JOURNAL OF COMPUTATIONAL PHYSICS 72, 467–485 (1987)

A New Finite Element Method for Spherically Symmetric Relativistic Collapse

PATRICK J. MANN

Department of Physics, Queen's University at Kingston, Kingston, Ontario, Canada, K7L 3N6

Received April 30, 1986; revised December 12, 1986

In a previous paper a mixed finite element–finite difference numerical method was used to model relativistic spherical collapse. The method was unsatisfactory in some cases, and a new method is described in this paper. Einstein's equations are written in a standard form, and a weighted residual, moving finite element method is applied to derive a discretization. Tests are made on static and pressureless collapse, in both Newtonian and relativistic situations. Shocks are smoothed using an artificial viscosity, and the results are compared to finite difference codes for the Riemann shock tube problem. Some collapse and bounce models are made, and the resulting shock is investigated. The code seems to be fast and accurate, with reasonable shock descriptions. © 1987 Academic Press, Inc.

1. INTRODUCTION

In a previous paper [7] the mixed finite element–finite difference method was used to compute solutions to Einstein's equations describing spherically symmetric collapse. The results were not completely satisfactory, and a new code has been developed. This scheme derives the fluid equations directly from Einstein's equations in differential form, rather than from a variational principle. A weighted residual finite element method is used for the radial differencing and a second-order finite difference method is used for the time differencing. Nodal movement is allowed through the use of a simplified moving finite element method (see [9, 10]). At all points the method has been designed to be generalizable to less symmetric configurations, where the finite element method can be expected to show its best performance.

This code has performed satisfactorily, with reasonable efficiency. It seems to be comparable to equivalent finite difference codes (for instance, [8] gives detailed descriptions and [11] gives some test results).

The plan of this paper is as follows. First the basic equations are given, and then the discretization is described. Some tests are made, and finally the code is used to produce a "bounce" which illustrates the development of shocks. In all the tests a simple polytropic equation of state is used.

0021-9991/87 $3.00

2. Basic Equations

The physical object to be modelled is a sphere of perfect fluid which evolves under the forces of gravity and pressure. The gravitational forces are calculated in a spacetime described by the metric:

$$ds^2 = -B^2(1-2m/r)\,dt^2 + (1-2m/r)^{-1}\,dr^2 + r^2\,d\Omega^2,$$

where $d\Omega^2 = d\theta^2 + \sin^2\theta\,d\phi^2$ and $m = m(r, t)$, $B = B(r, t)$. Geometric units in which the gravitational constant G and the speed of light c are set to unity are used throughout this paper. The quantity m is just the mass within a radius r, and B is the extra "potential" required by relativity.

The fluid is described by the stress-energy tensor

$$T^{\alpha\beta} = (p+\rho)\,u^\alpha u^\beta + pg^{\alpha\beta},$$

where u^α is the fluid four-velocity, p the pressure, ρ the total energy density, and $g^{\alpha\beta}$ the metric tensor. Greek indices take values from 0 to 3.

The metric satisfies Einstein's equations $G^{\alpha\beta} = 8\pi T^{\alpha\beta}$ ($G^{\alpha\beta}$ is the Einstein tensor) and the stress tensor satisfies the conservation equation $T^{\alpha\beta}_{\ ;\beta} = 0$. Also the baryon or mass conservation equation is needed, which is written as

$$(\rho_0 u^\alpha)_{,\alpha} = 0,$$

where ρ_0 is the rest mass density.

In spherical symmetry these equations are considerably simplified. In particular, the second derivatives in Einstein's equations can all be removed and the fluid equations become an obvious generalization of Euler's equations in Newtonian fluid theory. For convenience make the following definitions:

$$V \equiv u^r/(Bu^t) \qquad (u^\theta = u^\phi = 0) \tag{1}$$

$$A \equiv (1-2m/r)^{1/2}\,[(1-2m/r)^2 - V^2]^{-1/2} \tag{2}$$

$$w \equiv (p+\rho)/\rho_0 \qquad \text{(the relativistic enthalpy)}$$

$$\varepsilon \equiv \rho/\rho_0 - 1 \qquad \text{(specific internal energy density)}$$

$$\tilde{P} \equiv p/\rho_0$$

$$Z \equiv wV[(1-2m/r)((1-2m/r)^2 - V^2)]^{-1/2} \tag{3}$$

$$(\,)' \equiv \partial(\,)/\partial r$$

$$(\dot{}) \equiv \partial(\,)/\partial t.$$

Then the equations for the metric potentials are

$$m' = 4\pi r^2 \rho_0 (1 + \varepsilon + (1-2m/r)\,Z^2/w) \tag{4}$$

$$B'/B = 4\pi r \rho_0 (\tilde{P} + 1 + \varepsilon + 2(1-2m/r)\,Z^2/w)(1-2m/r)^{-1} \tag{5}$$

and for the fluid

$$r^2\rho_0\dot{Z} = -r^2\rho_0 VBZ' - r^2AB(\rho_0\tilde{P}' + \tilde{P}\rho_0') - \rho_0\frac{(\tilde{P}+1+\varepsilon)}{A}\left(r^2(1-2m/r)\,B' + B\left(\frac{V^2}{(1-2m/r)^2}\right)(m-rm')\right) \quad (6)$$

$$r^2\dot{\rho}_0 = -r^2VB\rho_0' - \rho_0[(r^2VB)' + r^2((\ln A)^{\cdot} + VB(\ln A)')] \quad (7)$$

$$r^2\rho_0\dot{\varepsilon} = -r^2\rho_0 VB\varepsilon' - p[(r^2VB)' + r^2((\ln A)^{\cdot} + VB(\ln A)')], \quad (8)$$

with unknowns m, B, ρ_0, ε, and Z.

These equations have been chosen through experience with various versions of the code. In particular a Newtonian test program showed that a momentum variable, $U \equiv V/\rho$. was susceptible to unstable behaviour near the surface where $\rho \to 0$. It was, in general, more robust to evolve V directly. In the relativistic case Z is a modification of the 3-velocity u^r/u^t. Note that $w = 1$ at the surface so Eq. (3) is well behaved.

It is also common in the literature to make a relativistic modification of ρ_0 and ε [14]. For instance, with $D = A\rho_0$ Eq. (7) can be written as

$$r^2\dot{D} = -(r^2VBD)'$$

which is identical to the Newtonian mass conservation equation. This simplifies the ρ evolution, but introduces a nonlinear equation for ρ_0 and ε which must be solved. In practice this has been a difficult and time-consuming problem for the methods of this paper, and considerable improvements in efficiency could be attained by using ρ_0 and ε directly. Comparisons between the methods have been made, showing insignificant differences.

The equations are split into two sets. Equations (4) and (5) are constraint equations which must be solved on a $t =$ constant slice, and (6), (7), and (8) are evolution equations which must be stepped from time slice to time slice.

Of course, an equation of state is required to complete the set of equations. It should be given in the form $p = p(\rho_0, \varepsilon)$ and, for the test purposes of this paper, the simple polytropic form $p = (\gamma - 1)\,\rho_0\varepsilon$, where γ is the polytropic index, is used.

