JOURNAL OF COMPUTATIONAL PHYSICS **48**, 412–422 (1982)

# Unigrid Methods for Boundary Value Problems with Nonrectangular Domains

W. HOLLAND

*National Center for Atmospheric Research, Boulder, Colorado*

AND

S. MCCORMICK AND J. RUGE

*Colorado State University, Fort Collins, Colorado 80523*

Received January 15, 1982

This paper illustrates the use of the so-called unigrid method as a multigrid simulator by applying it to an oceanographic problem with an irregular domain.

## 1. INTRODUCTION

Multigrid methods are generally very effective for solving differential boundary value problems. This is true because the smooth error, which is slow to converge during relaxation, is reduced by iterating on the problem projected onto coarser grids, where relaxation is both cheaper and more efficient. Fine grid relaxation can then be viewed as an attempt to eliminate the high frequency error.

In lieu of coarse grid iterations, one can, in fact, modify the fine grid relaxation process in order to reduce the smooth error directly on the fine grid (i.e., without the use of coarser grids at all). Under certain assumptions (see Section 3), the resulting methods, so-called *unigrid* [3], is theoretically equivalent to conventional multigrid in the sense that it produces the same results in exact arithmetic. Unigrid has different computational characteristics than multigrid, however, since it requires less storage and shorter code, but significantly more arithmetic work. More importantly, it is much easier to apply to a given problem because most of the design work for the grid transfers and coarse grid operators is automatic. Thus, existing software packages that solve possibly very complex problems by SOR, for example, can be easily modified for application of unigrid. This can usually be done by making a few changes in the relaxation routine without impacting any of the other software routines or data structures. These features make unigrid effective as a multigrid software simulator for quick and easy determination of the applicability of multigrid to a given problem.

Unigrid is developed in Section 2, its relationship to multigrid is described in

0021-9991/82/120412-11$02.00/0

Section 3, some simple theory is presented in Section 4, and its use is illustrated with a North Atlantic basin oceanographic model problem in Section 5. This application demonstrates how unigrid (ənd, hence, multigrid) can be used efficiently with vector computers on problems with irregular domains.

## 2. Unigrid

Assume given the $d$-dimensional operator equation

$$AU = F, \qquad U \in \mathscr{H}_1, \tag{2.1}$$

where $A: \mathscr{H}_1 \to \mathscr{H}_2$ is a linear operator and $\mathscr{H}_1$ and $\mathscr{H}_2$ are appropriate Hilbert spaces of functions defined on a region $\Omega$ in $R^d$, $d \geqslant 2$. Assume that (2.1) admits discretizations by a family of matrix equations, parameterized by *admissible grid sizes* $h > 0$ and given by:

$$A^h U^h = f^h, \qquad U^h \in \mathscr{H}^h, \tag{2.2}$$

where $\mathscr{H}^h = R^{n^h}$ and $n^h$ is an integer (approximately proportional to $h^{-d}$). Upper case $U^h$ will denote the exact solution of (2.2) and lower case $u^h$ its approximation. The grid transfers are full rank linear operators, represented by $I_{\bar{h}}^h: \mathscr{H}^{\bar{h}} \to \mathscr{H}^h$, that satisfy the consistency condition $I_{\bar{h}}^h = I_{h'}^h I_{\bar{h}}^{h'}$ for admissible $\bar{h}$, $h'$, and $h$ when $\bar{h} < h' < h$ or $\bar{h} > h' > h$.

The objective is to reduce the error from a current approximation $u^h$ in the subspace defined by a set of directions $\mathscr{D}^h = (d_1, d_2, \ldots, d_n) \subset \mathscr{H}^h$. Letting $D^h = [d_1, d_2, \ldots, d_n]$, then a Ritz projection can be performed that corrects $u^h$ by a function in the space of $\mathscr{D}^h$, so that the projection of the resulting residual over the subspace is zero. This leads to the problem of finding some $s = (s_1, \ldots, s_n)^{\mathrm{T}}$ so that

$$D^{h^{\mathrm{T}}}[A^h(u^h + D^h s) - f^h] = 0.$$

This can be rewritten as

$$D^{h^{\mathrm{T}}} A^h D^h s = D^{h^{\mathrm{T}}}[f^h - A^h u^h].$$

Gauss–Seidel relaxation on this system with some initial approximation $s$ and a new approximation $\bar{s}$ can be written as

$$\begin{aligned} \bar{s}_i = \Big(f^h - A^h u^h - \sum_{j>i} s_j A^h d_j \\ - \sum_{j<i} \bar{s}_j A^h d_j, d_i\Big) / (A^h d_i, d_i), \qquad i = 1, 2, \ldots, n^h. \end{aligned}$$

