TOWARD STANDARD TESTS FOR THE INITIAL/BOUNDARY PROBLEM IN NUMERICAl RELATIVITY

Apples Retreat, LSU 2004
(AppleswithApples)

PACS numbers:

I. INTRODUCTION

A plan for a round boundary tests has been formulated and the test details are being developed. Boundary tests can be classified by various stages, where Stage 1 corresponds to the first round of periodic boundary tests on a 3-torus T^3. In a Stage 2 test, one axis of the three torus is opened up to form a manifold with smooth T^2 boundaries. In Stage 3, all axes are opened up to form a manifold with cubic boundary. As either an alternative or addition to stage 3, the cubic boundary can be replaced by a smooth boundary of spherical topology in Stage 4. All first round Mexico tests on T^3 can be easily extended to either Stage 2 or Stage 3 tests. The first round of boundary tests will be restricted to Stage 2 in order to avoid the problems of dealing with sharp edges and corners (stage 3) or the interpolations necessary for describing spherical boundaries on a Cartesian grid (Stage 4). Below we give explicit specifications for extending the robust stability test,the linear wave test and the gauge wave test to stage 2. These tests have already been performed at Stage 1 so they have the advantage of differentiating between boundary and interior effects. In the robust stability test the constraints are not satisfied and all variables requiring boundary data should be given random data. In the linear, gauge and Gowdy wave tests, the spacetime metric is known so that the correct boundary data for all variables requiring boundary data can be explicitly prescribed.

Thus these tests can be run independently of a constraint preserving boundary condition. If a constraint preserving boundary condition is known then test comparisons between explicitly prescribed boundary data and constraint preserving boundary data would provide important additional information. Addition of a Rindler wedge boundary test is under consideration. The linear wave test may be dropped if the Stage 1 results show that nothing essential is learned.

The plan is to carry out preliminary exploration of a number of these tests to decide whether the suggested parameters and output data are appropriate. The general agreement is that these test should be run with a Runge Kutta update scheme rather than ICN.

II. TEST CASE SPECIFICATION

We specify the physical properties of each testbed by providing the complete 4-metric of the spacetime, or if this is not possible, the initial Cauchy and boundary data and choice of gauge for the evolution. In all cases, this provides the necessary data for 3-dimensional evolution in a Cartesian coordinate system x^i.

The first round of tests, designed to test evolution algorithms, was carried out on a cube $|x^i| \leq d/2$ with periodic boundary conditions, i.e. on a 3-torus T^3. In the current round, designed to test boundary algorithms, we open up the 3-torus to form a manifold with boundary. In the first of these tests one axis of the 3-torus is opened up to form the manifold with boundary $[-d/2, +d/2] \times T^2$. This has the advantage that the boundaries are smooth 2-tori.

For cross comparison of codes it is helpful to provide results from uniform numerical methods, In the first round of tests, evolution by a second order iterative Crank-Nicholson scheme was chosen for simplicity. After experimentation, it was found that Runge Kutta might have been a better choice. We have agreed that Runge-Kutta will be the standard scheme for the next round of boundary tests. There may be codes that cannot implement this type of numerical method. Similarly, a particular code may run better with an alternative numerical method such as a pseudo-spectral method. In such cases the relative performance of the code for these tests still offers a useful comparison, provided all parameters (such as the amount of artificial dissipation) are held constant over the tests.

In order to facilitate the implementation of boundary algorithms the grid structure is slightly changed from that used in the first round of tests. As in the first round, the grids extend an equal distance along the positive and negative coordinate axes. However, the boundaries are now located at grid points $x^i = \pm d/2$, rather than at midpoints. The "boundaries" for the coordinates running around the 2-torus are located at the first and last grid points along each axis, which points are identified. The true boundaries are also located at the first and last grid points. The number of grid points in a given direction is chosen to resolve the features of the initial data. Even though we are running three-dimensional codes, we gain efficiency for tests with only one-dimensional features by choosing n^i to be small in the trivial directions. As an example, for a wave propagating in the x-direction we use the minimum number of grid points in the trivial y and z directions that allow for non-trivial numerical second derivatives, typically 3 points for standard second order finite differencing.

The time step dt is given in terms of the grid size dx and chosen to lie within the CFL limit for an explicit evolution algorithm. For codes for which this is inappropriate some equivalent choice of time step should be made. A final time T, and intermediate times for data output, are specified for each test. The time T is chosen to incorporate all useful features of the test without prohibitive computational expense. The output times should be appropriately modified for codes that crash before time T.

