Cauchy boundaries in linearized gravitational theory

Bela Szilágyi\textsuperscript{1}, Roberto Gómez\textsuperscript{1}, Nigel T. Bishop\textsuperscript{2} and Jeffrey Winicour\textsuperscript{1,3}

\textsuperscript{1} Department of Physics and Astronomy
University of Pittsburgh, Pittsburgh, PA 15260, USA
\textsuperscript{2} Department of Mathematics, Applied Mathematics and Astronomy
University of South Africa, P.O. Box 392, Pretoria 0003, South Africa
\textsuperscript{3} Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut,
14476 Golm, Germany

Abstract

We investigate the numerical stability of Cauchy evolution of linearized gravitational theory in a 3-dimensional bounded domain. Criteria of robust stability are proposed, developed into a testbed and used to study various evolution-boundary algorithms. We consider a standard explicit finite difference code which solves the unconstrained linearized Einstein equations in the $3 + 1$ formulation and measure its stability properties under Dirichlet, Neumann and Sommerfeld boundary conditions. We demonstrate the robust stability of a specific evolution-boundary algorithm under random constraint violating initial data and random boundary data.
I. INTRODUCTION

The computational evolution of 3-dimensional general relativistic space-times by means of Cauchy evolution is a potentially powerful tool to study the gravitational radiation from black-hole/neutron-star binaries whose inspiral are expected to provide prominent signals to gravitational wave observatories. There are several 3-dimensional general relativistic codes under development to solve this problem. Boundary conditions are an essential part of these codes. At the outer boundary they must provide an outgoing radiation condition and extract the emitted waveform. For black-hole spacetimes, there is also an inner boundary, approximately given by the apparent horizon, where one excises the singular region inside a black hole. Instabilities or inaccuracies introduced at such boundaries have emerged as a major problem common to all code development. Historically, the first Cauchy codes were based upon the Arnowitt-Deser-Misner (ADM) formulation [1,2] of the Einstein equations. Recently there has been pessimism that such codes might be inherently unstable because of the lack of manifest hyperbolicity in the underlying equations. In order to shed light on this issue, we present here a study of ADM evolution-boundary algorithms in the simple environment of evolution in linearized gravity, where nonlinear sources of physical or numerical instability are not present. Our two main results are:

- On analytic grounds, ADM boundary algorithms which supply values for all components of the metric (or extrinsic curvature) are improperly posed.
- We present a boundary algorithm based upon the transverse-traceless components for which unconstrained, linearized ADM evolution can be carried out in a bounded domain for thousands of crossing times without any sign of instability.

This boundary algorithm differs fundamentally from previous approaches and offers fresh hope for robust nonlinear ADM evolution.

Our particular motivation for this work is the difficulty we have experienced implementing Cauchy-characteristic-matching (CCM) for 3-dimensional general relativity [3]. CCM provides a Cauchy boundary condition by matching the Cauchy evolution across the boundary to a characteristic evolution. For nonlinear scalar waves propagating in a flat 3-dimensional space, CCM has been demonstrated to be more accurate and efficient than all other existing boundary conditions for Cauchy evolution [4]. In addition, in the spherically symmetric case of a self gravitating scalar wave satisfying the Einstein-Klein-Gordon equations, CCM has been successfully applied at the inner boundary of a Cauchy evolution to excise the interior black hole region and, at the same time, at the outer boundary to provide a global evolution on a compactified grid extending to null infinity [5]. These successes are promising for the application of CCM to 3-dimensional problems in general relativity but this has not yet been borne out. One possible reason for this failure is an instability of the Cauchy boundary arising from an improperly posed numerical problem that cannot be cured by any boundary algorithm. Here we concentrate on Cauchy evolution based upon the ADM formulation of the Einstein equations which, at present, is the only formulation for which CCM has been attempted.

The stability of the Cauchy evolution algorithm itself is straightforward to investigate by carrying out a boundary-free evolution on a 3-torus (equivalent to periodic boundary
conditions). Such tests constitute Stage 1 of a 3-stage test bed for robust boundary stability proposed in Sec. 2. Stage 1 serves to cull out algorithms whose boundary stability is doomed from the start. In earlier work, robust stability for characteristic evolution with random data on an inner boundary was demonstrated for characteristic evolution using the PITT Null Code [6]. In the course of the present investigation we have reconfirmed this robustness of the PITT code using the same specifications proposed here for Cauchy codes.

CCM cannot work for linearized gravity unless the Cauchy code, as well as the characteristic code, has a robustly stable boundary. This is necessarily so because the interpolations between a Cartesian Cauchy grid and a spherical null grid continually introduce short wavelength noise into the neighborhood of the boundary. This is the rationale underlying the robustness criterion in our test bed. Robustness of the Cauchy boundary is a necessary (although not a sufficient) condition for the successful implementation of CCM.

Analytic studies of Cauchy evolution of linearized gravity with boundaries at infinity reveal modes which grow linearly in time, but none which grow exponentially [7]. The inaccuracy introduced by such secular modes can be controlled and is not of major concern, at least in the linearized theory. (Such secular modes can lead to exponential instabilities of numerical origin in the nonlinear theory if not properly treated. [8]) In the case of a finite boundary, there is further potential for unstable modes at the analytic level. For instance, consider evolution of the wave equation

\[ (-\partial_t^2 + \partial_x^2 + \partial_y^2 + \partial_z^2)\Phi = 0 \]  

in the half-space \( z > 0 \), with boundary at \( z = 0 \). This problem allows the exponentially growing solution \( \Phi = e^{t-z} \) to be initiated with data of finite norm. However, Dirichlet, Neumann or Sommerfeld boundary conditions, respectively

\[ \Phi(t, x, y, 0) = f(t, x, y) \]  

\[ \partial_z\Phi(t, x, y, 0) = f(t, x, y) \]  

\[ (\partial_t - \partial_z)\Phi(t, x, y, 0) = f(t, x, y) \]

all rule out such modes provided that the function \( f \) is bounded.

