

# CODE DEVELOPMENT OF THREE-DIMENSIONAL GENERAL RELATIVISTIC HYDRODYNAMICS WITH AMR (ADAPTIVE-MESH REFINEMENT) AND RESULTS FROM SPECIAL AND GENERAL RELATIVISTIC HYDRODYNAMICS

ORHAN DÖNMEZ

*Department of Physics, Faculty of Art and Science, Nigde University, Nigde, Turkey*

(Received 16 July 2003; accepted 23 January 2004)

**Abstract.** In this paper, the general procedure to solve the general relativistic hydrodynamical (GRH) equations with adaptive-mesh refinement (AMR) is presented. In order to achieve, the GRH equations are written in the conservation form to exploit their hyperbolic character. The numerical solutions of GRH equations are obtained by high resolution shock Capturing schemes (HRSC), specifically designed to solve nonlinear hyperbolic systems of conservation laws. These schemes depend on the characteristic information of the system. The Marquina fluxes with MUSCL left and right states are used to solve GRH equations. First, different test problems with uniform and AMR grids on the special relativistic hydrodynamics equations are carried out to verify the second-order convergence of the code in one, two and three dimensions. Results from uniform and AMR grid are compared. It is found that adaptive grid does a better job when the number of resolution is increased. Second, the GRH equations are tested using two different test problems which are Geodesic flow and Circular motion of particle. In order to do this, the flux part of GRH equations is coupled with source part using Strang splitting. The coupling of the GRH equations is carried out in a treatment which gives second order accurate solutions in space and time.

**Keywords:** numerical relativity, hydrodynamic, shock tube, fluid mechanic, adaptive-mesh refinement, black hole, accretion disk

## 1. Introduction

Numerical relativity in astrophysics is becoming an exciting area. The general relativistic observational data require integration of general relativistic hydrodynamical (GRH) equations with traditional tools which need high computational power. There are different numerical tools to solve the relativistic equations, such as general relativistic hydrodynamics, magnetohydrodynamics and nuclear astrophysics.

Relativity is necessary for describing astrophysical phenomena involving compact objects. Some of these phenomena are X-ray binary systems, coalescing compact binaries, black-hole formations, gamma-ray burst, core collapse supernovae, pulsars, microquasars and active galactic nuclei. The general relativistic effects must be considered when strong gravitational fields are encountered, such as near compact binaries. The significant gravitational wave signal produced by some of



these phenomena can also be understood in the framework of the general relativistic theory.

Numerical relativity in astrophysics is becoming a more exciting area of research due to the large amount of data taken from high-energy astronomical phenomena. Accordingly, the numerical relativistic astrophysics may help to understand some of the observations in high-energy and gravitational-wave astrophysics. In order to take advantage from observational data, relativistic equations must be solved to understand phenomena in the strong relativistic regions. The numerical solution of strong field astrophysical phenomena requires integration of the Einstein theory with other tools of astrophysics which are relativistic hydrodynamics and nuclear astrophysics. The Einstein theory describes the dynamics of the spacetime with a set of the most complicate equations. It is highly nonlinear and thousands of terms are coupled together in these equations.

The main issues in GRH are the numerical modeling of flows with large Lorentz factors (strong gravitational regions) and strong shock waves. An accurate description of relativistic flows with strong shocks is needed for study of important problems in astrophysics. General relativistic effects caused by the presence of strong gravitational fields appear to be connected to extremely fast flows in different astrophysical scenarios, such as accreting of compact objects, stellar collapse and coalescing compact binaries (Banyuls et al., 1997). In the last 10 years, much effort has been expended to develop accurate numerical algorithms to solve the equations of GRH in extreme conditions. The main conclusion that has emerged is that modern algorithms exploiting the hyperbolic and conservative character of the equations are more accurate at describing relativistic flows than traditional finite difference upwind techniques with artificial viscosity (Norman et al., 1986). To take advantage of the conservation properties of the system, modern algorithms are written in conservative form, in the sense that the variation of mean values of the conserved quantities within the numerical cells is given, in the absence of the sources, by the fluxes across the cell boundaries. Furthermore, the hyperbolic character of the system of equations allows one to treat physical discontinuities appearing in the flow consistently (the shock-capturing property)(Hirsch, 1992). Over the last few decades, many high-resolution shock capturing (HRSC) methods have arisen in classical computational fluid dynamics (Hirsch, 1992). Nowadays, the numerical simulations employ these HRSC methods. These techniques produce highly accurate numerical approximations in smooth regions of the flow and capture the motion of unresolved steep gradients without creating spurious oscillations.

Our goal is to understand how things formed in the universe. Our strategy is the application of known physics as it is dictated by the observations of the objects whose origin and evolution we wish to comprehend, using and creating advanced computer techniques. Our tactics are to proceed based on three-dimensional numerical hydrodynamics using any source for computing methods wherever

feasible, implementing the targeted physical processes in sequence, and proceeding to increasingly geometrically complex situations.

## 2. Formulations

### 2.1. DESCRIPTION OF HYDRODYNAMIC EQUATIONS

In this section, we give a general technical description of the hydrodynamic equation as they will be used in our code development and the analytic description of different problems.

The spacetime contains flow in 4-momentum. Each particle carries its 4-momentum vector along its worldline. The particles or fluid have a 4-velocity  $u^\mu$  ( $\mu = 0, 1, 2, 3$ ) at each point which is the 4-velocity of an observer co-moving with the fluid at that point. The unit of 4-velocity is unit of speed of light,  $c$ . At each point there are three scalar functions to define the characteristic of the fluid. The first of these is the rest mass density  $\rho$  (g/cm<sup>3</sup>). The other scalar functions are the specific internal energy  $\epsilon$  (ergs/g) and the pressure  $P$  in the units of  $\rho\epsilon$  which is the energy density. The pressure is connected to the other variables by the equation of state  $P = P(\rho, \epsilon)$ . The quantity  $\rho h$ , where  $h$  is called the relativistic enthalpy, is the total inertia carrying mass energy.

The time evolution of the fluid is governed by the general hydrodynamical equations. The first equation is the normalization of the 4-velocity, defined as

$$u^\mu u_\mu = -1. \quad (1)$$

The GRH equations in Font et al. (2000) and Donat et al. (1998), written in the standard covariant form, consist of the local conservation laws of the stress-energy tensor  $T^{\mu\nu}$  and the matter current density  $J^\mu$

$$\nabla_\mu T^{\mu\nu} = 0, \quad \nabla_\mu J^\mu = 0, \quad (2)$$

where Greek indices run from 0 to 3, Latin indices from 1 to 3, and units in which the speed of light  $c = 1$  are used;  $\nabla_\mu$  stands for the covariant derivative with respect to the 4-metric of the underlying spacetime,  $g_{\mu\nu}$ .