Boundary conditions are required for a complete specification. The surface r_s is defined to be the point at which the density vanishes, so at $r = r_s$, $\varepsilon = 0$, and $\rho_0 = 0$. At the center, symmetry can be applied to show that $u^r = m = 0$. In fact to ensure that singular behavior is avoided $m' = 0$ and $m'' = 0$. This is also a consequence of Eq. (4). The $u^r = 0$ condition gives $V = Z = 0$ at $r = 0$ and $A = 1$.

It is evident from the equations, and also from the form of the metric, that B is only defined up to multiplication by a function of t. This is equivalent to a normalization of the time coordinate and can be used to define a boundary condition on B. A useful condition is $B = 1$ at the surface. In vacuum equations (4) and (5) give $m =$ constant and $B =$ constant, so with this normalization, $B = 1$ everywhere outside the star and the metric is the familiar Schwarzschild form.

3. The Discretization

The mixed FEM–FDM approximates a typical function, for example Z, with a linear combination of basis functions,

$$Z(r, t) \cong \sum Z_i(t)\, N_i(r),$$

where the $Z_i(t)$ are values of the function at nodal points r_i [5]. In the rest of the paper i is assumed to run from 1 to n, with $r_1 = 0$ (the center) and $r_n = r_s$ (the surface). For convenience, the Einstein summation convention will be assumed for the nodal indices i, so the approximation can be written $Z \cong Z_i N_i$.

The simplest basis or shape functions are the "hat" functions which define a linear spline approximation. If $r \in [r_{i-1}, r_i]$ the formula is

$$\begin{aligned} N_{i-1} &= \frac{r_i - r}{r_i - r_{i-1}} \\ N_i &= \frac{r - r_{i-1}}{r_i - r_{i-1}} \\ N_j &= 0 \qquad \text{for all } j < i-1 \text{ and } j > i \end{aligned} \tag{9}$$

(see, for instance, [5] for a detailed description).

The discretization is derived from the weighted residual method as follows. A differential equation $D(Z) - F(Z) = 0$, where D is a differential operator, is evaluated with the approximation to give a residual $R = D(Z) - F(Z)$. This residual is multiplied by a set of weight functions W_i and then integrated over the region of interest, giving a set of equations for the unknowns Z_i. A standard choice of weights is the Galerkin choice, $W_i = N_i$, but, in general, W_i is chosen to localize the integration.

The quantities ρ_0, ε, Z, V, and $\ln A$ are approximated in this manner. However, m is inherently of cubic form in r, and B of quadratic form, so the linear approximation is unsuitable. A method of solving ordinary differential equations of the same form as (4) and (5) has been described in [6]. It uses cubic Hermite splines and is quite suitable for this problem.

The cubic Hermite spline approximation, with m as an example, is written

$$m(r, t) \cong m_{1i}(t)\, N_{1i}(r) + m_{2i}(t)\, N_{2i}(r),$$

where $m(r_i, t) = m_{1i}(t)$ and $m'(r_i, t) = m_{2i}(r_i, t)$. The shape functions N_{ai} are given explicitly in [3] and in most FEM texts (i.e., [5]). The typical equation $m' = f(m, r)$ is then applied directly to give

$$m_{2i} = f(m_{aj} N_{aj}(r_i), r_i) \qquad \text{(summation: } a = 1, 2 \text{ and } j = 1, \ldots, n)$$

and the weighted residual method applied with only one set of shape functions to give

$$\int_{r_{i-1}}^{r_i} (m_{ak} N'_{ak}(r))\, W_i(r)\, dr = \int_{r_{i-1}}^{r_i} f(m_{ak} N_{ak}(r), r)\, W_i(r)\, dr.$$

Here the weights W_i are assumed to vanish outside the region $[r_{i-1}, r_i]$, which will give a two-point discretization. Performing the integration gives a set of equations of the form:

$$\begin{aligned} m_{1i} &= F_1(m_{1i}, m_{2i}, m_{1i-1}, m_{2i-1}) \\ m_{2i} &= F_2(m_{1i}, m_{2i}, m_{1i-1}, m_{2i-1}). \end{aligned} \tag{10}$$

The method of solution is to assume m_{11} and m_{21} are known, and then to use Eq. (10) to step out, assuming the $i-1$ values are known at each step. If W_i is suitable then this is a robust and accurate method of solving these equations (see [7]). A suitable W_i is the Galerkin weight N_{1i} (restricted to $[r_{i-1}, r_i]$). Fortunately Eq. (4) is linear in m, so Eq. (10) are linear in m_{ai} and can be easily solved, giving a completely implicit radial discretization.

The B equation (5) is solved in an identical manner, although the boundary condition is incorporated by choosing B at the center, stepping out to the surface, and then normalizing by B_n to ensure that $B_n = 1$. Note also that Eq. (4) for m has no B dependence, so that it can be solved first. Then Eq. (5) is linear in B and the implicit equations can be easily solved.

Therefore, given the evolution variables V, ρ_0, and ε, the constraint equations (4) and (5) can be solved for m_{ai} and B_{ai}. It remains to evolve the evolution variables from a previous time slice $t = t^{r-1}$ to a new time slice $t = t^r$.

The Z equation is simplest, being of the form $\dot{Z} = f$, where f is a function of the unknowns and their radial derivatives only. Therefore the weighted residual method can be applied trivially to get

$$\int_{r_{i-1}}^{r_{i+1}} (Z_k N_k)^{\cdot}\, W_i\, dr = \int_{r_{i-1}}^{r_{i+1}} f W_i\, dr, \tag{11}$$

where the weights W_i are assumed to have support $[r_{i-1}, r_{i+1}]$.

The time derivative can be performed directly, but at this point the moving finite element method can be introduced [9, 10]. This is necessary because the body can be expected to change its size and structure drastically as it evolves, making a moving grid crucially important. Assume that the grid nodes r_i are themselves functions of t: $r_i = r_i(t)$. Then note that the N_i are functions of the r_j so the approximation can be written

$$Z(r, t) \cong Z_i(t)\, N_i(r, r_j(t)),$$

giving a time derivative

$$\dot{Z} \cong \dot{Z}_i N_i + Z_i \frac{\partial N_i}{\partial r_j} \dot{r}_j \tag{12}$$

with $\dot{r}_j$ being the grid velocity. This velocity will be calculated to model the collapse, but for this part of the paper it is just assumed to be a known function of t. For convenience write $\dot{N}_i \equiv (\partial N_i/\partial r_j)\, \dot{r}_j$ and apply this formula to (11) to get

$$\dot{Z}_k \int_{r_{i-1}}^{r_{i+1}} N_k W_i \, dr = \int_{r_{i-1}}^{r_{i+1}} (f - Z_k \dot{N}_k)\, W_i \, dr. \tag{13}$$

Both sides can be integrated, giving a set of equations of the form

$$c_{i-1} \dot{Z}_{i-1} + c_i \dot{Z}_i + c_{i+1} \dot{Z}_{i+1} = F_i.$$

At the center, $\dot{Z}_1 = 0$. At the surface, the weight function W_i is assumed to cut off at $r = r_n$, so that the $[r_n, r_{n+1}]$ contribution to the integral vanishes and the equation $c_{n-1} \dot{Z}_{n-1} + c_n \dot{Z}_n = F_n$ remains.

