LA-UR--83-23

DE83 006050

TITLE: A PRIORI LOCAL GRID REFINEMENT IN THE MULTIGRID METHOD

AUTHOR(S): Joel E. Dendy

SUBMITTED TO: To be published in Proceedings of Elliptic Problem Solvers Conference, January 10-12, 1983, Monterey, CA.

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.
The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

MASTER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

A PRIORI LOCAL GRID REFINEMENT IN THE MULTIGRID METHOD

Joel E. Dendy, Jr.
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM

In this work we study the possibility of a priori local grid refinement in the multigrid method for the single group neutron diffusion equation

$$\begin{aligned} -\nabla \cdot (D(x,y) \nabla U(x,y)) + \sigma(x,y) U(x,y) &= f(x,y), (x,y) \in \Omega \\ v(x,y) \cdot D(x,y) \nabla U(x,y) + \gamma(x,y) U(x,y) &= 0, (x,y) \in \partial\Omega \end{aligned} \quad (1)$$

Here Ω is a bounded region in R^2 with boundary $\partial\Omega$, v is the outward normal to $\partial\Omega$, D is positive, and σ and γ are nonnegative. D , σ , and f are piecewise constant with jumps across internal boundaries Γ of Ω , where U and $\mu \cdot (D\nabla U)$ are continuous at almost every $(x,y) \in \Gamma$ for a smooth normal vector μ to Γ . The emphasis is on jumps of orders of magnitude.

For simplicity we assume Ω to be a rectangle $(0,X) \times (0,Y)$ and Γ to be composed of horizontal and vertical line segments. We also assume that the grid G^M , on which (1) is approximated, has a uniform mesh size h_M and that Γ is a subset of the grid; relaxation of these

EJD

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

assumptions is discussed below. Let us write the difference approximation to (1) on a grid G^M as

$$L^M U^M = F^M ; \quad (2)$$

details of the difference approximation (2) may be found in (1,9).

In the simplest form of the multigrid method, one constructs a sequence of grids $G^1, \dots, G^M$ with corresponding mesh sizes $h_1, \dots, h_m$, where $h_{i-1} = 2h_i$. One does a fixed number, IM , of relaxation sweeps (Gauss-Seidel, for example) on (2) and then drops down to grid G^{M-1} and the equation

$$L^{M-1} v^{M-1} = f^{M-1} \equiv I_M^{M-1} (F^M - L^M v^M) , \quad (3)$$

where v^{M-1} is to be the coarse grid approximation to $v^M \equiv U^M - u^M$, where $v^M \equiv u^M$ is the last iterate on grid G^M , and where I_M^{M-1} is an interpolation operator from G^M to G^{M-1} . To solve equation (3) approximately, one resorts to recursion, taking ID relaxation sweeps on grid G^k before dropping down to grid G^{k-1} , $M-1 \geq k \geq 2$ and the equation

$$L^{k-1} v^{k-1} = f^{k-1} \equiv I_k^{k-1} (f^k - L^k v^k) . \quad (4)$$

When grid G^1 is reached, the equation $L^1 v^1 = f^1$ can be solved directly and $v^2 \leftarrow v^2 + I_1^2 v^1$ performed. Then one does IU relaxation sweeps on grid G^{k-1} before forming $v^k \leftarrow v^k + I_{k-1}^k v^{k-1}$, $3 \leq k \leq M$.

For equations with well behaved coefficients, it suffices to take I_{k-1}^k to be bilinear interpolation and I_k^{k-1} to be $(I_{k-1}^k)^T$ or just injection. However, because of the jump discontinuities in D , a more exotic choice of these operators is required (1,5): We take $I_{k-1}^k = J_{k-1}^k$ and $I_k^{k-1} = (J_{k-1}^k)^T$, where J_{k-1}^k is as follows. Suppose that at $(IF+1, JF) \in G^k$, L^k has the pointwise template

$$\begin{bmatrix} -T_{IF+1, JF+1}^k & -W_{IF+1, JF+1}^k & -R_{IF+2, JF+1}^k \\ -Q_{IF+1, JF}^k & S_{IF+1, JF}^k & -Q_{IF+2, JF}^k \\ -R_{IF+1, JF}^k & -W_{IF+1, JF}^k & -T_{IF+2, JF}^k \end{bmatrix} \quad (5)$$

and that $(IC, JC) \in G^{k-1}$ is the same point as $(IF, JF) \in G^k$. At (IF, JF) , J_{k-1}^k is just given by replacement:

