Louisiana State University
LSU Scholarly Repository

Faculty Publications | Department of Physics & Astronomy

1-21-2000

Adaptive mesh refinement approach to the construction of initial data for black hole collisions

Peter Diener
The University of Texas at Austin

Nina Jansen
Theoretical Astrophysics Center, Copenhagen

Alexei Khokhlov
Naval Research Laboratory

Igor Novikov
Theoretical Astrophysics Center, Copenhagen

Follow this and additional works at: https://repository.lsu.edu/physics_astronomy_pubs

Recommended Citation

Diener, P., Jansen, N., Khokhlov, A., & Novikov, I. (2000). Adaptive mesh refinement approach to the construction of initial data for black hole collisions. *Classical and Quantum Gravity, 17* (2), 435-451. https://doi.org/10.1088/0264-9381/17/2/312

This Article is brought to you for free and open access by the Department of Physics & Astronomy at LSU Scholarly Repository. It has been accepted for inclusion in Faculty Publications by an authorized administrator of LSU Scholarly Repository. For more information, please contact ir@lsu.edu.

arXiv:gr-qc/9905079v1 20 May 1999

Adaptive mesh refinement approach to construction of initial data for black hole collisions.

Peter Diener [1], Nina Jansen [2], Alexei Khokhlov [3], Igor Novikov [2,4,5,6]

March 7, 2018

Abstract

The initial data for black hole collisions is constructed using a conformal-imaging approach and a new adaptive mesh refinement technique, a fully threaded tree (FTT). We developed a second-order accurate approach to the solution of the constraint equations on a non-uniformly refined high resolution Cartesian mesh including second-order accurate treatment of boundary conditions at the black hole throats. Results of test computations show convergence of the solution as the numerical resolution is increased. FTT-based mesh refinement reduces the required memory and computer time by several orders of magnitude compared to a uniform grid. This opens up the possibility of using Cartesian meshes for very high resolution simulations of black hole collisions.

1 Introduction

This paper deals with the construction of initial data for black hole collisions on a high resolution Cartesian adaptive mesh. The problem of black hole collisions is an important problem of the dynamics of spacetime, and has applications to future observations of gravitational waves by gravitational observatories on Earth and in space.

The problem of black hole collisions is highly nonlinear and can only be solved numerically. A solution must be obtained within a large computational domain in order to follow the outgoing gravitational waves far enough from the source. At the same time, very high resolution is required near the black holes to describe the nonlinear dynamics of spacetime. Integration of the collision problem on a three-dimensional uniform mesh requires enormous computational resources, and this remains one of the major obstacles to obtaining an accurate solution.

Adaptive mesh refinement (AMR) can be used to overcome these problems by introducing high resolution only where and when it is required. AMR is widely used in many areas of computational physics and engineering. It has been applied in a more limited way in general relativity [1]. There are several types of AMR. In a grid-based AMR, a hierarchy of grids is created, with finer grids overlayed on coarser grids if a higher resolution is required [2]. An unstructured mesh approach uses meshes consisting of cells of arbitrary shapes and various sizes [3]. A cell-based approach to AMR uses rectangular meshes that are refined at the level of individual cells. This approach combines high accuracy of a regular mesh with flexibility of unstructured AMR [4]. The new introduced fully threaded tree (FTT) structure, which we use here, leads to an efficient, massively parallel implementation of a cell-based AMR [5].

[1]University of Texas at Austin, Austin, TX 78712
[2]Theoretical Astrophysics Center Juliane Maries Vej 30, 2100 Copenhagen, Denmark
[3]Code 6404, Naval Research Laboratory, Washington, DC 20375
[4]University Observatory, Juliane Maries Vej 30, 2100 Copenhagen, Denmark
[5]Astro Space Center of P.N. Lebedev Physical Institute, Profsoyuznaya 84/32, Moscow, 117810, Russia
[6]NORDITA, Blegdamsvej 17, 2100, Copenhagen, Denmark

The first step in solving the black hole collision problem is to construct the initial data. The widely used conformal-imaging approach has been proposed in [6],[7],[8] and developed in [9],[10]. Another approach to the construction of initial data was recently proposed in [11]. Approaches for constructing initial data for certain specific cases of black hole collisions were proposed in [12],[13], [14]. The conformal-imaging approach [6],[7],[8],[9],[10] consists of constructing the extrinsic curvature (momentum constraints equations) using an imaging technique and then solving a nonlinear elliptic equation for the conformal factor (energy constraint) with an appropriate mirror-image boundary condition. This approach is adopted in this paper.

A numerical technique for obtaining initial data for black hole collisions on a uniform Cartesian grid using conformal-imaging approach is described in [10]. Two major problems with this approach mentioned in [10] are the low resolution of a uniform grid near black holes, and low-order accuracy and programming complexity of the inner boundary conditions at the black hole throats. In [10], first-order accurate boundary conditions were implemented. The goal of this paper is to construct initial data for black hole collisions on a high resolution, Cartesian FTT adaptive mesh. In the process of realizing this goal, we found that the accuracy of the solution critically depend on the accuracy of the numerical implementation of the inner boundary condition. We developed a simple second-order accurate algorithm for boundary conditions to deal with this difficulty.

The paper is organized as follows. The next Section 2 presents the formulation of the problem and the equations solved. Section 3 describes the FTT technique, the finite-difference discretization of the problem, and the numerical solution techniques. Section 4 presents the results of the solutions for various configurations of two black holes configurations and compares these with existing solutions.

2 Formulation of the problem

The ADM or 3+1 formulation of the equations of general relativity works with the metric γ_{ij}^{ph} and extrinsic curvature K_{ij}^{ph} of three-dimensional spacelike hypersurfaces embedded in the four-dimensional space-time, where $i,j = 1,2,3$, and the superscript ph denotes the physical space. On the initial hypersurface, γ_{ij}^{ph} and K_{ij}^{ph} must satisfy the constraint equations [6]. The conformal-imaging approach assumes that the metric is conformally flat,

$$\gamma_{ij}^{ph} = \phi^4 \gamma_{ij} \quad , \tag{1}$$

where γ_{ij} is the metric of a background flat space. This conformal transformation induces the corresponding transformation of the extrinsic curvature

$$K_{ij}^{ph} = \phi^{-2} K_{ij} \quad . \tag{2}$$

With the additional assumption of

$$trK = 0 \quad , \tag{3}$$

the energy and momentum constraints are

$$\nabla^2 \phi + \frac{1}{8} \phi^{-7} K_{ij} K^{ij} = 0 \quad , \tag{4}$$

and

$$D_j K^{ij} = 0 \quad , \tag{5}$$

respectively, where ∇^2 and D_j are the laplacian and covariant derivative in flat space.