As is customary in numerical relativity, we monitor the existence of unstable modes by the growth of the Hamiltonian constraint. The constraints are not enforced during standard implementation of ADM evolution. In linearized gravity, it is easy to show for a variety of boundary conditions (see Sec. V) that there are no analytic modes in which this constraint grows exponentially. This implies that there are no unstable physical modes of the corresponding analytic problem. However, because the Riemann tensor does not vanish exactly for a pure gauge mode in the numerical problem, the Hamiltonian constraint is an effective sensor of numerical instabilities of either gauge or physical origin.

Stage 2 of the test bed is based on the simple boundary value problem obtained by opening one dimension of a 3-torus to form a 2-torus with plane boundaries normal to a Cartesian axis. Running a Cauchy-boundary algorithm with this topology and random initial and random boundary data forms the second stage of our test bed. Results are reported in Sec. V. We present several versions of a robustly stable boundary algorithm.

The third stage of the testbed is designed to test robustness of boundary conditions appropriate to an isolated system. In Sec. VI, we present results establishing Stage 3 robustness of a particular ADM boundary algorithm.
The main results presented here are experimental, in a computational sense. The difficulties encountered with finite Cauchy boundaries in general relativity have recently prompted some analytic investigations of the subject [9,10]. However, these have so far been confined to hyperbolic formulations, as opposed to the ADM formulation, and to the analytic problem, as opposed to the finite difference solution provided by computation. Therefore it is not possible to make a direct comparison but the nature of our results seem consistent with the general conclusions of these analytic studies.

There are several promising numerical approaches based upon hyperbolic (or “more hyperbolic”) formulations of the equations [11–20]. Here we concentrate on ADM schemes, which require the least memory because they have a small number of variables. Our results should provide useful benchmarks for other relativity codes. However, it should also be cautioned that the nature of a successful boundary algorithm is dependent on the form of the equations adopted. Results for ADM boundary algorithms do not necessarily apply to other formulations.

We use Greek letters for space-time indices and Latin letters for spatial indices. Four dimensional geometric quantities are explicitly indicated, such as \( (4)R_{\alpha\beta} \) and \( (4)G_{\alpha\beta} \) for the Ricci and Einstein tensors of the space-time, whereas \( R_{ij} \) and \( R \) refer to the Ricci tensor and Ricci scalar of the Cauchy hypersurfaces. Linearized versions of these quantities are denoted by \( (4)\tilde{R}_{\alpha\beta}, \tilde{R}_{ij}, \) etc. Three dimensional tensor indices are raised and lowered by the background Euclidean metric \( \delta_{ij} \). We write \( h = \delta^{ij}h_{ij} \) for 3-dimensional traces. We denote time derivatives by \( \dot{f} = \partial_t f \).

II. TEST BED FOR ROBUST STABILITY OF LINEARIZED CAUCHY EVOLUTION

For simplicity we consider a gauge in which the lapse is unity and the shift vanishes (linearized Gaussian coordinates), so that the linearized metric \( g_{\alpha\beta} = \eta_{\alpha\beta} + h_{\alpha\beta} \) satisfies \( h_{t\alpha} = 0 \). The linearized Einstein equations consist of six evolution equations \( E_{ij} = 0 \) along with the four constraint equations \( C = C_i = 0 \), where

\[
E_{ij} := (4)\tilde{R}_{ij} + \frac{1}{2}\lambda\delta_{ij}C, \tag{2.1}
\]

\( C := (4)\tilde{G}_{tt}, \ C_i := -(4)\tilde{G}_{ti} \) and the adjustable parameter \( \lambda \) allows mixing the (linearized) Hamiltonian constraint \( C \) into the evolution equations.

Codes under development for the evolution of 3-dimensional space-times without symmetry apply the constraint equations at the initial time but do not enforce them during the evolution. It is crucial for this approach that the constraints be stably propagated in time. An investigation by Frittelli [21] shows that the parameter \( \lambda \) in Eq. (2.1) must satisfy \( 1 + \lambda > 0 \) for a well-posed hyperbolic initial value problem for the system of equations governing constraint evolution. This follows from an analysis of the linearized Bianchi identities \( \partial_\beta(4)\tilde{G}^\beta_\alpha = 0 \), which imply that

\[
\dot{C} + (1 + \lambda) \partial^i C + \partial_j E^{ij} \equiv 0 \tag{2.2}
\]

\[
\dot{C} + \partial_i C^i \equiv 0. \tag{2.3}
\]
Thus if the evolution equations are satisfied then the Hamiltonian constraint satisfies the wave equation

\[ \ddot{C} - (1 + \lambda) \partial^k \partial_k C = 0 \quad (2.4) \]

and propagates with speed \( v_C = \sqrt{1 + \lambda} \). The classic case \( \lambda = 0 \) used in early evolution codes [2] (in which the Hamiltonian constraint propagates along the light cone) has a well-posed constraint problem but not the marginal case \( \lambda = -1 \) where evolution is based upon the spatial components of Einstein equations \((4) \mathcal{G}_{ij} = 0\). We consider here evolution equations with a range of \( \lambda \).