The output quantities are chosen with either some physical or numerical motivation in mind. Both quantitative and qualitative comparisons are used. We specify a minimum list of output quantities. Other quantities of interest for a specific code might include the Fourier transform of the error or curvature invariants.

A. Robust stability testbed

The robust stability test efficiently reveals instabilities which otherwise might be masked beneath a strong initial signal for a considerable evolution time. It is based upon small random perturbations of Minkowski space. The data consists of random numbers ϵ applied as a perturbation of Minkowski data at each grid point (1) at the initial time for every code variable requiring initialization and (2) at each time step at each boundary grid point for every code variable requiring boundary data. For example, the initial 3-metric is initialized as $h_{ij} = \delta_{ij} + \epsilon_{ij}$, where the ϵ_{ij} are independent random numbers. Similarly, for a code that requires a Sommerfeld type boundary condition based upon an outgoing characteristic direction ℓ^μ for a metric component h_{ij}, the Sommerfeld boundary data is given in the perturbed form $\ell^\mu \partial_\mu h_{ij} = \epsilon_{ij}$. Such non-zero boundary data is in general necessary when a Sommerfeld condition is not exact because of (1) the orientation of ℓ^μ, (2) the finite location of the boundary, (3) the inflow or backscattering of waves across the boundaries or (4) the characteristic or perturbative matching to data supplied by an exterior solution extending to infinity. Similarly, Dirichlet boundary data for a metric component would be prescribed in the form $\partial_t h_{ij} = \epsilon_{ij}$.

For economy, we fix the following parameters:

- Simulation domain: $d = 1$, $x \in [-0.5, +0.5]$.
- Grid: $x_n = -0.5 + ndx, \quad n = 0, 1 \ldots 50\rho, \quad dx = dy = dz = 1/(50\rho), \quad \rho \in \mathbb{Z}$
- Time step: $dt = dx/2 = 0.01/\rho$

The refinement parameter ρ takes values $\rho = 1, 2, 4$. we run the test in a long channel rather than a cube, using the minimum number of grid points in the y, z directions that allow for non-trivial numerical second derivatives, typically 3 points.

The amplitude of the random noise is scaled with the grid spacing as

$$\epsilon \in (-10^{-10}/\rho^2, +10^{-10}/\rho^2) \tag{2.1}$$

to ensure that the Hamiltonian constraint violation is (on average) the same for different refinements. The range of the random numbers ensures that ϵ^2 effects are below roundoff accuracy so that the evolution remains in the linear domain unless instabilities arise.The continuum limit is not a solution of the Einstein equations but "close" to one, as in a real numerical evolution where machine precision takes the place of ϵ. If a code cannot stably evolve the random noise then it will be unable to evolve a real initial/boundary data set.

The test should be run for a time of $T = 1000$ (corresponding to 1000 crossing times) or until the code crashes. Performance is monitored by outputting the L_∞ norm of the Hamiltonian constraint once per crossing time, i.e. at $t = 0, 1, 2, 3, ...$

B. Gauge wave testbed

These tests checks the ability of formulations to handle gauge dynamics. This is done by considering flat Minkowski space in coordinates generated by a propagating sine wave. The 4-metric is obtained from the Minkowski metric $ds^2 = -d\hat{t}^2 + d\hat{x}^2 + d\hat{y}^2 + d\hat{z}^2$ by the coordinate transformation

$$\begin{aligned} \hat{t} &= t + \tfrac{Ad}{4\pi}\cos\left(\tfrac{2\pi(x-t)}{d}\right), \\ \hat{x} &= x - \tfrac{Ad}{4\pi}\cos\left(\tfrac{2\pi(x-t)}{d}\right), \\ \hat{y} &= y, \\ \hat{z} &= z \end{aligned} \tag{2.2}$$

where d is the size of the evolution domain. This leads to the 4-metric

$$ds^2 = -H dt^2 + H dx^2 + dy^2 + dz^2, \tag{2.3}$$

where

$$H = H(x-t) = 1 - A\sin\left(\frac{2\pi(x-t)}{d}\right), \tag{2.4}$$

which describes a sinusoidal gauge wave of amplitude A propagating along the x-axis. The extrinsic curvature is given by

$$K_{xx} = -\frac{\pi A}{d}\frac{\cos\left(\frac{2\pi(x-t)}{d}\right)}{\sqrt{1 - A\sin\left(\frac{2\pi(x-t)}{d}\right)}}, \tag{2.5}$$

$$K_{ij} = 0 \qquad \text{otherwise.} \tag{2.6}$$

Since this wave propagates along the x-axis and all derivatives are zero in the y and z directions, the problem is essentially one dimensional and can simplify the system dramatically for certain formulations, as there is no finite-difference error in the orthogonal directions. A simple coordinate transformation causes the wave to propagate along a diagonal:

$$x = \frac{1}{\sqrt{2}}(x' - y'), \qquad y = \frac{1}{\sqrt{2}}(x' + y') \tag{2.7}$$