A solution to (5) for two black holes with masses M_δ, linear momenta $\mathbf{P}_\delta$ and angular momenta $\mathbf{S}_\delta$, where $\delta = 1,2$ is the black hole index, is [9]

$$K_{ij}(\mathbf{r}) = K_{ij}^{lin}(\mathbf{r}) + K_{ij}^{ang}(\mathbf{r}) \quad , \tag{6}$$

where

$$K_{ij}^{lin}(\mathbf{r}) = 3 \sum_{\delta=1}^{2} \left(\frac{1}{2R_\delta^2} \left(P_{\delta,i} n_{\delta,j} + P_{\delta,j} n_{\delta,i} - \left(\gamma_{ij} - n_{\delta,i} n_{\delta,j} \right) P_{\delta,k} n_\delta^k \right) \right) \tag{7}$$

and

$$K_{ij}^{ang}(\mathbf{r}) = 3 \sum_{\delta=1}^{2} \left(\frac{1}{R_\delta^3} \left(\epsilon_{kil} S_\delta^l n_\delta^k n_{\delta,j} + \epsilon_{kjl} S_\delta^l n_\delta^k n_{\delta,i} \right) \right) \quad . \tag{8}$$

In (7) and (8), the comma in the subscripts separates the index of a black hole from the coordinate component indices and is not a symbol of differentiation, $R_\delta = M_\delta/2$ is the black hole throat radius, and $\mathbf{n}_\delta = (\mathbf{r} - \mathbf{r}_\delta)/|\mathbf{r} - \mathbf{r}_\delta|$ is the unit vector directed from the center of the δ-th black hole $\mathbf{r}_\delta$ to the point $\mathbf{r}$. We work in units where $G = 1,\ c = 1$.

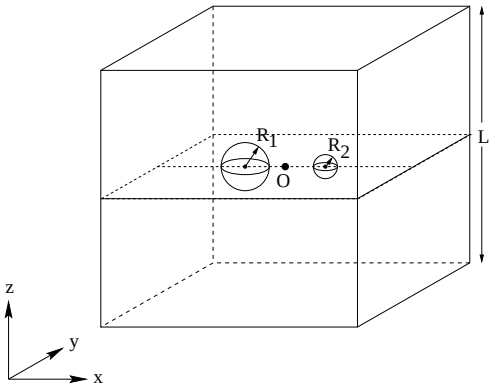

Figure 1: This figure shows schematically the computational domain used in the computations. The computational domain is a cubic box of size L. Two black holes with throat radii R_1 and R_2 are positioned on the X axis in the XY plane ($z = 0$) at equal distances from the origin, O, of the coordinate system.

The inversion-symmetric solution to (5) can be obtained from (6) by applying an infinite series of mirror operators to (6), as described in [9]. Note, that before applying the mirror operators to K_{ij}^{ang}, this term must be divided by 2 since the image operators will double its value. The series converges rapidly, and in practice only a few terms are taken. In this paper we take first five terms (for details see [15]).

After the isometric solution for K_{ij} is found, (4) must be solved subject to the isometry boundary condition at the black hole throats

$$n_\delta^i D_i \phi = -\frac{\phi}{2R_\delta} \quad , \tag{9}$$

and the outer boundary condition $\phi \to 1$ when $r \to \infty$. This boundary condition is represented by [9]

$$\frac{\partial \phi}{\partial r} = \frac{1-\phi}{r} \tag{10}$$

where r is the distance from the center of the computational domain to the boundary.

3 Numerical method

We work in the Cartesian coordinate system in the background flat space and find the solution within a cubic computational domain of size L. Figure 1 shows the schematic representation of the computational domain. Centers of the throats of black holes are located in the XY plane $z = 0$ on the line $y = 0$ at equal distances from the origin. The space inside the black hole throats is cut out. The inner boundary condition (9) is imposed at the surface of the two spheres, and the outer boundary condition is imposed at the border of the computational domain.

Figure 2 shows the cut of the computational domain through the $z = 0$ plane and gives an example of an adaptive mesh used in computations. The values of all variables are defined at cell centers. Figure 2 shows that coarse cells are used at large distances from the black holes, and the finest cells are used near the throats where the gradient of ϕ is large. The grid is refined to achieve a desired accuracy of the solution as described below.

There are three types of cells. The first type are internal cells that are actually used in computations (these cells are white in Figure 2). The layer of boundary cells which is one cell wide along the outer border of the computational domain is used to define the outer boundary conditions. The layers of boundary cells inside the throats, two cells wide, are used to define the inner boundary conditions. A cell is considered as located inside a throat if its center is located inside. Boundary cells are indicated as shaded on Figure 2.

We solve (4) as follows. Similar to [11], we introduce a new unknown variable

$$u = \phi - \alpha^{-1} \quad , \tag{11}$$

where

$$\alpha^{-1} = \sum_{\delta=1}^{2} \left(\frac{R_\delta}{|\mathbf{r} - \mathbf{r}_\delta|} \right) \quad . \tag{12}$$

The reason for using this transformation is that u varies slower than ϕ near the throats, and is

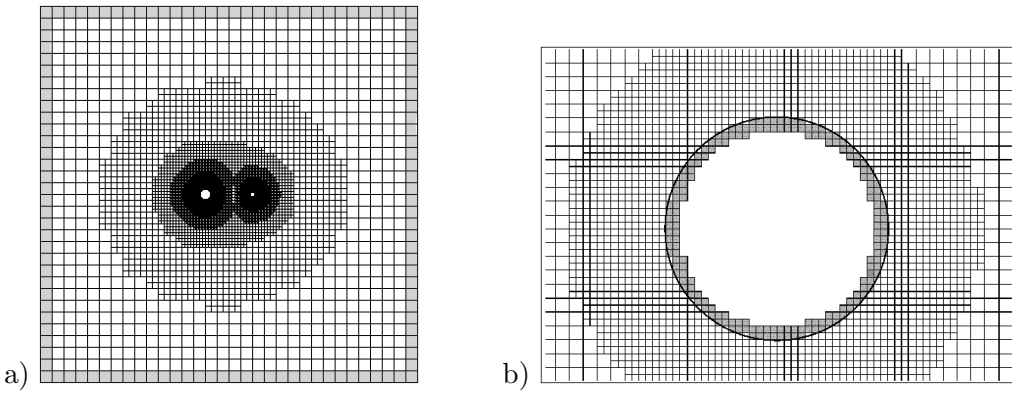

Figure 2: A cut of a computational domain along the XY plane ($z = 0$) showing an adaptive mesh. The resolution increases near the black holes. Internal cells are unshaded while boundary cells are shown shaded.

