IDAxiBrillBH

Paul Walker, Steve Brandt,
some cleanups by Jonathan Thornburg

Date: 2006/07/17 15:50:56

Abstract

Thorn **IDAxiBrillBH** provides analytic initial data for a vacuum black hole spacetime: a single Schwarzschild black hole in isotropic coordinates plus Brill wave. This initial data is provided for **ADMBase** 3-metric and extrinsic curvature, and optionally also for the **StaticConformal** conformal factor and its 1st and 2nd spatial derivatives.

1 Purpose

The pioneer, Bernstein, studied a single black hole which is non-rotating and distorted in azimuthal line symmetry of 2 dimensional case [1]. In this non-rotating case, one chooses the condition, $K_{ij} = 0$, and

$$\gamma_{ab} = \psi^4 \hat{\gamma}_{ab}, \tag{1}$$

where γ_{ab} is the physical three metric and $\hat{\gamma}_{ab}$ is some chosen conformal three metric.

The Hamiltonian constraint reduces to

$$\hat{\Delta}\psi = \frac{1}{8}\psi \hat{R}, \tag{2}$$

where $\hat{\Delta}$ is the covariant Laplacian and $\hat{R}$ is the Ricci tensor for the conformal three metric. This form allows us to choose an arbitrary conformal three metric, and then solve an elliptic equation for the conformal factor, therefore satisfying the constraint equations ($K_{ij} = 0$ trivially satisfies the momentum constraints in vacuum). This approach was used to create "Brill waves" in a spacetime without black holes [2]. Bernstein extended this to the black hole spacetime. Using spherical-polar coordinates, one can write the 3-metric,

$$ds^2 = \psi^4(e^{2q}(dr^2 + r^2 d\theta^2) + r^2 \sin\theta d\phi^2), \tag{3}$$

where q is the Brill "packet" which takes some functional form. Using this ansatz with (2) leads to an elliptic equation for ψ which must be solved numerically. Applying the isometry condition on ψ at a finite radius, and applying $M/2r$ falloff conditions on ψ at the outer boundary (the "Robin" condition), along with a packet which obeys the appropriate symmetries (including being invariant under the isometry operator), will make this solution describe a black hole with an incident gravitational wave. The choice of $q = 0$ produces the Schwarzschild solution. The typical q function used in axisymmetry, and considered here in the non-rotating case, is

$$q = Q_0 \sin^n \theta \left[\exp\left(\frac{\eta - \eta_0^2}{\sigma^2}\right) + \exp\left(\frac{\eta + \eta_0^2}{\sigma^2}\right) \right]. \tag{4}$$

Note regularity along the axis requires that the exponent n must be even. Choosing a logarithmic radial coordinate

$$\eta = \ln \frac{2r}{m}. \tag{5}$$

(where m is a scale parameter), one can rewrite (3) as

$$ds^2 = \psi(\eta)^4[e^{2q}(d\eta^2 + d\theta^2) + \sin^2\theta d\phi^2]. \tag{6}$$

The scale parameter m is equal to the mass of the Schwarzschild black hole, if $q = 0$. In this coordinate, the 3-metric is

$$ds^2 = \tilde{\psi}^4(e^{2q}(d\eta^2 + d\theta^2) + \sin^2\theta d\phi^2), \tag{7}$$

and the Schwarzschild solution is

$$\tilde{\psi} = \sqrt{2M}\cosh(\frac{\eta}{2}). \tag{8}$$

We also change the notation of ψ for the conformal factor is same as $\tilde{\psi}$ [3], for the η coordinate has the factor $r^{1/2}$ in the conformal factor. Clearly $\psi(\eta)$ and ψ differ by a factor of $\sqrt{r}$. The Hamiltonian constraint is

$$\frac{\partial^2\tilde{\psi}}{\partial\eta^2} + \frac{\partial^2\tilde{\psi}}{\partial\theta^2} + \cot\theta\frac{\partial\tilde{\psi}}{\partial\theta} = -\frac{1}{4}\tilde{\psi}(\frac{\partial^2 q}{\partial\eta^2} + \frac{\partial^2 q}{\partial\theta^2} - 1). \tag{9}$$