The resulting metric is a function of

$$\sin\left(\frac{2\pi(x' - y' - t'\sqrt{2})}{d'}\right), \quad \text{where} \quad d' = d\sqrt{2}\,. \tag{2.8}$$

Setting d' to the size of the evolution domain in the x' and y' directions gives periodicity along those directions. This test should be run in both axis-aligned and diagonal form.

We tentatively run the gauge wave with amplitudes $A = 10^{-1}$ and $A = 10^{-2}$. We have found that smaller amplitudes are quite simple for all codes. Larger amplitudes cause numerical error to trigger gauge pathologies more quickly and may turn out to be more efficient for code comparisons.

The specified wave has wavelength $d = 1$ in the 1D simulation and wavelength $d' = \sqrt{2}$ in the diagonal simulation. We find that 50 grid points are sufficient to resolve the profile and therefore make the following choices for the computational grid:

- Simulation domain:

$$\begin{array}{rllll} \text{1D:} & x \in [-0.5, +0.5], & y = 0, & z = 0, & d = 1 \\ \text{diagonal:} & x \in [-0.5, +0.5], & y \in [-0.5, +0.5], & z = 0, & d' = \sqrt{2} \end{array}$$

- Grid: $x_n = -0.5 + ndx, \quad n = 0, 1 \ldots 50\rho, \quad dx = dy = dz = 1/(50\rho), \quad \rho \in \mathbb{Z}$
- Time step: $dt = dx/4 = 0.005/\rho$

The 1D evolution is carried out for $T = 1000$ crossing times, i.e. $2 \times 10^5 \rho$ time steps (or until the code crashes), with output every 10 crossing times. The 2D diagonal runs are carried out for $T = 100$, with output every crossing time. We run using $\rho = 1, 2, 4$.

The boundary data are provided at $x = \pm.5$. For example, in the 1D simulation, for a formulation with a Sommerfeld boundary condition on the metric component g_{xx} the correct boundary data are

$$(\partial_t + \partial_x) g_{xx}|_{x=.5} = 0 \tag{2.9}$$

$$(\partial_t - \partial_x) g_{xx}|_{x=-.5} = 2\partial_t H(-.5 - t). \tag{2.10}$$

With this inhomogeneous Sommerfeld data, the wave enters through the boundary at $x = -.5$, propagates across the grid and exits through the boundary at $x = .5$.

We output the L_2-norm, the maxima and minima, and profiles along the x-axis through the center of the grid of g_{xx}, α, $tr(K)$, the Hamiltonian constraint and any other independent constraints that arise in a nontrivial way in a particular formulation. We also calculate the L_1-norm of the difference from the exact solution for g_{xx} and calculate the convergence factor.

C. Linear wave testbed

This test checks the ability of a code to propagate the amplitude and phase of a traveling gravitational wave. The test is run in the linear regime where there are no complications due to the toroidal topology implicit in periodic boundary conditions. It reveals effects of numerical dissipation and other sources of inaccuracy in the evolution algorithm. The evolution is meaningless once the accumulation of numerical error takes it out of the linear regime.

The initial 3-metric and extrinsic curvature K_{ij} are given by a diagonal perturbation with components

$$ds^2 = -dt^2 + dx^2 + (1+b)\,dy^2 + (1-b)\,dz^2, \tag{2.11}$$

where

$$b = A \sin\left(\frac{2\pi(x-t)}{d}\right) \tag{2.12}$$

for a linearized plane wave traveling in the x-direction. Here d is the linear size of the propagation domain, and the metric is written here in Gauss coordinates, i.e. with lapse $\alpha = 1$ and shift $\beta^i = 0$. The nontrivial components of extrinsic curvature are then

$$K_{yy} = \frac{1}{2}\partial_t b, \quad K_{zz} = -\frac{1}{2}\partial_t b. \tag{2.13}$$

As in the case of the gauge wave, by the simple coordinate transformation

$$x = \frac{1}{\sqrt{2}}(y' + x'), \qquad y = \frac{1}{\sqrt{2}}(y' - x') \tag{2.14}$$

the propagation direction can be aligned with a diagonal. Setting $d' = d\sqrt{2}$ to the size of the evolution domain in the x', y' directions gives periodicity along those directions. The amplitude of the wave is chosen as $A = 10^{-8}$, such that quadratic terms are of the order of numerical roundoff. Larger amplitudes mean that the solution does not stay in the linear regime sufficiently long.

