1 Equations

The ADM code solves the evolution equations,

$$\begin{aligned} \partial_t g_{ij} - \pounds_\beta g_{ij} &= -2\alpha K_{ij} && (1) \\ \partial_t K_{ij} - \pounds_\beta K_{ij} &= -D_i D_j \alpha + \alpha \left(R_{ij} + K K_{ij} - 2K_i^l K_{lj} \right), && (2) \end{aligned}$$

without imposing the constraint equations,

$$\begin{aligned} R - K_{ij}K^{ij} + K^2 &= 0 && (3) \\ D_j \left(K^{ij} - g^{ij} K \right) &= 0. && (4) \end{aligned}$$

The constraint equations are used to monitor the solution obtained from solving equations (1) and (2). In equation (1), g_{ij} is the 3-metric, K_{ij} is the extrinsic curvature, α is the lapse and β^i is the shift vector. The lie derivative of a rank 2 symmetric tensor, T_{ij}, is given by

$$\pounds_\beta T_{ij} = \beta^k \partial_k T_{ij} + T_{lj} \partial_i \beta^l + T_{il} \partial_j \beta^l. \quad (5)$$

D_i is the covariant derivative compatible with g_{ij} and R_{ij} is the 3-Ricci tensor, which is computed by contracting the 3-Riemann tensor, R_{ikjl} as $R_{ij} = g^{kl} R_{ikjl}$ where,

$$R_{ikjl} = \frac{1}{2} \left[\partial_l \partial_k g_{ij} - \partial_l \partial_i g_{kj} - \partial_k \partial_j g_{il} + \partial_i \partial_j g_{kl} \right] + \left[\Gamma_{nij} \Gamma^n_{kl} - \Gamma_{nil} \Gamma^n_{kj} \right]. \quad (6)$$

The symmetries of the Riemann tensor are taken into account when computing it. In 3-dimensions the number of independent components is 6, same as the number of independent components of the Ricci tensor. The Ricci tensor is then computed by contracting the Riemann tensor which is computed step by step from equation (6). Second and mixed derivatives of the 3-metric are computed as are both types of christoffel symbols.

The christoffel symbols,Γ^k_{ij} are computed from the associated Christoffel symbols,

$$\Gamma_{kij} = \frac{1}{2} \left(g_{ij,k} + g_{ik,j} - g_{jk,i} \right). \quad (7)$$

The first term on the right-hand-side of (2), $D_i D_j \alpha$ is given by

$$D_i D_j \alpha = \partial_i \partial_j \alpha - \Gamma^k_{ij} \partial_k \alpha. \quad (8)$$

Equations (1) and (2) then become

$$\begin{aligned}
\partial_t g_{ij} - \beta^k \partial_k g_{ij} &= -2\alpha K_{ij} + g_{lj}\partial_i \beta^l + g_{il}\partial_j \beta^l & (9)\\
\partial_t K_{ij} - \beta^k \partial_k K_{ij} &= -\partial_i \partial_j \alpha + \Gamma^k_{ij}\partial_k \alpha + \alpha \left[R_{ij} + K K_{ij} - 2K^l_i K_{lj}\right] \\
&\quad + K_{lj}\partial_i \beta^l + K_{il}\partial_j \beta^l, & (10)
\end{aligned}$$

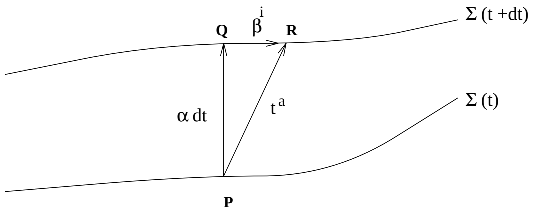
where R_{ij} and Γ^k_{ij} are computed as described above. Let ${}^{(g)}\mathcal{R}_{ij}$ denote the right-hand-side of equation (9) and ${}^{(K)}\mathcal{R}_{ij}$ the right-hand-side of equation (10). Note that we have placed the $\beta^k \partial_k g_{ij}$ term on the left-hand-side. We handle terms of this type differently in our computation from the right-hand-sides as shown. They are handled using causal differencing, which we shall describe in the next section. The spatial derivatives on the right-hand-side are all presently computed in the same manner at the interior and inner and outer boundaries. In the sections that follow we discuss the differencing of equations (9) and (10) and how the time updates are carried out to obtain the solution.