Therefore a tridiagonal matrix equation $C\dot{\mathbf{Z}} = \mathbf{F}$ must be solved to derive a set of equations $\dot{Z}_i = g_i$. This is not particularly time-consuming, but tests on a Newtonian collapse have shown that this method is particularly prone to numerical instability. Instead the finite element technique of "condensation" can be applied (see [1]). In this technique the time derivatives of Z_k for the ith equation are assumed to be equal, so that Eq. (13) becomes

$$\dot{Z}_i \int_{r_{i-1}}^{r_{i+1}} \sum_{k=i-1}^{i+1} N_k W_i \, dr = \int_{r_{i-1}}^{r_{i+1}} (f - \dot{Z}_k N_k)\, W_i \, dr. \tag{14}$$

Since $N_{i-1} + N_i = 1$ in the interval $[r_{i-1}, r_i]$, (14) becomes

$$\dot{Z}_i = \int_{r_{i-1}}^{r_{i+1}} (f - \dot{Z}_k N_k)\, W_i \, dr \left(\int_{r_{i-1}}^{r_{i+1}} W_i \, dr \right)^{-1}. \tag{15}$$

Then there is no matrix equation to solve, although some accuracy has been lost.

The choice of weight is very important if numerical instability is to be avoided. The standard weight $W_i = N_i$ invariably produces oscillations in Z both near the center and near the surface, caused by the r^{-2} and ρ_0^{-1} terms in the right-hand side expression f. It is necessary to use a weight $r^2 \rho_0 N_i$, and then to ensure that $r^2 \rho_0$ is multiplied into f directly. Equation (6) is written in a form which illustrates this multiplication.

Equation (15) is in the form $\dot{Z}_i = g_i$, so some standard finite difference method can be used to solve for the motion. The second-order predictor-corrector is simple and robust, giving

$$Z^t = Z^{t-1} + \Delta t\, g_i^{t-1} \qquad \text{(predictor)}$$

$$Z^t = Z^{t-1} + 0.5\, \Delta t (g_i^{t-1} + g_i^t) \qquad \text{(corrector)}.$$

The two equations (7) and (8) have the same overall form as the Z equation, but the left sides contain the derivative $(\ln A)^{\cdot}$. It is in theory possible to perform this derivative analytically and to derive expressions for $\dot{\rho}_0$ and $\dot{\varepsilon}$ which do not have time derivatives on the right side. In practice this is difficult, results in extremely complicated expressions, and is never used. Instead $\ln A$ must be differenced. The typical form can be written as

$$\dot{\varepsilon} = f + g(\ln A)^{\cdot}. \tag{16}$$

Then the weighted residual method can be applied as for the Z equation to get

$$\dot{\varepsilon}_k \int_{r_{i-1}}^{r_{i+1}} N_k W_i \, dr = \int_{r_{i-1}}^{r_{i+1}} (f - \varepsilon_k \dot{N}_k) W_i \, dr + \int_{r_{i-1}}^{r_{i+1}} g(\dot{a}_k N_k + a_k \dot{N}_k) W_i \, dr, \tag{17}$$

where $a \equiv \ln A$. Applying condensation to the left side only, and not to the a derivative, gives

$$\dot{\varepsilon}_i = \left(\int_{r_{i-1}}^{r_{i+1}} (f - \varepsilon_k \dot{N}_k) W_i \, dr + \int_{r_{i-1}}^{r_{i+1}} g(\dot{a}_k N_k + a_k \dot{N}_k) W_i \, dr \right) \div \left(\int_{r_{i-1}}^{r_{i+1}} W_i \, dr \right)$$

The form is then $\dot{\varepsilon}_i = F_i + G_{ik} \dot{a}_k$, and the time differencing can proceed in the same fashion previously described. Only one modification is required as an extra step to predict $\dot{a}_k$. The method is as follows:

(a) estimate $\dot{a}_k$ at time t^{t-1}. (18)

(b) $\varepsilon_i^t = \varepsilon_i^{t-1} + \Delta t(F_i^{t-1} + G_{ik}^{t-1} \dot{a}_k^{t-1})$.

(c) $\varepsilon_i^t = \varepsilon_i^{t-1} + (\Delta t/2)(F_i^{t-1} + F_i^t) + (1/2)(G_{ik}^{t-1} + G_{ik}^t)(a_k^t - a_k^{t-1})$.

Part (a) just uses the estimate for $\dot{a}_k$ from the previous time step. Part (b) is a predictor and (c) is the corrector.

The weight for the ε discretization must also remove the r^{-2} and ρ_0^{-1} terms, so it is the same as the weight for Z: $W_i = r^2 \rho_0 N_i$.

The ρ_0 discretization is identical to the ε discretization except for the weight. There is no ρ_0^{-1} term in (7) so it is sufficient to use $W_i = r^2 N_i$.

An evolution step can now be outlined:

(1) A complete data set is given on time slice $t = t^{t-1}$.

(2) A predictor step is made to predict values of ρ_0, ε, and Z at $t = t^t$.

(3) On this t^t slice the pressure p and its associated variable $\tilde{P}$ can be calculated at any point from the equation of state. This in turn allows w to be evaluated as needed.

(4) V_i is calculated through the inversion of Eq. (3),

$$V = Z(1 - 2m/r) \left(\frac{1 - 2m/r}{w^2 + Z^2(1 - 2m/r)} \right)^{1/2},$$

and then a_i through the definition of A:

$$\ln A = (\tfrac{1}{2})(\ln(1-2m/r) - \ln[(1-2m/r)^2 - V^2]).$$

(5) m_{ai} and B_{ai} are computed with the constraint solver.

(6) A corrector step is made to refine the ρ_0, ε, and Z estimates, and steps (3), (4), and (5) are repeated to compute the other variables.

The V and a calculation of step (4) require a knowledge of m, which is only calculated in step (5). Fortunately m occurs in the combination $1-2m/r$, which is insensitive to changes in m unless the body is highly relativistic. Therefore the m used in step (4) is the previous m^{t-1}, or the m from the predictor. In highly relativistic situations, step (4) and the m solution can be iterated but comparison tests have shown this to be unnecessary.

The evolution method outlined above is explicit, which means that the time step Δt must be limited for stability. It is possible to make it implicit by iterating the corrector until convergence occurs, but this puts convergence constraints on the size of the time step which are equivalent to the stability constraints.

The stability constraint is in the familiar Courant–Friedrich–Hilbert form (see [1]), which is written as

$$\Delta t < \delta_c \min_i (h_i/c_i), \tag{19}$$

where c_i is the characteristic velocity at node i, and δ_c is a constant which includes all the details of a complete analysis. This form is not complete, as the equations have both parabolic and hyperbolic behavior, so terms of the form $\Delta t < \delta_c h_i^2$ can be expected. In practice, Eq. (19) seems sufficient if $c_i = |BV_i| + c_{si}$, where c_s is the speed of sound. Note that BV is the characteristic velocity for the ρ_0 and ε equations, and c_s is the characteristic velocity for the Z equation.