In numerical evolution, even if the initial Cauchy data were to satisfy the constraints exactly, each subsequent time step would introduce computational error. Any unstable mode would eventually grow to swamp the evolution. The presence of such an instability may not be evident if masked by a strong signal in the initial data, depending upon the evolution time. The first stage of the testbed is an efficient test for unstable evolution modes.

**Stage 1:** Run the evolution code on on a 3-torus with random initial Cauchy data. The stage is passed if the Hamiltonian constraint \( C \) does not exhibit exponential growth.

The Cauchy data, which consists of \( h_{ij} \) and \( \partial_t h_{ij} \), can be initialized as random numbers in any interval \((-A, A)\), since the system is linear. Here we use the interval \((-10^{-6}, +10^{-6})\). The 3-torus is determined by the periodicity conditions \( h_{ij}(x, y, z) = h_{ij}(x + L, y, z) = h_{ij}(x, y + L, z) = h_{ij}(x, y, z + L) \). We find that an evolution time of 2000L on a uniform 48³ spatial grid with a time step slightly less than one half the Courant-Friedrichs-Lewy limit is practical and sufficient to reveal exponential growth of the Hamiltonian, as measured by the \( \ell_\infty \) norm. **Unless otherwise noted, all runs reported in this paper are made with these specifications.**

Whereas Stage 1 tests stability of the evolution algorithm, Stage 2 is designed to test stability of the boundary algorithm under simple conditions. The three torus is opened up in the \( z \)-direction to form a space of topology \( T^2 \times (0, L) \), with boundaries at \( z = 0 \) and \( z = L \) coinciding with planes of grid points. A boundary algorithm for these points is necessary in order to update the evolution at points neighboring the boundary. For evolution of a scalar wave, the boundary algorithm can be of various types, e.g. Dirichlet, Neumann or Sommerfeld. In the gravitational case, many more versions are possible, corresponding to conditions on the various components of the metric. In order to classify the possibilities, we denote by \( h_{TT} \) the traceless part of the components transverse to the boundary, i.e. \( (h_{xx} - h_{yy}) \) and \( h_{xy} \). Since, our gauge choice \( h_{\mu\nu} = 0 \) is consistent with the radiation gauge subclass of harmonic coordinates, these represent the free modes of waves propagating normal to the boundary. We make the hypothesis that the boundary values of \( h_{TT} \) should be freely specified in either Dirichlet, Neumann or Sommerfeld form. This is justified in the Dirichlet case by the robust stability of characteristic evolution where the free data on a worldtube corresponds to Dirichlet data for \( h_{TT} \) in the linearized approximation.

Given this choice of boundary condition on \( h_{TT} \), the boundary algorithm must be completed to determine the remaining components. Various possibilities are examined in Sec. 4.

One purpose of the testbed is to measure suitability for matching the Cauchy evolution to an exterior numerically generated solution, such as in CCM. In such a case, interpolations...
from the exterior grid to the interior grid continually introduce short wavelength error at the 
Cauchy boundary. In practice, unstable boundary modes often also have short wavelength 
and are triggered quickly by this noise. We introduce random boundary data in our test bed 
to simulate such noise due to matching to an exterior code. As an example of how this is 
implemented, rather than giving smooth Dirichlet data, such as \( h_{TT}(x, y, 0) = 0 \) which 
would represent a reflecting boundary, we require that \( h_{TT} \) be prescribed as a random number at 
each boundary point. Similarly, in the Neumann or Sommerfeld cases, \( \partial_z h_{TT} \) or \( (\partial_\nu - \partial_z) h_{TT} \) 
are prescribed as random numbers. This motivates the following test of the robustness of 
the boundary algorithm.

**Stage 2:** Run the evolution-boundary code on \( T^2 \times (0, L) \) with random initial Cauchy 
data and random boundary data for \( h_{TT} \). The stage is passed if the Hamiltonian constraint 
\( C \) does not exhibit exponential growth.

Here, in order to avoid inconsistencies, the initial and boundary data are both set to 0 
in a few grid zones near the intersection of the initial Cauchy surface with the boundary.

Finally, since boundary algorithms are designed to handle isolated systems, a test with 
either a spherical or cubic boundary is necessary to verify stability. Here we consider a 
cubic boundary, as this is the geometry assumed in standard Cauchy evolution schemes. For 
CCM, a spherical boundary is necessary. This will be the subject of future work.

**Stage 3:** Run the evolution-boundary code with a cubic boundary with random initial 
Cauchy data and random boundary data for \( h_{TT} \). The stage is passed if the Hamiltonian 
constraint \( C \) does not exhibit exponential growth.

### III. EVOLUTION ALGORITHMS

We consider ADM evolution schemes in which the variables consist of \( h_{ij} \) and their asso-
ciated momentum \( K_{ij} = -\dot{h}_{ij}/2 \). The possible finite difference algorithms can be discussed 
in reference to the scalar wave Eq. (1.1), rewritten in the form

\[
\dot{\Phi} = -2K \\
\dot{K} = -\frac{1}{2} \partial_m \partial^m \Phi. 
\]

The evolution is implemented on a uniform spatial grid \((x_i, y_j, z_k) = (j \Delta x, k \Delta x, l \Delta x)\) with 
time levels \( t^n = n \Delta t \). We denote \( \Phi_{j,k,l}^n = \Phi(t^n, j \Delta x, k \Delta x, l \Delta x) \). We consider three different 
finite difference algorithms to carry out the evolution.