$$(J_{k-1}^k v^{k-1})_{IF, JF} = v_{IC, JC}^{k-1}.$$

At $(IF+1, JF)$, we have

$$(J_{k-1}^k v^{k-1})_{IF+1, JF} = \frac{\tilde{Q}_{IF+1, JF} v_{IC, JC}^{k-1} + \tilde{Q}_{IF+2, JF} v_{IC+1, JC}^{k-1}}{\tilde{S}_{IF+1, JF}^k},$$

where

$$\tilde{Q}_{IF+1, JF}^k = T_{IF+1, JF+1}^k + Q_{IF+1, JF}^k + R_{IF+1, JF}^k,$$

$$\tilde{S}_{IF+1, JF}^k = W_{IF+1, JF}^k + S_{IF+1, JF}^k + W_{IF+1, JF+1}^k,$$

$$\bar{Q}_{IF+1,JF}^k = T_{IF+2,JF}^k + Q_{IF+1,JF}^k + R_{IF+2,JF+1}^k$$

(We have just summed (5) vertically to average out its y-dependence.) A similar formula is used for points like (IF, JF+1). In each case, one performs $v^k + v^k + J_{k-1}^k v^{k-1}$. Enough information is now present to use the difference equation at points like (IF+1, JF+1) to solve for $v_{IF+1,JF+1}^k$ in terms of its eight neighbors. Finally, we need to specify L^k , $k < M$, and this is done by taking $L^k = (J_k^{k+1})^T L^{k+1} J_k^{k+1}$.

Let us now consider an example. Take $\Omega = (0., 48.) \times (0., 48.)$, $\Omega' = (23., 25.) \times (23., 25.)$,

$$D(x,y) = \begin{cases} 1., & \text{if } (x,y) \in \Omega' \\ 1000., & \text{otherwise} \end{cases},$$

$$\sigma(x,y) = \begin{cases} 1/3, & \text{if } (x,y) \in \Omega' \\ 0., & \text{otherwise} \end{cases}, \quad (6)$$

$$f(x,y) = \begin{cases} 1., & \text{if } (x,y) \in \Omega' \\ 0., & \text{otherwise} \end{cases}.$$

The boundary conditions are $\frac{\partial u}{\partial v} = 0$ along $y = 0.$ and along $x = 0.$ and $\frac{\partial u}{\partial v} = -\frac{1}{2D}u$ along $x = 48.$ and along $y = 48.$ Results of solving this problem with a uniform grid are described below.

In (9), Wachspress derived an accuracy criterion for problems like the one above. This criterion gives the mesh size h necessary for accuracy in terms of σ and D ; it is $\sqrt{\frac{\sigma}{D}} h \ll 1$. A theoretical explanation for this criterion was given only for one space dimension, but its

utility has been verified many times for multi-dimensional problems. For the problem above, it suggests that the mesh size in Ω' should be less than $\sqrt{3}$ but that in $\Omega \setminus \Omega'$ the mesh size can be much larger. Such problems have been solved traditionally by using Cartesian products of nonuniform meshes; for many practical problems, however, the number and placement of regions needing small mesh sizes is such that this approach leads to a globally small mesh size. Thus, although the Cartesian product approach is quite useful in the above problem, we will retain this problem as a model problem for an illustration of how multigrid and local grid refinement can be successfully combined.

The multigrid local grid refinement we use is the composite grid approach (2,3). In this approach, only the coarsest grid, G^1 , needs to cover all of Ω ; see, for example Figure 1. As in (2) we use an adaptation of the full approximation storage (FAS) version of multigrid instead of the correction version described above.

Let G_m^k denote the set of points of G^k which are inner points of a finer level G^m (that is, points where the G^m difference equations are defined); then the difference equations on G^k are

$$L^k U^k = \bar{F}^k ,$$

where

$$\bar{F}^k = F^k \text{ in } G^k \setminus G_{k+1}^k , \quad (7a)$$

$$\bar{F}^k = F_{k+1}^k = I_{k+1}^k (\bar{F}^{k+1} - L^{k+1} u^{k+1}) + L^k (\tilde{I}_{k+1}^k u^{k+1}), \text{ in } G_{k+1}^k \quad (7b)$$

where $\tilde{I}_{k+1}^k u^{k+1}$ is injection from G^{k+1} to G^k . Thus on the subdomain where G^{k+1} is not defined, G^k plays the role of the finest grid, and on the subdomain where finer subgrids are present, G^k serves for calculating the coarse grid correction. In interpolation, $u^{k+1} + u^{k+1} + I_k^{k+1} (u^k - \tilde{I}_{k+1}^k u^{k+1})$ is used in G_{k+1}^k and $u^{k+1} + u^{k+1} + I_k^{k+1} u^k$ is used on ∂G^{k+1} . Thus the approximate solution u^{k+1} of (7a) is interpolated to provide boundary data for the difference equations on G^{k+1} .