The time step should also be limited by accuracy considerations, but formal estimates of accuracy are difficult to derive for such complicated nonlinear equations. Instead ρ_0 and ε are allowed to change only by some maximum amount, which is given in this code as a percentage of central ρ_0 and ε. In general, a value of 10% has been sufficient.

The grid velocity can be used to great advantage in following steep gradients and shocks. Such methods are detailed in, for instance, [9, 10] but they invariably result in the necessity for completely implicit differencing. Therefore a simple Lagrangian movement is used in this code. This is just $\dot{r}_i = V_i B_i$, the three velocity in the slice. However, this couples the nodal spacing to the velocity, and oscillatory instability nodes can be disastrously amplified. To stop this, some sort of damping is required. After some experimentation, the formula

$$\dot{r}_i = V_i B_i + \frac{2}{3}|V_i B_i| \left(\frac{h_{i+1}}{h_i - h_{\min}} - \frac{h_i}{h_{i+1} - h_{\min}} \right),$$

with $\dot{r}_n = V_n B_n$ and $\dot{r}_1 = 0$, has proved successful; $h_{\min}$ is some minimum allowed spacing. A suitable value for $h_{\min}$ is $r_n(20n)^{-1}$. For convenience $\dot{r}_i$ is calculated only once on the t^{t-1} slice, and then assumed to be constant in the interval $[t^{t-1}, t^t]$.

All the integrals have been computed with a two-point Gaussian quadrature formula. At regular periods, 5-point Gaussian quadratures were tested but, in all cases, there have been no significant differences.

3. Simple Initial Tests

There are many choices which must be made to produce the FEM code. If therefore seemed reasonable to test the general method first on a Newtonian set of equations. This considerably simplifies the equations and allows the development of a test code which can be run many times. This code was written, and some necessary choices analyzed and made. First it showed that using a momentum $U = V\rho$ was unstable and that V had to be evolved directly. Second, the weights $r^2\rho_0 N_i$ were indicated and, third, a time-step limit of $\delta_c = 0.2$ was necessary.

With these choices the method successfully evolved static initial data and pressureless "dust" collapse. The static evolutions showed an oscillation in surface velocity, which has also been seen in the full relativistic case. This is almost certainly due to a poorly defined "surface" which couples through the grid velocity directly to the V equation. It should be possible to damp the surface movement, but a suitable technique has not been found. Other researchers have used either a point where ρ_0 reaches a given nonzero value or have worked in co-moving coordinates where the problem never arises. Since the surface oscillation does not seem to affect the rest of the object the code has not been modified.

The pressureless collapse and, indeed, any collapse with $\gamma \leqslant \frac{4}{3}$, has shown the development of a ρ spike at the center of the body. These models are physically unstable so such behavior is to be expected and is not serious. In particular, the spike only became significant (visible on a graph) when the object was well within its gravitational radius $r = 2m$ and so does not affect the relativistic runs.

In the relativistic case, a global instability appeared during the evolution of initially static, and physically stable, data. Experiment showed that the ε evolution was not reacting to the velocity evolution correctly. This meant that changes in the pressure were lagging behind density changes, allowing unstable amplification of the gravitational terms, and corresponding collapse. A solution requires that ε react faster to velocity changes, and this can be implemented by using a forward-weighted time differencing for ε,

$$\begin{aligned} &\text{(a)} \quad \text{estimate } \dot{a}_k \text{ at time } t^{t-1}, \\ &\text{(b)} \quad \varepsilon_i^t = \varepsilon_i^{t-1} + \Delta t(F_i^{t-1} + G_{ik}^{t-1}\dot{a}_k^{t-1}), \\ &\text{(c)} \quad \varepsilon_i^t = \varepsilon_i^{t-1} + \Delta t\, F_i^t + G_{ik}^t(a_k^t - a_k^{t-1}), \end{aligned} \tag{20}$$

in contrast to the center-weighted scheme of Eq. (18). This is of first-order accuracy

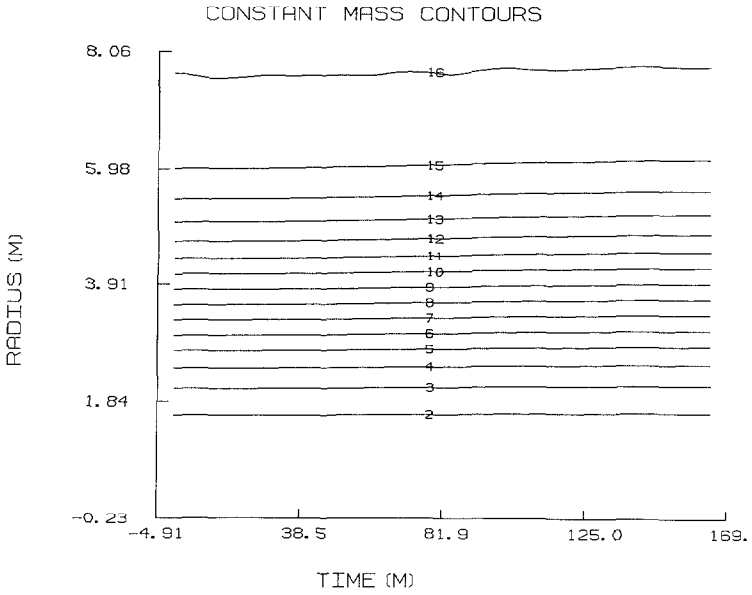

FIG. 1. Constant mass (m) curves in the r, t plane are graphed for an initially static relativistic evolution. The contours are at intervals of $\frac{1}{15}$ of the total mass of the object. The run uses a $\gamma = \frac{5}{3}$ polytrope initial model with 32 nodes. This model has central enthalpy $w = 1.35$ and is physically stable, but close to being unstable.

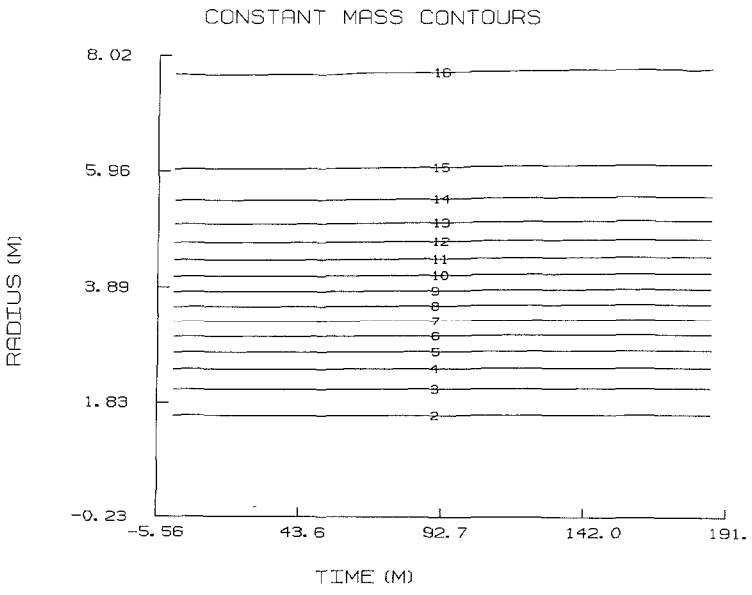

FIG. 2. A 128-node model equivalent to that of Fig. 1.

in Δt but it is the simplest way of stabilizing the method. It is interesting to note that this affect is not noticeable in the Newtonian code, but is strictly a relativistic property.