**A. Standard leapfrog**

The first algorithm, which we refer to as LF1, is a standard leapfrog implementation of 
Eq’s (3.1):

\[
\Phi_{j,k,l}^{n+1} = \Phi_{j,k,l}^{n-1} - 4K_{j,k,l}^n \Delta t \\
K_{j,k,l}^{n+1} = K_{j,k,l}^{n-1} - \nabla^2 \Phi_{j,k,l}^n \Delta t, 
\]

\(3.2\)
where $\nabla^2$ is the second order accurate centered difference approximation to the Laplacian. It is known that this algorithm has a time-splitting instability in the presence of dissipative and nonlinear effects [22].

B. Staggered leapfrog

The second algorithm, which we refer to as LF2, is a staggered in time leapfrog scheme which is not subject to the time-splitting instability:

$$\Phi_{j,k,l}^{n+1} = \Phi_{j,k,l}^{n} - 2K_{j,k,l}^{n+1/2}\Delta t$$  \hspace{1cm} (3.3)

$$K_{j,k,l}^{n+1/2} = K_{j,k,l}^{n-1/2} - \frac{1}{2} \nabla^2 \Phi_{j,k,l}^{n}\Delta t.$$  \hspace{1cm} (3.4)

Here $K$ is evaluated on the half grid. By subtracting the equation

$$\Phi_{j,k,l}^{n} = \Phi_{j,k,l}^{n-1} - 2K_{j,k,l}^{n-1/2}\Delta t$$  \hspace{1cm} (3.5)

from Eq. (3.3) and using Eq. (3.4) to eliminate $K$, we see that LF2 is equivalent to the standard leapfrog scheme for the second differential order in time form of the wave equation Eq. (1.1), in which $\Phi$ lies on integral time levels and $K$ is not introduced.

C. Iterative Crank-Nicholson

The third algorithm, which we refer to as ICN, is a two-iteration Crank-Nicholson algorithm. The following sequence of operations is executed for each time-step:

1. Compute the first order accurate quantities

$$\Phi_{j,k,l}^{n+1} = \Phi_{j,k,l}^{n} - 2K_{j,k,l}^{n}\Delta t$$  \hspace{1cm} (3.6a)

$$K_{j,k,l}^{n+1} = K_{j,k,l}^{n} - \frac{1}{2} \nabla^2 \Phi_{j,k,l}^{n}\Delta t.$$  \hspace{1cm} (3.6b)

2. Starting with $(i) = 0$, compute the midlevel values

$$\Phi_{j,k,l}^{n+1/2} = \frac{1}{2} \left\{ \Phi_{j,k,l}^{n} + \Phi_{j,k,l}^{n+1} \right\}$$  \hspace{1cm} (3.7a)

$$K_{j,k,l}^{n+1/2} = \frac{1}{2} \left\{ K_{j,k,l}^{n} + K_{j,k,l}^{n+1} \right\}.$$  \hspace{1cm} (3.7b)

3. Update using levels $n$ and $n+1/2$,

$$\Phi_{j,k,l}^{n+1} = \Phi_{j,k,l}^{n} - 2K_{j,k,l}^{n+1/2}\Delta t$$  \hspace{1cm} (3.8a)

$$K_{j,k,l}^{n+1} = K_{j,k,l}^{n} - \frac{1}{2} \nabla^2 \Phi_{j,k,l}^{n+1/2}\Delta t.$$  \hspace{1cm} (3.8b)
4. Increment \( i \) by one and return to step 2 until \( i = 2 \) is reached.

A discretized stability analysis shows that the evolution scheme is stable for 2 and 3 iterations, unstable for 4 and 5 iterations, stable for 6 and 7 iterations, etc [23].

For LF2 and ICN we set \( \Delta t = \Delta x/4 \) and for LF1 we set \( \Delta t = \Delta x/8 \), in all cases slightly less than half the Courant-Friedrich-Lewy condition for the algorithm.

\section*{IV. STAGE 1 TESTS}

The gravitational evolution equations take the form

\[ \dot{h}_{ij} = -2K_{ij} \]  

and

\[ \dot{K}_{ij} = -\frac{1}{2} \partial_m \partial^m h_{ij} + \frac{1}{2} (\partial_i H_j + \partial_j H_i) + \frac{1}{2} \delta_{ij} \lambda C, \]  

where

\[ H_i = \partial^j (h_{ij} - \frac{1}{2} \delta_{ij} h). \]  

and we can express the Hamiltonian as

\[ C = \frac{1}{2} \partial_i H^i - \frac{1}{4} \partial_m \partial^m h. \]  

All terms on the right hand side of Eq. (4.2) are calculated as second spatial derivatives of \( h_{ij} \) in centered form.

The scalar wave equation is a known hyperbolic system in either second differential form or reduced to symmetric hyperbolic first differential form by introducing auxiliary variables. In the hybrid form of Eq. (3.1), which is first differential order in time and second differential order in space, there is no standard classification. Experimentally, we find that the three algorithms LF1, LF2 and ICN pass the stage 1 test for evolution of a scalar wave by means of discretizing Eq. (3.1).