2 Differencing

In the sections that follow we discuss the discrete form of equations (9) and (10) and their solution. The notation that we adopt to show time grid-index the space grid-indices is (1) raised indices with parentheses on the right side of any quantity denotes the time index and (2) lowered indices in capital letters denote the spatial indices. For example at time level n and spatial grid point I, J, K the metric may be written as $g^{(n)}_{ij(I,J,K)}$. Our computational domain, $\hat{\Sigma}$, consists of a 3D uniform Cartesian grid. Typically if black hole data is being evolved the interior of the black hole apparent horizon is excised from the computational domain. We define $\hat{S}$ as the set of Cartesian gridpoints that defines the inner boundary(a 2-surface). Points contained in $\hat{S}$ are excluded from the computational domain. The 3-metric and extrinsic curvature data are defined everywhere on $\hat{\Sigma}$ except interior to $\hat{S}$.

We use a 3-level scheme to update data at the $n+1$ timelevel. The time derivatives in equations (9) and (10) are differenced to $O(\delta t^2)$. The difference equations are then of the form,

$$u^{(n+1)} = u^{(n-1)} + 2\delta t \mathcal{R}^{(n)}. \quad (11)$$



Where $\mathcal{R}^{(n)}$ is the right-hand-side evaluated at time-level n. We use a non-staggered leapfrog scheme to update the 3-metric and extrinsic curvature at the $n+1$ level. There are two sets of steps involved in updating the data. The first involves evaluating the right-hand-sides of equations (9) and (10). The second step is the causal difference step that takes the $\beta^k\partial_k$ terms into account. The first step will be discussed in more detail. For now we focus on the causal difference approach that we use.

2.1 Causal Differencing

Causal differencing handles the $\beta^k\partial_k$ terms in the Lie derivative terms in the Einstein equations. On the left-hand-sides of (9) and (10) define

$$\partial_0 = \partial_t - \beta^k\partial_k. \tag{12}$$

Then the evolution equations can be rewritten as

$$\partial_0 g_{ij} = {}^{(g)}\mathcal{R}_{ij} \tag{13}$$

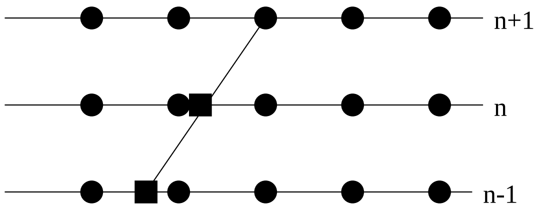
$$\partial_0 K_{ij} = {}^{(K)}\mathcal{R}_{ij}. \tag{14}$$

∂_0 is along the time-like normals, n^μ to the spacelike hypersurfaces. Let $\tilde{x}^i$ be the spatial coordinate system for observers that follow the worldlines of n^μ. Let x^i be the spatial coordinate system for observers that follow the integral curves of $t^\mu = \alpha n^\mu + \beta^\mu$. The shift vector is defined in terms of the coordinates as

$$\beta^i = -\frac{\partial x^i}{\partial \tilde{t}}. \tag{15}$$

The approach then is to evolve the 3-metric and extrinsic curvature in the frame of observers moving along n^μ, evolve the coordinates x^i as well, interpolate the values of g_{ij} and K_{ij} at the new coordinates. The approach taken in the context of a leapfrog update scheme then involves the following steps:

Figure 1: The circles in this figure show the coordinate points in the frame of observers moving along the time-like normal, n^μ, to Σ. The squares show the interpolation points required and the straight line denotes the world line of coordinate observers.