Then $u^h$ can be corrected by

$$u^h \leftarrow u^h + D^h \bar{s}.$$

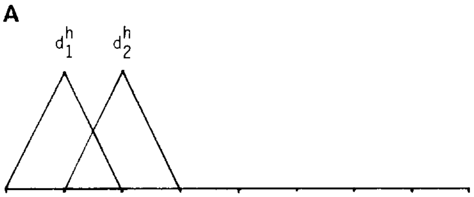

FIGURE 1A

Since $A^h$ is linear, then corrections can be made to $u^h$ directly, rather than to $s$, resulting in the *directional iteration*

$$u^h \leftarrow u^h + [(f^h - A^h u^h, d_i)/(A^h d_i, d_i)]\, d_i, \tag{2.3}$$

where left arrow denotes replacement. (Note that these are simply the basic steps of Gauss–Seidel applied to $s$.) Rewriting (2.3) as

$$u^h \leftarrow G^h(u^h, d), \tag{2.4}$$

then one *sweep* with initial guess $u^h$ consists of iterating with (2.4) in sequence over $d_k$, $k = 1, 2,..., n^h$. For example, Gauss–Seidel is specified by the choice $d_k = e_k^h$, the $k$th coordinate vector in $\mathscr{H}^h$.

To define unigrid for a given admissible grid size $h = H_0$, suppose $m \geqslant 1$ is an integer so that $H_q = 2^q H_0$ are admissible, $q \leqslant m$. Now define the direction sets for unigrid according to

$$\mathscr{D}_q^h = (d_1^{H_q}, d_2^{H_q},..., d_{n^{H_q}}^{H_q}), \qquad 0 \leqslant q \leqslant m, \tag{2.5}$$

where $d_k^{H_q} = I_{H_q}^h e_k^{H_q}$. Thus, the directions on *level* $q$ are just the relaxation directions on grid $H_q$ transferred to grid $h = H_0$.

One of the many possible unigrid schemes is described in terms of the relaxation parameters $\nu$ and $\nu_c$ and the cycling parameter $\mu$. The unigrid cycles are then defined recursively by: one unigrid cycle on level $q$ consists first of $\nu$ unigrid relaxation sweeps via (2.4) with directions $d_k^{H_q}$, $k = 1, 2,..., n^{H_q}$, followed for $q < m$ by $\mu$ cycles on level $q + 1$ and for $q = m$ by $\nu_c$ more sweeps via (2.4).

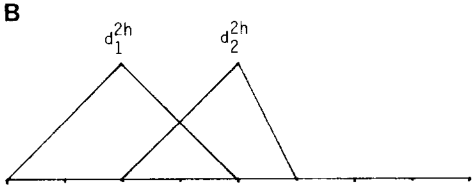

FIGURE 1B

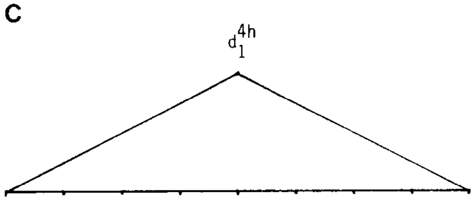

FIGURE 1C

*Remark.* The directions defining unigrid depend not directly on the operator $A$ but rather on the domain $\Omega$. Using linear interpolation, then these directions $d_i^{\bar{h}}$ are in fact the $i$th grid $\bar{h}$ coordinate vectors interpolated to grid $H_0$. In one dimension, this is illustrated by Figs. 1A–1C. For rectangular $\Omega$ in two dimensions, each direction is a product of two such functions, one in $x$ and the other in $y$, resulting in the "tent" function described as follows:

