General Relativity and Gravitation, Vol. 18, No. 8, 1986

An Introduction to the Multi-Grid Method for Numerical Relativists

Matthew Choptuik[1] **and W. G. Unruh**[1]

Received August 30, 1985

The multi-grid method, which has made a considerable impact on both theoretical and applied numerical analysis in the past decade, is reviewed within the context of the solution of boundary value problems in $3+1$ numerical relativity. The basic principles of operation of a multi-grid algorithm are discussed and, with the aid of numerical experiments on exactly soluble model problems, the method is compared to more traditional techniques such as SOR. The results of application of the method to a set of axisymmetric problems for black hole initial data, previously determined by relaxation techniques, are presented.

1. INTRODUCTION

Boundary value problems are commonplace in numerical relativity. Elliptic partial differential equations are encountered in both the initial value problem and the evolution problem, the two stages in the $3+1$ construction of a space-time. In York's procedure for determining appropriate initial data for a space-time, all four quantities which are not freely specified are constrained by such equations. When the initial data is time evolved, as many as eight separate boundary value problems may have to be solved at each time step, depending on the particular evolution scheme employed. The complexity and nonlinearity of these equations for a generic space-time implies that they will have to be treated numerically.

Most numerical relativists would agree that given a well-posed boundary value problem (BVP), determining an approximate numerical

[1] Department of Physics, University of British Columbia, Vancouver, B.C. V6T 2A6, Canada.

0001-7701/86/0800-0813$05.00/0

842/18/8-3

solution is not, in principle, an extraordinarily difficult task. The straightforward combination of finite difference techniques and successive over-relaxation (SOR) seems to have served researchers well. Because of the "steady state" nature of elliptic problems, one can be almost certain that the end point of a converging relaxation process will be a reasonable approximation to the true solution.

There are difficulties with this standard approach, however, which will become more pronounced as increasingly realistic space-times are constructed. In terms of computer resources, SOR is not an efficient way of solving difference systems. Anyone who has used the method knows that many passes over the grid may be required to produce an acceptable solution. Furthermore, the number of passes needed increases if the grid is made finer in an effort to produce a better approximation. Another difficulty with SOR lies in the fact that the user typically has very little idea about when to stop the iteration. If it is terminated too soon, the solution may not be very reliable; if it is continued too long, much computational work may be wasted computing corrections to the estimated solution, which are negligible in comparison to the unavoidable error which results from the discretization process. Finally, for many boundary value problems, given a difference approximation of fixed order, the grid spacing required to yield a solution of uniform accuracy may vary significantly from place to place in the solution domain. If the grid used is too coarse, accuracy will be compromised. On the other hand, if a grid fine enough to adequately resolve all of the solution detail is used, work may again be wasted in regions of the domain where a coarser grid would suffice. Apart from a change of coordinates or the use of a nonuniform grid, there is little that can be done about this problem using standard relaxation techniques.

One of the purposes of this paper is to review a relatively new method which goes a long way toward remedying all three of these difficulties. In applications from many fields of study, the multi-grid method [3–6], combined with adaptive discretization techniques has provided accurate, efficient solutions to boundary value problems. The method has recently been used [7] to reconstruct initial data for black hole space-times originally calculated by Bowen and York [2], and York and Piran [19]. The results of this study, which include comparisons with relaxation methods, are also presented here. Although knowledge of the formalism used in $3+1$ numerical relativity is assumed for Sections 3 and 4, Sections 2 and 3 are reasonably self-contained and may be of interest to others who have boundary value problems to solve.

2. THE MULTI-GRID METHOD

The multi-grid method is a general numerical technique for solving continuous problems such as boundary value problems or functional integral equations. There are many separate aspects and extensions of the method and the study of multi-grid is an active area of numerical analysis research. The aim of this section is only to introduce the basic principles of a multi-grid solver which suffice to demonstrate why the method represents such an improvement over more well-known methods such as SOR. Much of the material below is originally due to A. Brandt and various collaborators [3–6].

Consider the following boundary value problem in d dimensions

$$\begin{aligned} L[u(x)] &= f(x) \qquad x \in \Omega \subseteq \mathbb{R}^d \\ B[u(x)] &= g(x) \qquad x \in \partial\Omega \end{aligned} \tag{1}$$

where Ω is a subspace of d-dimensional (Euclidean) space with boundary $\partial\Omega$, and L and B are differential operators corresponding to the interior and boundary equations for the unknown $u(x)$. Typically, L will be a second order, elliptic operator; B will usually be the identity operator, the normal derivative operator, or some combination of the two. Numerical schemes for solving such systems fall into two general classes: finite difference methods [10] and finite element methods [11]. Here, only the application of the multi-grid method to finite difference equations will be considered, although the method may also be used in conjunction with finite element techniques.

The first step in the approximate solution of the system (1) by finite difference methods involves the introduction of a grid (mesh) on Ω. The points in Ω which the grid defines comprise the discrete domain Ω^h with boundary $\partial\Omega^h$. With the multi-grid method, it is almost always the case that Ω^h is defined by a uniform grid, that is, a mesh with constant spacing between lines in each of the coordinate directions.

The next step in the solution of (1) involves the introduction of finite difference approximations L^h, B^h of the differential operators L and B. A properly constructed difference scheme results in a system of algebraic equations

$$\begin{aligned} L^h u^h &= f^h \\ B^h u^h &= g^h \end{aligned} \tag{2}$$

Here, u^h is to be thought of as n-component vector of unknowns, where n is the number of points in Ω^h. f^h and g^h are n-component restric-

tions of the given functions f and g onto Ω^h and $\partial\Omega^h$, respectively. Assuming that this algebraic system can be solved, the success of the finite difference scheme is dependent upon two main factors: the order of approximation of L^h and B^h and the density of grid points used. In theory, these factors should be chosen so as to provide a solution of acceptable accuracy with a minimum of computation. In most applications of the finite difference method, however, they are determined by more ad hoc considerations.

There are many techniques which have been developed to solve the systems of algebraic equations resulting from the discretization of boundary values problems via finite difference techniques. Of these, only relaxation methods have been used extensively in numerical relativity [8, 9, 12, 14, 15, 19]. Because of this, and because relaxation processes play a crucial role in the operation of a multi-grid algorithm, a brief review of relaxation methods is in order.

It will be assumed, for simplicity, that both L and B are linear operators. In this case, the solution of the difference scheme satisfies a linear system which will be written as

$$A^h u^h = b^h \tag{3}$$

It will also be assumed that the typical number of grid points in any particular coordinate direction is roughly $N \simeq n^{1/d}$, where as before, n is the total number of grid points and d is the dimension of the problem. Relaxation methods for solving (3) belong to a general class called iterative methods. In these methods, a sequence of iterates $u^{(k)}$, $k = 1, 2, \ldots$, is generated from some initial estimate $u^{(0)}$ such that

$$\lim_{k \to \infty} u^{(k)} = u^h \tag{4}$$

In practice, the iterative procedure is terminated for some finite k when it is felt that the solution estimate is "close enough" to u^h. One possibility is to stop the process when the change in norm of two successive iterates is less than some prescribed amount. Another possibility is to compute, or approximately compute, at each iteration, the residual vector $r^{(k)}$

$$r^{(k)} = A^h u^{(k)} - b^h \tag{5}$$

If the iteration converges, the residual vector tends to the zero vector and the process can be terminated when the norm of $r^{(k)}$ is less than some given tolerance.

Relaxation method also fall into two general classes: point and line methods. A typical point relaxation scheme is the Gauss–Seidel (GS)

method. Let the n points of the grid be numbered in some fashion. Denote the value of the unknown at grid position i by u_i^h and the elements of the matrix A^h by a_{ij}. Then the GS iteration is given by

$$u_i^{(k+1)} = \left[-\left(\sum_{j=1}^{i-1} a_{ij} u_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij} u_j^{(k)} \right) + b_i \right] \Big/ a_{ii} \tag{6}$$

Thus, each component $u_i^{(k+1)}$ is determined by the demand that it instantaneously satisfy the ith equation using the most recently computed values of the other unknowns. Although this method is easy to implement and economical of memory storage, it usually converges very slowly, and is rarely used in practice. It typically requires $O(N^2)$ iterations, or relaxation sweeps, to reduce the error in the approximation by an order of magnitude [16].