The geometry of the grid is chosen identical to the 1D gauge wave test, with $d = 1$ in the 1D case and $d' = \sqrt{2}$ in the diagonal case:

- Simulation domain:

$$\begin{array}{rllll} \text{1D:} & x \in [-0.5; +0.5], & y = 0, & z = 0, & d = 1 \\ \text{diagonal:} & x \in [-0.5; +0.5], & y \in [-0.5; +0.5], & z = 0, & d' = \sqrt{2} \end{array}$$

- Grid: $x_n = -0.5 + n dx, \quad n = 0, 1 \ldots 50\rho, \quad dx = dy = dz = 1/(50\rho), \quad \rho \in \mathbb{Z}$
- Time step: $dt = dx/4 = 0.005/\rho$

As in the gauge wave case, the 1D evolution is carried out for $T = 1000$ crossing times with boundary data prescribed at $x = \pm).5)$, with output every 10 crossing times. The 2D diagonal runs are carried out for $T = 100$, with output every crossing time. For the trivial directions (y and z for the wave propagating along the x axis and z for the wave propagating along the diagonal) we use the minimum number of grid points in the y, z directions that allow for non-trivial numerical second derivatives. For standard second order finite differencing this implies that we use 3 points in the appropriate directions. We run using $\rho = 1, 2, 4$.

The output quantities are similar to those for the gauge wave: the L_∞-norms, the maxima and minima, and profiles along the x-axis through the center of the grid of g_{yy}, g_{zz}, $tr(K)$, the Hamiltonian and any other nontrivial constraints, and the L_∞-norm of the difference from the linear exact solution for g_{zz}.

D. Polarized Gowdy wave testbed

Here we use a polarized Gowdy spacetime to test codes in a genuinely curved, strong field context. The polarized Gowdy T^3 spacetimes are solutions of the vacuum Einstein equations which describe an expanding universe containing plane polarized gravitational waves with metric

$$ds^2 = t^{-1/2}e^{\lambda/2}(-dt^2 + dz^2) + t\,dw^2, \tag{2.15}$$

where

$$dw^2 = e^P dx^2 + e^{-P} dy^2. \tag{2.16}$$

The quantities λ and P are functions of z and t only and are periodic in z. We choose the particular solution

$$P = J_0(2\pi t)\cos(2\pi z), \tag{2.17}$$

which yields

$$g_{xx} = te^P, \; g_{yy} = te^{-P}, \; g_{zz} = t^{-1/2}e^{\lambda/2}, \tag{2.18}$$

$$\begin{aligned} K_{xx} &= -\frac{1}{2}t^{1/4}e^{-\lambda/4}e^P(1 + tP_{,t}), \\ K_{yy} &= -\frac{1}{2}t^{1/4}e^{-\lambda/4}e^{-P}(1 - tP_{,t}), \\ K_{zz} &= \frac{1}{4}t^{-1/4}e^{\lambda/4}(t^{-1} - \lambda_{,t}), \end{aligned} \tag{2.19}$$

with

$$\begin{aligned} \lambda = & -2\pi t J_0(2\pi t)J_1(2\pi t)\cos^2(2\pi z) + 2\pi^2 t^2\left[J_0^2(2\pi t) + J_1^2(2\pi t)\right] \\ & - \tfrac{1}{2}\left\{(2\pi)^2\left[J_0^2(2\pi) + J_1^2(2\pi)\right] - 2\pi J_0(2\pi)J_1(2\pi)\right\}. \end{aligned} \tag{2.20}$$

The shift vanishes, and the lapse is given as

$$\alpha = \sqrt{g_{zz}} = t^{-1/4}e^{\lambda/4}. \tag{2.21}$$

We set the boundaries at $z = \pm.5$, as in the Stage 1 periodic test. As a result, $A = \int_{-.5}^{.5} tP_{,t}\,dz$ remains a constant of motion and can be used to monitor code accuracy. In our case A is set to zero by choice of initial data. Here the time coordinate t is chosen such that time increases as the universe expands. The metric is singular at $t = 0$. For code testing, it is quite interesting to compare collapsing and expanding situations, we will thus carry out our tests in *both* time directions.

The (coordinate) velocity of light is constant in the coordinates chosen in eq. (2.15), and with a fixed spatial discretization size Δz the Courant condition is consistent with a fixed timestep Δt. This makes it convenient to choose the gauge (2.15) for evolving in the *expanding* direction. We will see below however, that this leads to exponential growth in the metric component g_{zz}. For the collapsing direction, this would lead to a singularity at $t = 0$, so we will evolve this case with a different slicing as discussed below.