In order to quantify the fluid or particles which are moving in their worldline, the stress-energy tensor  $T^{\mu\nu}$  is defined. The stress-energy tensor exists at each event in spacetime. The stress-energy density is a quantity which contains information about energy density, momentum density and stress as measured by any and all observers at that event. The stress-energy tensor for an imperfect fluid is defined by Misner et al. as

$$T^{\mu\nu} = \rho(1 + \epsilon)u^\mu u^\nu + (P - \zeta\theta)h^{\mu\nu} - 2\eta\sigma^{\mu\nu} + q^\mu u^\nu + q^\nu u^\mu. \quad (3)$$

where the scalar functions  $\eta$  and  $\zeta$  are the shear and bulk viscosities, respectively. This is the stress-energy tensor for a viscous fluid with heat conduction. The scalar  $\theta$  is defined as

$$\theta = u^\alpha_{;\alpha}. \quad (4)$$

It describes the divergence of the fluid worldlines. The symmetric, trace-free and spatial shear tensor  $\sigma^{\mu\nu}$  is defined by

$$\sigma^{\mu\nu} = \frac{1}{2}(u^\mu_{;\gamma} h^{\gamma\nu} + u^\nu_{;\gamma} h^{\gamma\mu}) - \frac{1}{3}\theta h^{\mu\nu}, \quad (5)$$

where  $h_{\gamma\mu} = u_\gamma u_\mu + g_{\gamma\mu}$  is the spatial projection tensor. The vector  $q^\mu$  is the energy flux vector.

Eqs. (3)–(5) describe the motion of imperfect fluid flow for a fixed metric in complete generality. Entropy for imperfect fluid is influenced by viscosity and heat flow.

Defining the characteristic waves of the GRH equations is not trivial with imperfect fluid stress-energy tensor. The viscosity and heat conduction effects are neglected. This defines the perfect fluid stress-energy tensor. This stress-energy tensor is used to derive the hydrodynamical equations. Using perfect fluid stress-energy tensor, we can solve some problems which are solved by the Newtonian hydrodynamics with viscosity, such as those involving angular momentum transport and shock waves on an accretion disk, etc. Entropy for perfect fluid is conserved along the fluid lines. The stress-energy tensor for a perfect fluid is given as

$$T^{\mu\nu} = \rho h u^\mu u^\nu + P g^{\mu\nu}. \quad (6)$$

A perfect fluid is a fluid that moves through spacetime with a 4-velocity  $u^\mu$  which may vary from event to event. It exhibits a density of mass  $\rho$  and isotropic pressure  $P$  in the rest frame of each fluid element. Here,  $h$  is the specific enthalpy, defined as

$$h = 1 + \epsilon + \frac{P}{\rho}. \quad (7)$$

Here,  $\epsilon$  is the specific internal energy. The equation of state might have the functional form  $P = P(\rho, \epsilon)$ . The perfect gas equation of state,

$$P = (\Gamma - 1)\rho\epsilon, \quad (8)$$

is such a functional form.

The conservation laws in the form given in Eq. (2) are not suitable for use in advanced numerical schemes. In order to carry out numerical hydrodynamic evolutions such as those reported in Font et al. (2000), and to use HRSC methods,

the hydrodynamic equations after the 3 + 1 split must be written as a hyperbolic system of first-order flux conservative equations. Eq. (2) is written in terms of coordinate derivatives, using the coordinates  $(x^0 = t, x^1, x^2, x^3)$ . Eq. (2) is projected onto the basis  $\{n^\mu, (\frac{\partial}{\partial x^i})^\mu\}$ , where  $n^\mu$  is a unit timelike vector normal to a given hypersurface. After a straightforward calculation, we get (see Font et al., 2000)

$$\partial_t \vec{U} + \partial_i \vec{F}^i = \vec{S}, \quad (9)$$

where  $\partial_t = \partial/\partial t$  and  $\partial_i = \partial/\partial x^i$ . This basic step serves to identify the set of unknowns, the vector of conserved quantities  $\vec{U}$  and their corresponding fluxes  $\vec{F}(\vec{U})$ . With the equations in conservation form, almost every high-resolution method devised to solve hyperbolic systems of conservation laws can be extended to GRH.

The evolved state vector  $\vec{U}$  consists of the conservative variables  $(D, S_j, \tau)$  which are conserved variables for density, momentum and energy, respectively; in terms of the primitive variables  $(\rho, v^i, \epsilon)$ , this becomes (Font et al., 2000)

$$\vec{U} = \begin{pmatrix} D \\ S_j \\ \tau \end{pmatrix} = \begin{pmatrix} \sqrt{\gamma} W \rho \\ \sqrt{\gamma} \rho h W^2 v_j \\ \sqrt{\gamma} (\rho h W^2 - P - W \rho) \end{pmatrix}. \quad (10)$$

Here,  $\gamma$  is the determinant of the 3-metric  $\gamma_{ij}$ ,  $v_j$  is the fluid 3-velocity and  $W$  is the Lorentz factor,

$$W = \alpha u^0 = (1 - \gamma_{ij} v^i v^j)^{-1/2}. \quad (11)$$

The flux vectors  $\vec{F}^i$  are given by (Font et al., 2000)

$$\vec{F}^i = \begin{pmatrix} \alpha(v^i - \frac{1}{\alpha}\beta^i)D \\ \alpha\{(v^i - \frac{1}{\alpha}\beta^i)S_j + \sqrt{\gamma}P\delta_j^i\} \\ \alpha\{(v^i - \frac{1}{\alpha}\beta^i)\tau + \sqrt{\gamma}v^i P\} \end{pmatrix}. \quad (12)$$

The spatial components of the 4-velocity  $u^i$  are related to the 3-velocity by the following formula:  $u^i = W(v^i - \beta^i/\alpha)$ .  $\alpha$  and  $\beta^i$  are the lapse function and the shift vector of the spacetime, respectively. The source vector  $\vec{S}$  is given by (Font et al., 2000)

$$\vec{S} = \begin{pmatrix} 0 \\ \alpha\sqrt{\gamma}T^{\mu\nu}g_{\nu\sigma}\Gamma_{\mu j}^\sigma \\ \alpha\sqrt{\gamma}(T^{\mu 0}\partial_\mu\alpha - \alpha T^{\mu\nu}\Gamma_{\mu\nu}^0) \end{pmatrix}, \quad (13)$$

where  $\Gamma_{\mu\nu}^\alpha$  is the 4D Christoffel symbol

$$\Gamma_{\mu\nu}^\alpha = \frac{1}{2}g^{\alpha\beta}(\partial_\mu g_{\nu\beta} + \partial_\nu g_{\mu\beta} - \partial_\beta g_{\mu\nu}). \quad (14)$$

## 2.2. SPECTRAL DECOMPOSITION AND CHARACTERISTIC FIELDS

The use of HRSC schemes requires the spectral decomposition of the Jacobian matrix of the system,  $\partial \vec{F}^i / \partial \vec{U}$ . The spectral decomposition of the Jacobian matrices of the GRH equations with a general equation of state was reported in Font et al. (2000). They display the spectral decomposition, valid for a generic spatial metric, in the  $x$ -direction,  $(\partial \vec{F}^x / \partial \vec{U})$ ; permutation of the indices yields the other two directions.