Historically, it was discovered that the convergence of GS could often be accelerated by modifying the iteration so that

$$u_i^{(k+1)} = \omega \hat{u}_i^{(k+1)} + (1 - \omega) u_i^{(k)} \tag{7}$$

where $\hat{u}_i^{(k+1)}$ is determined from the right hand side of (6) and ω is called the relaxation parameter. For $0 \leqslant \omega \leqslant 2$, (7) defines the well-known successive overrelaxation (SOR) iteration [16]. An optimal value of ω usually must be determined by numerical experimentation. Given a good choice of ω, the number of sweeps needed to solve the system may be reduced to $O(N)$.

As mentioned previously, the above two relaxation methods are properly referred to as point-GS and point-SOR. Line relaxation methods involve updating entire groups of unknowns, such as all of the unknowns along one grid line, at a time. These methods may converge somewhat faster for some problems and have been found to be applicable (i.e., convergence is guaranteed) for a larger class of problems [16].

Perhaps the most distressing feature of relaxation methods such as SOR is that they require an increasing amount of computational work per grid point as the grid is made finer. This shortcoming can be partially circumvented by noticing that, as with any iterative procedure, a good initial estimate is highly desirable, and that such an estimate may be generated by first solving an equivalent difference problem on a coarser grid, Ω^H. For example the grid lines of Ω^H might consist of every other line of Ω^h. Thus, the system

$$\begin{aligned} L^H u^H &= f^H \\ B^H u^H &= g^H \end{aligned} \tag{8}$$

is solved, and the coarse grid solution, u^H, can then be interpolated to initialize the estimate, $\tilde{u}^h$, of the fine grid unknown. That is

$$\tilde{u}^h := I\!I_H^h u^H \tag{9}$$

where $I\!I_H^h$ is a coarse-to-fine interpolation operator. In general, determining u^H will require only a fraction of the computational work needed to calculate u^h and the hope is that this additional work will be more than offset by work saved in the subsequent computation of u^h. A logical extension of this idea is to solve the discretized version of (1) on a sequence of increasingly finer grids, starting with as coarse a grid as is feasible. The successfulness of this technique depends on the degree of smoothness of the exact solution, but in any case, given that one wants to choose a good starting point for an iterative process, one is naturally led to the notion of using a related, but less costly, process.

This idea of successive approximations using multiple grids is, as the reader might suspect, a part of many multi-grid applications. However, the feature which is most characteristic of multi-grid methods is the use of coarse grids to *accelerate* a fine grid solution process.

To see how this is accomplished, again assume that L^h is linear and that an estimate $\tilde{u}^h$ of the fine grid unknown has been determined. Then the residual vector, r^h, is given by

$$r^h = L^h \tilde{u}^h - f^h \tag{10}$$

and (2) is solved by finding the correction, v^h, which satisfies

$$\begin{aligned} L^h v^h &= -r^h \\ u^h &= \tilde{u}^h + v^h \end{aligned} \tag{11}$$

Now, if there were some way of representing (11) accurately on a coarser grid, Ω^h, then an estimate of v^h could be obtained on Ω^H and then interpolated to Ω^h in a fashion analogous to the determination of $\tilde{u}^h$ from the solution of coarse-grid difference equations. In general, (11) cannot be well represented on Ω^H since r^h may be highly oscillatory from point to point on Ω^h. That is, r^h may have high frequency components which could not be reproduced on a coarser grid. In fact, a highly oscillatory residual is just what one gets from a fine grid solution estimate which has been generated from interpolation of a coarse-grid solution. Thus, before (11) can be well approximated on Ω^H, the residual must be smoothed, so as to remove high frequency components. Fortunately, for most problems, such smoothing may be accomplished simply by employing some form of relaxation. A simple example best illustrates this.

Consider the one-dimensional boundary value problem

$$\begin{aligned} -\Delta u(x) &\equiv -\frac{\partial^2 u(x)}{\partial x^2} = f(x) \qquad x \in [0, 1] \\ u(0) &= u(1) = 0 \end{aligned} \tag{12}$$

Discretize the problem by introducing the uniform grid

$$\begin{aligned} \Omega^h &= \{x_i \mid x_i = ih;\ i = 0,..., N\} \\ hN &= 1 \end{aligned} \tag{13}$$

Denoting $u(x_i)$ by u_i, replace the second derivative with a secoond-order centered difference approximation [10] to get the set of difference equations

$$\begin{aligned} \frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} &= f_i \qquad i = 1,..., N-1 \\ u_0 &= u_N = 0 \end{aligned} \tag{14}$$

which may be written in the form

$$L^h u^h = f^h \tag{15}$$

Here, L^h is an $(N-1)\times(N-1)$ matrix; u^h and f^h are $(N-1)$-component vectors. Now, consider the iteration for the solution of the above system

$$\begin{aligned} u^{(k+1)} &= u^{(k)} - \omega D^{-1}(L^h u^{(k)} - f^h) \\ &= u^{(k)} - \omega D^{-1} r^{(k)} \end{aligned} \tag{16}$$

where D is the main diagonal of L^h and ω is a parameter which will be chosen from the interval [0, 1]. Examine the effect of this relaxation method on the residual vector

$$r^{(k+1)} = L^h u^{(k+1)} - f^h \tag{17}$$

Using (16), this becomes

$$\begin{aligned} r^{(k+1)} &= G r^{(k)} \\ G &\equiv I - \omega L^h D^{-1} \end{aligned} \tag{18}$$

where I is the $(N-1)\times(N-1)$ identity matrix. Clearly

$$r^{(k)} = G^{(k)} r^{(0)} \tag{19}$$

where $G^{(k)}$ is the kth power of the matrix G, and $r^{(0)}$ is the residual corresponding to the initial solution estimate. G has a complete set of orthogonal eigenvectors ϕ_m, $m = 1,..., N-1$, with corresponding eigenvalues λ_m so $r^{(0)}$ may be expressed as

$$r^{(0)} = \sum_{m=1}^{N-1} c_m \phi_m \tag{20}$$

where the c are coefficients. It then follows from (19) that

$$r^{(k)} = \sum_{m=1}^{N-1} c_m (\lambda_m)^k \phi_m \tag{21}$$

Now, taking $\omega = \frac{1}{2}$, the eigenvectors and eigenvalues of G are readily calculated

$$\begin{aligned} &\phi_m = (\sin(\pi mh), \sin(2\pi mh),..., \sin(N-1)\pi mh) \\ &\lambda_m = \cos^2(m\pi h/2) \qquad m = 1,..., N-1 \end{aligned} \tag{22}$$

Now consider the eigenvalue corresponding to the lowest frequency eigenvector

$$\lambda_1 = \cos^2(\pi h/2) = 1 - O(h^2) \tag{23}$$

For small h, it is clear that the lowest frequency component of the residual will be damped very slowly by the iteration (16) resulting in the slow asymptotic convergence rate characteristic of relaxation methods. This is not the case, however, for the high-frequency components of the residual. Consider those components of $r^{(k)}$ which could not be represented on a grid Ω_H with $H = 2h$. These components correspond to eigenvectors having wavelengths greater than $4h$, that is with $mh \geqslant \frac{1}{2}$. The corresponding eigenvalues satisfy

$$\lambda_m \leqslant \cos^2(\pi/4) = 0.5 \tag{24}$$

This result shows that the high-frequency residual components will be damped very effectively by the iteration. A few relaxation sweeps will virtually eliminate them. Moreover, the rate at which the high-frequency components are reduced, which is called the *smoothing rate*, is independent of h. It can also be shown that the error vector

$$e^{(k)} \equiv u^h - u^{(k)} \tag{25}$$

is also smoothed by this relaxation process.

The smoothing of the residual and error vectors is a characteristic property of relaxation methods. However, the smoothing rate of a given method is dependent on the particular system of difference equations being solved and the determination of an appropriate scheme is crucial to the proper operation of a multi-grid algorithm [3].

Assume that a relaxation scheme with satisfactory smoothing properties has been determined for the system (2). Then after the application of a few relaxation sweeps on Ω^h the high-frequency components of r^h will be essentially eliminated. At this point, the system (11) may be accurately represented on the coarser grid Ω^H since the desired correction, v^h, will also be smooth. Thus, on Ω^H, the following problem is solved

$$L^H v^H = -I_h^H r^h \tag{26}$$

Here, I_h^H is a restriction operator which produces a coarse-grid function from a fine-grid function. Once v^H has been determined, the approximation to the fine-grid unknown is updated as

$$\tilde{u}^h := \tilde{u}^h + II_H^h V^H \tag{27}$$

II_H^h is an interpolation operator which, in practice, usually performs linear interpolation. The interpolation process may introduce high-frequency components in the residual, but these may be effectively eliminated with a few more relaxation sweeps on the fine grid. The process of using a coarse grid to compute an approximation to v^h is called a coarse grid correction.