With the usual double subscript notation and $n^h = N^h \times N^h$, then $d_{k,l}^h = e_{k,l}^h$ for $h = H_0$, $1 \leqslant k, l \leqslant N^h$. The coarse grid directions are defined so that the $i$, $j$ component of $d_{k,l}^{H_q}$ is

$$\begin{aligned} d_{k,l}^{H_q}(i,j) &= (2^q - |k-i|)(2^q - |l-j|), && \text{if } \ |k-i|, \ \ |l-j| \leqslant 2^q, \\ &= 0, && \text{otherwise.} \end{aligned} \tag{2.7}$$

This assumes that the point denoted by $(k, l)$ is a point of the $H^q$ grid.

In irregular regions where boundaries do not lie on coarse grid lines, there are several options possible for treating these boundaries. The most obvious, which is analogous to the usual multigrid approach, is to define the directions as the interpolated coarse grid coordinate vectors and use the (zero) boundary conditions properly in interpolation. This is illustrated in one dimension by $d_2$ in Fig. 2. Note that this requires special handling of the coarse grid points that are adjacent to the boundary. Another approach is simply to ignore those directions which would overlap the boundary so that $d_2$ is suppressed as in Fig. 3. In Section 5, this will be referred to as the *contracted boundary* method. This means that some points near the boundary are not corrected by smooth error iterations, so the danger is that convergence is slowed (see Section 5).

Another possibility is to enlarge the region $\Omega$ so that it is aligned with the coarse grid directions, but ignore correcting that part of the expanded region $\Omega$ that does not

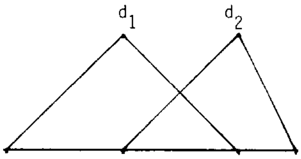

FIGURE 2

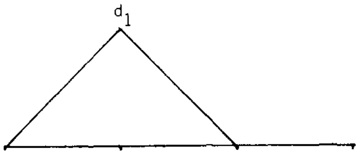

FIGURE 3

lie in the interior of $\Omega$. This is illustrated in Fig. 4. Note that this method, which here is called the *expanded boundary* approach, does not require extra information at the boundaries, so the directions can be computed once for each grid over the entire domain $\Omega$ and stored in the form of a matrix stencil.

## 3. MULTIGRID

### 3.1. *Conventional Multigrid*

One multigrid cycle on problem (2.2) with present approximation $u^h$, right-hand side $\bar{f}^h$, and $h = H_q$ is denoted by $\mathrm{MG}_h(u^h, \bar{f}^h)$ and defined recursively by:

(i) For $q = m$, $\mathrm{MG}_h(u^h, \bar{f}^h)$ consists of $\nu + \nu_c$ relaxation sweeps via (2.4) with directions $e_k^h$, $k = 1, 2,..., n^h$.

(ii) For $q < m$, $\mathrm{MG}_h(u^h, \bar{f}^h)$ consists of:

Step 1. Perform $\nu$ relaxation sweeps via (2.4) with directions $e_k^h$, $k = 1, 2,..., n^h$.

Step 2. Let $r^h = \bar{f}^h - A^h u^h$, $r^{2h} = I_h^{2h} r^h$, $u^{2h} \leftarrow 0$, and perform $\mu$ grid $2h$ cycles via $\mathrm{MG}_{2h}(u^{2h}, \bar{f}^{2h})$ with $\bar{f}^{2h} = r^{2h}$.

Step 3. Set $u^h \leftarrow u^h + I_{2h}^h u^{2h}$.