By comparing Eq’s. (3.1) and (4.1) - (4.2), it is evident that the \( H_i \) and \( C \) terms remove any similarity between the gravitational evolution equations and the scalar wave equation. Nevertheless, for gravitational evolution with \( \lambda \) equal to 0, 2 and 4, the three algorithms LF1, LF2 and ICN again pass the stage 1 test! For runs with \( \lambda \) equal to -0.1, 4.1 and 5.0 these three algorithms exhibited exponential growth. These results indicate a range of stability for \( 0 \leq \lambda \leq 4 \).

In this range, \( 0 \leq \lambda \leq 4 \), it is notable that the norm of the Hamiltonian constraint grows linearly in time for LF1 and LF2 but decays exponentially for ICN. This apparently results from the artificial dissipation of ICN. Along the same lines, for \( \lambda = -1 \), algorithms LF1 and LF2 showed exponential growth whereas the norm of the Hamiltonian only grows linearly for ICN. However, for \( \lambda = -1.01 \) or \( \lambda = -0.99 \) this norm grows exponentially for ICN. It appears that the damping of ICN is capable of producing anomalous performance.
The upper limit of the window of stability at $\lambda = 4$ is related to the size of the time step. For algorithm LF2, a run with $\Delta t = \Delta x/8$ (half the time step of the standard runs) and $\lambda = 20$ showed no exponential growth. This seems to arise from the increase of the constraint propagation speed with $\lambda$, which makes the Courant-Friedrich-Lewy condition more stringent.

V. STAGE 2 TESTS

For an asymptotically flat system with the boundary condition that the metric and its derivatives vanish at infinity, it has been shown that the equations (4.1) - (4.2) have no exponentially growing analytic solutions [7]. Here we consider evolution on a Euclidean 3-space with topology $T^2 \times (0, L)$ which has a finite boundary. Before carrying out the computational Stage 2 test, it is important to note that the analogous analytic problem satisfies this test. For this purpose, first consider solutions of the wave equation. The plane wave solutions have the form $\Phi = A e^{i(\alpha t + \beta z + k_x x + k_y y)}$ where the periodicity of functions on $T^2$ requires that $k_x$ and $k_y$ be real. We express $\alpha = -\omega - i\sigma$ and $\beta = k_z + i\ell$, in terms of real quantities and, without loss of generality, choose $\omega \geq 0$. The wave equation implies that

$$-\omega^2 + \sigma^2 + k^2 - \ell^2 = 0$$

(5.1)

$$\omega\sigma - \ell k_z = 0,$$

(5.2)

where $k^2 = k_x^2 + k_y^2 + k_z^2$.

An exponentially growing mode is described by $\sigma > 0$ so that $\ell k_z > 0$. (Thus if the $z$-component $\beta$ of the complex wave number gives rise to an unstable mode, so does $-\beta$.) First consider homogeneous Dirichlet data $\Phi(t, x, y, 0) = \Phi(t, x, y, L) = 0$. This can be satisfied at $z = 0$ by choosing a linear combination of incident and reflected waves

$$\Phi(t, x, y, 0) = A e^{i(\alpha t + \beta z + k_x x + k_y y)} - A e^{i(-\alpha t - \beta z + k_x x + k_y y)},$$

(5.3)

with arbitrary $\beta$, and hence arbitrary $\sigma$. However, the boundary condition at $z = L$ requires that $\ell = 0$, so that exponentially growing modes with positive values of $\sigma$ are not allowed. With the inhomogeneous Dirichlet boundary condition (1.2), the same result holds as long as the boundary data $f(t, x, y)$ is bounded. It is easy to verify that this argument also extends to Neumann and Sommerfeld boundary conditions.

Alternatively, if $\Phi \sim e^{\sigma t}$ then the energy grows as $e^{2\sigma t}$, which requires an energy flux $\sim e^{2\sigma t}$ across the boundary. However, the energy flux is proportional to $\Phi \partial_z \Phi$. Bounded Dirichlet or Neumann data at the boundary rule out an exponential growth of either $\Phi$ or $\partial_z \Phi$ and cannot produce the required energy flux. However, purely local energy considerations do not rule out exponential growth in the Sommerfeld case.

We apply this scalar wave result to the gravitational case by noting that for $\lambda > -1$ the evolution equations imply that the Hamiltonian constraint satisfies the wave equation (2.4). Thus, for $\lambda > -1$, the scalar wave result implies that the analytic problem has no mode for which the Hamiltonian constraint exhibits exponential growth.

The boundary data for $h_{TT}$ is implemented computationally in the following way, which we illustrate in terms of the scalar boundary conditions (1.2) - (1.4) applied at $z = 0$. The Dirichlet condition (1.2) is straightforward to implement as
The Neumann condition (1.3) is implemented as a 3-point one-sided derivative

\[ \Phi_{j,k,0}^n = \phi_{j,k}^n. \]  

The Sommerfeld condition (1.4) is implemented in the interpolative form used in several relativity codes [11,13,19] by modeling the field in the neighborhood of the boundary as \( \Phi(t+z,x,y) \) and using a 3-point spatial interpolation to obtain

\[ \Phi_{j,k,0}^n = \frac{1}{2} \left( 2 - \frac{\Delta t}{\Delta x} \right) \Phi_{j,k,0}^{n-1} + \frac{\Delta t}{\Delta x} \left( 2 - \frac{\Delta t}{\Delta x} \right) \Phi_{j,k,1}^{n-1} - \frac{1}{2} \left( 1 - \frac{\Delta t}{\Delta x} \right) \frac{\Delta t}{\Delta x} \Phi_{j,k,2}^{n-1} + \Delta t \phi_{j,k}^{n-1}. \]  

In the gravitational case, we apply the analogues of these boundary conditions to \( h_{TT} \) at both boundaries, with \( f \) chosen as random numbers.

