEBERG, *Multigrid methods: Fundamental algorithms, model problem analysis and application*, in *Multigrid Methods*, W. Hackbusch and U. Trottenberg, eds., Springer-Verlag, Berlin, 1982, pp. 1–176.
- [9] I. YAVNEH, *Multigrid smoothing factors of red-black Gauss–Seidel Relaxation applied to a class of elliptic operators*, SIAM J. Numer. Anal., 32 (1995), pp. 1126–1138.
- [10] I. YAVNEH, *On red-black SOR smoothing in multigrid*, SIAM J. Sci. Comput., 17 (1996), pp. 180–192.
- [11] I. YAVNEH, *Coarse-grid correction for nonelliptic and singular perturbation problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1682–1699.