### 3.2. *Immediate Replacement Multigrid*

Multigrid is theoretically equivalent to unigrid if, as is henceforth assumed, the formulation of the coarser grid equations satisfies the *variational conditions*

$$A^{2h} = I_h^{2h} A^h I_{2h}^h \tag{3.1}$$

$$I_h^{2h} = \alpha^h (I_{2h}^h)^{\mathrm{T}}, \tag{3.2}$$

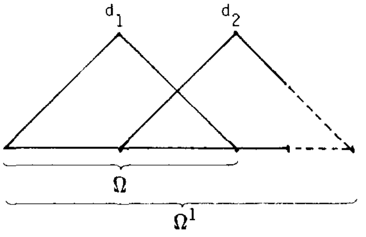

FIGURE 4

where $\alpha^h$ is a scalar. To see this, it is convenient to inroduce a modified multigrid algorithm that depends directly on the fine grid right-hand side $f^{H_0}$. Its cycles on grid $h = H_q$ are denoted by $\mathrm{MGIR}_q(f^0)$, where $q$ is used in place of $H_q$ as a subscript or superscript. It is characterized as a modification of conventional multigrid applied to (2.2) in which all coarse grid changes are immediately reflected in the fine grid approximation and the fine grid residual is recomputed and used to redefine the coarse grid equations. *Immediate replacement multigrid* is then defined in terms of $\mathrm{MGIR}_q(f^0)$ by:

(i) For $q = m$, $\mathrm{MGIR}_q(f^0)$ consists of performing $\nu + \nu_0$ relaxation sweeps via

$$u^0 \leftarrow u^0 + [\langle I_0^q(f^0 - A^0 u^0), e_k^{H_q}\rangle / \langle A^q e_k^{H_q}, e_k^{H_q}\rangle]\, I_q^0 e_k^{H_q},$$
$$k = 1, 2, \ldots, n^q.$$

(i) For $q < m$, $\mathrm{MGIR}_q(f^0)$ consists of $\nu$ relaxation sweeps via (3.3) followed by $\mu$ level $q-1$ cycles via $\mathrm{MGIR}_{q-1}(f^0)$.

Note that the immediate fine grid correction is incorporated in the relaxation scheme. This scheme on a level $q > 0$ is just (2.3) with $u^p = 0$, $0 < p \leqslant q$, and $r^q = I_0^q(f^0 - A^0 u^0)$, followed by interpolation of the correction directly to the finest grid.

### 3.3 *Theoretical Equivalence*

It is not difficult to see [3] that MGIR is fully equivalent to MG under Conditions (3.1), (3.2). This is done by noting what the status of intermediate MG calculations would be if coarse grid changes were immediately reflected in the fine grid approximation. By examining the iterative formulae, it is easy to see that MGIR and unigrid are identical, from which it follows that multigrid designed according to variational conditions (3.1) and (3.2) is theoretically equivalent to unigrid.

### 3.4. *Implementation Differences* (See [3] for more details.)

Unigrid code is typically very compact, partly because it lacks the modular structure of multigrid software. This is one reason that unigrid code can be developed very quickly. Also, there are fewer design choices with unigrid, since the coarse grid and grid transfer operators are automatically determined. This also adds to ease of programming, but restricts the flexibility of the method. The design of unigrid also guarantees convergence independently of the choice of the coarse level iteration directions and cycling scheme, so mistakes may slow convergence but do not result in divergence as often as for multigrid.

This ease of programming and small program size make unigrid an effective method to test the convergence behavior of multigrid for many application problems. For many existing codes, it is relatively easy to replace a conventional linear equation subroutine (especially if it is based on some form of relaxation) in order to perform such a feasibility test. Of course, the amount of work involved makes any comparison of solution time meaningless, but actual multigrid efficiency can be predicted by applying the usual multigrid operation counts to the unigrid cycling scheme. Since the methods are equivalent in terms of results when multigrid is implemented according to (3.1), (3.2), then unigrid will accurately represent the numerical performance of such a variationally formulated multigrid scheme.

## 4. Theory

Assuming that $A^h$ is symmetric and positive definite, define the *energy inner product* and *norm* on $H^h$ by