In Figs. 1 and 2, two representative static evolutions are illustrated. Both are $\gamma = \frac{5}{3}$ relativistic polytropes, with a central enthalpy of 1.35. The turnover from a stable polytrope to an unstable one occurs at $w = 1.38$, so the model is almost physically unstable, and numerical errors will be particularly obvious. A 32-node run is graphed in Fig. 1, and it is evident that the surface is oscillating, but with constant amplitude, as mentioned previously. Mass is conserved to 0.02% after 1000 time steps. In Fig. 2 a 128-node run is graphed, with an obvious increase in accuracy. After 4000 steps, mass is conserved to better than one part in 10^6 (the accuracy of the graphing program).

4. Shocks

Evolving fluids can produce shocks. The numerical method is not capable of modelling a perfect shock because of the restricted resolution of the grid. Shock energy is channelled by the fluid equations down to the smallest scales allowed by the grid and produces unacceptable oscillations in the results. A method to smooth these energies over some reasonable width is required. Many schemes are available in the literature, but the simplest one is still the introduction of a viscosity term in the fluid equations. A standard bulk viscosity can be added to $T^{\alpha\beta}$, with the result that a viscous "pressure" Q must be added to p wherever p occurs. A study of this addition has been made for the relativistic collapse case in [8], and the method of this paper uses essentially the same techniques.

Some researchers, (see [2–4, 13]) have introduced a viscosity or, at least, a special differencing which is an implicit viscosity, into particular terms only. Such methods will not satisfy the Rankine–Hugoniot jump conditions across discontinuities, but tests by [4] show little difference in practice and give excellent results. It is difficult to extend their use to finite element methods. Advanced FEM techniques are available in [1] but have not yet been used for relativistic fluid problems.

Therefore, the only modification which has been made is to include a term Q to p everywhere. The Q is large where there are steep gradients in V and is given by

$$Q_{i-1/2} = -K_q \rho_0 h_i^2 |(VB)'_{i-1/2}| \, (VB)'_{i-1/2},$$

where K_q is a constant and the $i-\frac{1}{2}$ indicates that Q is defined on each element $[r_{i-1}, r_i]$. The derivatives are calculated using the spline approximations on the element, each evaluated at the midpoint. This formula will smooth the shock over about K_q grid nodes, with values of 2 or 3 for K_q considered to be quite reasonable. The formula is not usuable directly, because Q is not continuous and can upset the FEM integration. Therefore a cubic Hermite spline is fitted through the points with

$$Q_{1i} \equiv 0.5(Q_{i-1/2} + Q_{i+1/2})$$

and

$$Q_{2i} \equiv \frac{Q_{i+1/2} - Q_{i-1/2}}{0.5(h_{i+1/2} + h_{i-1/2})}.$$

A cubic interpolation could be used, but this method is fast and has produced reasonable results. Note that the amplitude of Q will, in general, be decreased so larger values of K_q are required.

A considerable literature on Newtonian shocks and numerical methods is available, and there are some papers for relativistic shocks. A particular test case is that of a one-dimensional Riemann shock tube. The Newtonian case can be solved

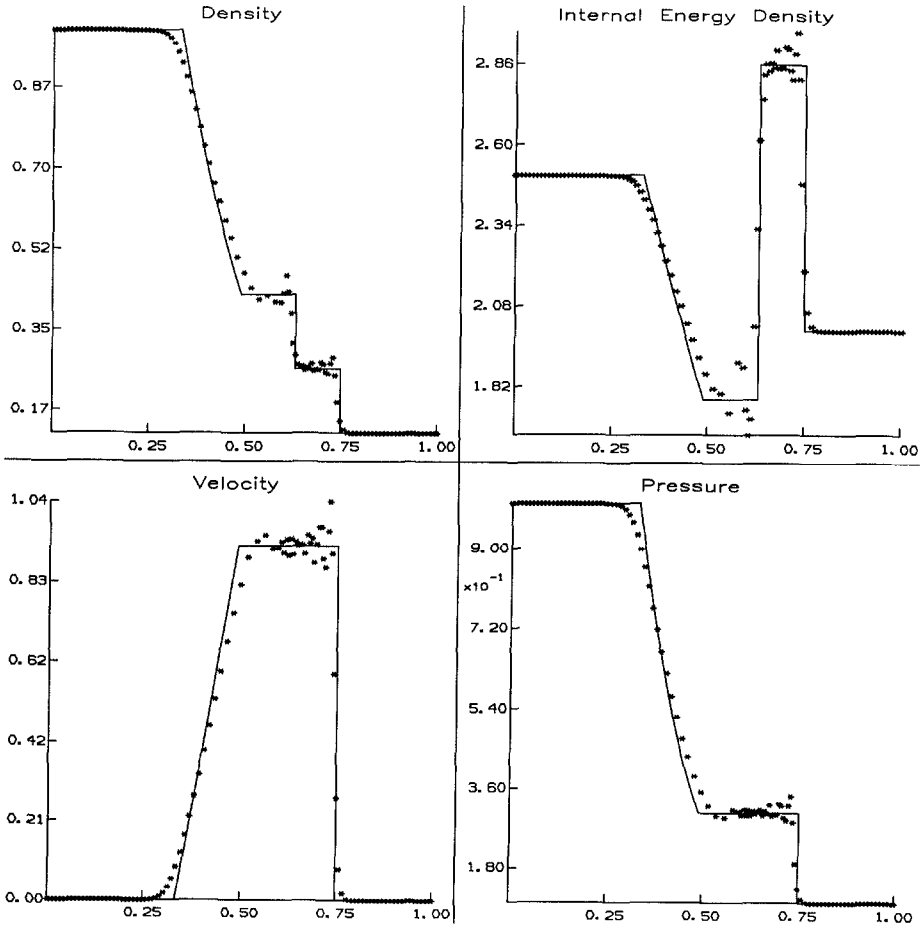

FIG. 3. The Newtonian shock tube modelled with 101 nodes, at a time $t = 0.15154$. The courant multiplier (δ_c) is 0.2, and $K_q = 2$. All nodal values are graphed as asterisks, and the analytic solution is graphed as a solid line. Initial conditions are those of [13].

analytically (see [12], for instance), and the relativistic case can be reduced to the point where only one ordinary differential equation must be solved numerically [2]. Therefore this example is well suited for comparison with numerical techniques. A comprehensive test of finite difference methods is made by [13], and some finite element methods are tested in [1], for the Newtonian case. Some FDM's for the relativistic case are described in [2–4].