Clearly, the same technique may be used to solve the coarse-grid system (26). Relaxation sweeps over Ω^H, which update the approximation $\tilde{v}^H$ of v^H are performed until the corresponding residual

$$r^H = I_h^H r^h - L^H \tilde{v}^H \tag{28}$$

is smoothed. Then, an even coarser grid may be used to compute a good approximation to the defect $v^H - \tilde{v}^H$. The process continues, using coarser and coarser grids until eventually, on the coarsest grid, a problem results, which can be solved very inexpensively without the aid of another grid. Once this problem has been solved, a descent toward the finest grid is initiated, using a series of interpolations of the various computed coarse grid corrections. Each interpolation is followed by a few more relaxation sweeps to remove high-frequency error components. This entire process is called a coarse-grid correction cycle. At the end of such a cycle, all components of r^h will essentially have been damped by the same factor, and if the initial estimate of u^h was good, the fine grid problem may be solved to

the desired tolerance. If an even better approximation of u^h is desired, another coarse grid correction cycle may be performed.

Even though attention has been restricted here to the case of linear difference equations, the preceding description illustrates the key features of a generic multi-grid algorithm. There are three major ideas involved: (1) A sequence of grids with geometrically decreasing mesh sizes is employed. On each successive grid, the finite difference equivalent of (1) is solved to yield an initial estimate of the unknown on the next finer grid. (2) In the process of solving any system of equations on any particular grid, relaxation sweeps are performed solely for the purpose of smoothing the residual of the system. (An exception is made for the coarsest grid, where relaxation may be used to actually solve the system.) (3) Once the residual of a given system is sufficiently smooth, the problem of computing the necessary correction to the grid function is transferred to a coarse grid.

Because the role of relaxation in the multi-grid method is to smooth the system, rather than solve it, the method does not suffer from the slow convergence rate characteristic of relaxation methods. As long as a relaxation scheme with a smoothing rate independent of the mesh size can be found, the actual number of relaxation sweeps applied on the finest grid in the course of solving the fine grid equations will also be independent of the mesh size. All of the additional work performed in the solution process will amount to the equivalent of a few additional sweeps on the fine grid. Thus, a properly constructed multi-grid solver can solve the system (3) using an amount of computational work per grid point which is independent of the density of grid points. That is, in comparison to the $O(N)$ relaxation sweeps typically needed to solve (3) by SOR, multi-grid requires the equivalent of $O(1)$ sweeps (typically 4–10 in two or more dimensions [3]).

The multi-grid method is equally applicable to nonlinear equations. The algorithm described above must be modified, but the same basic principles apply. In particular, most nonlinear difference systems are as readily smoothed by some form of relaxation as linear systems, so that nonlinear equations are also typically soluble by multi-grid in time proportional to the number of unknowns. Equations resulting from the discretization of elliptic *systems* have also been successfully solved using the method. Provided an appropriate form of relaxation is chosen, a multi-grid program can be fully parallelized to take advantage of any available parallel processing capability. There are many other attractive features of the multi-grid method; some of these will be mentioned in Section 5.

At the present time the major drawback to the use of the multi-grid method would seem to be its implementation. Coding a multi-grid algorithm is a nontrivial task—especially when compared to the coding of

an SOR routine. General purpose multi-grid software is available [6], but its usefulness for the class of BVPs encountered in numerical relativity has yet to be investigated. The next section is intended to give the interested reader some ideas regarding the construction of a simple, yet representative multi-grid program. The description of an actual implementation also has the benefit of clarifying the various multi-grid processes which have been described in a somewhat abstract fashion thus far.

3. A SIMPLE MULTI-GRID EXAMPLE

Again for simplicity, consider the one-dimensional BVP described in the previous section:

$$\begin{aligned} -\frac{\partial^2 u}{\partial x^2} &= f(x) \qquad x \in [0, 1] \\ u(0) &= u(1) = 0 \end{aligned} \tag{12}$$

As before, the discrete domain is a uniform grid

$$\begin{aligned} \Omega^h &= \{x_i \mid x_i = ih;\ i = 0, \ldots, N\} \\ hN &= 1 \end{aligned} \tag{13}$$

and the second-order centered difference approximation is employed, resulting in the set of algebraic equations

$$\begin{aligned} h^{-2}(-u_{i+1} + 2u_i - u_{i-1}) &= f_i \qquad i = 1, \ldots, N-1 \\ u_0 &= u_N = 0 \end{aligned} \tag{14}$$

The solution of these equations is an approximation to the solution of (1) whose error is $O(h^2)$ as $h \to 0$, for f sufficiently smooth.

In what follows, the reader should be aware that use of the multi-grid method for the solution of such a simple BVP is not being advocated. This type of problem—if it cannot be treated analytically—will probably be soluble by any one of a number of commonly available ODE solvers. In addition, (14) is a tridiagonal system which can be solved with $O(N)$ calculations [10]. This does not imply, however, that the multi-grid solution of this problem is valueless. The program described below already has many of the algorithmic and data structuring complexities of a more realistic application. At the same time, exact solutions of (12) are easily constructed for testing purposes, and because the problem is one-dimensional, computer resource constraints are not really a concern. More

importantly, this type of exercise provides the opportunity to gain a feeling for the operation of a multi-grid process while minimizing the effects of programming difficulties induced by complexities in the BVP itself.

The task of programming a multi-grid algorithm is considerably simplified if proper attention is paid to the organization of the data objects which the program is to manipulate. The basic data structure used in the current application is depicted in the form of pseudo-code in Fig. 1a. It consists of an aggregate of substructures each of which contains the data objects associated with a specific level of discretization. The ease of implementation of such a structure varies from language to language—the important thing is that at the top levels of the program, some facility should exist for communicating arbitrary data objects from arbitrary grids to lower level routines which will perform the actual multi-grid processes. As a final point regarding the data structure, it is implicit in the following that all grids cover the unit interval and that the relation $n_j = 2n_{j-1}$ holds for all $j > 1$.

Given the main data structure, the highest level routine is very straightforward and is described in the form of pseudo-code in Fig. 1b. Four parameters are supplied to the main routine: ngrid and ncycle determine the number of levels of discretization to be employed and the number of coarse grid corrections to be made at each level, respectively; the roles of p and q will be explained below.

On the coarsest grid, the components of the unknown grid function are (arbitrarily) initialized to 0—on subsequent levels the initial estimate is cubically interpolated from the final solution on the previous level. The use of cubic interpolation is dictated by the known smoothness of solutions to (12) for smooth f—in general the appropriate degree of interpolation is a function of both the order of the differential equation and the order of the difference approximation [3]. Polynomial interpolation is a very common and well-studied numerical operation (see for example [1]) and the "Cubically-Interpolate" routine is easily written.

The recursive routine "Solve," given in pseudo-code in Fig. 1c is the heart of the multi-grid solver. The parameters j and m indicate which problem is being solved and on which grid it is being solved, respectively. Thus if $m < j$, then "Solve" is performing a coarse grid correction. On the coarsest grid, relaxation sweeps are always used to *solve* the system to some tolerance c_1. In the runs described below, where there was only one equation to be solved on the coarsest level, a single relaxation sweep would always solve the system, but in general, many sweeps can be performed on the coarsest level, using a negligible portion of the total computational effort.