1 Evaluate the right-hand-sides ${}^{(g)}\mathcal{R}_{ij}^{(n)}$ and ${}^{(K)}\mathcal{R}_{ij}^{(n)}$ at level n.

2 Compute shifted coordinates at levels $n-1$ and n (shown as squares in figure (1)

3 Interpolate the right-hand-sides to these new shifted coordinates at the n timelevel.

4 Interpolate g_{ij} and K_{ij} at the shifted coordinates on the $n-1$ timelevel.

5 Carry out a leapfrog update.

In terms of difference equations this can be written as

$$g_{ij}^{(n+1)} = \mu(y^{l(n-1)})g_{ij}^{(n-1)} + 2\delta t\mu(y^{l(n)})^{(g)}\mathcal{R}_{ij}^{(n)} \tag{16}$$

$$K_{ij}^{(n+1)} = \mu(y^{l(n-1)})K_{ij}^{(n-1)} + 2\delta t\mu(y^{l(n)})^{(K)}\mathcal{R}_{ij}^{(n)} \tag{17}$$

where $\mu(y^l)$ is an interpolation operator that interpolates to the point y^l. In our case, $y^{l(n-1)}$ defines the coordinates at the $n-1$ time level to be interpolated to and $y^{l(n)}$ the coordinates to be interpolated to at the n time level. This handles the $\beta^k\partial_k$ terms in the evolution equations. We obtain a $O(h^2)$ solution for the Schwarzschild black hole in Eddington-Finkelstein coordinates using this method.

2.2 Spatial differencing

The spatial derivatives in ${}^{(g)}\mathcal{R}_{ij}$ and ${}^{(K)}\mathcal{R}_{ij}$ are all evaluated using centered differencing in the interior of the computational domain to second order in the grid spacing, h. At the boundaries derivatives are computed to $O(h^2)$ using either backwards or forwards differencing. Currently, we are working on evaluating spatial derivatives to first order in h at the inner and outer boundaries. In the following discussion we list out the difference stencils as we use them in the ADM code.

2.2.1 Interior Differencing

First derivatives in the interior are all evaluated using centered differences. For example an x-derivative would be evaluated using,

$$\Delta_x^h u_{(I,J,K)}^{(n)} = \frac{1}{2h}\left(u_{(I+1,J,K)}^{(n)} - u_{(I-1,J,K)}^{(n)}\right). \tag{18}$$

Second derivatives would be evaluated as

$$\Delta_x^h \Delta_x^h u_{(I,J,K)}^{(n)} = \frac{1}{h^2}\left(u_{(I+1,J,K)}^{(n)} + u_{(I-1,J,K)}^{(n)} - 2u_{(I,J,K)}^{(n)}\right). \tag{19}$$

Mixed derivatives are computed for example as

$$\begin{aligned}\Delta_x^h \Delta_x^h u_{(I,J,K)}^{(n)} = & \frac{1}{4h^2}\left(u_{(I+1,J+1,K)}^{(n)} - u_{(I+1,J-1,K)}^{(n)} - u_{(I-1,J+1,K)}^{(n)}\right. \\ & \left. + u_{(I-1,J-1,K)}^{(n)}\right).\end{aligned} \tag{20}$$

2.2.2 Inner boundary Differencing

The differencing at the inner boundary must keep track of which points are the computational domain and can be used to compute derivatives. The location of the inner boundary gridpoints are stored in a one-dimensional list. The l-th point in this list has grid indices, (I_l, J_l, K_l). For each boundary point we need to know which finite difference stencil to use. This information is stored inside another one-dimensional list d_l; where the subscript l denotes the l-th boundary point. Each time the inner boundary geometry changes these lists are reconstructed.