They started the solution by considering an equation of state in which the pressure  $P$  is a function of  $\rho$  and  $\epsilon$ ,  $P = P(\rho, \epsilon)$ . The relativistic speed of sound in the fluid  $C_s$  is given by (Font et al., 2000)

$$C_s^2 = \left. \frac{\partial P}{\partial E} \right|_s = \frac{\chi}{h} + \frac{P\kappa}{\rho^2 h}, \quad (15)$$

where  $\chi = \partial P / \partial \rho|_{\epsilon}$ ,  $\kappa = \partial P / \partial \epsilon|_{\rho}$ ,  $S$  is the entropy per particle and  $E = \rho + \rho\epsilon$  is the total rest energy density.

A complete set of the right eigenvectors and  $[\vec{r}_i]$  and corresponding eigenvalues  $\lambda_i$  along the  $x$ -direction obeys

$$\left[ \frac{\partial \vec{F}^x}{\partial \vec{U}} \right] [\vec{r}_i] = \lambda_i [\vec{r}_i], \quad i = 1, \dots, 5. \quad (16)$$

The solution contains triply degenerate eigenvalues (Font et al., 2000),

$$\lambda_1^x = \lambda_2^x = \lambda_3^x = \alpha v^x - \beta^x. \quad (17)$$

The other eigenvalues are given by (Font et al., 2000)

$$\lambda_{\pm}^x = \frac{\alpha}{1 - v^2 C_s^2} \left\{ v^x (1 - C_s^2) \pm \sqrt{C_s^2 (1 - v^2) [\gamma^{xx} (1 - v^2 C_s^2) - v^x v^x (1 - C_s^2)]} \right\} - \beta^x. \quad (18)$$

A set of linearly independent right eigenvectors spanning this space is given by (Font et al., 2000)

$$\vec{r}_1^x = \left[ \frac{\kappa}{hW(\kappa - \rho C_s^2)}, v_x, v_y, v_z, 1 - \frac{\kappa}{hW(\kappa - \rho C_s^2)} \right]^T, \quad (19)$$

$$\vec{r}_2^x = [Wv_y, h(\gamma_{xy} + 2W^2 v_x v_y), h(\gamma_{yy} + 2W^2 v_y v_y), h(\gamma_{yz} + 2W^2 v_y v_z), v_y W(2Wh - 1)]^T, \quad (20)$$

$$\vec{r}_3^x = [Wv_z, h(\gamma_{xz} + 2W^2v_xv_z), h(\gamma_{yz} + 2W^2v_yv_z), h(\gamma_{zz} + 2W^2v_zv_z), v_zW(2Wh - 1)]^T \quad (21)$$

$$\vec{r}_\pm^x = \left[ 1, hW \left( v_x - \frac{v^x - (\lambda_\pm + \beta^x)/\alpha}{\gamma^{xx} - v^x(\lambda_\pm + \beta^x)/\alpha} \right), hWv_y, hWv_z, \frac{hW(\gamma^{xx} - v^xv^x)}{\gamma^{xx} - v^x(\lambda_\pm + \beta^x)/\alpha} - 1 \right]^T. \quad (22)$$

Here, the superscript T denotes the transpose.

Now, the eigenvalues and eigenvectors are computed in the  $y$ - and  $z$ -directions using the information in the  $x$ -direction. Let  $M^1(U)$  be the matrix of the right eigenvectors, that is, the matrix having as columns the right eigenvectors with the standard ordering (Banyuls et al., 1997),  $M^1 = (\vec{r}_-, \vec{r}_1, \vec{r}_2, \vec{r}_3, \vec{r}_+)$ . To obtain the spectral decomposition in the other spatial directions  $q = 2, 3$  ( $\equiv y, z$ ), it is enough to take into account the following symmetry relations:

- (1) The eigenvalues are easily obtained by carrying out a simple substitution of indices  $x$  by  $q$  in Eqs. (17) and (18).
- (2) The matrix  $M^1$  in the  $x$ -direction is permuted according to  $M^q = \mathcal{P}_{q1}(M^1)$ , where the two operators  $\mathcal{P}_{q1}$  act on their arguments by the following sequential operations:
  - to permute the second and  $(q + 1)$ th rows,
  - to interchange indices  $1 \equiv x$  with  $q$ .

From Ibanez et al. (1998), the left eigenvectors in the  $x$ -direction are

$$\mathbf{l}_{0,1} = \frac{W}{\mathcal{K} - 1} \begin{bmatrix} h - W \\ Wv^x \\ Wv^y \\ Wv^z \\ -W \end{bmatrix}, \quad (23)$$

$$\mathbf{l}_{0,2} = \frac{1}{h\xi^x} \begin{bmatrix} -\gamma_{zz}v_y + \gamma_{yz}v_z \\ v^x(\gamma_{zz}v_y - \gamma_{yz}v_z) \\ \gamma_{zz}(1 - v_xv^x) + \gamma_{xz}v_zv^x \\ -\gamma_{yz}(1 - v_xv^x) - \gamma_{xz}v_yv^x \\ -\gamma_{zz}v_y + \gamma_{yz}v_z \end{bmatrix}, \quad (24)$$

$$\mathbf{l}_{0,3} = \frac{1}{h\xi^x} \begin{bmatrix} -\gamma_{yy}v_z + \gamma_{zy}v_y \\ v^x(\gamma_{yy}v_z - \gamma_{zy}v_y) \\ -\gamma_{zy}(1 - v_xv^x) - \gamma_{xy}v_zv^x \\ \gamma_{yy}(1 - v_xv^x) + \gamma_{xy}v_yv^x \\ -\gamma_{yy}v_z + \gamma_{zy}v_y \end{bmatrix}, \quad (25)$$

$$\mathbf{l}_{\mp} = (\mp 1) \frac{h^2}{\Delta^x} \begin{bmatrix} hW\mathcal{V}_{\pm}^x \xi^x + (\pm \frac{\Delta}{h^2}) l_{\mp}^{(5)} \\ \Gamma_{xx}(1 - \mathcal{K}\tilde{\mathcal{A}}_{\pm}^x) + (2\mathcal{K} - 1)\mathcal{V}_{\pm}^x(W^2v^x\xi^x - \Gamma_{xx}v^x) \\ \Gamma_{xy}(1 - \mathcal{K}\tilde{\mathcal{A}}_{\pm}^x) + (2\mathcal{K} - 1)\mathcal{V}_{\pm}^x(W^2v^y\xi^x - \Gamma_{xy}v^x) \\ \Gamma_{xz}(1 - \mathcal{K}\tilde{\mathcal{A}}_{\pm}^x) + (2\mathcal{K} - 1)\mathcal{V}_{\pm}^x(W^2v^z\xi^x - \Gamma_{xz}v^x) \\ (1 - \mathcal{K})[-\gamma v^x + \mathcal{V}_{\pm}^x(W^2\xi^x - \Gamma_{xx})] - \mathcal{K}W^2\mathcal{V}_{\pm}^x\xi^x \end{bmatrix}. \quad (26)$$