An additional specification is needed in the above description, and that is for the definition of L^k on ∂G_{k+1}^k . L^k can be defined directly in $G^k \setminus G_{k+1}^k$ and can be defined as $(I_k^{k+1})^T L^{k+1} I_k^{k+1}$ in G_{k+1}^k . If it were not for requirements of symmetry and energy conservation, it could also be defined directly on ∂G_{k+1}^k . Instead we use a hybrid definition, which can be illustrated by considering a point P of a vertical part of ∂G_{k+1}^k which is not a corner point. Assume also that G_{k+1}^k lies to the east of ∂G_{k+1}^k and that part of $G^k \setminus G_{k+1}^k$ lies to the west of ∂G_{k+1}^k . Then at P we define the north, northwest, west, southwest, and south coefficients directly; the northeast, east, and southeast coefficients are defined from $(I_k^{k+1})^T L^{k+1} I_k^{k+1}$; the central coefficient is defined by requiring consistency.

It is clear from the above description that there is a limit to the width of refinement near an interface. If

G^{k+1} , for example, ends too close to an interface, then the boundary data from G^k will be inaccurate since it will be playing the role of the finest grid and using a mesh size that is too large. The minimum width to avoid such a situation is $2h_{k+1}$, and in the situation depicted in Fig. 1a, $3h_{k+1}$ is necessary in order that G^{k+1} terminates on a grid line of G^k . In the example (6) note that the fine grids do not always need to cover Ω' . If Ω' were $(17., 31.) \times (17., 31.)$ instead, then refining in the neighborhood of $\partial\Omega'$ might give a picture like Fig. 1b.

We now give a comparison of uniform grid multigrid and a priori local grid refinement multigrid for problem 6. In the uniform case we did problem 6 three times, with $h = 1., \frac{1}{2},$ and $\frac{1}{4}$ respectively. We used Richardson extrapolation on $h = \frac{1}{2}$ and $h = \frac{1}{4}$ to obtain an answer at $(24., 24.)$ of .346818, which we accept as the true answer for purposes of comparison. In each case we solved the problem until the discrete L^2 norm of the residual was less than 10^{-6} . We also used the capability of multigrid to compare residuals on grid G^M and G^{M-1} to solve just to truncation error (2,5). In this case we list the number of times, NCYC, that the finest grid was visited.

In the local grid refinement case, we worked the problem with $M = 5$ and 6 respectively. The grids used were:

- G^1 : uniform with $h_1 = 16.,$
- G^2 : uniform with $h_2 = 8.,$

G^3 : (8., 40.) x (8., 40.) with $h_3 = 4.$,
 G^4 : (12., 32.) x (12., 32) with $h_4 = 2.$,
 G^5 : (20., 28.) x (20., 28.) with $h_5 = 1.$,
 G^6 : (21., 27) x (21., 27) with $h_6 = \frac{1}{2}$;

the $M = 6$ case is shown in Fig. 1a. The comparisons are given in Table 1; T.E. stands for "truncation error". Both methods start on the coarsest grid and "bootstrap" themselves up to the finest grid before beginning simple cycling (5).

TABLE 1.

Case	Answer, Uniform grids	Error, Uniform grids	Answer Local grids	Error, Local grids
$M = 5, -6$ TOL= 10^{-6}	.306501	.040	.307590	.039
$M = 6, -6$ TOL= 10^{-6}	.334839	.012	.334776	.012
$M = 7, -6$ TOL= 10^{-6}	.343824	.003	---	---
$M = 5,$ T.E.	.306501, NCYC=3	.040	.307590, NCYC=3	.039
$M = 6,$ T.E.	.334858, NCYC=3	.012	.334486, NCYC=3	.012
$M = 7,$ T.E.	.343841, NCYC=3	.003	---	---

The comparison between answers for the local grid method versus the uniform grid method is rather good, and a comparison between the number of unknowns suggests that the local grid method should be about eight and twenty times faster than the uniform grid method in the $M = 5$ and 6 case, respectively.