Little modification is reqired to specialize the FEM of this paper to the shock tube. In the Newtonian case the results are summarized in Fig. 3. This figure can be compared directly to those of [13]. It is evident that advanced FDM methods are much better and that the FEM is roughly equivalent to the Crank–Nicholson FDM. This is to be expected as the weighted residual, condensed FEM, does

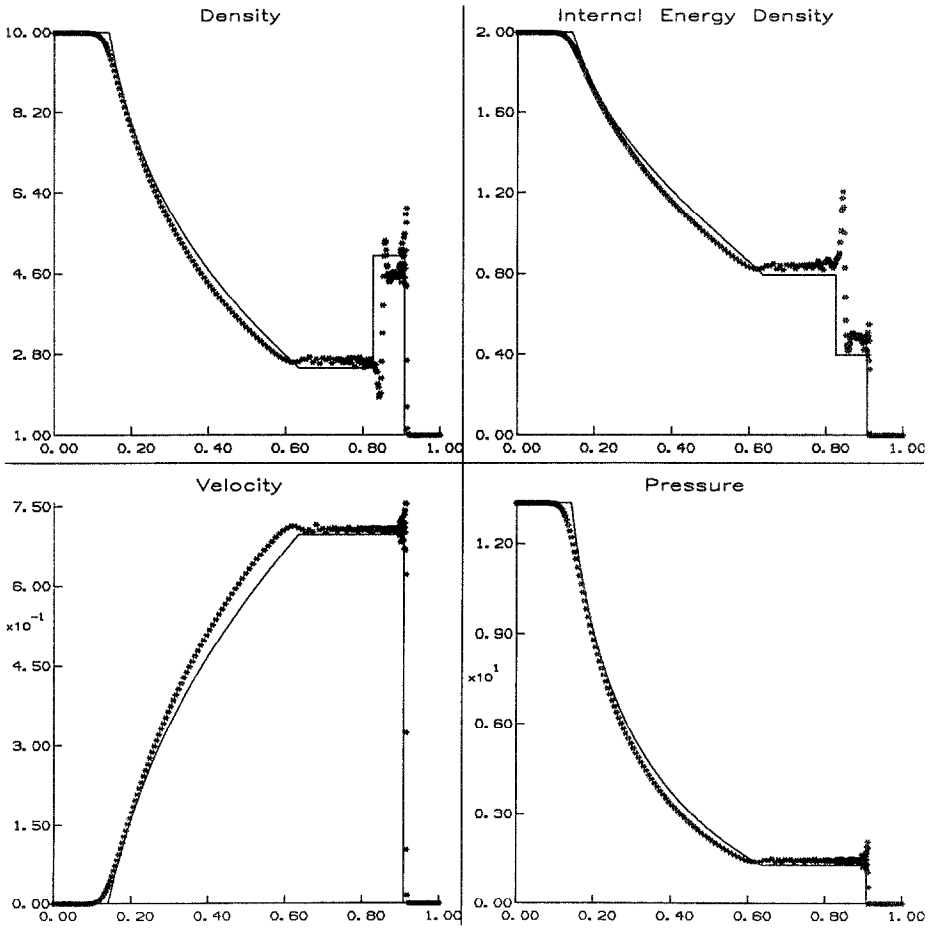

FIG. 4. The relativistic shock tube modelled with 256 nodes, at a time $t = 0.5$, $\delta_c = 0.5$, and $K_q = 2$, with initial conditions those of [2].

produce difference schemes which are similar to the Crank–Nicholson. It should be noted here that there is no smoothing on the internal energy equation. The artificial viscosity operates directly on V oscillations only. The large internal energy oscillations near the shock and the contact discontinuity are due to this lack of smoothing, but a useful FEM solution for the relativistic case is not yet available.

The relativistic shock of [2] is very strong, and gives a stringent test of any numerical method. Unfortunately the shock has a structure which requires rather small nodal spacing, but results from the FEM code are summarized in Fig. 4. Again it is evident that the refined techniques of [4] are better, but that the overall characteristics of the shock are adequately modelled. One obvious problem is in the consistent overestimate of the velocity. This affect has also been seen in [2] but its cause is unknown.

The advanced FEM of [1] are, in the Newtonian shock tube, as good as the best of the FDM's, and further development of the relativistic FEM is indicated. The general behavior of the shock is adequately modelled. however, and the artificial viscosity is both simple and easy to implement in relativity and in less symmetric configurations.

5. Some Evolutions

A complete collapse of a pressureless fluid is illustrated in Figs. 5 and 6 (for 32 and 128 nodes, respectively). The initial model is the $w = 1.35$, $\gamma = \frac{5}{3}$ static one, but

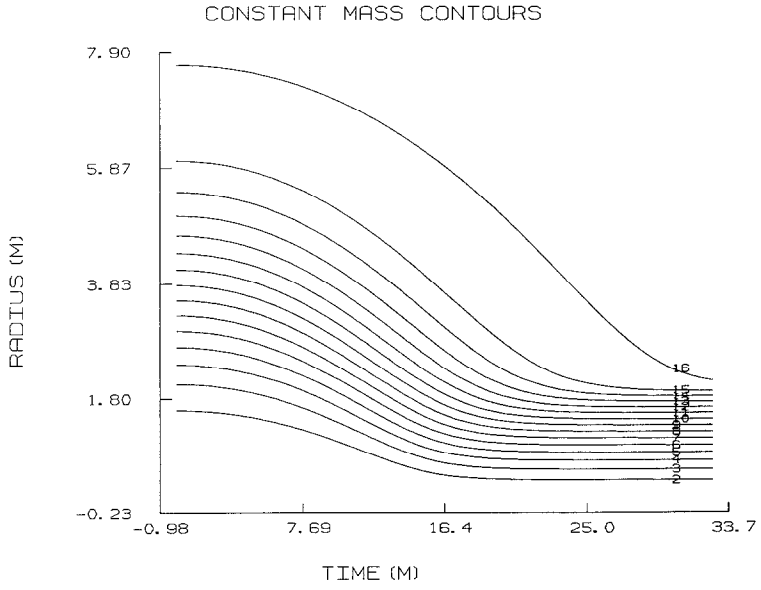

FIG. 5. The mass curves from a $p = 0$ relativistic collapse model with 32 nodes. The initial data is the relativistic polytrope of Fig. 1 with the internal energy set to zero.