For solving this Hamiltonian constraint numerically. At first we substitute

$$\begin{aligned} \delta\tilde{\psi} &= \tilde{\psi} + \tilde{\psi}_0 && (10)\\ &= \tilde{\psi} - \sqrt{2m}\cosh(\frac{\eta}{2}). && (11) \end{aligned}$$

to the equation (9), then we can linearize it as

$$\frac{\partial^2\delta\tilde{\psi}}{\partial\eta^2} + \frac{\partial^2\delta\tilde{\psi}}{\partial\theta^2} + \cot\theta\frac{\partial\delta\tilde{\psi}}{\partial\theta} = -\frac{1}{4}(\delta\tilde{\psi} + \tilde{\psi}_0)(\frac{\partial^2 q}{\partial\eta^2} + \frac{\partial^2 q}{\partial\theta^2} - 1). \tag{12}$$

For the boundary conditions, we use for the inner boundary condition an isometry condition:

$$\frac{\partial\tilde{\psi}}{\partial\eta}|_{\eta=0} = 0, \tag{13}$$

and outer boundary condition, a Robin condition:

$$(\frac{\partial\tilde{\psi}}{\partial\eta} + \frac{1}{2}\tilde{\psi})|_{\eta=\eta_{max}} = 0. \tag{14}$$

2 The Resulting Slice

This thorn normalizes things so that if there is no perturbation, it produces a Schwarzschild (= Brill-Lindquist) slice of the $m = 2$ Schwarzschild spacetime.[1] You can't change this mass. :(

In any case (perturbation or not), this thorn also reports an ADM mass for the slice. This seems to be pretty reliable.

[1] This slice has an apparent horizon at a coordinate radius $r = m/2 = 1$.

3 2-D Grid and Interpolation Parameters

This thorn solves equation (12) on a 2-D (η, θ) grid. However, Cactus needs a 3-D grid, typically with Cartesian coordinates. Therefore, this thorn *interpolates* ψ and its (η, θ) derivatives to the Cartesian grid. More precisely, for each Cactus grid point, this thorn calculates the corresponding (η, θ) coordinates, and interpolates the 2-D solution to that point.

3.1 Size of the 2-D Grid

Because of the isometry condition (13), the 2-D grid need only cover the region $\eta \geq 0$; the code just takes the absolute value of η before interpolating.

The 2-D grid covers the region $|\eta| \in [0, \texttt{etamax}]$, $\theta \in [0, \pi]$, where the parameter `etamax` defaults to 5. If any 3-D grid point's $(|\eta|, \theta)$ is outside this 2-D grid, this thorn will abort with a fatal error message from the interpolator. In practice, the most common cause of such an out-of-range point is the 3-D grid having a grid point at, or very near to, the origin. For example:

```
WARNING level 1 in thorn AEILocalInterp processor 0 host ic0087 (line 1007
of /nfs/nethome/pollney/runs/CactusDev/arrangements/AEIThorns/AEILocalInterp/src
/Lagrange-tensor-product/../template.c):
  ->
   CCTK_InterpLocalUniform():
        interpolation point is either outside the grid,
        or inside but too close to the grid boundary!
        0-origin interpolation point number pt=307062 of N_interp_points=614125
        interpolation point (x,y)=(36.1875,0.955317)
        grid x_min(delta_x)x_max = -0.0199336(0.0199336)6.01993
        grid y_min(delta_y)y_max = -0.0290888(0.0581776)3.17068

WARNING level 0 in thorn IDAxiBrillBH processor 0 host ic0087
  (line 484 of IDAxiBrillBH.F):
  -> error in interpolator: ierror=   -1002
```

Here the 3-D grid had a point right at the origin (which by virtue of (5) would have given $\eta = -\infty$), but some software moved the grid point by $10^{-16}m$ or so (the standard Cactus work-around to try to avoid divisions by zero), giving $\eta \equiv \ln(2 \times 10^{-16}) \approx -36$. The absolute value of this is the "x" coordinate the interpolator is complaining about.

In an ideal world, someone would enhance **IDAxiBrillBH** so it could handle a grid point at (or very near to) the origin.[2] However, so far noone has volunteered to do this.