$$\langle x^h, y^h \rangle_{A^h} = \langle A^h x^h, y^h \rangle$$

and

$$\| x^h \|_{A^h} = \langle A^h x^h, x^h \rangle^{1/2},$$

respectively. Let $\mathscr{W}^h$ denote the set of all $A^h$-unit vectors in $\operatorname{span}\{w^h : A^h w^h = \mu w^h, \mu \leqslant \lambda\}$. Let $\mathscr{V}^h_\lambda$ denote the $A^h$-orthogonal complement of span $\mathscr{W}^h$ and let $\mathscr{G}^h$ denote one pass of (2.4) over the fine grid directions $\mathscr{D}^h$ which is assumed to span $\mathscr{H}^h$. For each integer $v \geqslant 1$, define $\mathscr{G}^h_{\lambda,v}$ as the restriction of $(A^h)^{-1/2}\,((G^h)^v)^{\mathrm{T}}\ A^h (G^h)^v\ (A^h)^{-1/2}$ to $V^h_\lambda$. (For Jacobi-type versions of (2.4), this latter operator simplifies to $(G^h)^{2v}$.) Then, with $2M$ the degree of the differential operator in (2.1), assume (cf., [2])

(A1) There exist constants $a > 0$ and $c_0 < \infty$ independent of $h$ and such that:

$$\| w^h - R(I^h_{2h}) \|^2_{A^h} \leqslant c_0 (\lambda h^{2M})^a$$

for all admissible $h$ and all $w^h \in \mathscr{W}^h_\lambda$, where $R(I^h_{2h})$ is the range of $I^h_{2h}$. (Note that $R(I^h_{2h}) = \operatorname{Span}(\mathscr{D}^h_1)$, the coarse grid directions.)

(A2) There exist constants $c_1, c_2 > 0$ with $c_1 < 1$ and $c_2 \leqslant (c_1 + 1)\, h^{-2M} / \rho(A^h)$, where $\rho(A^h)$ is the spectral radius of $A^h$, such that

$$\rho(\mathscr{G}^h_{\lambda,v}) \leqslant \max\{c_1^v, |1 - c_2 \lambda h^{2M}|^v\}.$$

Theorem. *Suppose $\mu > 1$ and $m$ and $v_0$ are such that the error from the coarsest level does not significantly contribute to the finest level error. Then there exists a $v$ independent of $h$ such that unigrid converges to the solution of* (2.2) *by a fixed linear rate independent of $h$.*

*Proof.* This theorem follows from the results of Section 3 that relate unigrid to multigrid and from the theory of [2] slightly modified to account for the class of relaxation methods depicted in (2.4).

Relaxation does not generally minimize the residual error, although it should approximately. In fact, when direct application of unigrid to (2.2) exhibits convergence but does not monotonically reduce the residual error on the coarse levels, this is a signal that the directions for relaxation are improperly defined. They should be chosen to approximate the *smooth* eigenvectors of $A^h$, that is, those that belong to the lower end of its spectrum. This would ensure that relaxation quickly eliminates the *oscillatory* eigenvector components of the error with little effect on the smooth ones. Since the spectrum of $A^h$ that corresponds to these oscillatory components is *relatively* narrow, there is a close relationship between error in the energy norm, for which relaxation is a minimizer, and the residual error norm. The residual norm is not generally *minimized* by relaxation, but a proper choice of directions coupled with a good smoothing rate ensures that it will be monotonically *reduced.*

## 5. Numerical Results

This section contains a report on numerical experiments with unigrid applied to the solution of the model problem

$$-\nabla^2 u + \lambda u = f \quad \text{in} \quad \Omega, \qquad u = g \quad \text{on} \quad \partial\Omega, \tag{5.1}$$

where $\Omega$ is an irregular domain used to describe the North Atlantic basin. In this case, $\Omega$ is rectangular on three sides but irregular on the fourth, as depicted by Fig. 5, and $\lambda$ is a given function which is set to the constant 64 in the following experiments. (Such a value for $\lambda$ results in strong positive definitiveness of the operator in (5.1), leading to very fast convergence rates for multigrid. Such a value, however, is fairly realistic for this application and sharply depicts the disadvantage of using the contracted boundary method described in Section 3.) In these experiments, the fine grid spacing is $h = 0.0625$ and the rectangle encompassing $\Omega$ is $|0, 3| \times |0, 2|$. In