The variables used for the left eigenvectors in the  $x$ -direction are defined as follows:

$$\begin{aligned} \Lambda_{\pm}^x &\equiv \tilde{\lambda}_{\pm} + \tilde{\beta}^x, \quad \tilde{\lambda} \equiv \lambda/\alpha, \quad \tilde{\beta}^x \equiv \beta^x/\alpha, \\ \mathcal{K} &\equiv \frac{\tilde{\kappa}}{\tilde{\kappa} - C_s^2}, \quad \tilde{\kappa} \equiv \kappa/\rho, \\ \mathcal{C}_{\pm}^x &\equiv v_x - \mathcal{V}_{\pm}^x, \quad \mathcal{V}_{\pm}^x \equiv \frac{v^x - \Lambda_{\pm}^x}{\gamma^{xx} - v^x \Lambda_{\pm}^x}, \\ \tilde{\mathcal{A}}_{\pm}^x &\equiv \frac{\gamma^{xx} - v^x v^x}{\gamma^{xx} - v^x \Lambda_{\pm}^x}, \\ 1 - \tilde{\mathcal{A}}_{\pm}^x &= v^x \mathcal{V}_{\pm}^x, \quad \tilde{\mathcal{A}}_{\pm}^x - \tilde{\mathcal{A}}_{\mp}^x = v^x(\mathcal{C}_{\pm}^x - \mathcal{C}_{\mp}^x), \\ (\mathcal{C}_{\pm}^x - \mathcal{C}_{\mp}^x) + (\tilde{\mathcal{A}}_{\mp}^x \mathcal{V}_{\pm}^x - \tilde{\mathcal{A}}_{\pm}^x \mathcal{V}_{\mp}^x) &= 0, \\ \Delta^x &\equiv h^3 W(\mathcal{K} - 1)(\mathcal{C}_{+}^x - \mathcal{C}_{-}^x)\xi^x, \\ \xi^x &\equiv \Gamma_{xx} - \gamma v^x v^x, \\ \gamma &\equiv \det \gamma_{ij} = \gamma_{xx}\Gamma_{xx} + \gamma_{xy}\Gamma_{xy} + \gamma_{xz}\Gamma_{xz}, \\ \Gamma_{xx} &= \gamma_{yy}\gamma_{zz} - \gamma_{yz}\gamma_{zy}, \\ \Gamma_{xy} &= \gamma_{yz}\gamma_{zx} - \gamma_{yx}\gamma_{zz}, \\ \Gamma_{xz} &= \gamma_{yx}\gamma_{zy} - \gamma_{yy}\gamma_{zx}, \\ \Gamma_{xx}v_x + \Gamma_{xy}v_y + \Gamma_{xz}v_z &= \gamma v^x. \end{aligned}$$

Now, the left eigenvectors of the Jacobian matrix are computed in the  $y$  and  $z$ -directions using the left eigenvectors in the  $x$ -direction. Let  $\mathbf{L}^1(U)$  be the matrix of left eigenvectors, that is, the matrix having as rows the left eigenvectors with the standard ordering. The eigenvalues of the left eigenvectors are the same with the eigenvalues of the right eigenvectors. To obtain the spectral decomposition in the other spatial directions  $q = 2, 3$  ( $\equiv y, z$ ), it is enough to take into account the following symmetry relations

The matrix  $M^1$  in the  $x$ -direction is permuted according to  $\mathbf{L}^q = \mathcal{P}_{q1}(\mathbf{L}^1)$ , where the two operators  $\mathcal{P}_{q1}$  act on their arguments by the following sequential operations:

- to permute the second and  $(q + 1)$ th columns,
- to interchange indices  $1 \equiv x$  with  $q$ .



### 3. Numerical Method

#### 3.1. MARQUINA FLUXES

Approximate Riemann solver failures and their respective corrections (usually adding a artificial dissipation) have been studied in the literature (Quirk, 1994). Motivated by the search for a robust and accurate approximate Riemann solver that avoids these common failures, Shu and Osher (1989) have proposed a numerical flux formula for scalar equations. Marquina flux is generalization of flux formula in Shu and Osher (1989). In the scalar case and for characteristic wave speeds which do not change sign at the given numerical interface, Marquina's flux formula is identical to Roe's flux (Toro, 1999). Otherwise, scheme is more viscous, entropy satisfying local Lax-Friedrichs scheme (Shu et al., 1989). The combination of Roe and Lax-Friedrichs schemes is carried out in each characteristic field after the local linearization and de-coupling of the system of equations. However, contrary to other schemes, the Marquina's method is not based on any averaged intermediate state.

The Marquina fluxes (Font et al., 2000) with MUSCL left and right states are used to solve the 3D relativistic hydroequation. In Marquina's scheme there are no Riemann solutions involved (exact or approximate) and there are no artificial intermediate states constructed at each cell interface.

To compute the Marquina fluxes, first, the sided local characteristic variables and fluxes are computed. For the left and right sides, the characteristic variables are

$$w_l^p = L^p(U_l) \cdot U_l, \quad w_r^p = L^p(U_r) \cdot U_r \quad (27)$$

and the characteristic fluxes are

$$\Phi_l^p = L^p(U_l) \cdot F(U_l), \quad \Phi_r^p = L^p(U_r) \cdot F(U_r), \quad (28)$$

where the number of conservative variables  $p = 1, \dots, 5$ ;  $U_l$  and  $U_r$  are conservative variables at the left and right sides, respectively;  $L^p(U_l)$  and  $L^p(U_r)$  are the left eigenvectors of the Jacobian matrice,  $\partial F^i / \partial U$ .

The left and right fluxes are defined depending on the velocities of the fluid for each specific grid zone. The prescription given in Donat et al. (1998) is as follows:

For all conserved variables  $p = 1, \dots, m$   
 if  $\lambda_p(U)$  does not change sign in (if  $(\lambda_p(U_L) \times \lambda_p(U_R) \geq 0)$ ), then  
 if  $\lambda_p(U_l) > 0$  then

$$\begin{aligned} \Phi_+^p &= \Phi_l^p \\ \Phi_-^p &= 0 \end{aligned}$$

else

$$\Phi_+^p = 0$$

$$\Phi_-^p = \Phi_r^p$$

end if

else

$$\alpha_p = \max_{U \in \Gamma(U_l, U_r)} |\lambda_p(U)|$$

$$\Phi_+^p = 0.5(\Phi_l^p + \alpha_k w_l^p)$$

$$\Phi_-^p = 0.5(\Phi_r^p + \alpha_p w_r^p)$$

end if

where  $\lambda_p$  is an eigenvalue of the Jacobian matrix and,

$$\alpha_k = \max\{|\lambda_p(U_l)|, |\lambda_p(U_r)|\}. \quad (29)$$

The numerical flux that corresponds to the cell interface separating the states  $U_l$  and  $U_r$  is then given by Donat et al. (1998):

$$F^M(U_l, U_r) = \sum_{p=1}^m (\Phi_+^p r^p(U_l) + \Phi_-^p r^p(U_r)). \quad (30)$$

Marquina's scheme can be interpreted as a characteristic-based scheme that avoids the use of an averaged intermediate state to perform the transformation to the local characteristic fields.