What about the restriction that Γ lie on grid lines of the finest grid? Certainly there is no difficulty about using the above approach with nonuniform grids adjusted so as to satisfy this constraint, but we claim that such an approach is not necessary. Another common approach to interfaces with uniform grids is to assume Γ lies halfway between grid lines; in this case the difference equation coefficients are formed using harmonic averages of values of D instead of arithmetic averages. A comparison of the two approaches in (5) suggested equivalent accuracy. It is easy to devise schemes--and we have--which use a combination of arithmetic and harmonic averaging, reducing to the first if Γ lies on grid lines and to the second if Γ lies halfway between grid lines. Thus one need only come within $h/2$ of Γ to assure accuracy and can thereby use only locally uniform grids.

What about the restriction that Γ consist of horizontal and vertical segments? In this application, this restriction is no undue burden, but in other applications--such as front tracking in time dependent codes--being able to refine locally only horizontally and vertically has been shown to be inefficient.(8) One solution is to allow grids with local coordinates, one of which is nearly parallel to the front.(4) Given the formidable logical difficulties of communicating between such grids and the global uniform grids, it seems more attractive to us to abandon uniform grids in such cases but to retain the logically rectangular structure of the

grids; this leads to difference schemes on quadrilaterals whose vertices can be moved with the front. This technique leads to unacceptable tangling of the mesh in many cases, but recent ideas by J. M. Hyman (6) may solve this difficulty, in which case the above multigrid local mesh refinement techniques--requiring a logically rectangular structure--may be used.

We have called the local mesh refinement in this paper a priori refinement since one knows beforehand where to place the refinement. This is in contrast to adaptive refinement, where decisions of where to refine are made in the course of computation. We have examined such procedures (2,3) in the case of (1), and it is reassuring that the refinements they would dictate are not far from the refinements that we choose in an ad hoc manner. It is clear, therefore, that adaptive refinements are inefficient in this case since they expend unnecessary work to obtain refinements that one could have chosen a priori anyway.

In this regard, we comment briefly on the data structure we use for local grid refinement. It is basically the same linked list structure described in (3,7). Each grid has a pointer to the beginning of storage for that grid, and each grid is organized as a collection of strings. There is one row string which gives the beginning and end J values for the grid. For each J there is a beginning and end I value and a pointer to the next I string. There may be more than one I string for each J, as in Fig. 1b. In the adaptive mode,

when additional points need to be added to an I string, the pointer to that I string is simply changed to point to the next available storage location and the augmented I string is stored beginning there. The pointer to the next I string is unchanged. This system is efficient for serial machines but not apparently so for vector machines like the Cray 1 (which demands vectors stored in contiguous storage locations or, at best, a constant offset) since relaxation and other procedures require a given I string to communicate with I strings above and below it. The finest grid in Fig. 1b could be organized into the bottom six rows, then an intermediate row, then two of seven rows, then an intermediate row, then the top six rows. Four of these six sets would vectorize well. Thus, in the a priori mode, it appears to be possible to take advantage of vector efficiency. In the adaptive mode however, such a conclusion is not clear. Suppose additional points must be added to an I string. Then it appears necessary to move all storage locations after this I string to make room for it, in order to keep storage contiguous. Using vector moves will amortize part of the expense of this operation, but in the worst case the expense could be proportional to the number of grid points. This matter needs further investigation, and in any case the development of an adaptive procedure for the above data structure still awaits the development of suitable software, such as that being developed by Ophir (7).

References

1. R. E. Alcouffe, A. Brant, J. E. Dendy, Jr., and J. W. Painter, SIAM J. Sci. Stat. Comput., 2, 430.
2. A. Brandt, Math. Comp. 31, 333.
3. A. Brandt, in "Mathematical Software III" (John R. Rice, ed.), 273. Academic Press, New York.
4. A. Brandt, in "Multigrid Methods," (W. Hackbush, V. Trottenberg, eds.), Springer-Verlag.
5. J. E. Dendy, Jr., Black box multigrid, to appear in J. Comp. Phys.
6. James M. Hyman, Self-adjusting grid methods for one-dimensional hyperbolic conservation laws, contributed talk at SIAM meeting, Palo Alto, CA., June, 1982.
7. D. Ophir, Language for processes of numerical solutions to differential equations, Ph.D. thesis, The Weizmann Institute of Science, Rehovat, Israel (1979), to appear.
8. B. Swartz, Courant-like conditions limit reasonable mesh refinement to order h^2 to appear in SIAM J.
9. E. L. Wachspress, "Iterative solution of elliptic systems and applications to the neutron diffusion equations of reactor physics". Prentice-Hall, Englewood Cliffs, N.J.

Figure 1a

Figure 1b