more convenient for numerical calculations (see Section 4.1). In Cartesian coordinates, (4) then becomes

$$\nabla^2 u + F(u) = 0 \quad , \tag{13}$$

where

$$F(u) = \frac{\beta}{(1+\alpha u)^7} \quad , \tag{14}$$

and

$$\beta = \frac{1}{8}\,\alpha^7 K_{ij} K^{ij} \quad . \tag{15}$$

Equation (13) is a nonlinear elliptic equation. Below we describe the numerical procedure of finding its solution. First consider a cell of size Δ which has six neighbors of the same size. Let us number this cell and its neighbors with integers from 0 to 6, respectively. Then the discretized form of (13) is

$$\frac{u_1+u_2+u_3+u_4+u_5+u_6-6u_0}{\Delta^2} + F(u_0) = 0 \tag{16}$$

A finite-difference form of (13) is more complicated for cells that have neighbors of different sizes and may involve larger number of neighbors in order to maintain second order accuracy. This is described in Appendix A. In general, for every internal cell, the finite-difference discretization may be written as

$$f(u_0, u_1, u_2, ..., u_n) = 0 \quad , \tag{17}$$

where $u_1, ..., u_n$ are the values of u in n neighboring points chosen to represent the finite-difference stencil of a cell. In the particular case of a cell with all equal neighbors, f is defined by the left-hand side of (16).

We solve the set of (17) by the Newton Gauss-Seidel method [16], that is, we obtain a new guess of u_0^{new} using Newton iteration with respect to the unknown u_0

$$u_0^{new} = u_0 - f(u_0, ...) \left(\frac{\partial f(u_0, ...)}{\partial u_0} \right)^{-1} \quad . \tag{18}$$

Then we accelerate the convergence by using a successive overrelaxation (SOR)

$$u_0^{new} = \omega u_0^{new} + (1-\omega) u_0 \quad , \tag{19}$$

where ω is the overrelaxation parameter. For a simple case of all equal neighbors, (18) can be written as

$$u_0^{new} = u_0 + \frac{(u_1+u_2+u_3+u_4+u_5+u_6-6u_0)}{6 - \Delta^2 \left(\frac{dF(u_0)}{du_0} \right)} + \frac{\left(\Delta^2 F(u_0) \right)}{6 - \Delta^2 \left(\frac{dF(u_0)}{du_0} \right)} \quad . \tag{20}$$

For stencils with non-equal neighbors the discretization of equation (13) is given in Appendix A, from which the expressions for the Newton Gauss-Seidel iterations in those cases can be written explicitly.

We select the value of ω from the interval $[1, \omega_{max}]$ by the method described in [17]. The value of ω_{max} is initially set to $\omega_{max} = 1.995$. If the solution begins to diverge during the iterations, ω is reset to 1, ω_{max} is decreased by 2%,

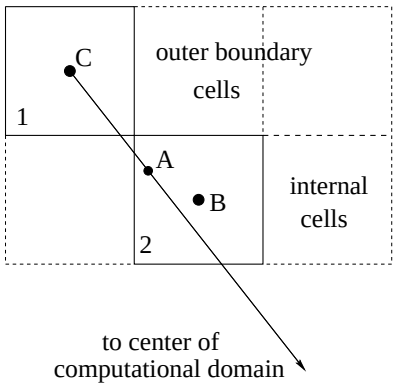

Figure 3: An example of the computation of outer boundary values. Outer boundary cells are the three cells located at the top of the figure. Cell 1 is an outer boundary cell, with center at point C. Point A is the point located one cell size away from the C on the line connecting point C with the center of the computational domain. Point A is located inside cell 2, which is an internal cell centered on point B. The value of ϕ at point A, to be used in the outer boundary condition, is interpolated from point B with second order accuracy using the derivatives evaluated at point B. These derivatives depend on ϕ at Cell 2 and it's neighboring cells, including outer boundary cells.

and the relaxation is continued allowing ω to increase up to the new ω_{max} or until the solution starts to diverge again. Iterations are terminated when

$$\frac{u_0^{new} - u_0}{u_0} < \varepsilon \qquad (21)$$

for all u_0, where ε is a predefined small number. In this paper, we do not attempt to accelerate the iterations using, for example, a multigrid or other sophisticated techniques since the initial value problem must be solved only once.

The procedure described above assumes that the values of u are known at all neighbors. For internal cells that are close to a boundary, these values are substituted with the values of u in boundary cells. Now we describe the procedure of assigning values of u to boundary cells. A similar technique was used for fluid flow simulations about complex bodies [18].

Figure 3 illustrates the process for the outer boundary. In this figure, cell 1 is a boundary cell. We need to define a new value u_C^{new} in its center, point C. We find another point, A, which is located at a distance Δ (equal to the size of cell 1) from point C along the line that connects C with the center of the computational domain. The value of u_A is found by the second order interpolation using the values of u in cell 2 and all of its neighbors. The interpolation involves old values of u in both internal cells and boundary cells. Then ϕ_A is computed using equation (11) with α^{-1} evaluated in point A. The finite-difference expression of (10) can be written as

$$\frac{\phi_C^{new} - \phi_A}{\Delta} = \frac{1 - (\phi_C^{new} + \phi_A)/2}{(r_C - \Delta/2)} \quad , \qquad (22)$$

and then solved for ϕ_C^{new}. The value of u_C^{new} is then finally found using equation (11) with α^{-1} calculated at point C.