In carrying out Marquina's scheme, we have to compute intermediate states and the Jacobian matrix of the states at each cell interface. So we need to know the left and right states,  $U_L$  and  $U_R$ , at each interface. To construct the second-order scheme, the MUSCL left and right states are used,

$$\begin{aligned} u_i^L &= u_i(0) = u_i - \frac{1}{2} \Delta_i, \\ u_i^R &= u_i(\Delta x) = u_i + \frac{1}{2} \Delta_i. \end{aligned} \quad (31)$$

To avoid the appearance of oscillations around discontinuities in MUSCL-type schemes, slope limiters in the reconstruction stage (Hirsch, 1992; Toro, 1999) are used. A common TVD limiter is based on the minmod function (Hirsch et al., 1992). The standard minmod slope provides the desired second-order accuracy for smooth solutions, while still satisfying the TVD property. Minmod slope function is given as

$$\Delta_i = \text{minmod}(U_i - U_{i-1}, U_{i+1} - U_i), \quad (32)$$

where the minmod function of two arguments is defined by

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0. \end{cases} \quad (33)$$

### 3.2. STRANG SPLITTING

HRSC methods are applicable without source term. We now turn our attention to handling the source and flux terms. To solve the GRH equations numerically Strang splitting is used, in which Eq. (9) is split into two parts. The first is given by the flux terms (Bona et al., 1998)

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}^x}{\partial x} + \frac{\partial \vec{F}^y}{\partial y} + \frac{\partial \vec{F}^z}{\partial z} = 0. \quad (34)$$

The evolution operator of Eq. (34) can be described by an  $F$ , so the solution of Eq. (34) becomes

$$U(t + \Delta t) = FU(t). \quad (35)$$

The second has the source terms

$$\frac{\partial \vec{U}}{\partial t} = \vec{S}. \quad (36)$$

With the evolution operator for Eq. (36),  $S$ , the solution of Eq. (36) is

$$U(t + \Delta t) = SU(t). \quad (37)$$

In the numerical solution, this splitting is performed by a combination of both flux and source operators. Denoting by  $U(\Delta t)$  the numerical evolution operator for the system in a single time step, the evolution operator for solution (Toro, 1999) is written as

$$U(t + \Delta t) = S\left(\frac{\Delta t}{2}\right)F(\Delta t)S\left(\frac{\Delta t}{2}\right)U(t). \quad (38)$$

The resulting update of  $U(t + \Delta t)$  is second-order accurate in space and time if and only if both the operators  $S$  and  $F$  are  $O(\Delta x^2, \Delta t^2)$ .

This kind of splitting allows easy implementation of the different numerical treatments of the principal part of the system without worrying about the sources of the equations. It also allows choosing the best scheme for each type of problem.

### 3.3. UPDATING THE EQUATIONS IN TIME

The HRSC methods will be used to solve the hydrodynamical equations. This requires that the hydrodynamical equations are written in a flux form.

Solution of the relativistic hydrodynamical equations using MUSCL left and right states with Marquina fluxes gives second-order accurate solutions in space. In order to get solutions that are also second-order accurate in time, the solution is evolved in the following way.

To solve Eq. (38) first, the solution of Eq. (37) should be computed, which is called a system of ordinary differential equations (ODEs). Using the two stage *Runge–Kutta method* (Toro, 1999), which gives a second-order accurate solution in time, we get

$$\begin{aligned} K_1 &= \Delta t S(t^n, U^n), \\ K_2 &= \Delta t S(t^n + \Delta t, U^n + K_1), \\ U^{n+1} &= U^n + \frac{1}{2}[K_1 + K_2]. \end{aligned} \quad (39)$$

Explicit methods are much simpler to use than implicit methods. The latter require the solution of nonlinear algebraic equations at each time step and are therefore much more expensive. If ODEs which have no spatial variations,  $\bar{F}(U) = 0$ , are solved by the implicit method, then there will be no stability restriction on the time step. Since an explicit method is used to solve the ODEs here, the time step is limited by the Courant condition (Hirsch, 1992),

$$\Delta t = C \frac{\Delta x}{(|v^x| + C_s)_{\max}}, \quad (40)$$

where  $C$  is called Courant number. It can be  $0 < C \leq 1$ . Eq. (40) means that fastest wave at a given time travels for at most one cell length  $\Delta x$  in the sought time step  $\Delta t$ .

Second, the solution of Eq. (34) is computed using a predictor–corrector time update method, as follows.

Initially, the solution at the half time step is computed,

$$\begin{aligned} U_{i,j,k}^{n+\frac{1}{2}} &= U_{i,j,k}^n - \frac{\Delta t}{2\Delta x} \left( (f^n)_{i+\frac{1}{2},j,k} - (f^n)_{i-\frac{1}{2},j,k} \right) \\ &\quad - \frac{\Delta t}{2\Delta y} \left( (f^n)_{i,j+\frac{1}{2},k} - (f^n)_{i,j-\frac{1}{2},k} \right) \\ &\quad - \frac{\Delta t}{2\Delta z} \left( (f^n)_{i,j,k+\frac{1}{2}} - (f^n)_{i,j,k-\frac{1}{2}} \right). \end{aligned} \quad (41)$$

Then, the solution using the full time step is computed and new fluxes ( $f^{n+\frac{1}{2}}$ ) which are functions of  $U_{i,j,k}^{n+\frac{1}{2}}$ ,

$$\begin{aligned}
 U_{i,j,k}^{n+1} = U_{i,j,k}^n & - \frac{\Delta t}{\Delta x} \left( \left( f^{n+\frac{1}{2}} \right)_{i+\frac{1}{2},j,k} - \left( f^{n+\frac{1}{2}} \right)_{i-\frac{1}{2},j,k} \right) \\
 & - \frac{\Delta t}{\Delta y} \left( \left( f^{n+\frac{1}{2}} \right)_{i,j+\frac{1}{2},k} - \left( f^{n+\frac{1}{2}} \right)_{i,j-\frac{1}{2},k} \right) \\
 & - \frac{\Delta t}{\Delta z} \left( \left( f^{n+\frac{1}{2}} \right)_{i,j,k+\frac{1}{2}} - \left( f^{n+\frac{1}{2}} \right)_{i,j,k-\frac{1}{2}} \right). \quad (42)
 \end{aligned}$$

In order to solve general relativistic hydrodynamical equation in 3 D, the following procedure is followed. First Eq. (36) is solved using second-order Runge–Kutta method, Eq. (39), at half time step. Later Eq. (34) is solved using predictor–corrector method, Eqs. (41) and (42), at the full time step. Finally, the source term is solved another half time step using the Runge–Kutta method. At the end of the calculation, a second-order accurate solution for GRH equation is obtained in space and time.

#### 4. Adaptive-Mesh Refinement

Adaptive mesh refinement (AMR) is needed to solve some problems in numerical relativity, such as coalescing compact binaries, accreting compact objects, etc. A key difficulty in calculating accurate waveforms from these simulations lies in the fact that the length and time scales for these problems can vary by an order of magnitude or more within the computational domain. Adequate resolution is needed to model strong field regions. Adaptive techniques are thus crucial for success in modeling such systems.

##### 4.1. PARAMESH PACKAGE

The *Paramesh* package (MacNeice et al., 2000) is used to make to implement AMR and parallelization in the code. The package runs on any parallel platform which supports either the shmem (shared memory) or MPI libraries. This package is a set of FORTRAN 90 routines designed to enable AMR for a parallel distributed memory machine.

