Schwarzschild Spacetime in Novikov Coordinates: a Numerical Solution

Randall Correll Mijan Huq Pablo Laguna
Richard Matzner

4 July 1995

1 Overview

We have been developing a numerical scheme to evolve blackhole spacetimes using the standard ADM 3+1 formulation.[1] That is, we consider a spacelike hypersurface defined by its 3-metric and its extrinsic curvature, the momentum conjugate to the 3-metric. We then evolve the hypersurface according to the Einstein equations in the 3+1 formalism. An interesting test case is a Schwarzschild spacetime described by Novikov coordinates because the 3-metric components in these components are time dependent even though the spacetime is static. This allows us to test our numerical evolution against a known, time-dependent solution.

2 Novikov Coordinates

Novikov coordinates represent observers who are freely falling into the black hole. MTW[2] has a thorough description from which we summarize the main features. The Novikov radial coordinate, R, is related to the Schwarzschild radial coordinate, r, at the height of its trajectory by

$$R = (r_{max}/2M - 1)^{1/2},$$

while the Novikov time coordinate is simply the proper time, τ, of the infalling observer. This transformation leads to the Schwarzschild spacetime given in Novikov coordinates by

$$ds^2 = -d\tau^2 + (\frac{R^2+1}{r^2})(\frac{\partial r}{\partial R})^2 dR^2 + r^2(d\theta^2 + \sin\theta^2 d\phi^2).$$

[1] Misner, C. W., Thorne, K. S., and Wheeler, J. A., *Gravitation , 1973, pp 505-525.*
[2] Misner, Thorne, and Wheeler, ref cited, p 826.

But now the metric function r^2 is given by the equations

$$r = M(R^2+1)(a+\cos\eta),$$

$$\tau = M(R^2+1)^{3/2}(\eta+\sin\eta),$$

which has an implicit time dependence. Thus given any values for R and τ, we can solve for η from the second of the two equations and compute r from the first equation. Using a Newton's root solving method, we created a lookup table for the value of r^2 for all values of R and τ in our problem space.

3 Numerical Evolution

Our evolution scheme is based on a computer code developed to study periodic, inflationary cosmologies[3]. The code evolves spacelike hypersurfaces according to the ADM 3+1 formulation of the Einstein equations. The initial data was computed through the standard technique of computing the ϕ^4 conformal factor for a spherically symmetric spacetime. The code is fully three dimensional. No special symmetries in the spacelike hypersurface were assumed.

We choose the 3-metric, h_{ij}, the square root of its determinant, $h^{1/2}$, the traceless form of the extrinsic curvature, $A^i_j = K^i_j - \frac{1}{3}\delta^i_j K$, and the trace of the extrinsic curvature, K, as our dynamic variables. The evolution scheme is essentially a leap-frog in time scheme with centered differences computed for spatial derivatives. The form of the evolution equation facilitates a splitting of the dynamic variables between alternating time slices.

We have modified the code to include spacetimes with finite spatial boundaries and with singularities. Field values at grid points that bound the computational domain can't be computed by centered differences; we use second or third order extrapolation to update these field values. The singularity is excised from the calculation through the use of a masking function. The excised region is initially set to be the interior of the event horizon.

We can compute the Novikov metric components with our numerical evolution scheme by choosing the appropriate lapse and shift vector. The Novikov metric has zero shift vector, while unit lapse corresponds to a freely falling observer. As the evolution proceeds, the radial Novikov coordinate value of the event horizon increases, so the event horizon appears to expand in our computational grid. We update the masking function after each iteration to continue excising grid points inside the event horizon.

[3] Kurki-Suonio, H., Laguna, P., Matzner, R. A., 1993, "Inhomogeneous inflation: Numerical Evolution, *Phys. Rev. Vol. 48, No. 8, 3611-3624*

4 Results

Our evolution was executed on the Cray C90 at the Pittsburg Supercomputing Center. The initial data was computed in isotropic coordinates and then mapped to the 3D cartesian computational grid. Runs were accomplished using 33^3 and 65^3 grid points with the spatial dimensions extending to $\pm 1.0M$, where M is the mass of the black hole. At each iteration, the Novikov radial and angular components of the 3-metric and extrinsic curvature were transformed to their values in the cartesian coordinates of our numerical evolution. This allowed us to compute the L^2-norms of the error between our numerically evolved hypersurface and the exact Novikov hypersurface.

The convergence test showed $O(h^2)$ convergence out to only approximately $\tau = 0.5M$. A plot of the L^2-norms versus τ show the error growing in an unbounded manner. An inspection of the data show good agreement in the interior of the computational domain, but rapidly growing instabilities at the inner boundary near the event horizon and at the outer boundary of our computational grid. Thus we have an inherent instability in our method of updating the field variables by extrapolation.

We also completed one run of the code with 65^3 grid points and with the outer boundary extended to $\pm 2.0M$. This delayed the growth of the error at the outer boundary, while our method of excising grid points that have fallen into the event horizon intermittently tames the growth of errors at that inner boundary. In this case the code runs out to $\tau = 2.93M$ before halting due to floating point exception. The results are visibly good throughout the interior regions of the computational grid as late as $\tau = 2.5M$. The L^2-norm of the metric is approximately 5%, while the error in the extrinsic curvature values are approximately 10%.

We are currently extending our work through the incorporation of an apparent horizon solver developed by Huq which will allow us to track apparent horizons in arbitrary, dynamic spacetimes. We are also investigating stable methods of updating the field values at the boundaries of our computational domain.