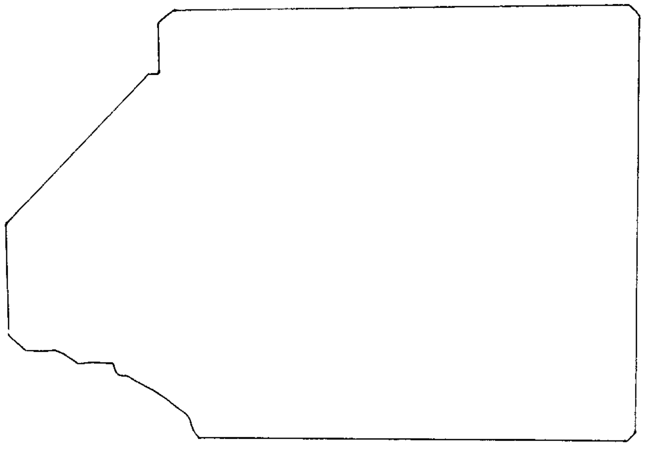

FIGURE 5

each case, a very simple grid cycling scheme with four grids is used, where each cycle involves three relaxations, each performed in turn on grids $8h$, $4h$, $2h$, and $h$. Four cycles are made for each of the three problems, with $u = 0$ as the initial guess. The usual central five-point stencil was used to discretize (5.1).

The main feature of the discretization of (5.1) is that the boundary is enforced to pass through grid $h$ vertices. Although this represents only an approximation to the actual boundary (of reduced order), it has conservative properties that are not easily obtained any other way. More specifically, conservation of kinetic energy, vorticity, and enstrophy in a dissipationless finite difference discretization of atmospheric diffusion problems can be easily guaranteed when the grid points and irregular boundary points coincide (cf., [4]). Although this is an advantage when used on a

TABLE I

| | | Dynamic residual error | | |
|---|---|---|---|---|
| Cycle number | Relaxation level | Multigrid on $[0,3] \times [0,2]$ | Contract multigrid on $\Omega$ | Expanded multigrid on $\Omega$ |
| 1 | $8h$ | 7.402E + 03 | 6.027E + 03 | 7.418E + 03 |
| | | 1.197E + 03 | 1.147E + 03 | 2.465E + 03 |
| | | 2.745E + 02 | 1.753E + 02 | 1.037E + 03 |
| | $4h$ | 2.617E + 03 | 2.384E + 03 | 2.364E + 03 |
| | | 7.758E + 01 | 8.864E + 01 | 2.598E + 02 |
| | | 3.416E + 00 | 3.927E + 00 | 2.187E + 02 |
| | $2h$ | 4.270E + 02 | 5.663E + 02 | 4.311E + 02 |
| | | 8.350E + 01 | 1.359E + 02 | 1.002E + 02 |
| | | 2.200E + 00 | 4.139E + 01 | 2.602E + 01 |
| | $h$ | 1.090E + 02 | 2.716E + 02 | 1.437E + 02 |
| | | 1.784E + 01 | 8.348E + 01 | 2.630E + 01 |
| | | 7.771E + 00 | 3.836E + 01 | 9.057E + 00 |
| 2 | $8h$ | 3.620E + 01 | 3.602E + 01 | 3.749E + 01 |
| | | 9.004E + 00 | 6.381E + 00 | 9.451E + 00 |
| | | 2.383E + 00 | 1.145E + 00 | 4.826E + 00 |
| | $4h$ | 3.236E + 01 | 3.607E + 01 | 3.166E + 01 |
| | | 8.235E − 01 | 1.383E + 00 | 4.491E + 00 |
| | | 3.198E − 02 | 6.425E − 02 | 2.348E + 00 |
| | $2h$ | 6.099E + 00 | 4.050E + 01 | 6.358E + 00 |
| | | 1.139E + 00 | 1.006E + 01 | 1.373E + 00 |
| | | 2.956E − 01 | 3.073E + 00 | 3.161E − 01 |
| | $h$ | 1.363E + 00 | 2.030E + 01 | 2.116E + 00 |
| | | 2.232E − 01 | 9.095E + 00 | 4.554E − 01 |
| | | 1.048E − 01 | 4.764E + 00 | 1.694E − 01 |