The basic AMR strategy is that the computational domain is covered with a hierarchy of numerical sub-grids. These sub-grids form the nodes of a tree data-structure and are distributed among the processors. When more spatial resolution is required at some location, the highest resolution sub-grid covering that point spawns child sub-grids (two in 1D, four in 2D and eight in 3D), which together cover the line, area or volume of their parent, but now with twice its spatial resolution. They refer to the block with the highest refinement level at a given

physical location as a ‘leaf block’. All sub-grids have identical logical structure (i.e., the same number of grid points in each dimension, the same aspect ratios, the same number of guard cells, etc). They differ from each other in their physical sizes and locations, and in their list of neighbors, parents and children. The package assumes that the application will use logically Cartesian grids (i.e., grids that can be indexed  $i, j, k$ ). It works for 1D, 2D and 3D problems.

The package provides routines which perform all the communication required between sub-grids and between processors. It manages the refinement and de-refinement processes, and can balance the workload by reordering the distribution of blocks among the processors. It also provides routines to enforce conservation laws at the interfaces between grid blocks at different refinement levels. Flux conservation at different refinement level interfaces will be explained in Section 4.2.

The top panel of Fig. 1 shows a  $6 \times 4$  grid structure in 2D on each block. The numbers assigned to each block represent the block location in the three structure seen in the bottom panel of Fig. 1. Notice that in Paramesh the refinement level is not allowed to jump by more than one level at any location of the spatial domain.

#### 4.2. FLUX CONSERVATION

Conservation of conserved variables in the hydrodynamical code is an important physical result. To guarantee conservation of variables on the AMR grid, some special operation at block boundaries of different refinement levels are needed. Conservative hydrodynamics codes require that the fluxes entering or leaving a grid cell through a common cell face shared with the four refined neighbors, equal the sum of the fluxes across the appropriate faces of the four smaller cells; Fig. 2 (MacNeice et al., 2000). This is achieved by taking the average of the fine grid fluxes and copying them into the coarse grid to correct fluxes at different refinement levels. For more information about this flux conservation, see MacNeice et al. (2000).

#### 4.3. REFINEMENT CRITERIA

A refinement criterion needs to be defined to analyze any problem using Paramesh package. We have chosen to use a refinement criterion based on the first derivative of density, which has different dynamics during the evolution for the shock tube problem. When density is refined or de-refined, the other variables, such as pressure and velocity, will be refined or de-refined simultaneously. Two different functions are used to define refinement criteria, which are given as follows:

$$\tau_1(x, y) = \left| \frac{h}{f} \left( \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \right) \right| \quad (43)$$

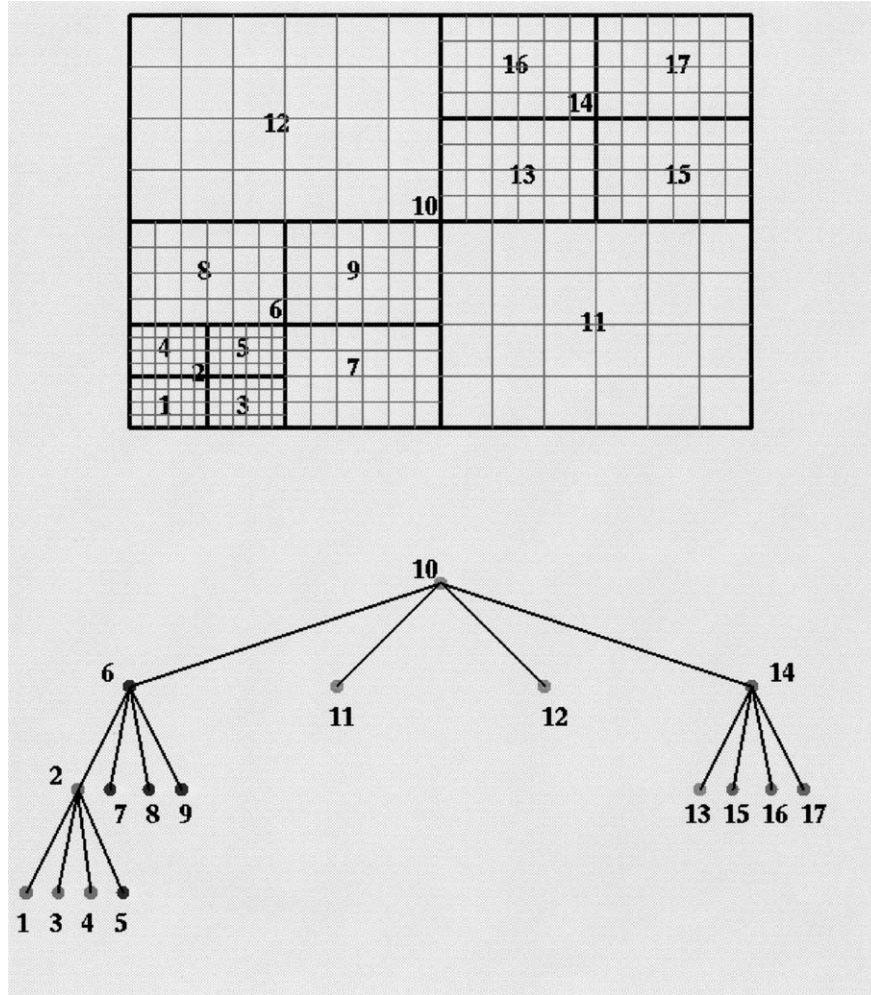


Figure 1. Representation of adaptive-mesh refinement grid and tree structure. (Figure is taken from MacNeice et al. (2000).)

and

$$\tau_2(x, y) = \left| \frac{1}{f} (f(i, j) - f(i + 1, j + 1)) \right|. \quad (44)$$

where Eqs. (43) and (44) are the first derivative and diagonal derivative of a function  $f$ , respectively. The function  $f$  can be chosen any variable in the simulation depending on the change of dynamics during the numerical simulation. Refinement and de-refinement of blocks depend on refinement criteria and parameters. If the