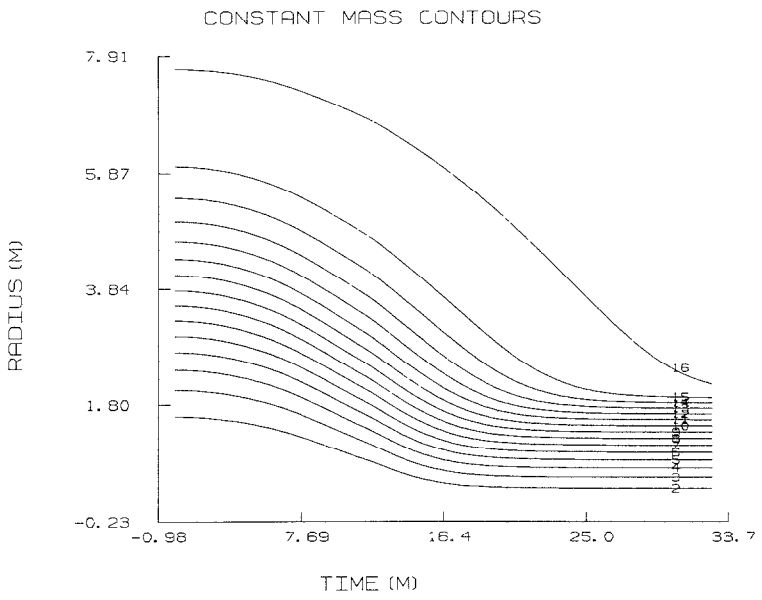

FIG. 6. A 128-node model equivalent to that of Fig. 5.

the internal energy ε is set to zero. With $Q=0$ this ensures that the collapse is indeed pressureless. Comparison of the two figures shows excellent agreement with correct timescales. The 128-node run conserves mass to less than 0.3%, while the 32-node run is slightly worse at 0.6%. In Fig. 7 the grid nodes are graphed, showing clearly the grid movement as the Lagrangian nodes accumulate against the forming horizon.

Shock production is more difficult due to the use of simple polytropic equations of state. However, it is possible to set up initial conditions which result in a bounce of the surface layers off a hard inner core. As an example, a static model with $w=1.02$ and $\gamma=\frac{5}{3}$ can be perturbed by reducing the internal energy to 40% of its original value. This is illustrated for 32 nodes (Fig. 8) and for 128 nodes (Fig. 9). The runs are similar, showing a collapse and then a bounce and shock near the core which propagates out and forces the surface to explode. It is evident that the finer grid allows a sharper shock with a harder bounce, so the 128-node run expands at a significantly higher speed (see Fig. 10).

The mass is conserved to 0.1% in the 128-node run, and to 0.2% in the 32-node run, so the harder bounce is probably more accurate. In both cases $K_q=2$, but the grid spacing is larger in the 32-node run, so there is more smoothing. Indeed, increasing K_q in the 128-node run softened the bounce, giving results closer to the 32-node run but, at the same time, the mass increased.

There is also a post-shock oscillation evident in the velocity (Fig. 10). The artificial viscosity included is insufficient to completely damp this but, as mentioned

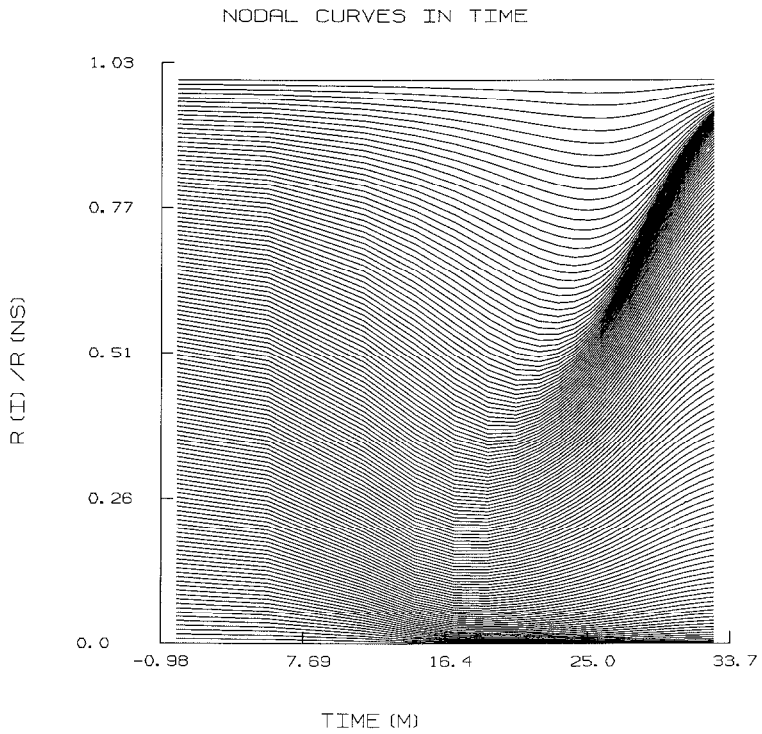

FIG. 7. Grid node curves for the $p=0$ collapse of Fig. 6. Initially the nodes are equally spaced, but then they collapse and concentrate near the center. The buildup of matter on the horizon, which is an object of the coordinate choice, is clearly seen.

in the previous paragraph, increasing Q results in a softer bounce which seems less accurate. Decreasing Q ($K_q = 1$, for instance) makes little difference to the overall characteristics, but the oscillation becomes huge and, when the shock reaches the surface, the velocity suddenly increases and the mass increases again. Therefore a value for Q which allows some oscillation seems to be necessary.

6. Accuracy and Efficiency

The nonlinearity of the equations make it extremely difficult to get estimates of the accuracy. Simple expansions can be made, indicating, as expected, that the Z and ρ_0 equations are accurate to order Δt^2, the ε equation evolution to order Δt, and the spatial parts accurate to at least Δr^2.

However, it is evident that the mass conservation of the 128-node runs is only half that of the 32-node runs, which is not in agreement with the order of accuracy estimates. These estimates, therefore, give only rough ideas of the accuracy.

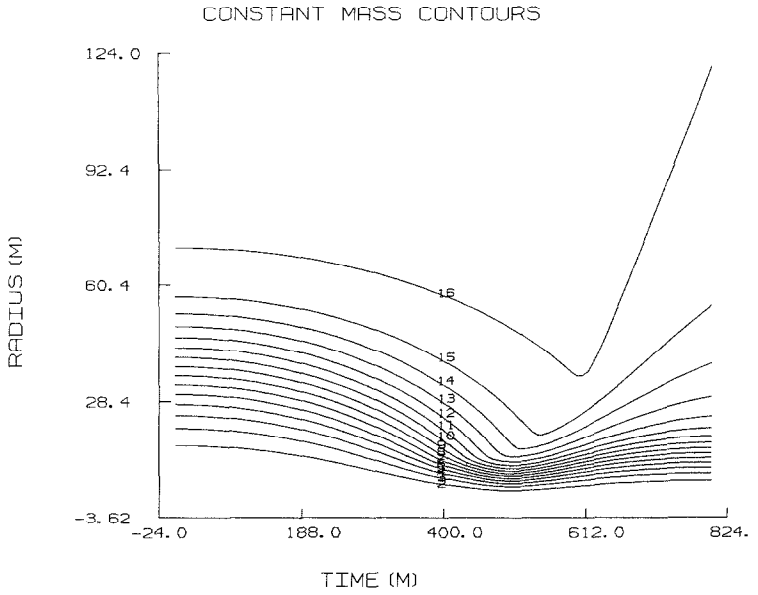

FIG. 8. The mass curves from a collapse and bounce model with 32 nodes. The initial data is a perturbed $w = 1.02$ polytrope, with the internal energy decreased to 40% of its initial value with $\delta_c = 0.2$ and $K_q = 2$.