Figure 4 illustrates the process of defining u for the inner boundary. Cell 1 is located inside a throat, and we need to define a new value u_C^{new} in its center, point C. The point A is a point outside the throat that has the same distance to the inner boundary as C, along the normal to the throat passing through point C. Let us denote the distance between C and A as Δ_{AC}. Again, the value of u_A is found by second-order interpolation using old values of u in cell 2 and its neighbors, and ϕ_A is calculated using equation (11). Then the boundary condition (9) becomes

$$\frac{\phi_C^{new} - \phi_A}{\Delta_{AC}} = \frac{\phi_A + \phi_C^{new}}{4R_\delta} \quad , \qquad (23)$$

which can be solved for ϕ_C^{new}; R_δ is the radius of the throat that contains point C. As before the value of u_C^{new} is then finally found using equation (11) with α^{-1} evaluated at point C.

After all inner and outer boundary points are defined, the next iteration (21) is performed for internal cells and so on, until the iterations converge. The advantage of the method described above is that the new method applies the same

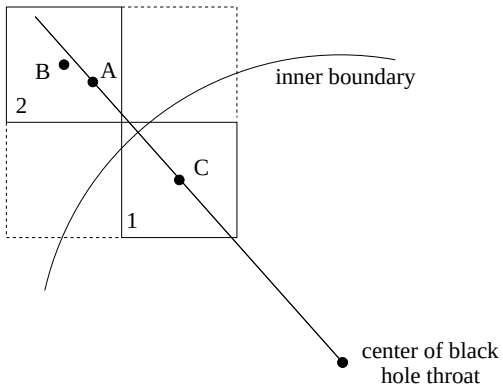

Figure 4: An example of the computation of inner boundary values. Cell 1, centered on point C, is the inner boundary cell. Point A is the point located outside of the black hole throat on the line between the center of the black hole throat and point C that has the same distance to the boundary as point C. Point A is located inside cell 2, centered on point B. The value at point A, to be used in the inner boundary condition, is interpolated from point B with second order accuracy using the derivatives evaluated at point B. These derivatives depend on the value of Cell 2 and it's neighboring cells, including inner boundary cells.

numerical algorithm to all cells, and that this algorithm is second-order accurate. The method described in [10] required 77 different numerical stencils corresponding to different relative positions of the boundary and interior cells, and was only first-order accurate.

The computational domain used by the FTT is a cube of size L. It can be subdivided to a number of cubic cells of various sizes $1/2,\ 1/4,\ 1/8, ...$ of L. Cells are organized in a tree with the direction of thread pointers inverted. These pointers are directed from children to neighbors of parent cells, as described in [5]. The most important property of an FTT data structure is that all operations on it, including tree refinement and derefinement, can be performed in parallel. A computer memory overhead of FTT is extremely small: two integers per computational cell. All coding was done by using the FTTLIB software library [5] that contains functions for refinement, derefinement, finding neighbors, children, parents, coordinates of a cell, and performing parallel operations.

We characterize an FTT mesh by the minimum and maximum levels of leaves (unsplit cells) present in the tree, l_{min} and l_{max}. We construct an adaptively refined mesh by starting with one computational cell representing the entire computational domain and then by subsequently subdividing it by a factor of two until we reach the level l_{min}. We find a coarse solution for two black holes at level l_{min} using $u = 1$ as an initial guess. After this solution is obtained, we identify the regions that require more refined cells. These regions are then refined once, and a new converged solution is obtained. The old coarse solution is used as an initial guess for the new one. The procedure is repeated until the level of refinement reaches l_{max}.

4 Test results

4.1 One Schwarzschild black hole: one-dimensional test

Before presenting the results of three-dimensional test computations on an FTT mesh, we will discuss the accuracy of our numerical method using a simpler one-dimensional test problem. A single Schwarzschild black hole at rest has an analytic solution for the conformal factor

$$\phi = 1 + \frac{R}{r} \quad , \tag{24}$$

where R is the throat radius and r is the distance in the background space from the center of the black hole throat. The one-dimensional test was performed on a uniform grid in order to assess the influence of our treatment of boundary conditions (22) and (23), and the effects of changing the variable ϕ to a new variable u (11) on the accuracy of the solution. In these calculations, we used a uniform one-dimensional grid consisting of $n = 16$, 32, and 64 cells. Cells 2 through $n - 1$ were interior cells. Cells 1 and n

were the inner and outer boundary cells, respectively. The computations were performed for the grid size $L = 10$, throat radius $R = 1$, and using convergence criterion $\varepsilon = 6 \times 10^{-15}$. The throat was located between the first and second cell centers at a distance Δr from the center of the border cell 1. Three values of $\Delta r = 0.05\Delta$, 0.5Δ, and 0.95Δ were considered, where Δ is the cell size. When $\Delta r = 0.5\Delta$, the throat is located exactly in the middle between the points A and C in (22), and the inner boundary condition (22) becomes second-order accurate regardless of the order of interpolation that is used for finding u_A. For $\Delta r = 0.05\Delta$ and 0.95Δ, the overall accuracy of the inner boundary condition depends on the interpolation used for finding u_A.

Numerical solutions were obtained using three different methods: *(a)* solving the finite-difference form of (4) for the original unknown variable ϕ and using *first order* interpolation to find ϕ_A (the rest of the boundary condition procedure was identical to that described in Section 2), *(b)* solving for the original variable ϕ but using second order interpolation, and finally, *(c)* using both the second order interpolation and solving for the new unknown u as described in Section 2. Table 1 compares the numerical and analytical solutions for these three cases by showing the maximum relative deviation of the numerical solution from the analytical solution for the interior points of the grid. As can be seen from Table 1, the accuracy varies with the grid resolution (n), the method of interpolation, and the choice of the unknown variable (ϕ or u). It also depends on the exact location of the throat relative to grid points ($\Delta r/\Delta$). As we expect various relative locations of the throats relative to grid points in three-dimensional calculations, we need a numerical procedure that provides a second-order accuracy in all cases.

Results using method *(a)* show that the accuracy of the solution using first order interpolation for ϕ_A is unacceptable. The third row of Table 1 shows that the accuracy does not improve with increasing number of cells. Results using method *(b)* show that second order interpolation for ϕ_A (method *(b)*) leads to the overall second-order algorithm. The accuracy of the solution increases roughly by a factor of four when the grid resolution is doubled. Results for method *(c)* shows that the accuracy is further improved for small Δr.