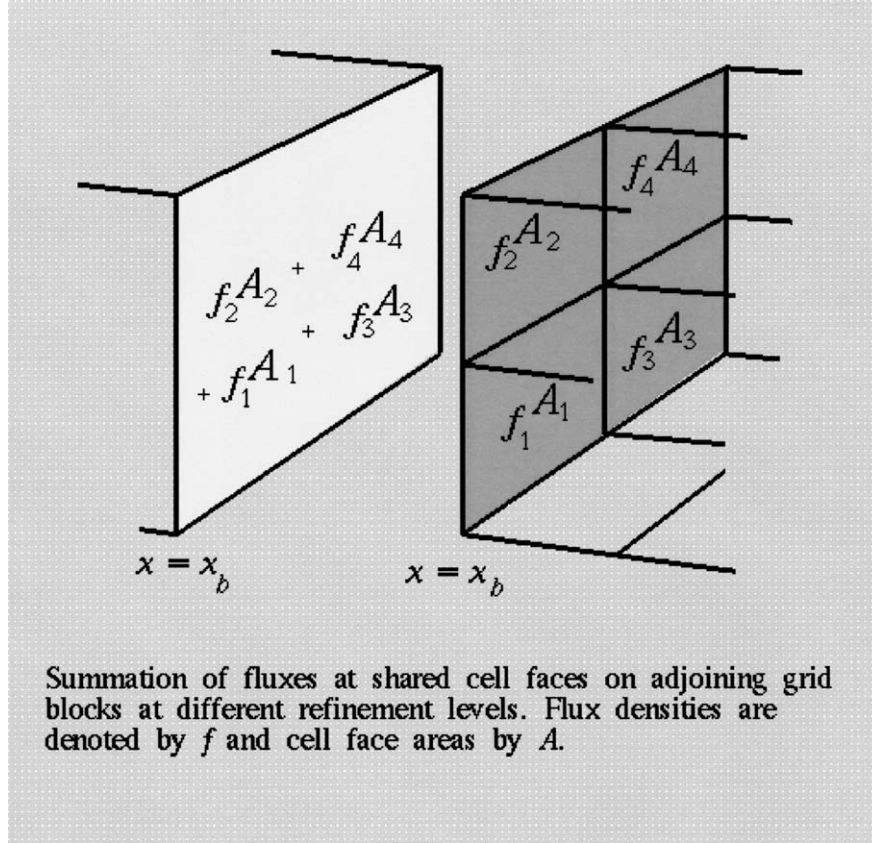


Figure 2. Flux conservation is fixed at different refinement level boundaries. (Figure is taken from MacNeice et al. (2000).)

value of the refinement functions, Eq. (43) or (44), are bigger than refinement parameter, core specified by user based on problem, in any zone in the same block, that block will be refined. If the value of the refinement functions are smaller than de-refinement parameter,  $ctode$ , that block will be de-refined. Refinement parameter has to be bigger than de-refinement parameter. These refinement and de-refinement criteria are checked on each time step during the numerical simulation. The refinement criteria and parameters play an important role in AMR. Note that both refinement and de-refinement in Paramesh are restricted by the requirement that the grid can change by only one refinement level at each block.

## 5. Results from Special Relativistic Shock Tube Problem

The Riemann shock tube problem (Hirsch, 1992; Font et al., 2000) is used as a test case. In this problem, the fluid is initially in two different thermodynamical states



on either side of a membrane. The membrane is then removed. In the test problem, it is assumed that the fluid initially has  $\rho_L > \rho_R$ , where the subscripts  $L$  and  $R$  refer to the left and right sides of the membrane. With this assumption, a rarefaction wave travels to the left, and a shock wave and contact discontinuity travel to the right. The fluid in the intermediate state moves at a mildly relativistic speed which is  $v = 0.725c$  to the right. Flow particles accumulate in a dense shell behind the shock wave compressing the fluid and heating it up to values of the internal energy much larger than the rest-mass energy. Hence, the fluid is extremely relativistic from a thermodynamical point of view, but only mildly relativistic dynamically. The Riemann shock tube is a useful test problem, because it has an exact time-dependent solution, and tests the ability of the code to evolve both smooth and discontinuous flows. In the case considered here, the velocities are special relativistic and the method of finding the exact solution differs somewhat from the standard nonrelativistic shock tube.

The initial values of the fluid variables for 1D test are:  $\rho_l = 10.0$ ,  $\rho_r = 1.0$ ,  $u_l = 0.0$ ,  $u_r = 0.0$ ,  $p_l = 13.3$ ,  $p_r = 0.66 \times 10^{-6}$ . The computational domain is  $0 \leq x \leq 1$  and at  $t = 0$  the membrane is placed at  $x = 0.5$ . We use the perfect fluid equation of state Eq. (8) with  $\Gamma = 5/3$ .

### 5.1. UNIFORM SOLUTION OF 3D SHOCK TUBE

The 3D special relativistic shock tube problem is solved using the initial data which are given above. The MUSCL left and right states and the minmod slope function with the Marquina fluxes are used. This method gives a second-order accurate solution for the 3D special relativistic hydrodynamical equations. The  $L_1$ -norm errors and convergence rates are computed and our results are also compared with the existing literature. In order to compare the 3D numerical with 1D analytic solution for the shock tube, the initial discontinuity is set up along the diagonal of the computational domain. The  $128^3$  zones are used with interval of length  $1/\sqrt{3}$  in each directions. The diagonal of the cube therefore has unit length. The maximum time,  $t = 0.4$ , is used to compute analytic and numerical calculations and the numerical data are extracted along the main diagonal of the cube.

Table I shows the  $L_1$ -norm errors of different hydrodynamical quantities. These are the density, pressure and velocity. We present our results, those from Font et al. (2000) and GENESIS code (Aloy et al., 2000) in Table I; all of these are 3D simulations. Our code and the code of Font et al. (2000) use the Marquina method with MUSCL left and right states which give a second-order accurate solution. The Genesis code uses Marquina method with the PPM reconstruction procedure which gives third-order accurate solutions. Table I shows that the results are comparable. Another difference between the Genesis and our code is that we compute  $L_1$ -norm errors at  $t = 0.4$ , but they compute at  $t = 0.5$ . Marquina method with PPM left and right states is computationally more expensive than our code. Due to more reconstruction process, Marquina method with PPM left and right states needs two

TABLE I  
Comparing our code  $L_1$  norm errors with the other hydrodynamic code results from 3D

Code	$npts$	$L_1(\rho)$	$L_1(p)$	$L_1(v)$
OUR CODE	$32^3$	0.188	0.1798	$1.82 \times 10^{-2}$
	$64^3$	$9.813 \times 10^{-2}$	$9.199 \times 10^{-2}$	$6.178 \times 10^{-3}$
	$128^3$	$5.439 \times 10^{-2}$	$4.595 \times 10^{-2}$	$3.86 \times 10^{-3}$
Font, Miller Suen, Tobias	$128^3$	$9.23 \times 10^{-2}$	$7.98 \times 10^{-2}$	$9.66 \times 10^{-3}$
GENESIS (Aloy et al., 2000)	$40^3$	$1.1 \times 10^{-1}$	$8.0 \times 10^{-2}$	$0.9 \times 10^{-2}$
	$60^3$	$9.8 \times 10^{-2}$	$5.2 \times 10^{-2}$	$1.1 \times 10^{-2}$
	$80^3$	$9.2 \times 10^{-2}$	$4.5 \times 10^{-2}$	$1.1 \times 10^{-2}$
	$100^3$	$7.0 \times 10^{-2}$	$3.7 \times 10^{-2}$	$7.0 \times 10^{-3}$
	$150^3$	$4.8 \times 10^{-2}$	$2.5 \times 10^{-2}$	$5.0 \times 10^{-3}$

more evolution steps to get the third-order accuracy in time. The PPM interpolation algorithm described in Colella and Woodward (1984) gives monotonic conservative parabolic profiles of variables within a numerical zones. In the relativistic version of PPM, the original interpolation algorithm is applied to zone-averaged values of the primitive variables ( $\rho$ ,  $v$ ,  $p$ ), which are obtained from zone-averaged values of the conserved variables  $U$ .