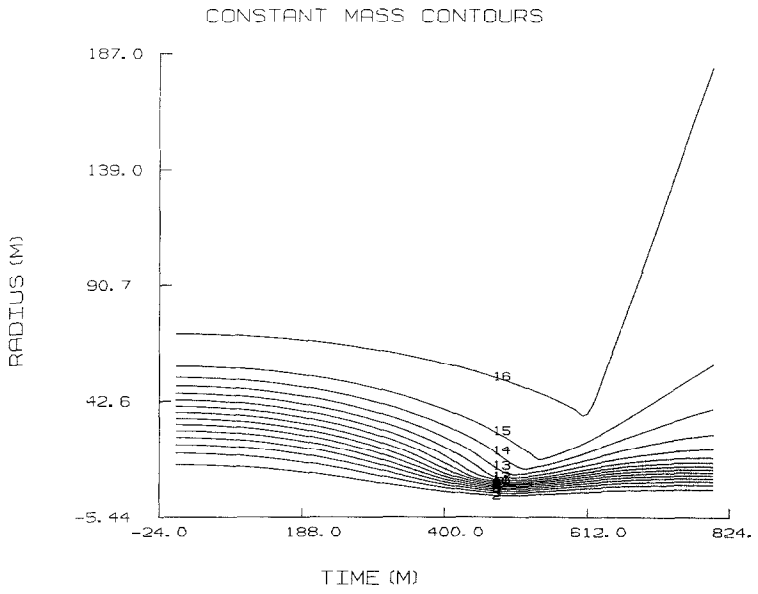

FIG. 9. A 128-node model equivalent to that of Fig. 8. The time scales are similar, but it is clear that this run has bounced much harder.

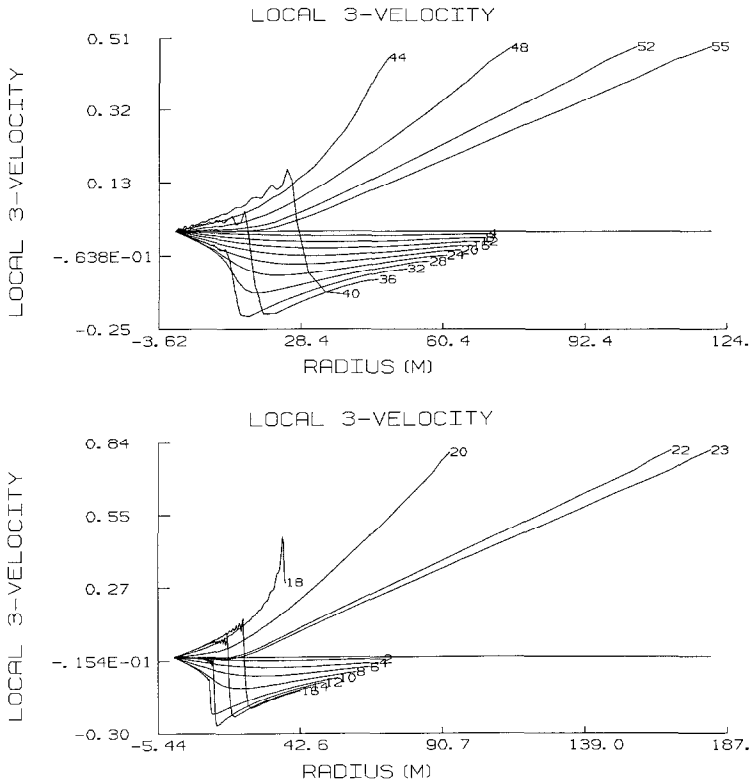

FIG. 10. Velocity snapshots for the two runs shown in Figs. 8 and 9. The upper curve has 32 nodes and the lower has 128 nodes. It is clear that the finer grid shows a sharper shock with a harder bounce, giving outflow velocities 50% greater than the 32-node run.

The code is written in Fortran, and is run on an IBM 3081G computer at Queen's University, Canada. It is written in an easily modifiable and understandable style but with little regard for efficiency. However, the IBM VS Fortran compiler with its optimizer decreases runtimes by about 40% and further significant decreases are not expected. Graphics routines are not included in the basic code. With 64-bit arithmetic, the runs took a consistent 0.0017 s per time step per node, with typical runs taking a few minutes of cpu time to complete.

7. CONCLUSIONS

The original purpose of this work was as an introduction to the axi-symmetric collapse problem, and all the techniques used in this paper are, in theory, straightforward to extend to that problem. Indeed the finite element method can be expected to show better performance in such less-symmetric configurations. Since

the spherically symmetric code seems to be comparable in accuracy and efficiency to similar finite difference codes, it is reasonable to continue toward the axi-symmetric case.

The axi-symmetric case will be a large numerical problem, so efficiency is important. The easiest way to get short run times is through the use of coarse grids but, as shown in the collapse results, the outgoing shock can be smoothed to much. [redacted] would be advantageous, especially with respect to the global effects of artificial viscosity.

Further refinement is also required in the shock modelling. A better moving grid should be implemented and an investigation of standard Newtonian smoothing methods made.

ACKNOWLEDGMENTS

I thank Professor D. W. Sciama for his continuing help and encouragement, which have been invaluable. The work was partially completed during my stay at SISSA, in Trieste, Italy, and I particularly acknowledge the technical expertise of Dr. A. Nobile.

REFERENCES

1. A. J. BAKER, *Finite Element Computational Differential Equations in Science and Engineering* (Wiley, New York, 1983).
2. J. CENTRELLA AND J. R. WILSON, *Astrophys. J. Suppl. Ser.* **54**, 229 (1984).
3. J. F. HAWLEY, L. L. SMARR, AND J. R. WILSON, *Astrophys. J.* **277**, 296 (1984).
4. J. F. HAWLEY, L. L. SMARR, AND J. R. WILSON, *Astrophys. J. Suppl. Ser.* **55**, 211 (1984).
5. L. LAPIDUS AND G. F. PINDER, *Numerical Solutions of Partial Differential Equations in Science and Engineering* (Wiley, New York, 1982).
6. P. J. MANN, *Comput. Phys. Commun.* **30**, 127 (1983).
7. P. J. MANN, *J. Comput. Phys.* **58**, 377 (1985).
8. M. MAY AND R. H. WHITE, *Math. Comput. Phys.* **7**, 219 (1967).
9. K. MILLER AND R. N. MILLER, *SIAM J. Numer. Anal.* **18**, 1019 (1981).
10. K. MILLER, *SIAM J. Numer. Anal.* **18**, 1033 (1981).
11. S. L. SHAPIRO AND S A. TEUKOLSKY, *Astrophys. J.* **235**, 199 (1980).
12. A. H. SHAPIRO, *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Vol. II (Ronald Press, New York, 1953), p. 1007.
13. G. A. SOD, *J. Comput. Phys.* **27**, 1 (1978).
14. J. R. WILSON, in *Sources of Gravitational Radiation*, edited by L. Smarr (Univ. of Cambridge Press, Cambridge, England, 1979), p. 423.