For the forward evolution, we set initial data from the exact solution at $t = 1$, which yields initial data of order unity, and evolve with any lapse condition which is equivalent in the continuum limit to the exact lapse given in eq. (2.21). Due to the exponential growth in the metric variables, such evolutions may crash rather soon but allow the testing of accuracy in a rather harsh situation.

In order to evolve in the backward time direction, we choose harmonic time slicing, which is marginally singularity avoiding such evolutions should only asymptotically reach the singularity at $t = -\infty$.

It turns out, that it is actually quite simple to write down an exact solution for harmonic slicing, which greatly simplifies the task of choosing appropriate gauge source functions for various formalisms. Starting with the Gowdy metric, as given by eq. (2.15), we look for a coordinate transformation $(t, x^i) \rightarrow (\tau, x^i)$, with $t = F(\tau)$. In the new coordinates, the lapse becomes $\hat{\alpha} = F(\tau)^{-1/4}\partial_\tau F(\tau) e^{\lambda/4}$. The harmonicity condition $\Box t = 0$ then implies

$$e^{-\lambda/2}\left[F\partial_{\tau\tau}F - \partial_\tau F^2\right] = \sqrt{F}\partial_\tau F^3 \tag{2.22}$$

with the solution $F(\tau) = ke^{c\tau}$, where c and k are free constants. The lapse in this new gauge is

$$\hat{\alpha}(\tau) = ck^{3/4}e^{3c\tau/4+\lambda(F(\tau),z)/4}. \tag{2.23}$$

In order to start the collapse slowly, and to simplify initial data, we choose the constants c, k in such a way that $\hat{\alpha} = 1$ at the initial time $t = t_0$. Picking a value t_0 for which $J_0(2\pi t_0) = 0$, eq.(2.20) implies that $\hat{\alpha}$ is independent of z. Using

$$\tau_0 = \frac{1}{c}\ln\left(\frac{t_0}{k}\right), \quad \lambda(ke^{c\tau_0}, z) = \lambda_0$$

we obtain

$$\hat{\alpha}_0 = c\, t_0^{3/4}\, e^{\lambda_0/4}. \tag{2.24}$$

Given our requirement $\hat{\alpha}_0 = 1$, and choosing $t_0 = \tau_0$, i.e. $F(\tau_0) = \tau_0$, we get

$$c = t_0^{-3/4}\, e^{-\lambda_0/4}, \quad k = t_0 e^{-ct_0}. \tag{2.25}$$

We will choose a particular value of t_0 such as to start off far from the cosmological singularity, but not so far that we have to deal with extremely large numbers. We pick zero number 20 of the Bessel function $J_0(2\pi t_0)$, which yields $t_0 \sim 9.8753205829098$, corresponding to

$$c \sim 0.0021195119214617, \quad k \sim 9.6707698127638.$$

For the collapsing direction we thus initialize our runs taking data from the metric (2.15) at time $t = t_0 \sim 9.8753205829098$. The values of the metric components at $t = t_0$ are then $g_{xx} = g_{yy} = t_0$, $g_{zz} \sim 2.283\times 10^3$. This choice challenges a numerical code to accurately track a small effect (dynamics in g_{xx}, g_{yy} together with a larger effect (dynamics in $g_z z$). Other choices are of course possible, and certainly worth exploring. For the purpose of a standard testbed, which should provide tests which are able to discriminate well between different formulations, the current choice seems appropriate.

The grid is chosen analogous to the 1D gauge wave test:

- Simulation domain: $z \in [-0.5; +0.5]$, $x = y = 0$
- Grid: $x_n = -0.5 + ndx, \quad n = 0, 1 \ldots 50\rho, \quad dx = dy = dz = 1/(50\rho), \quad \rho \in \mathbb{Z}$
- Time step: $dt = dz/4 = 0.005/\rho$
- Run time: $T = 1000$, i.e. 1000 crossing times or until code crash.

We output the L_∞ and L_2-norms, the maxima and minima, and profiles along the z-axis through the center of the grid of g_{zz}, α, $tr(K)$, the Hamiltonian constraint and all other nontrivial constraints of the formulation, and some typical evolution variables, depending on the evolution system chosen. We output norms every crossing time, and profiles either every 10 crossing times or once per crossing time for some initial time, depending on the behavior of the solution. We also calculate the L_∞-norms of the difference from the exact solution for g_{xx} and g_{zz} for the expanding direction.