It is possible to give an analytical estimate for an error introduced by the numerical inner boundary condition (22) for the Schwarzschild black hole case. In this case, the general solution of (4), limited at infinity, is $\phi = c + b/r$. Let us assume that the numerical outer boundary condition does not introduce any error, and the solution in interior points is found exactly. Thus, the numerical approximation of the inner boundary condition is a unique source of numerical error. Then the numerical solution would have the form $\phi = 1 + b/r$. The difference between b and R in (24) then will determine the overall error in the solution. We can find b by substituting $\phi = 1 + b/r$ into (22). The estimate of the relative error then is

$$\text{Relative error} = \frac{R-b}{R} = \left(\frac{\Delta r}{2R}\right)^2 \quad . \tag{25}$$

The estimate of the error using (25) is given in Table 1 in brackets for the method *(c)*. The comparison with the numerical error indicates that for method *(c)*, the error in the solution is second-order and is determined by the accuracy of the inner boundary condition rather than by errors of numerical calculations for internal cells.

4.2 Time-symmetric initial data for two black holes.

Next, we consider the case of two black holes with $\mathbf{P}_\delta = \mathbf{S}_\delta = 0$ that have masses $M_1 = 1$, $M_2 = 2$ and located (positions of the centers of their throats) at $\mathbf{r}_1 = (-4, 0, 0)$, $\mathbf{r}_2 = (4, 0, 0)$ with finite separation $|\mathbf{r}_1 - \mathbf{r}_2| = 8$. The size of the computational domain is $L = 64$. Numerical solutions were obtained using FTT adaptive meshes with different increasing resolutions near the black hole throats. We characterize the resolution by specifying the minimum and maximum levels of cells in the tree, l_{min} and l_{max}. The cell size at a given level l is $\Delta_l = L \cdot 2^{-l}$. The computations were performed on meshes with $l_{min} = 4, 5, 6$ and $l_{max} = 6, 7, 8, 9, 10, 11$. The refinement criterion for this case was the requirement that

$$\eta = \frac{\Delta}{\phi^4}\left(\left(\frac{\partial \phi^4}{\partial x}\right)^2 + \left(\frac{\partial \phi^4}{\partial y}\right)^2 + \left(\frac{\partial \phi^4}{\partial z}\right)^2\right)^{1/2} < 0.05 \tag{26}$$

in every cell, where partial derivatives in (26) are determined by the numerical differentiation. We used $\varepsilon = 6 \times 10^{-7}$ in (21) to terminate iterations.

Accuracy of one-dimensional computations.

Method	$\Delta r/\Delta$	n= 16	n=32	n=64
	0.95	-2.9×10^{-1}	-7.2×10^{-2}	-1.8×10^{-2}
(a)	0.5	-6.1×10^{-2}	-1.7×10^{-2}	-5.0×10^{-3}
	0.05	-3.8×10^{-1}	-4.6×10^{-1}	-4.7×10^{-1}
	0.95	-2.9×10^{-1}	-7.0×10^{-2}	-1.7×10^{-2}
(b)	0.5	-6.1×10^{-2}	-1.7×10^{-2}	-5.0×10^{-3}
	0.05	-3.0×10^{-1}	-1.2×10^{-1}	-4.2×10^{-2}
	0.95	$-3.4(3.5)\times10^{-1}$	$-8.7(8.8)\times10^{-2}$	$-2.2(2.2)\times10^{-2}$
(c)	0.5	$-7.4(9.8)\times10^{-2}$	$-2.1(2.4)\times10^{-2}$	$-5.7(6.1)\times10^{-3}$
	0.05	$-9.9(9.8)\times10^{-4}$	$-1.4(2.4)\times10^{-4}$	$-5.7(6.1)\times10^{-5}$

Table 1: Relative accuracy of the numerical solutions of (4) for a Schwarzschild black hole obtained using three different methods (a,b,c), different resolutions (n), and different location of the throat relative to a grid (Δr). Numbers in parentheses for the method (c) is the accuracy estimate based on the consideration of the error introduced by the inner boundary condition (see Section 4.1).

The mirror-image symmetric analytic solution for two time-symmetric black holes is given in Appendix B (for details of derivation see [15]). Table 2 gives the comparison of the numerical solutions with fixed $l_{min} = 5$ and varying $l_{max} = 5 - 11$ with the analytical solution. The table shows the maximum deviation of a numerical solution from the analytical one. It also shows the level of a cell where the maximum error was found. From the table we see that the accuracy of the solution increases approximately linearly with increasing l_{max}, and that the maximum error is located at maximum level of refinement near the throats. When we compare solutions obtained on different meshes on which the resolution was increased on all levels simultaneously (Table 3), we observe better than linear convergence, as it should be expected.

The computations performed on an adaptive mesh allow us to save a significant amount of computational resources. For example, our solution obtained on the $l_{min} = 5$, $l_{max} = 11$ adaptive mesh used 6×10^5 computational cells. An equivalent uniform-grid computation with the same resolution near the throats would have required using a 2048^3 uniform Cartesian grid with $\simeq 8\times10^9$ cells. That is, in this case the computational gain was $\sim 10^4$.

4.3 Two black holes with linear and angular momenta

Cook et al. [10] considered the initial conditions for two black holes of equal mass $M_1 = M_2 = 2$ with non-zero linear and angular momenta (case A1B8 in Table 3 of [10]). In our coordinate system (Figure 1), the components of linear and angular momenta of the black holes for the case A18B are $\mathbf{P}_1 = -\mathbf{P}_2 = (0, 0, -14)$, and $\mathbf{S}_1 = (280, 280, 0)$ and $\mathbf{S}_2 = (0, 280, 280)$, respectively. The throats are located at $\mathbf{r}_1 = (4, 0, 0)$ and $\mathbf{r}_2 = (-4, 0, 0)$ with a relative separation equal eight. We computed the A1B8 case using the same size of the computational domain, $L = 28.8$ as that used in [10], and using a series of refined meshes with increasing resolution near the throats, $l_{min} = 5$, $l_{max} = 5 - 11$. We used the same value of ε as in Section 4.2 but used a modified mesh refinement criterion

$$\eta = \max\left(\frac{\Delta}{\phi^4}\left(\left(\frac{\partial\phi^4}{\partial x}\right)^2 + \left(\frac{\partial\phi^4}{\partial y}\right)^2 + \left(\frac{\partial\phi^4}{\partial z}\right)^2\right)^{1/2}, |K_{ij}|\right) < 0.05 \qquad (27)$$

Table 4 compares coarse solutions $l_{min} = 5$, $l_{max} = 5 - 10$ with the finest solution obtained on the $l_{min} = 5$, $l_{max} = 11$ mesh. It shows that both the maximum and the average deviation of the solutions decreases by more than two orders of magnitude when the maximum resolution near the throats is increased by a factor of 32. The solutions in [10] did not show improvement with increasing resolution (see their Table 3). Figure 5 shows the comparison of ϕ on the line passing through the centers of the throats computed in this paper with resolution

Accuracy of two black hole time-symmetric computations.