As a first set of experiments, we have confirmed that scalar wave evolution with algorithms LF2 and ICN passes Stage 2 for both a Dirichlet and a Sommerfeld boundary; but the results for a Neumann boundary were ambiguous showing a growth that was not exponential but stronger than linear with time. The algorithm LF1 passed Stage 2 for a Dirichlet boundary, failed clearly for a Sommerfeld boundary, and for a Neumann boundary showed the same ambiguous behavior as the other algorithms.

For a system of equations in diagonalizable, strongly hyperbolic form there is a standard way of deciding which variables require a boundary condition at a given boundary [24]. Variables propagating along future directed characteristics which emanate from the boundary require a boundary condition but assigning a boundary conditions to the other variables is inconsistent with the evolution equations. Although the gravitational system (4.1) - (4.2) is not symmetric hyperbolic, it would be surprising if all metric variables (or their associated momentum variables) could be freely assigned boundary values. As an example, consider homogeneous boundary conditions corresponding to setting \( f = 0 \) in Eq’s (1.2) - (1.4). For evolution with \( \lambda = 0 \), we have confirmed that the algorithms LF1 LF2 and ICN show exponential growth on the order of 10 crossing times for homogeneous Dirichlet, Neumann or Sommerfeld boundary conditions with only one exception. The ICN algorithm shows no exponential growth (after 2000 crossing times) for a homogeneous Sommerfeld boundary. However, below we shall see that this stability is not robust.

In order to recognize how the system (4.1) - (4.2) constrains the boundary algorithm first consider the linearized Einstein equation component

\[ 2 \left( \tilde{G}_{zz} \right)^n = -\tilde{h}_A^A + \partial^B \partial_B h_A^A - \partial_A \partial_B h^{AB} = 0. \]  

where \( x^A = (x,y) \). Because this component contains no \( z \)-derivatives it can be applied on the boundary to evolve the transverse trace \( h_A^A = h_{xx} + h_{yy} \), given \( h_{TT} \). Any boundary algorithm for \( h_A^A \) must be consistent with this equation.

In addition, the linearized Ricci tensor equation

\[ (4) \tilde{R}_i^i = (4) \tilde{R}_k^k - 2C \equiv \frac{1}{2} \tilde{h} = 0. \]
contains no $z$-derivatives and can be applied on the boundary to evolve the trace $h$, thus determining $h_{zz}$ in terms of transverse components.

The Einstein equation components
\begin{equation}
2^{(4)} \tilde{G}_{z}^{A} \equiv \tilde{h}_{z}^{A} - \partial_{B}(\partial^{B} h_{z}^{A} - \partial^{A} h_{z}^{B}) - \partial_{z} \partial^{A} h_{B}^{B} + \partial_{z} \partial^{B} h_{B}^{A} = 0. \tag{5.9}
\end{equation}

contain no $z$-derivative of $h_{z}^{A}$ but do contain $z$-derivatives of $h_{AB}$. Thus application of this component can serve as a boundary algorithm for $h_{z}^{A}$, given boundary data for $h_{TT}$.

Alternatively, application of the linearized momentum constraint
\begin{equation}
-2 C^{A} \equiv \partial_{z} \dot{h}_{A}^{z} + \partial_{B} \dot{h}_{AB} - \partial^{A} \dot{h} = 0 \tag{5.10}
\end{equation}
or the combination of the time derivative of the momentum constraints with Eq. (5.8),
\begin{equation}
2(-\tilde{C}^{A} + \partial^{A}(4) \tilde{R}_{t}^{t}) := \partial_{z} \dot{h}_{z}^{A} + \partial_{B} \dot{h}_{AB} = 0, \tag{5.11}
\end{equation}
give other ways to update the Neumann boundary data for $h_{z}^{A}$ in terms of Dirichlet boundary values of $h_{AB}$. Also, the combination
\begin{equation}
2(-\tilde{C}_{z} + \partial^{z}(4) \tilde{R}_{t}^{t}) := \partial_{z} \ddot{h}_{z}^{z} + \partial_{A} \ddot{h}_{z}^{A} = 0 \tag{5.12}
\end{equation}
could be used to update the Neumann boundary data for $h_{zz}$.

These considerations can be used to show that certain boundary conditions give rise to an ill-posed problem. For instance, consider the Dirichlet algorithm consisting of setting all components of $h_{ij}$ to zero on the boundary, which is a perfectly good (reflecting) boundary algorithm for a scalar field. In the gravitational case, Eq. (5.9) gives a constraint on the normal derivative of $h_{AB}$ which implies that $\Psi := \partial_{A} \partial^{A} h_{B}^{B} - \partial^{A} \partial_{B} h_{AB}$ and its normal derivative $\partial_{z} \Psi$ both vanish on the boundary. But it is easy to verify, in the case $\lambda = 0$, that the evolution equations for $h_{ij}$ imply that $\Psi$ satisfies the scalar wave equation. Thus the vanishing Dirichlet data for $h_{ij}$ generates, for any initial data, a solution $\Psi$ of the wave equation whose Dirichlet and Neumann boundary data both vanish. This a classic example of an inconsistent boundary value problem for the scalar wave $\Psi$.