In the meantime, a staggered grid is the "standard" work-around for this problem.

[2] See the file `doc/TODO` in the **IDAxiBrillBH** source code for some ideas on how this might be done.

3.2 Resolution of the 2-D Grid

The parameters neta and nq specify the resolution of this thorn's 2-D grid in η and θ respectively.[3] The default values are a reasonable starting point, but you may need to increase them substantially if you need very high accuracy (very small constraint violations).

To help judge what resolution may be needed, this thorn has an option to write out $\psi(\eta)$ and ψ on the 2-D grid to an ASCII data file where it can be examined and/or plotted. To do this, set the Boolean parameter output_psi2D, and possibly also the string parameter output_psi2D_file_name.

3.3 Interpolation Parameters

This thorn uses the standard Cactus CCTK_InterpLocalUniform() local interpolation system for this interpolation. The interpolation operator is specified with the interpolator_name parameter (this defaults to "uniform cartesian", the interpolation operator provided by thorn **CactusBase/LocalInterp**).

The interpolation order and/or other parameters can be specified in either of two ways:[4]

- The integer parameter interpolation_order may be used directly to specify the interpolation order.
- More generally, the string parameter interpolator_pars may be set to any nonempty string (it defaults to the empty string). If this is done, this parameter overrides interpolation_order, and explicitly specifies a parameter string for the interpolator.

Note that the default interpolator parameters specify *linear* interpolation. This is rather inaccurate, and (due to aliasing effects between the 2-D and 3-D grids) will give a fair bit of noise in the metric components. You may want to specify a higher-order interpolator to reduce this noise.

For example, for one test series where I (JT) needed very accurate initial data (I wanted the initial-data errors to be much less than the errors from 4th order finite differencing on the 3-D Cactus grid), I had to use a resolution of 1000 in η and 2000 in θ, together with either 4th order Lagrange or 3rd order Hermite interpolation (provided by thorn **AEIThorns/AEILocalInterp**) to get sufficient accuracy.

One problem with such high resolutions is that **IDAxiBrillBH** uses an internal multigrid solver which allocates local arrays on the stack, whose size depends on the η and θ resolutions. For high resolutions these arrays may exceed system- and/or shell-imposed limits on the maximum stack size, causing the code to crash (core-dump). In an ideal world, someone would fix the offending code to allocate large arrays on the heap. Unless/until that happens, you can either use lower resolution :(, or try raising the operating-system and/or shell stack-size limits. For example, using tcsh the shell command limit shows the current limits, and limit stacksize unlimited raises your limit to as much as the operating system will allow. Using bash the corresponding commands are ulimit -a and ulimit -s unlimited.

[3] Internally, this thorn uses "q" to refer to θ in Fortran code, with the q function of (4) being hidden in the Mathematica files (and not present in the Fortran code). Noone seems to know *why* the code does things this way... Unfortunately, this renaming has leaked out into the parameter names...

[4] Notice that, for historical reasons, the interpolation parameter names are a bit inconsistent: interpolat<u>ion</u>_order versus interpolat<u>or</u>_name and interpolat<u>or</u>_pars.

4 Physical or Conformal Metric

By default, **IDAxiBrillBH** generates initial data which uses a nontrivial static conformal factor (as defined by thorn **StaticConformal**). This initial data includes both the conformal factor and its 1st and 2nd spatial derivatives, so **IDAxiBrillBH** sets `conformal_state` to 3.

However, if the Boolean parameter `generate_StaticConformal_metric` is set to `false`, then **IDAxiBrillBH** generates a pure physical 3-metric (and sets `conformal_state` to 0). This is useful if you have other thorns which don't grok a conformal metric.

5 Debugging Parameters

This thorn has options to print very detailed debugging information about internal quantities at selected grid points. This is enabled by setting the integer parameter `debug` to a positive value (the default is 0, which means no debugging output). See `param.ccl` and the source code `src/IDAxiBrillBH.F` for details.

References

[1] D. Bernstein, Ph.D thesis University of Illinois Urbana-Champaign, (1993)

[2] D. S. Brill,Ann. Phys.**7**, 466 (1959)

[3] K. Camarda, Ph.D thesis University of Illinois Urbana-Champaign, (1998)