$l_{min} - l_{max}$	Max. error	l_{err}
5 - 5	3.3×10^{-2}	5
5 - 6	4.8×10^{-2}	6
5 - 7	2.9×10^{-2}	7
5 - 8	1.0×10^{-2}	8
5 - 9	1.2×10^{-2}	9
5 - 10	3.9×10^{-3}	10
5 - 11	8.9×10^{-4}	11

Table 2: The table shows the maximum relative deviation of the numerical solutions for the two Schwarzschild black holes (Section 4.2), and the level of cells l_{err} where the maximum error is located for computations with different maximum resolution.

Accuracy of two black hole time-symmetric computations.

$l_{min} - l_{max}$	Max. error
4 - 8	3.6×10^{-2}
5 - 9	1.2×10^{-2}
6 - 10	3.7×10^{-3}

Table 3: The table shows the maximum relative deviation of the numerical solutions for the two Schwarzschild black holes (Section 4.2) for computations where both minimum and maximum resolutions were increased simultaneously.

Accuracy of computations of two black hole with non-zero linear and angular momenta.

$l_{min} - l_{max}$	Max. error	Avg. error
5 - 5	1.1×10^{-1}	1.1×10^{-2}
5 - 6	6.5×10^{-2}	9.0×10^{-3}
5 - 7	1.6×10^{-2}	1.8×10^{-3}
5 - 8	1.7×10^{-2}	2.1×10^{-3}
5 - 9	5.3×10^{-3}	6.4×10^{-4}
5 - 10	7.5×10^{-4}	6.7×10^{-5}

Table 4: Comparison of coarse solutions to the finest solution on the $l_{min} = 5$, $l_{max} = 11$ mesh. The table gives the maximum relative deviation (Max. error) and the average relative deviation (Avg. error) of the numerical solutions computed in Section 4.3.

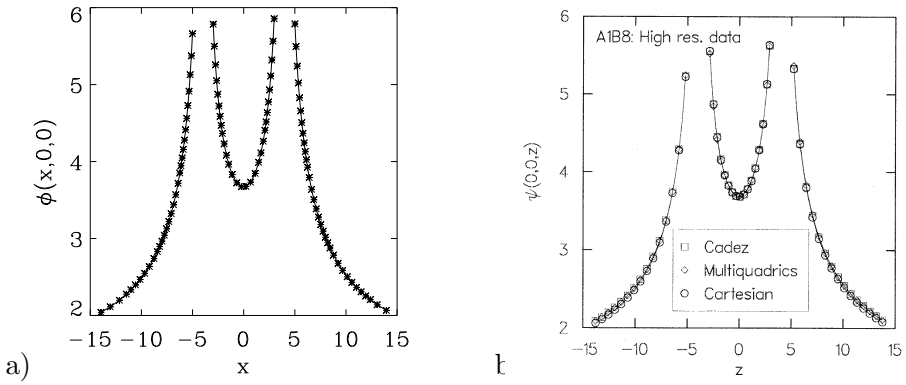

Figure 5: A comparison of ϕ for the A1B8 case presented in [10] (their Figure 4.) (b) and our computations with resolution $l_{min} = 5$ and $l_{max} = 8$ (a). Differences are not visible on the scale of these graphs, except for the fact that we have better resolution near the holes which is why we have higher values close to the holes.

$l_{min} = 5$, $l_{max} = 8$ with the results presented in [10] (their Figure 4). In our computations, finer cells cluster near the throats where the gradient in the solution is larger, whereas in [10], cells have the same size and are uniformly distributed in space. This combined with the overall second-order accuracy of our method is the reason why the adaptive mesh refinement solution improves when the mesh is refined (see Table 4).

5 Conclusions.

In this paper, we applied a new adaptive mesh refinement technique, a fully threaded tree (FTT), for the construction of initial data for the problem of the collision of two black holes. FTT allows mesh to be refined on the level of individual cells and leads to efficient computational algorithms. Adaptive mesh refinement is very important to the problem of black hole collisions because a very high resolution is required for obtaining an accurate solution.

We have developed a second-order approach to representing both the inner boundary conditions at the throats of black holes and the outer boundary conditions. This allowed us to implement an approach to the solution of the energy constraint that is formally second-order accurate. We presented results of tests for two black holes that demonstrated a good improvement of the accuracy of the solution when the numerical resolution was increased.

The FTT-based AMR approach gives a gain of several orders of magnitude in savings of both memory and computer time (Section 4.2), and opens up the possibility of using Cartesian meshes for very high resolution simulations of black hole collisions. A second-order boundary condition technique similar to that developed in this paper can be applied for the integration of initial conditions in time. We plan to use these techniques for time integration of the black hole collision problem.

Acknowledgments. This paper is based in part on the Master Thesis of Ms. Nina Jansen [15]. We thank J. Craig Wheeler, Elaine S. Oran and Jay P. Boris for their support, encouragement, and discussions, and Almadena Yu. Chtchelkanova for help with FTT. The work was supported in part by the NSF grant AST-94-17083, Office of Naval Research, DARPA, Danish Natural Science Research Council grant No.9401635, and by Danmarks Grundforskningsfond through its support for the establishment of the Theoretical Astrophysics Center.