To compute first spatial derivatives in the x-direction for the lth boudary point to $O(h^2)$ we use for instance,

$$\begin{aligned}\Delta_x^h u_{(I_l,J_l,K_l)}^{(n)} = & \frac{1}{2h}\left\{d_l\left[-3u_{(I_l,J_l,K_l)}^{(n)} + 4u_{(I_l+d_l,J_l,K_l)}^{(n)} - u_{(I_l+2d_l,J_l,K_l)}^{(n)}\right] + \right. \\ & \left.(1-\|d_l\|)\left[u_{(I_l+1,J_l,K_l)}^{(n)} - u_{(I_l-1,J_l,K_l)}^{(n)}\right]\right\}.\end{aligned} \tag{21}$$

d_l can have three values, 0 for centered diferencing, -1 for backwards differencing and $+1$ for forward differencing. This is similarly the case for y and z derivatives.

To compute second spatial derivatives the same approach is taken, for example:

$$\begin{aligned}\Delta_x^h \Delta_x^h u^{(n)}_{(I_l,J_l,K_l)} &= \frac{\|d\|}{h^2}\left\{-u^{(n)}_{(I_l-3d_l,J_l,K_l)} + 2u^{(n)}_{(I_l,J_l,K_l)} - 5u^{(n)}_{(I_l+d_l,J_l,K_l)} + 4u^{(n)}_{(I_l+2d_l,J_l,K_l)}\right\} + \\ &\quad \frac{(1-\|d_l\|)}{h^2}\left\{u^{(n)}_{(I_l+1,J_l,K_l)} + u^{(n)}_{(I_l-1,J_l,K_l)} - 2u^{(n)}_{(I_l,J_l,K_l)}\right\}. \end{aligned} \tag{22}$$

Where here d_l has the same range of values of $-1, 0$ and 1.

For mixed derivatives the difference operators are constructed from repeated operations of forward, centered and backwards differences. That is, at a particular boundary point it may be possible only to take a forward difference in the x-direction and centered difference in the y-direction. This would mean that the mixed derivative would be computed from a combination of forwards and centered differences. There are nine such possible combinations. For example for the xy derivative we have the following combinations: $\Delta_x^h\Delta_y^h$, $\Delta_x^h\Delta_y^{\pm h}$, $\Delta_x^{\pm h}\Delta_y^h$, $\Delta_x^{\pm h}\Delta_y^{-h}$ and $\Delta_x^{\pm h}\Delta_y^{+h}$. Δ_x^h denotes centered differencing in the x-direction, Δ^{-h} denotes backwards differencing and Δ_x^{+h} denotes forwards differencing. The choice of which difference stencil to use is based upon the geometry of the inner boundary at that point. For mixed derivatives d_l has a range of values of 1 through 9, where each stencil is chosen through a case statement.

$O(h)$ derivatives are computed in a similar fashion. Currently fully $O(h)$ inner and outer boundary derivatives are not implemented fully.

2.2.3 Outer boundary Differencing

At the outer boundaries of $\hat{\Sigma}$ we currenty use $O(h^2)$ differencing. For first and second derivatives we use purely backwards or forwards differencing to $O(h^2)$. For the mixed derivatives, we use centered differencing for one of the two derivatives where possible. For example at $I = n$(where n^3 is the size of our Cartesian grid) we would compute the first derivative in the x-direction from

$$\Delta_x^{+h} u^{(n)}_{(n,J,K)} = \frac{1}{2h}\left[3u^{(n)}_{(n,J,K)} - 4u^{(n)}_{(n-1,J,K)} + u^{(n)}_{(n-2,J,K)}\right]. \tag{23}$$

Similarly the xx derivative,

$$\Delta_x^{+h}\Delta_x^{+h}u^{(n)}_{(n,J,K)} = \frac{1}{h^2}\left[-u^{(n)}_{(n-3,J,K)} + 2u^{(n)}_{(n,J,K)} - 5u^{(n)}_{(n-1,J,K)} + 4u^{(n)}_{(n-2,J,K)}\right]. \quad (24)$$