Similarly, suppose the Sommerfeld condition $(\partial_{t} - \partial_{z}) h_{ij} = 0$ were applied to all metric components on the plane boundary at $z = 0$. If $h_{ij}$ were a global solution consistent with this boundary data then, since the equations are linear and have space-time translational symmetry, $\tilde{h}_{ij} = (\partial_{t} - \partial_{z}) h_{ij}$ would also be a global solution but with vanishing Dirichlet data for all components at the boundary. Thus, as in the Dirichlet problem, a Sommerfeld boundary condition (or by the same argument a Neumann boundary condition) applied to all components of the metric leads to an inconsistent boundary value problem. In order to shed light on this issue we conducted the following two computational experiments, using ICN evolution with $\lambda = 0$ and random initial data. First, we applied homogeneous Sommerfeld boundary data to all components (the analogue of setting $f = 0$ in Eq. (1.4)). The runs in this case were well behaved for 2000 crossing times. Apparently, any effects of inconsistency in the boundary value problem are squelched in the finite difference approximation by the vanishing boundary data. Next, we applied random Sommerfeld boundary data to all components (the analogue of choosing $f$ randomly in Eq. (1.4)). A log plot of the Hamiltonian
constraint is shown in Fig. 1. While not an exponential, it grows by 5 orders of magnitude in 2000 crossing times, clearly unacceptable for numerical application and showing that the results for homogeneous data are not reliable.

We again note that these conclusions apply to the ADM system and not necessarily to other versions of the evolution equations. It is apparent that the ADM evolution equations (4.1) - (4.2), along with the constraints, must be properly applied along with boundary data for \( h_{TT} \) in order to obtain suitable performance. Of the many combinations we have tried, many show exponential growth after only \( \approx 20 \) crossing times. The following Dirichlet boundary algorithms exhibit Stage 2 robust stability for the ICN evolution algorithm.

### A. Robust Dirichlet stability

For the ICN evolution algorithm, we apply the foregoing analysis to construct 5 robust Dirichlet boundary algorithms for the system of equations (4.1) - (4.2). The algorithms supply boundary values for the extrinsic curvature \( K_{ij} \), with boundary values for the metric perturbation updated by the centered difference version of (4.1). Given random initial and boundary data for the transverse-traceless components \( K_{TT} \), all five algorithms update the boundary values of the trace \( K_A^A \) via integration of Eq. (5.7).

**Algorithm 1**: We apply Eq. (5.8) to update boundary values for \( K_{zz} \) and the momentum constraint Eq. (5.10) to supply boundary values for \( \partial_z K_z^A \) which, expressed as a 3-point sideways finite difference, supplies \( K_z^A \) on the boundary.

**Algorithm 2**: We apply Eq. (5.8) to update boundary values for \( K_{zz} \) and Eq. (5.11) to supply boundary values for \( \partial_z K_z^A \) which, as in Algorithm 1, is used to update the boundary values for \( K_z^A \) using a centered time difference.

**Algorithm 3**: We apply Eq. (5.12) to update boundary values for \( K_{zz} \) (with finite difference stencils as above) and Eq. (5.11) to update boundary values for \( K_z^A \).

**Algorithm 4**: We apply Eq. (5.8) to update boundary values for \( K_{zz} \) and Eq. (5.9) to update boundary values for \( K_z^A \).

**Algorithm 5**: We apply Eq. (5.12) to update \( K_{zz} \) and Eq. (5.9) to update \( K_z^A \).

All five boundary conditions satisfied Stage 2 robust stability for evolution with \( \lambda = 2 \). Algorithms 1 and 5 were also found to be robustly stable for \( \lambda = 0 \) and \( \lambda = 4 \). (The other algorithms were not checked for these cases in order to conserve supercomputing time).

For the case \( \lambda = 2 \), Fig. 2 shows the behavior of the Hamiltonian constraint for these five algorithms. Note that algorithms 1 and 2 have identical performance, as might be expected as they differ only with respect to details of initializing the boundary routine. Algorithms 4 and 5 show less noise in the Hamiltonian constraint than the others.

Algorithm 5 gave the best performance, with the Hamiltonian constraint actually showing a slow decrease at late times. This algorithm embodies the “boundary constraints” \( \tilde{G}_{zi} \) and a linear combination of \( \partial_t \tilde{G}_{zt} \) and \( \partial_z \tilde{G}_{tt} \). Note that use of \( \tilde{G}_{zt} \) by itself, in addition to \( \tilde{G}_{zi} \), would be similar to using the Hamiltonian constraint along with the momentum constraints - four conditions that are not independent [8].

While these 5 Dirichlet boundary algorithms were robust for ICN evolution, they failed Stage 2 for either LF1 or LF2 leapfrog evolution with \( \lambda = 2 \). A selection of tests with \( \lambda = 0 \) and \( \lambda = 4 \) also failed. In this range, the exponential growth rate typically decreases with

12
increasing $\lambda$. As an example, for boundary algorithm 1 with either LF1 or LF2 evolution, the time scale for exponential growth is 100 crossing times (CT) for $\lambda = 0$, 300 CT for $\lambda = 2$ and 1000 CT for $\lambda = 4$. The failure of these algorithms for leapfrog evolution, but not for ICN, emphasizes that the finite difference problem is much more complex than the corresponding analytic problem.