A Appendix: Finite-difference formulas on the Fully Threaded Tree

On the FTT mesh, we use the four different types of stencils shown in Figure 6 when a cell has zero, one, two, or three neighbors that are two times larger. These stencils involve nine neighbors of a cell and the cell itself. Let us introduce the vector of partial derivatives of u at the center of

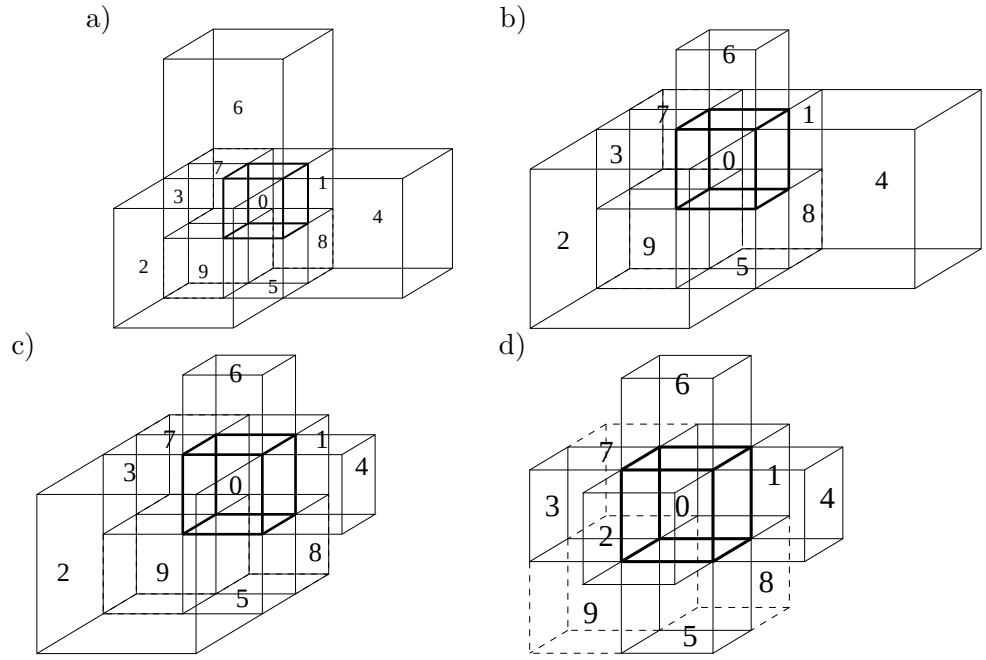

Figure 6: Examples of the different stencils used for computing derivatives are shown for the 4 different cases: a) 3 large neighbors, b) 2 large neighbors, c) 1 large neighbor and d) 0 large neighbors

cell 0

$$Du_k = (u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial x \partial z}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 u}{\partial y \partial z}, \frac{\partial^2 u}{\partial z^2}) \tag{28}$$

which includes the value of the function itself as the zeroth component. We can express the values of u_i in $i = 1, ..., 9$ neighboring cells with the second-order accurate Taylor expansion using Du_k. This leads to a linear system of ten equations for the ten Du_k unknowns. From this system, Du_k can be expressed as a weighted sum of the values of u_i in the cell 0 and its neighbors

$$Du_k = \sum_{i=0}^{9} w_k^i u_i \quad . \tag{29}$$

Due to the limited number of stencils encountered in the FTT structure, this can be done once and for all and the weights can be stored in an array.

For three bigger neighbors in the positive x-, y- and z-directions the weights are:

$$\begin{aligned}
w_1 &= \frac{1}{52\Delta}(-48, -14, 54, -1, 15, \\
&\quad -1, 15, -11, -11, 2), \\
w_2 &= \frac{1}{52\Delta}(-48, -1, 15, -14, 54, \\
&\quad -1, 15, -11, 2, -11), \\
w_3 &= \frac{1}{52\Delta}(-48, -1, 15, -1, 15, \\
&\quad -14, 54, 2, -11, -11), \\
w_4 &= \frac{1}{26\Delta^2}(-4, 14, -2, 1, -15, \\
&\quad 1, -15, 11, 11, -2), \\
w_5 &= \frac{1}{\Delta^2}(1, 0, -1, 0, -1, 0, 0, 1, 0, 0), \\
w_6 &= \frac{1}{\Delta^2}(1, 0, -1, 0, 0, 0, -1, 0, 1, 0), \\
w_7 &= \frac{1}{26\Delta^2}(-4, 1, -15, 14, -2, \\
&\quad 1, -15, 11, -2, 11), \\
w_8 &= \frac{1}{\Delta^2}(1, 0, 0, 0, -1, 0, -1, 0, 0, 1), \\
w_9 &= \frac{1}{26\Delta^2}(-4, 1, -15, 1, -15, \\
&\quad 14, -2, -2, 11, 11).
\end{aligned} \tag{30}$$

For two bigger neighbors in the positive x- and

y-directions the weights are:

$$\begin{aligned}
w_1 &= \frac{1}{56\Delta}(-48,-15,57,-1,15,\\
&\qquad -2,14,-12,-11,3),\\
w_2 &= \frac{1}{56\Delta}(-48,-1,15,-15,57,\\
&\qquad -2,14,-12,3,-11),\\
w_3 &= \frac{1}{2\Delta}(0,0,0,0,0,-1,1,0,0,0),\\
w_4 &= \frac{1}{28\Delta^2}(-8,15,-1,1,-15,\\
&\qquad 2,-14,12,11,-3),\\
w_5 &= \frac{1}{\Delta^2}(1,0,-1,0,-1,0,0,1,0,0),\\
w_6 &= \frac{1}{\Delta^2}(1,0,-1,0,0,0,-1,0,1,0),\\
w_7 &= \frac{1}{28\Delta^2}(-8,1,-15,15,-1,\\
&\qquad 2,-14,12,-3,11),\\
w_8 &= \frac{1}{\Delta^2}(1,0,0,0,-1,0,-1,0,0,1),\\
w_9 &= \frac{1}{\Delta^2}(-2,0,0,0,0,1,1,0,0,0).
\end{aligned} \tag{31}$$