For $m > 1$, the role of relaxation sweeps is to *smooth* the residuals,

```
main_structure IS
    ngrid : INTEGER                          {number of grids}
    grid_j ; j := 1...ngrid : STRUCTURE
grid_j IS
    n_j : INTEGER                            {number of grid points}
    h_j : REAL                               {grid spacing}
    u_j[0...n_j]   : REAL ARRAY              {unknown vector}
    rhs_j[0...n_j] : REAL ARRAY              {right hand side vector}
    r_j[0...n_j]   : REAL ARRAY              {residual vector}
```

(a)

```
PROCEDURE Main_mg(ngrid,ncycle,p,q)
    Create_main_structure(ngrid)
    DO j := 1...ngrid
        Initialize(grid_j)
        IF j = 1
        THEN
            u_j := 0
        ELSE
            u_j := Cubically_Interpolate(u_j-1)
        END IF
        DO ncycle TIMES
            u_j := Solve(j,j,p,q)
        END DO
    END DO
END PROCEDURE
```

(b)

```
PROCEDURE Solve(j,m,p,q)
    IF m = 1
    THEN
        DO ( e := Relax(m) ) UNTIL ( e ≤ c_1) END DO
    ELSE
        DO p TIMES ( e := Relax(m) ) END DO
        r_m := Residuals(m)
        rhs_m-1 := Restrict(-r_m)
        u_m-1 := 0
        Solve(j,m-1,p,q)
        u_m := u_m + Linearly_Interpolate(u_m-1)
        DO q TIMES ( e := Relax(m) ) END DO
    END IF
END PROCEDURE
```

(c)

```
Legend:
    PSEUDO-RESERVED WORDS
    Procedures (may be null valued, array valued, etc.)
    identifiers
    {comments}
    := means "is assigned the value"
```

(d)

Fig. 1. Pseudo-code form of the basic multi-grid algorithm described in the text.

and, as discussed previously, if a good form of relaxation has been chosen, adequate smoothing can be accomplished at any level with a constant number of sweeps. Thus, p Gauss–Seidel (point-GS) relaxation sweeps are performed by the routine "Relax" before the coarse grid correction is started. The residual vector r_m is calculated and then its negative is injected into the right hand side vector, rhs_{m-1} of the coarse grid. The "Restrict" routine coded here was the simplest possible—the components of rhs_{m-1} are every second component of $-r_m$. This is an entirely adequate procedure for the simple problem solved here; other problems may require more sophisticated transferring schemes [3]. Following the residual injection, the vector u_{m-1}, which will eventually contain the coarse grid correction, is zeroed and then "Solve" is recursively invoked with the same j, but with m decremented by one. When this invocation of "Solve" returns, the coarse grid correction is complete and the solution estimate u_m is updated by addition of the correction, which is linearly interpolated from u_{m-1}. Again, the routine "Linearly-Interpolate" is very easy to write in this case. Following the coarse grid correction, q additional point-GS sweeps are performed, and the routine terminates with u_m containing either an estimate of the solution of (12) on level j if $m = j$ or a smoothed correction to a level $m+1$ unknown if $m < j$. As with the data structure for the program, the ability to implement "Solve" recursively is language dependent; however, an iterative version may also be coded without too much trouble.

The results obtained from a FORTRAN-66 version of this program as well as an SOR routine for a problem with

$$f = -e^x((1-\pi^2)\sin(\pi x) + 2\pi\cos(\pi x)) \tag{29}$$

corresponding to an exact solution

$$u = e^x \sin(\pi x) \tag{30}$$

are summarized in Fig. 2. The systems solved ranged in size from $N = 2$ to $N = 1024$. The left y axis shows the amount of relaxation work W_R invested per grid point to solve the system. For the SOR method, this represents virtually all the computational work; for the multi-grid method, at least 70% of the total work is invested in relaxation (except for very small N). The dotted line shows a measure of the average relative error

$$e_{r_j} = |u_{\text{exact}_j} - u_{\text{calc}_j}|/u_{\text{exact}_j}$$

at each level as labeled on the right axis. For all MG runs p and q were both 3 and ncycle was 1.

An examination of the slope of the dotted line shows that the error is going to 0 slightly faster than h^2, which suggests that the single cycle of the

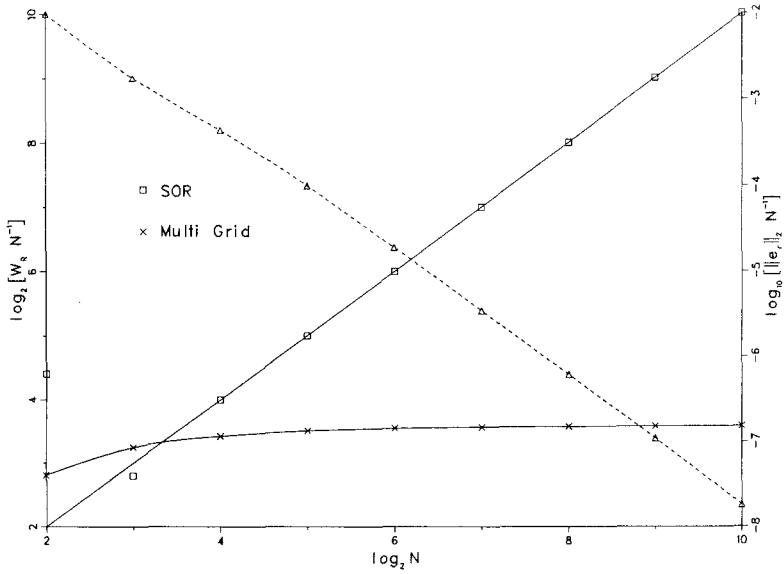

Fig. 2. Comparison of the solution efficiencies of SOR and multi-grid solvers.

MG program solves each problem to the level of inherent discretization error. The ability of the method to solve the difference equations with constant relaxation work per grid point is evident, as is the linear increase in $W_R N^{-1}$ for the SOR method. It should be remarked that because of the simple nature of (12) and the discretization procedure, the optimal relaxation parameter can be exactly calculated [16]

$$\omega_{\text{opt}} = 2(1 + \sin(\pi/N))^{-1} \tag{31}$$

and this value was used in the SOR routine. In addition, the initial estimate given to the SOR program was the same cubically interpolated function that the corresponding MG routine began with, and the SOR process was terminated as soon as the norm of the residual was less than the norm of the residual at the end of the MG run.

The reader might suspect that even through the SOR routine takes more sweeps to converge, it might be terminated early without a corresponding decrease in accuracy. That this is not the case is suggested by Fig. 3 which shows a plot of the absolute value of the SOR residual vector for a $N = 256$ system, throughout the solution process. It would seem that stopping the solution early is likely to result in a solution whose accuracy is good in one part of the domain and questionable, at best, in the other.

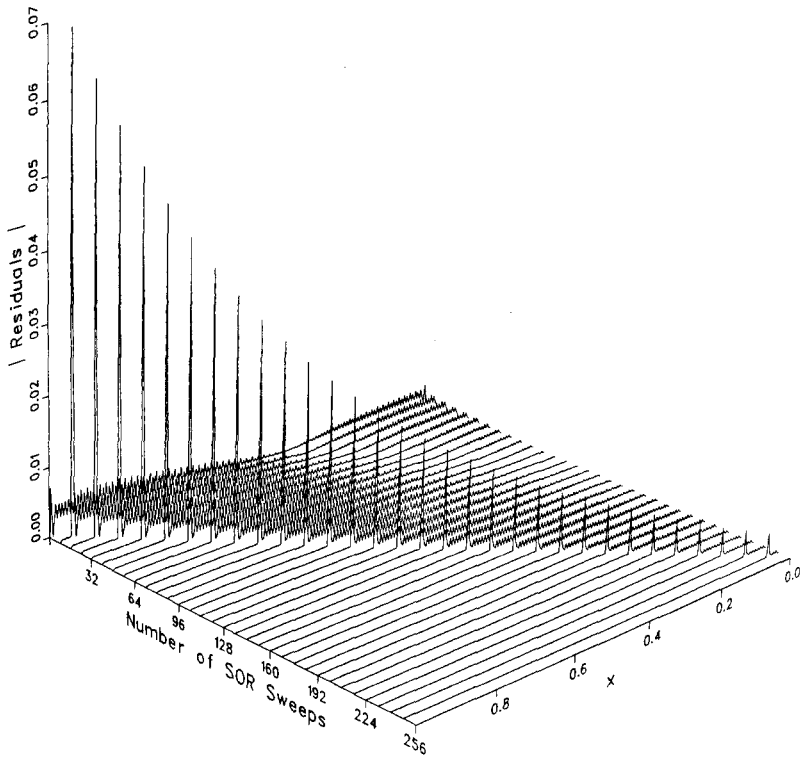

Fig. 3. Effect of SOR solver on residual vector.

Finally, Fig. 4 shows a similar plot of the residuals generated in the course of an MG solution of the $N = 64$ problem. The residuals after any relaxation sweep on any level are displayed as solid lines. The spacing between any residual vector and its predecessor reflects the amount of work, in units of a $N = 64$ relaxation pass, performed to produce the residual. Thus, for example, the dark strip around 6 on the right-hand axis represents all of the relaxation work expended on the three coarsest grids. The dotted line shows the SOR residual after 12 SOR sweeps. The contrasting nature of the solution processes is quite evident.