**B. Neumann and Sommerfeld boundaries**

We have been unsuccessful in attempts to convert any of the 5 Dirichlet boundary algorithms (listed in the previous section) to obtain robust stage 2 Neumann or Sommerfeld evolution. We based these attempts as follows. In the Neumann case, suppose that all components of the metric have been determined at time level $N - 1$ and the evolution has been applied to update all components at level $N$ except at the boundary. Then we put this Neumann data into Eq. (5.5) to update $h_{TT}$ on the boundary at level $N$. This supplies the necessary Dirichlet data to apply the Dirichlet algorithms to update all remaining components. Similarly, in the Sommerfeld case, given that the metric has been determined at level $N - 1$ and the evolution has been applied to update all components at level $N$ except at the boundary, we apply the interpolative Sommerfeld approximation (5.6) to update $h_{TT}$ on the boundary at level $N$. Again this supplies the necessary Dirichlet data to apply the Dirichlet algorithms to update all remaining components.

Since the robustness of Neumann boundary conditions was already ambiguous for a scalar field, our attempts were limited in this case. However, in the Sommerfeld case, we tried an extensive, if not exhaustive, combination of boundary conditions 1 - 5 and values of $\lambda$. The failure of these schemes indicates that a successful boundary algorithm does not readily translate to other forms of prescribing boundary data.

**VI. STAGE 3 TEST ROBUST STABILITY**

In view of the stage 2 results, we confine our stage 3 investigation of robust boundaries to ICN evolution with a Dirichlet boundary condition. In applying a constrained boundary condition to the faces of the cube, we have chosen to use algorithm 5 on all faces. The edges and corners must be handled separately. The two components $K_{TT} = -\frac{1}{2} \dot{h}_{TT}$ are treated as free data (i.e. are specified randomly) on all faces, edges and corners. While this means two free quantities and four constraints on the faces, the number of free quantities on the edges is four, i.e. one needs two constraints only. Similarly, on the corners, there are five free quantities, for the identity $[K_{xx} - K_{yy}] + [K_{yy} - K_{zz}] + [K_{zz} - K_{xx}] = 0$ reduces the total number of six $TT$ components to five that are independent. Thus only one constraint is needed at the corners.

As already stated, on the corners all non-diagonal components are provided as data. Given $[K_{xx} - K_{yy}]$ and $[K_{zz} - K_{xx}]$ the missing ingredient of the diagonal components is $K_{xx}$ which we compute from

$$3K_{xx} = K + [K_{xx} - K_{yy}] + [K_{xx} - K_{zz}].$$
The trace $K$ is updated using the condition

$$(4) \tilde{R}_t^t = -\dot{K} = 0.$$

Next we give the algorithm for the edges. On the edges parallel to the $x$-axes one already has $K_{xy}$ and $K_{xz}$ as boundary data. The missing $K_{yz}$ is computed using $(4)\tilde{G}_{yz}$, an equation that is also used on both neighboring faces. Derivatives in the $y$- and $z$- directions are computed by sideways, 3-point finite difference formulae. The diagonal components of the 3-metric are computed the same way as on the corners.

We should note that the routine that solves the constraint

$$(\tilde{C}^n + \partial^n(4)\tilde{R}_t^t) = 0$$

on a face of the cube with normal in the $n$-direction must be called after the missing non-diagonal components have been updated on the edges surrounding that face. Otherwise, in the case of the $z = \text{const}$ face, when computing the quantity $K_{yz,y}$ on the top time-level, with centered finite differencing, one might use values of $K_{yz}$ on the edge parallel to the $x$-axis that had not yet been updated.

To show that the above algorithm is robustly stable we performed runs with $\lambda = 0, 2, 4$. All three runs were given random initial and boundary data. A graph showing the Hamiltonian constraint as a function of time is shown in Fig. 3.

VII. CONCLUSION

We have shown that linearized ADM evolution can be carried out with long term stability in a test bed consisting of random constraint violating initial data and random boundary data applied to the trace-free-transverse modes. The success of the new formulation of the ADM boundary algorithm presented here offers new hope both for the long term stability of nonlinear ADM evolution and for CCM applied at an ADM boundary. The extension of the boundary algorithm to the nonlinear case is in principle straightforward since it is based upon well defined components of the metric and field equations. However, choice of the correct numerical stencils to treat the nonlinear terms may not be so obvious. Adding the Hamiltonian constraint (with $\lambda > 0$) to the Ricci system of linearized equations gives better performance but does not drastically affect overall robustness. However, in the nonlinear case such techniques have been shown to suppress the secular modes in the linear theory from becoming exponential [8]. The extension of the boundary algorithm to a spherical boundary, as would be necessary for CCM, again seems straightforward in principle but also entails many numerical complications. Such studies are now underway.

ACKNOWLEDGMENTS

We thank B. Schmidt for reading the manuscript and suggesting improvements. This work has been supported by NSF PHY 9510895, NSF PHY 9800731 and NSF INT 9515257 to the University of Pittsburgh. N.T.B. thanks the Foundation for Research Development,
South Africa, for financial support, and the University of Pittsburgh for hospitality. Computer time for this project was provided by the Pittsburgh Supercomputing Center and by NPACI.
REFERENCES

FIG. 1. A log plot of the $\ell_{\infty}$ norm of the Hamiltonian constraint as a function of crossing time for a Stage 2 test of random Sommerfeld boundary conditions on all metric components.
FIG. 2. Stage two performance of the Hamiltonian constraint as a function of crossing time for the five robustly stable boundary algorithms.
FIG. 3. Behavior of the Hamiltonian constraint for a Stage 3 test with cubic boundary.