For one big neighbor in the positive x direction the weights are:

$$\begin{aligned}
w_1 &= \frac{1}{30\Delta}(-24,-8,30,-1,7,-1,7,-6,-6,2),\\
w_3 &= \frac{1}{2\Delta}(0,0,0,-1,1,0,0,0,0,0),\\
w_3 &= \frac{1}{2\Delta}(0,0,0,0,0,-1,1,0,0,0),\\
w_4 &= \frac{1}{15\Delta^2}(-6,8,0,1,-7,1,-7,6,6,-2),\\
w_5 &= \frac{1}{\Delta^2}(1,0,-1,0,-1,0,0,1,0,0),\\
w_6 &= \frac{1}{\Delta^2}(1,0,-1,0,0,0,-1,0,1,0),\\
w_7 &= \frac{1}{\Delta^2}(-2,0,0,1,1,0,0,0,0,0),\\
w_8 &= \frac{1}{\Delta^2}(1,0,0,0,-1,0,-1,0,0,1),\\
w_9 &= \frac{1}{\Delta^2}(-2,0,0,0,0,1,1,0,0,0).
\end{aligned} \tag{32}$$

For zero big neighbors the mixed second order derivatives are given by the same weights as for the other three cases, while all other derivatives are given by the standard formulas for central differences on a uniform grid. Equation (13) then can be expressed as

$$Du_4 + Du_7 + Du_9 + F(u_0) = 0 \quad . \tag{33}$$

B Appendix: Conformal factor of two time-symmetric black holes.

A solution for the conformal factor of two time-symmetric black holes [19], [9] can be written in the Cartesian coordinates as [15]

$$\phi(\mathbf{r}) = 1 + \sum_{n=1}^{\infty} (F_1^n + F_2^n) \quad , \tag{34}$$

with

$$\begin{aligned}
F_1^n &= \begin{cases} F_1^{n-1}\left(R_1/\rho_{11}^{n-1}\right) & \text{for } n \text{ odd;} \\ F_1^{n-1}\left(R_2/\rho_{12}^{n-1}\right) , & \text{for } n \text{ even;} \\ 1 , & \text{for } n = 0, \end{cases}\\
F_2^n &= \begin{cases} F_2^{n-1}\left(R_2/\rho_{21}^{n-1}\right) , & \text{for } n \text{ odd;} \\ F_2^{n-1}\left(R_1/\rho_{22}^{n-1}\right) , & \text{for } n \text{ even;} \\ 1 , & \text{for } n = 0, \end{cases}
\end{aligned} \tag{35}$$

where

$$\rho_{\alpha\delta}^{n-1} = |\mathbf{x}_\alpha^{n-1} - \mathbf{r}_\delta| \ , \tag{36}$$

$\alpha = 1,2$, $\delta = 1,2$, $\mathbf{r}_\delta$ are the positions of the centers of black hole throats, R_δ are the throat radii,

$$\begin{aligned}
\mathbf{x}_1^n &= \begin{cases} \mathbf{J}_1(\mathbf{x}_1^{n-1}) , & \text{for } n \text{ odd;} \\ \mathbf{J}_2(\mathbf{x}_1^{n-1}) , & \text{for } n \text{ even;} \\ \mathbf{r} , & \text{for } n = 0, \end{cases}\\
\mathbf{x}_2^n &= \begin{cases} \mathbf{J}_2(\mathbf{x}_2^{n-1}) , & \text{for } n \text{ odd;} \\ \mathbf{J}_1(\mathbf{x}_2^{n-1}) , & \text{for } n \text{ even;} \\ \mathbf{r}, & \text{for } n = 0, \end{cases}
\end{aligned} \tag{37}$$

and

$$\mathbf{J}_\delta(\mathbf{x}) = \left(\frac{R_\delta^2}{|\mathbf{x} - \mathbf{r}_\delta|^2}\right)(\mathbf{x} - \mathbf{r}_\delta) + \mathbf{r}_\delta \quad . \tag{38}$$

References

[1] M. Ghoptuik, Phys. Rev. Lett., **70**, 9 (1993); B.M. Parashar and J.C. Browne, *An Infrastructure for Parallel Adaptive Mesh-Refinement Techniques* , Technical Report, Department of Computer Sciences, University of Texas at Austin, 2.400 Taylor Hall, Austin, TX 78712, 1995; Bruegmann, Int. J. Mod. Phys, D8,**85**, (1999); P. Papadopoulos, E. Seidel and L. Wild, gr-qc/9802069 and Phys. Rev. D, in press.

[2] For example, M.J. Berger and J. Oliger, J. Comput. Phys. **53**, 484 (1984).

[3] For example, D.J. Mavriplis, Ann. Rev. Fluid Mech., **29**, 473 (1997).

[4] For example, D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussoletti, J. Comput. Phys. **92**, 1 (1991).

[5] A.M. Khokhlov, J. Comput. Phys. **143**, 519 (1998); A.M. Khokhlov and A.Yu. Chtchelkanova, *Fully Threaded Tree Algorithms for Massively Parallel Computations*, Proceedings of the Ninth SIAM Conference on Parallel Processing, March 22-24, 1999, San Antonio, TX USA.

[6] J.M. Bowen and J.W. York, Jr., Phys. Rev. D **21**, 2047 (1980).

[7] J.W. York, Jr., J. Math. Phys. **14**, 456 (1973).

[8] J.W. York, Jr. and T. Piran, Spacetime and geometry, edited by R. Matzner and L. Shepley (University of Texas Press, Austin, 1982), pp. 147-176.

[9] G.B. Cook, Phys. Rev. D**44**, 2983 (1991).

[10] G.B. Cook, M.W. Choptuik, M.R. Dubal, S. Klasky, R.A. Matzner, and S.R. Oliveira, Phys. Rev. D**47**, 1471 (1993).

[11] S. Brandt and B. Brugmann, Phys. Rev. Lett. **78**, 3606 (1997).

[12] C.O. Lousta and R.H. Price, Phys. Rev. D**57**, 1073 (1998).

[13] J. Baker and R.S. Puzio, Phys. Rev. D**59**:044030 (1998).

[14] W. Krivan and R.H. Price, Phys. Rev. D**58**:104003 (1998).

[15] N. Jansen, *The initial value problem of general relativity*, Master Thesis, Theoretical Astrophysics Center, Copenhagen, Denmark; `http://www.tac.dk/~jansen/thesis.ps.gz`

[16] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes in FORTRAN: The art of scientific computing, Cambridge, England: Cambridge University Press, 1992.

[17] L.A. Hageman and D.M. Young., Applied Iterative methods, New York, Academic Press, 1981.

[18] S. Hu, T. Aslam, and S. Stewart, Combust. Theory Modeling, **1** 113 (1997).

[19] C.W. Misner, Ann. Phys. (N.Y.) **24**, 102 (1963).