4. A SPECIFIC BOUNDARY VALUE PROBLEM

As mentioned in the introduction, the determination of initial data for a space-time via York's procedure [13, 17–19] is one part of numerical relativity which requires the solution of elliptic boundary value problems.

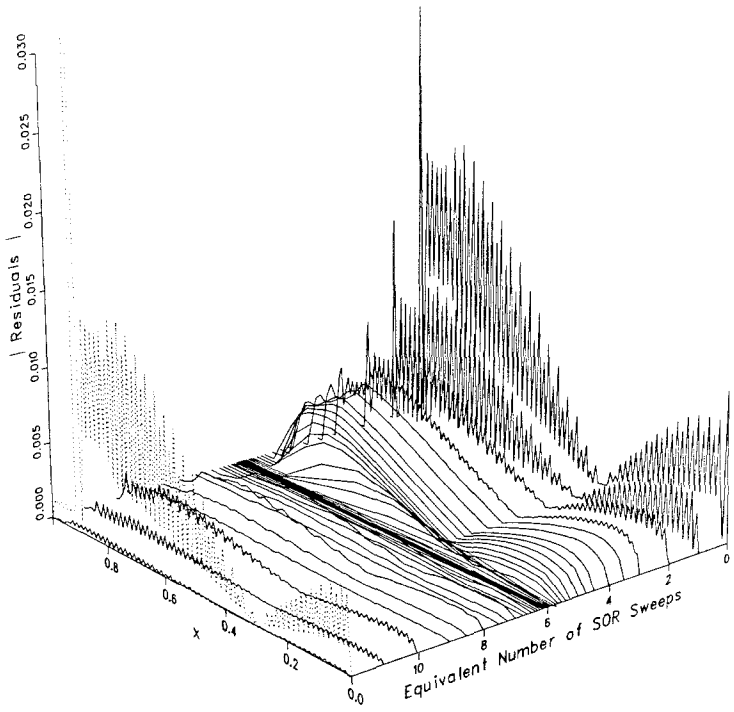

Fig. 4. Effect of multi-grid solver on residual vector.

We have used the multi-grid method to determine a portion of the initial data for space-times containing boosted and spinning black holes. This data was originally constructed by Bowen and York [3] and York and Piran [19]. The current work relies heavily on some of the results in these papers which will be reviewed briefly below. Our approach to the discretization of the Hamiltonian constraint, which differs from that used in the previous work, will also be described. Familiarity with York's formulation of the initial value problem is assumed for this section.

In order to simplify the solution of the initial value problem, Bowen and York restricted attention to maximal, conformally flat, vacuum hypersurfaces. Thus, the mean extrinsic curvature of the slices vanishes

$$\mathrm{Tr}\, K \equiv g_{ij} K^{ij} = 0 \tag{32}$$

The 3-metric is of the form

$$g_{ij} = \psi^4 f_{ij} \tag{33}$$

where f_{ij} is a flat metric, and ψ is the conformal factor. The energy and momentum densities on the slices vanish, and the slices are also assumed to be asymptotically flat. By further demanding that the conformally scaled extrinsic curvature, $\hat{K}^{ij}$

$$\hat{K}^{ij} = \psi^{10} K^{ij} \tag{34}$$

where K^{ij} is the physical extrinsic curvature, be purely longitudinal, the momentum constraints were reduced to

$$D_j \hat{K}^{ij} = 0 \tag{35}$$

D_j is the covariant derivative compatible with the conformal metric. Of the many possible solutions of the above equations, Bowen and York chose

$$\hat{K}^{\pm}_{ij} = \frac{3}{2r^2}\left[P_i n_j + P_j n_i + (f_{ij} - n_i n_j)\, P^k n_k\right] \mp \frac{3a^2}{2r^4}\left[P_i n_j + P_j n_i + (f_{ij} - 5 n_i n_j)\, P^k n_k\right] \tag{36}$$

$$\hat{K}_{ij} = \frac{3}{r^2}\left[\varepsilon_{kil} J^l n^k n_j + \varepsilon_{kjl} J^l n^k n_i\right] \tag{37}$$

Here, r is the Euclidean distance from some arbitrary point on the hypersurface, n^i is the unit normal to a $r =$ constant 2-sphere, P^i and J^i are constant vectors, ε_{ijk} is the permutation tensor, and a is a free parameter having units of length.

With the simplifying assumptions outlined above, the Hamiltonian constraint for the conformal factor reduces to

$$\Delta\psi + \frac{\hat{K}^{ij}\hat{K}_{ij}}{8\psi^7} = 0 \tag{38}$$

with $\hat{K}^{ij}$ given by (36) or (37). Asymptotic flatness of the slices provides the boundary condition for ψ

$$\lim_{r \to \infty} \psi = 1 + O\left(\frac{1}{r}\right) \tag{39}$$

Equations (38) and (39) do not yield a well posed boundary value problem for ψ due to the irregular behavior of the $\hat{K}^{ij}$'s at $r = 0$. To deal with this problem, Bowen and York demanded that the slices be isometric with respect to a mapping through a 2-sphere of radius a. The $\hat{K}^{ij}$'s of (36) and

(37) are compatible with this requirement. The isometry condition yields an inner boundary condition for ψ

$$\left.\frac{\partial\psi}{\partial r}+\frac{\psi}{2r}\right|_{r=a}=0 \tag{40}$$

Physically, the isometric condition leads to the identification of the regions $0<r\leqslant a$, $r\geqslant a$ as two separate, but identical asymptotically flat sheets joined in a smooth fashion by a "throat" at $r=a$. Because of the similarity to the case of the Schwarzschild black hole at a moment of time symmetry, Bowen and York concluded that their initial data represented time instants of black holes. Furthermore, it was shown that a slice with $\hat{K}^{ij}$ given by (36) or (37) was a time instant of a space-time having linear momentum P^i or angular momentum J^i, repectively.

Equation (38) for $r\geqslant a$, together with the boundary conditions (39) and (40) constitute a well-posed boundary value problem for ψ which can be solved numerically by finite difference techniques. Introducing spherical polar coordinates (r, θ, ϕ) on the slice, the flat metric is

$$f_{ij}=\mathrm{diag}(1, r^2, r^2\sin^2\theta) \tag{41}$$

The explicit form of the $\hat{K}^{ij}\hat{K}_{ij}$ term in (38) is easily calculated from (36) and (37). For the boosted hole

$$\begin{aligned} H_{\mathrm{B}}^{\pm} &\equiv \hat{K}^{ij}\hat{K}_{ij} \\ &= \frac{9}{2}\frac{P^2}{r^4}\left[\left(1\mp\frac{a^2}{r^2}\right)^2+2\cos^2\left(1\pm\frac{4a^2}{r^2}+\frac{a^4}{r^4}\right)\right] \\ P &\equiv (f_{ij}P^iP^j)^{1/2} \end{aligned} \tag{42}$$

and for the spinning hole

$$\begin{aligned} H_{\mathrm{s}} &= \frac{18\,J^2\sin^2\theta}{r^6} \\ J &\equiv (f_{ij}J^iJ^j)^{1/2} \end{aligned} \tag{43}$$

$H_{\mathrm{B}}^{\pm}$ and H_{s} have no ϕ dependence (the space-times are axisymmetric), and are reflection symmetric about $\theta=\pi/2$. Thus, it suffices to solve the boundary value problem on the domain $r\geqslant a$, $0\leqslant\theta\leqslant\pi/2$. The symmetry conditions provide the additional boundary conditions for ψ

$$\left.\frac{\partial\psi}{\partial\theta}\right|_{\theta=\pi/2,0}=0 \tag{44}$$

The unboundedness of the domain in the radial direction presents a problem numerically, since any computational domain is necessarily finite. York and Piran replaced (39) with a boundary condition, derived from the known asymptotic behavior of ψ, which could be imposed at a finite radius. A different approach is used here [20] which involves the introduction of a new radial coordinate:

$$s \equiv 1 - \frac{a}{r} \tag{45}$$

Thus, the interval $a \leqslant r < \infty$ is mapped onto $0 \leqslant s < 1$. This transformation proved to be quite advantageous from a numerical point of view.