*Table continued*

TABLE I (*continued*)

| | | Dynamic residual error | | |
|---|---|---|---|---|
| Cycle number | Relaxation level | Multigrid on [0,3] × [0,2] | Contract multigrid on $\Omega$ | Expanded multigrid on $\Omega$ |
| 3 | 8*h* | 3.602E − 01 | 9.720E − 01 | 5.507E − 01 |
| | | 8.874E − 02 | 3.538E − 02 | 1.325E − 01 |
| | | 2.134E − 02 | 7.674E − 03 | 9.404E − 02 |
| | 4*h* | 3.599E − 01 | 3.644E + 00 | 4.807E − 01 |
| | | 8.122E − 03 | 1.276E − 01 | 1.743E − 01 |
| | | 3.198E − 04 | 6.083E − 03 | 1.064E − 01 |
| | 2*h* | 1.127E − 01 | 4.954E + 00 | 1.354E − 01 |
| | | 2.044E − 02 | 1.126E + 00 | 3.299E − 02 |
| | | 5.259E − 03 | 3.595E − 01 | 9.587E − 03 |
| | *h* | 2.329E − 02 | 2.687E + 00 | 4.209E − 02 |
| | | 4.590E − 03 | 1.272E + 00 | 1.035E − 02 |
| | | 2.076E − 03 | 7.181E − 01 | 4.465E − 03 |
| 4 | 8*h* | 4.434E − 03 | 1.617E − 01 | 8.480E − 03 |
| | | 9.947E − 04 | 4.012E − 03 | 4.178E − 03 |
| | | 2.325E − 04 | 5.582E − 04 | 3.872E − 03 |
| | 4*h* | 4.774E − 03 | 6.814E − 01 | 9.658E − 03 |
| | | 1.055E − 04 | 2.275E − 02 | 4.446E − 03 |
| | | 4.965E − 06 | 1.167E − 03 | 3.061E − 03 |
| | 2*h* | 2.096E − 03 | 7.891E − 01 | 3.475E − 03 |
| | | 3.632E − 04 | 1.697E − 01 | 7.559E − 04 |
| | | 9.150E − 05 | 5.688E − 02 | 2.512E − 04 |
| | *h* | 6.079E − 04 | 4.213E − 01 | 1.248E − 03 |
| | | 1.561E − 04 | 2.089E − 01 | 3.633E − 04 |
| | | 6.461E − 05 | 1.243E − 01 | 1.590E − 04 |

vector processor, however, coarse grids in the usual multigrid process will not generally share this simplified property. The question then is whether or not one of the means for preserving this feature on coarser grids (namely, boundary contraction or expansion) will maintain the efficiency of the usual multigrid process. Such is the objective of the experiments reported in this section.

To compare the contracted and expanded boundary methods with the usual multigrid, unigrid was used on the Cray 1 at NCAR as a simple tool to simulate multigrid performance. Instead of comparisons with the usual multigrid on the irregular region, it was much simpler to compare the two methods with the analogous (i.e., naturally extended) problem defined on the entire rectangle [0, 3] × [0, 2]. Thus, a function $U$ on this rectangle was chosen to determine $f$ and the usual unigrid algorithm was run on the full rectangle. The results are depicted in the first column of Table I. Both the contracted and expanded boundary methods were also tried with the same $f$, but with $f$ restricted to the irregular region $\Omega$. The results are depicted in the second and third columns of the table, respectively. Note the severe degradation in convergence for the contracted boundary method. As might be expected, however, there is almost no loss of efficiency with the expanded boundary approach.

Although these are admittedly very limited experiments, they represent the numerical experience with several such tests that were conducted. Generally, although full multigrid (FMG) vastly and expectedly improves the performance of the contracted boundary method, it remains somewhat less efficient than conventional multigrid. On the other hand, the expanded boundary method seems generally as (or nearly as) efficient as, and therefore preferrable, to the usual multigrid approach, expecially for use on vector processors such as the Cray 1.

## References

1. A. Brandt, *Math. Comp.* **31** (1977), 333.
2. S. F. McCormick, and J. W. Ruge, *SIAM J. Numer. Anal.*, to appear.
3. S. F. McCormick, and J. W. Ruge, *Math. Comp.*, to appear.
4. H. Arakawa, *J. Comput. Phys.* **1** (1966), 119.