The Laplacian of ψ in (s, θ) coordinates is

$$\Delta\psi = \frac{(1-s)^4}{a^2}\frac{\partial^2\psi}{\partial s^2} + \frac{(1-s)^2}{a^2 \sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\psi}{\partial\theta}\right) \tag{46}$$

At $\theta = 0$, the second term in the above is singular. Accordingly, it is replaced by $2(1-s)^2/a^2 \cdot \partial^2\psi/\partial\theta^2$ which follows from an application of L'Hopital's rule and the boundary condition (44). In terms of the s coordinate, the inner boundary condition (40) becomes

$$\left.\frac{\partial\psi}{\partial s} + \frac{\psi}{2}\right|_{s=0} = 1 \tag{47}$$

and the outer boundary condition is simply

$$\psi|_{s=1} = 1 \tag{48}$$

A uniform grid is now introduced on the domain of the boundary value problem as illustrated in Fig. 5. The points marked with $\times$ in the diagram, which comprise the set

$$\begin{aligned} \{(s_i, \theta_i) \,|\, & s_i = i\Delta s, \;\; i = 0, 1, \ldots, n_s - 1; \\ & \theta_j = j\Delta\theta, \;\; j = 0, 1, \ldots, n_\theta - 1\} \\ & n_s = (\Delta s)^{-1} \\ & n_\theta = \pi(2\Delta\theta)^{-1} \end{aligned} \tag{49}$$

are the points at which a difference analog of the interior equations is to be satisfied. The points marked with $\square$, which lie outside of the continuous domain, are introduced so that centered difference approximations may be employed throughout the domain and on the boundaries. Finally, the circled points are locations where the value of ψ is known.

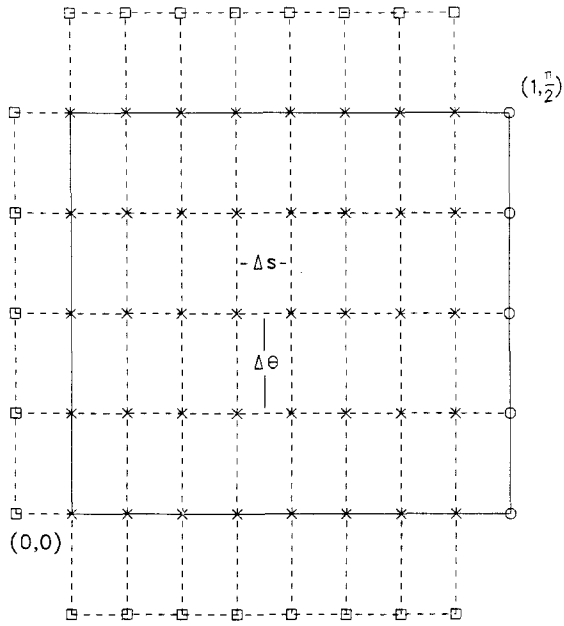

Fig. 5. Uniform grid for discretization of the BVP described in the text.

Using (46), and replacing all derivatives by conservative second order differences [10], the discrete version of (38) is

$$\frac{(1-s_i)^4}{a^2\,\Delta s^2}\,[\psi_{i+1,j}-2\psi_{i,j}+\psi_{i-1,j}]$$

$$+\frac{(1-s_i)^2}{a^2\,\Delta\theta^2}\,[C_j^+(\psi_{i,j+1}-\psi_{i,j})-C_j^-(\psi_{i,j}-\psi_{i,j-1})]+\frac{H_{ij}}{8\psi_{i,j}^7}=0$$

where $i=0,\ 1,...,\ n_s-1,\ j=0,...,\ n_\theta$

$$C_j^{\pm}=\begin{cases}\dfrac{\sin\theta_{j\pm 1/2}}{\sin\theta_j}, & j=1,...,\ n_\theta\\ 2, & j=0\end{cases} \tag{50}$$

where, for example, $\psi_{i,j}=\psi(i\Delta s,\ j\Delta\theta)$. The boundary conditions (44) and (47) are discretized in the same fashion

$$\frac{\psi_{i,1}-\psi_{i,j-1}}{2\Delta\theta}=0, \qquad i=0,\ldots, n_s-1$$

$$\frac{\psi_{i,n_\theta+1}-\psi_{i,n_\theta-1}}{2\Delta\theta}=0, \qquad i=0,\ldots, n_s-1 \tag{51}$$

$$\frac{\psi_{1,j}-\psi_{-1,j}}{2\Delta s}+\frac{\psi_{0,j}}{2}=0, \qquad j=0,\ldots, n_\theta$$

Finally, (48) simply becomes

$$\psi_{n_s,j}=1 \tag{52}$$

The last five expressions define the system of nonlinear equations whose solution via the multi-grid method is discussed in the next section.

5. NUMERICAL EXPERIMENTS AND RESULTS

In order to evaluate the performance of the multi-grid method with respect to relaxation methods, a series of numerical experiments using a modified version of the system of equations described in the previous section were performed. York and Bowen had discovered that if the following form for $\hat{K}^{ij}\hat{K}_{ij}$ is used in (38)

$$H_{\text{MODEL}}=\frac{6P^2}{r^4}\left(1-\frac{a^2}{r^2}\right)^2 \tag{53}$$

then the conformal factor is given by

$$\psi_{\text{MODEL}}=\left(1+\frac{2E}{r}+\frac{6a^2}{r^2}+\frac{2a^2E}{r^3}+\frac{a^4}{r^4}\right)^{1/4}$$
$$E=(P^2+4a^2)^{1/2} \tag{54}$$

Here, E is the ADM energy of the space-time. It was originally intended to solve the system given by equations (50)–(52) using both the multi-grid and relaxation methods. However, it was found that the inner boundary condition (47) makes the system of difference equations nondefinite. That is, the system has both positive and negative eigenvalues. When relaxation sweeps are applied to such a system, residual components corresponding to the negative eigenvalues tend to be magnified so that convergence to a solution is not achieved. For this reason, the difference equations were modified by specifying values for the unknowns on the inner boundary (Dirichlet conditions) to restore definiteness.

Table I. Multi-Grid Results for Modified Test Problem

Grid	n_R	Time (msec)	Storage (kbytes)	$\|r\|$
9×9	7.0	80	25	8.7 (−4)
17×17	7.0	145	40	3.1 (−4)
33×33	7.5	390	90	4.4 (−5)
65×65	8.5	1540	260	9.4 (−6)
127×127	7.5	5200	960	1.4 (−6)

Tables I, II, and III list the results of applying the multi-grid, point-SOR and line-SOR methods, respectively, to test problems of various sizes, all with $a = 1$ and $P = 2$. (For the last method, lines of constant radial coordinate were used.) All routines were coded in FORTRAN-66 using 8-byte variables, and compiled with the IBM H Optimizing compiler. The tests were performed on an Amdahl 470 V/8 CPU, operating under the MTS time-sharing system. The tables list the size of the grid, the number of relaxation sweeps, n_R, required to solve the system, the CPU time and memory storage required, the L_1 norm of the residual when the solution process was stopped, and for the relaxation methods, the over relaxation parameter which was determined by numerical experiment. Although the multi-grid algorithm used more time for the smaller grids, it is clearly superior for the larger problems. Notice that the constancy of n_R in Table I, as well as the timing figures demonstrate the ability of the multi-grid method to solve the nonlinear difference equations with an amount of work which is proportional to the number of grid points. It should be noted that the line-SOR method failed to converge on the 65×65 grid for any $\omega > 1$, and converged so slowly for $\omega = 1$ that it was terminated before a solution with the required accuracy had been attained. Finally, although the multi-grid method used about four times as much storage as the relaxation

Table II. Point-SOR Results for Modified Test Problem

Grid	ω	n_R	Time (msec)	Storage (kbytes)	$\|r\|$
9×9	1.55	10	20	1	5.6 (−4)
17×17	1.60	10	64	4	2.8 (−4)
33×33	1.60	48	1170	16	4.3 (−5)
65×65	1.60	170	17000	64	9.4 (−6)

Table III. Line-SOR Results for Modified Test Problem

Grid	ω	n_R	Time (msec)	Storage (kbytes)	$\|r\|$
9×9	1.35	8	16	1	2.9 (−4)
17×17	1.45	9	71	4	3.1 (−4)
33×33	1.45	38	1100	16	4.4 (−5)
65×65	1.00	[100]	[11300]	64	[1.8 (−5)]

methods, at least 40% of the storage was unnecessary and could have been eliminated at the expense of a 5–10% increase in execution time.

Before proceeding to a discussion of the multi-grid solution of the actual difference equations described in the previous section, a few comments about the specific algorithm used should be made. Because of the nonlinearity of the boundary value problem, a multi-grid variant called the full approximation scheme (FAS) [3] was used. In an FAS algorithm, coarse grid corrections to fine grid unknowns are not calculated explicitly as in the procedure discussed in Sections 2 and 3. Instead, the basic grid function which is manipulated on any level during a coarse grid correction represents a full approximation to the finest level unknown. One attractive feature of the FAS scheme, as well as other multi-grid methods, is that estimates of the *local truncation error* of the approximate solution can be routinely generated. Given the differential operator L of (1) and a corresponding difference operator L^h, the local truncation error, τ^h, defined on Ω^h is given by

$$\tau^h = L^h I^h u - I^h L u \tag{55}$$

where u is the exact solution of (1) and I^h restricts a function on Ω to Ω^h. The value of τ^h at any point of Ω^h provides a measure of how well the difference operator approximates the differential operator at that point.

The ability to estimate the local truncation error provides a way of defining natural terminating criteria for the multi-grid algorithm. The basic rule is simple and logical: once the size of the residual is comparable to the size of the estimated truncation error, the problem is essentially solved, and any additional computational work expended on the solution may well be wasted.

Another place where truncation error estimation is useful is in the implementation of adaptive discretization techniques [3–6]. The basic aim of such methods is to provide uniformly accurate solutions with a minimum of wasted work by introducing fine grids only in those areas of

the problem domain where they are really needed. The multi-grid program used to obtain the results described below accepted a parameter, τ_c, which represented an upper bound on the desired level of truncation error. When the submitted problem had been solved on some grid Ω^h, a local truncation error estimate τ^h was generated. This estimate was then examined to determine if there was some region in Ω^h where τ^h consistently exceeded τ_c. If such a region was found, a finer grid was introduced, only in that region. Finer and finer grids were introduced only where they were needed until the estimated truncation error was less than τ_c everywhere.

Because of the nondefiniteness of the difference equations (50)–(52), systems on the coarsest grid could not be solved by relaxation. Therefore, a scheme which combined an n-dimensional version of Newton's method with a direct (i.e., Gaussian elimination) solution of the resulting linear systems was employed. The basic relaxation scheme used in the algorithm was line-GS with the lines of unknowns having constant radial coordinate. Point-GS could not be effectively used, since the coupling between unknowns in the angular direction is much greater than that in the radial direction, and point methods smooth predominantly along directions of strong coupling. Finally, the boundary equations (51) were treated according to Brandt's suggestions for non-Dirichlet conditions [4]. That is, a process which aimed only to smooth the boundary residuals, independently of the interior relaxation, was employed.

To test the operation of the adaptive multi-grid algorithm, two additional sets of numerical experiments were performed using the actual difference equations (50)–(52) and the model H of (53). The results of a series of test runs with fixed momentum, $P=4$ $(a=1)$, and varying values of τ_c are listed in Table IV. The first column lists the supplied values of τ_c. The next six columns show the extent of the various grids used for each problem. The number heading each column is the maimum number of grid points available in the radial direction at each level; the numbers beneath indicate the position of the outer boundary of the corresponding grid for

Table IV. Effect of Varying Truncation Error Parameter

	Extent of grids						Time	Error (%)	
τ_c	9	17	33	65	129	257	(msec)	$\|\cdot\|_\infty$	$\|\cdot\|_1$
10^{-3}	9	—	—	—	—	—	360	.707	.698
10^{-4}	9	17	21	—	—	—	1130	.178	.170
10^{-5}	9	17	33	53	73	—	3200	.041	.036
10^{-6}	9	17	33	65	113	201	6900	.009	.007

each run. The execution times listed are for the solution of the entire problem, which in each case began on a 5×5 grid. Because the test problem is spherically symmetric, only four angular zones were used at each level. In the final two columns, the relative percentage errors in the computed solutions, calculated as

$$\left\| \frac{\psi_{i,i}^{\text{exact}} - \psi_{i,j}^{\text{computed}}}{\psi_{i,j}^{\text{exact}}} \right\| \tag{56}$$

using both L_1 and L_∞ norms are listed. Notice how the error in the computed solution uniformly decreases as the convergence criteria become more stringent, even though the entire domain is not being discretized in a uniform fashion.

Table V shows the results of a series of runs in which τ_c was held constant at 10^{-5} and the momentum was varied. The format of this table is the same as Table IV. The execution time needed increases with momentum, but remains essentially linear in the total number of grid points used at the various finest levels. The error in the computed solution as a function of momentum is quite constant, varying from 0.04% to 0.12% in the L_1 norm. Again, this demonstrates the effectiveness of the adaptive procedure in producing solutions of uniform accuracy with a minimum of wasted work.

Table VI displays results obtained with the multi-grid algorithm for the case of the boosted black holes, for both H_B^+ and H_B^-. The problem was solved for a series of linear momentum values chosen to correspond

Table V. Multi-Grid Test Results for Varying Momentum

	Extent of grids					Time	Error (%)	
P	9	17	33	65	129	(msec)	$\|\cdot\|_\infty$	$\|\cdot\|_1$
2	9	17	25	45	—	2000	.058	.052
4	9	17	33	53	73	3225	.041	.036
6	9	17	33	57	93	3700	.081	.062
8	9	17	33	57	101	3750	.132	.090
10	9	17	33	57	105	3800	.196	.121
12	9	17	33	65	109	4200	.081	.046
14	9	17	33	65	113	4300	.106	.053
16	9	17	33	65	113	4500	.133	.062
18	9	17	33	65	117	4400	.165	.070
20	9	17	33	65	117	4850	.194	.078

Table VI. Total Energy Versus Momentum for Boosted Holes

P	$E(H_B^+)$	$E(H_B^-)$
1.0	2.35	2.33
2.5	3.59	3.55
5.0	6.15	6.09
7.5	8.87	8.81
10.0	11.6	11.5
12.5	14.4	14.3
15.0	17.2	17.1
17.5	20.0	20.0

with those used by Piran in his work on the problem. Again, E is the ADM energy, which was calculated from the expression

$$E = \frac{1}{16\pi} \int_{r \geq a} \frac{\hat{K}_{ij}\hat{K}^{ij}}{\psi^7} dv + \frac{a}{2} \int_0^\pi \psi \sin\theta \, d\theta \tag{57}$$

via numerical integration. Judging from the results for the test problem, the energies are probably accurate to within a percent or two. The results agree with those obtained previously to within 2%. All of the solutions were very nearly spherically symmetric.

Another series of runs was performed to determine the conformal factor for the spinning black holes, using expression (43). The calculated total energy as a function of momentum appears in the second column of Table VII. The range of J values used was again chosen to facilitate comparison with Piran's results. The computed solutions for large values of J

Table VII. Angular Momentum and Energy Parameters for Spinning Holes

J	E	M_{AH}	$M(J, M_{AH})$	$\overline{\Delta E}$ (%)
1.0	2.05	2.03	2.05	0
3.0	2.33	2.23	2.33	0
10.0	3.48	3.03	3.45	1
30.0	5.76	4.67	5.67	2
100.0	10.4	8.07	10.2	2
300.0	18.0	13.7	17.5	3
1000.0	32.9	24.7	31.9	3
10000.0	104.	77.6	101.	3

show significant departure from spherical symmetry near the inner boundary. Figure 6 shows a plot of ψ for $J=1000$, in the region $1 \leqslant r \leqslant \infty$, $0 \leqslant \theta \leqslant \pi/2$. The asymmetry near $r=1$ is clearly visible.

Following York and Piran, the values of three parameters characterizing the rotating black holes are plotted in Fig. 7. The quantities

$$\varepsilon_l \equiv J/E^2 \tag{58}$$

and

$$\varepsilon_u \equiv J/M^2(M_{AH}, J) \tag{59}$$

are analogous to the usual Kerr angular momentum parameter which tends to unity for an extreme Kerr hole. $M(M_{AH}, J)$ is defined as

$$M(M_{AH}, J) \equiv \left(M_{AH}^2 + \frac{J^2}{4M_{AH}^2} \right)^{1/2} \tag{60}$$

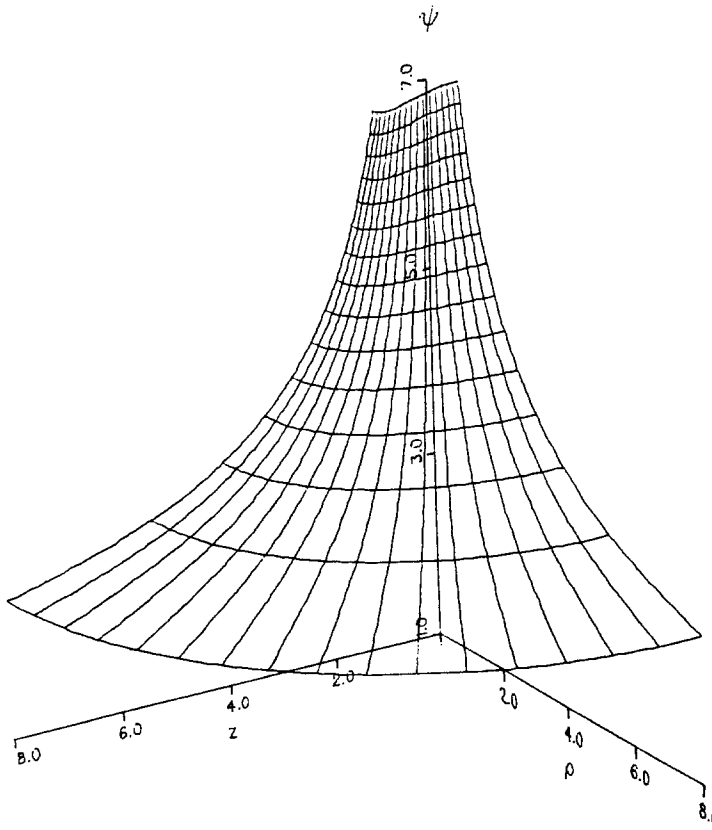

Fig. 6. Conformal factor of spinning black hole ($J=1000$).

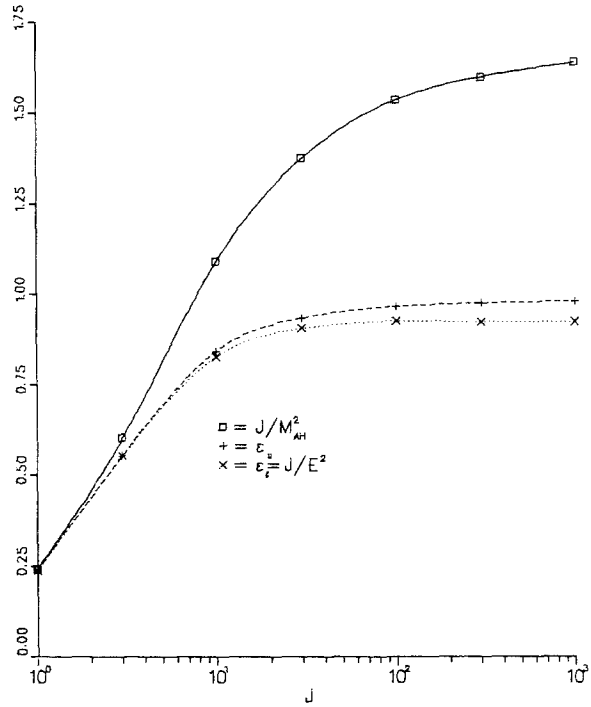

Fig. 7. Angular momentum and energy parameters for spinning holes.

where M_{AH} is related to the area of the apparent horizon of the hole, A_{AH} by

$$M_{AH} = \left(\frac{A_{AH}}{16\pi}\right)^{1/2} \tag{61}$$

The apparent horizon for the current data is located at $r = a$, and A_{AH} is readily calculated once ψ has been determined. Assuming that the new black holes are asymptotically Kerr and that they do not radiate away angular momentum, York and Piran conclude that the quantity

$$\overline{\Delta E} = E - M(M_{AH}, J) \tag{62}$$

is an upper limit on the amount of gravitational radiation present in the space-time. The last three columns of Table VII show the computed values of M_{AH}, $M(M_{AH}, J)$, and E for the various values of J.

The results show in Figure 7 and Table VII differ from those found by York and Piran. They found ε_l and ε_u tending to limiting values of about

0.33 and 0.55, respectively, while Figure 7 suggests asymptotic limits of approximately 0.92 and 0.98. An examination of the code for their solution of the Hamiltonian constraint (for the rotating holes only) suggests that the factor of 1/8 appearing in the nonlinear term of (38) may have been omitted. Because H_s is quadratic in J, the values of J supplied to the code probably corresponded to actual values larger by a factor of $\sqrt{8}$. Among other things, this implies that the asymptotic limit for ε_l should be multiplied by $\sqrt{8}$ which yields a value of 0.93, in good agreement with the current result.

Finally, it would appear that $\overline{\Delta E}$ is considerably smaller for large values of J than calculated previously. Here, the maximum value is 3% of $\overline{\Delta E}$ for the three configurations of greatest angular momentum; Piran and York had obtained a figure of about 25% for $J = 1000$. This would seem to indicate that this family of black holes may radiate less than had been expected, although no firm conclusions can be drawn without actually evolving the initial data.

Although the multi-grid algorithm performed quite well in solving the Hamiltonian constraint for the new black hole families, the performance appeared to be suboptimal. Three, rather than one, coarse grid correction cycles were usually needed to solve a problem to the level of truncation error. This behavior seemed related to the nondefiniteness of the problem induced by the boundary condition (47). It is probable that the relaxation scheme used, or the treatment of the boundary conditions, or a combination of both caused excessive amplification of the lowest frequency residual components, thus degrading the overall convergence. It is quite possible that a different smoothing technique would remedy this problem; this area of the analysis needs additional research.

6. CONCLUSIONS

It seems fairly clear that the multi-grid method should be seriously considered by numerical relativists, especially when the solution of elliptic equations is anticipated to consume an undue amount of computer resources. The combination of generality, adaptability, and efficiency is unique to multi-grid among current methods for solving boundary value problems. It is not an overstatement to say that for large problems a multi-grid solution will be orders of magnitude less costly than a solution obtained by SOR. Thus, making the SOR/multi-grid transition is probably as significant as moving from a VAX to a Cray.

ACKNOWLEDGMENTS

One of us (M.C.) would like to thank Professor U. Ascher for an introduction to the multi-grid method. We would also like to thank T. Piran for supplying us with a copy of his code. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

1. Abramowitz, M., and Stegun, I. A. (1972). *Handbook of Mathematical Functions* (Dover, New York).
2. Bowen, J. M., and York, J. W. (1980). *Phys. Rev. D* **21**, 2047.
3. Brandt, A. (1977). *Math. Comput.* **31**, 333.
4. Brandt, A. (1977). In *Numerical Software III*, John Rice, ed. (Academic Press, New York), pp. 277–318.
5. Brandt, A. (1979). In *Numerical Analysis of Singular Perturbation Problems*, P. W. Hemker and J. J. Miller, eds. (Academic Press, New York/London), pp. 53–147.
6. Brandt, A. (1982). In *Lecture Notes in Mathematics: Multi Grid Methods*, Vol. 960, W. Hackbusch and U. Trottenberg, eds. (Springer Verlag, New York).
7. Choptuik, M. (1982). M.Sc. Thesis (University of British Columbia).
8. Eppley, K. (1977). *Phys. Rev. D* **16**, 1609.
9. Evans, C. R., Smarr, L., and Wilson, J. R. (1983). Preprint.
10. Griffiths, D. F., and Mitchell, A. R. (1980). *The Finite Difference Method in Partial Differential Equations* (John Wiley, New York).
11. Mitchell, A. R., and Wait, R. (1977). *The Finite Element Method in Partial Differential Equations* (John Wiley, New York).
12. Nakamura, Maeda, T. K., Miyami, S., and Sasaki, M. (1980). *Progr. Theor. Phys.* **63**, 1229.
13. O. Murchadha, N., and York Jr., J. W. (1974). *Phys. Rev. D* **10**, 428.
14. Piran, T. (1983). In *Gravitational Radiation*, N. Deruelle and T. Piran, eds. (North Holland, New York), pp. 203–256.
15. Smarr, L. (1979). In *Sources of Gravitational Radiation*, L. Smarr, ed. (Cambridge University Press, Cambridge), pp. 139–159.
16. Varga, R. S. (1962). *Matrix Iterative Analysis* (Prentice–Hall, Englewood Cliffs, N.J.).
17. York Jr., J. W. (1973). *J. Math. Phys.* **14**, 456.
18. York Jr., J. W. (1979). In *Sources of Gravitational Radiation*. L. Smarr, ed. (Cambridge University Press, Cambridge, pp. 83–126.
19. York Jr., J. W., and Piran, T. (1982). In *Spacetime and Geometry*. R. Matzner and L. Shepley, eds. (University of Texas Press, Austin), pp. 145–176.
20. Gassman, G. Private communication.