The convergence factors are shown in Table II. It should approach 1 when the higher order methods are used. Fig. 3 shows the result obtained for different numbers of points with this method. The solid line represents the exact solution while different symbol represents the numerical results for different resolution for the density. As can be seen from Fig. 3 that when the number of resolutions are increased, numerical solution converge to analytic one.

## 5.2. AMR RESULTS

We have first carried out simulations of the 2D shock tube using three levels of refinements. To solve this problem, initial discontinuity has been placed along the diagonal axis of the computational domain (a square). The evolution of the system along the direction orthogonal to this diagonal is purely an one-dimensional problem. It allows us to compare one-dimensional analytic solution with the numerical

TABLE II  
Convergence rate from 3D special relativistic hydrocode

$npts-2 * npts$	$r(\rho)$	$r(p)$	$r(v)$
$32^3-64^3$	0.94	0.9668	1.563
$64^3-128^3$	0.851	1.001	0.68

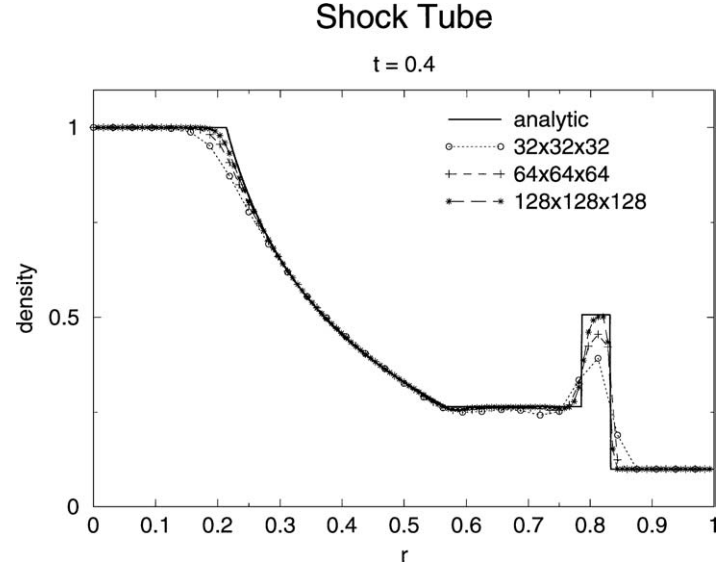


Figure 3. The analytic and numerical solutions of the relativistic shock tube problem for 3D.

results. The results of shock tube problem with AMR are compared with analytic solution at each grid zone of the computational domain. The error in each grid zone and the refinement levels are plotted in Fig. 4. The finest level is replaced around  $r = 0.8$  where shock wave is produced. Eq. (43) is used as a refinement function.

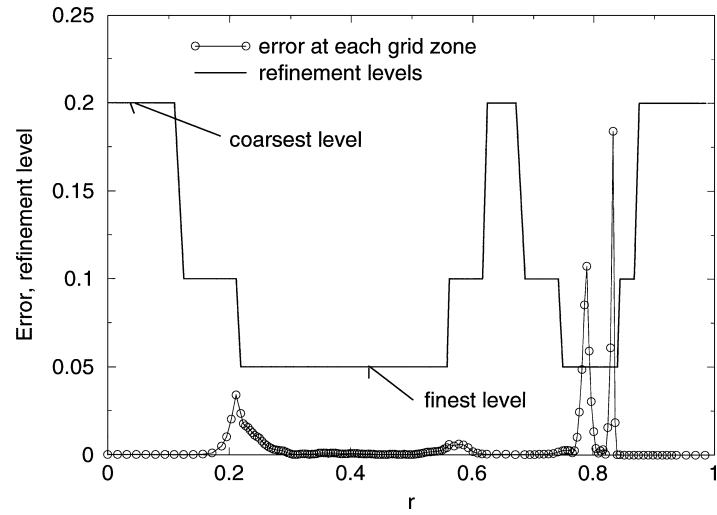


Figure 4. Plotting the error and the refinement levels in each zone. Eq. (43) is used as the refinement function with parameters:  $c_{\text{tore}}$  (refinement parameter) = 0.035,  $c_{\text{tode}}$  (de-refinement parameter) = 0.005.

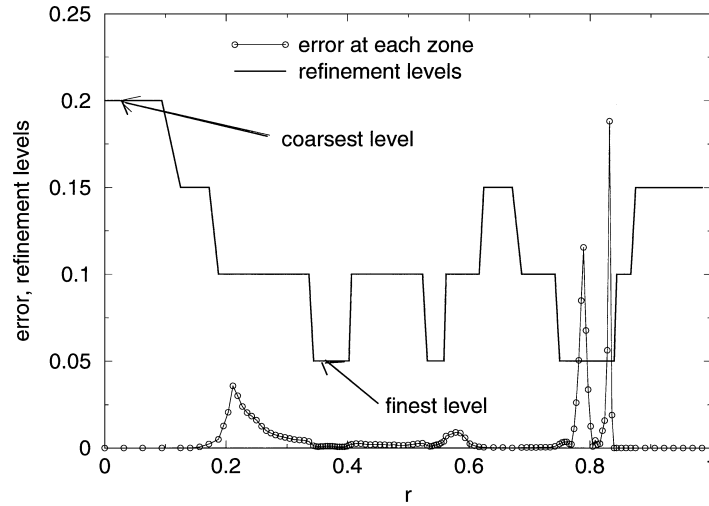


Figure 5. Plotting the error and refinement levels in each zone. Eq. (44) is used as the refinement function with parameters:  $c_{\text{tore}}$  (refinement parameter) = 0.035,  $c_{\text{tode}}$  (de-refinement parameter) = 0.005.

Refinement parameters for this problem are:  $c_{\text{tore}}$  (refinement parameter) = 0.035,  $c_{\text{tode}}$  (de-refinement parameter) = 0.005.

The refinement function in Eq. (44) is used to solve the same problem with the same refinement parameters of Fig. 4. The numerical results are given at Fig. 5. The results show that Fig. 5 has more de-refined regions and uses less number of resolutions.

In this part of test problem same refinement function is used in Fig. 5, but refinement parameters for this problem are:  $c_{\text{tore}}$  (refinement parameter) = 0.035,  $c_{\text{tode}}$  (de-refinement parameter) = 0.007.

It is seen from Figs. 4–6 that refinement levels are different for each cases. Fig. 6 has more de-refine regions than others. If the more de-refine regions are observed, this indicates that a less number of points are used during the evolutions. The highest refinement level is replaced at discontinuity. As a conclusion, AMR grid structure depends on refinement functions and parameters, which need to be defined carefully for different problems.

Here, the Riemann shock tube problem is solved in 2D using seven different refinement levels with the best refinement parameters which are  $c_{\text{tore}}$  (refinement parameter) = 0.035 and  $c_{\text{tode}}$  (de-refinement parameter) = 0.007. In Fig. 7, density is displayed on the  $x$ - $y$  plane at the time when the shock tube structure is well developed. The AMR run of the shock tube problem creates a nice grid structure that follows the density gradients in the problem well. In Fig. 8, the error function for each grid zone is given with refinement level to see the behavior of the adaptive

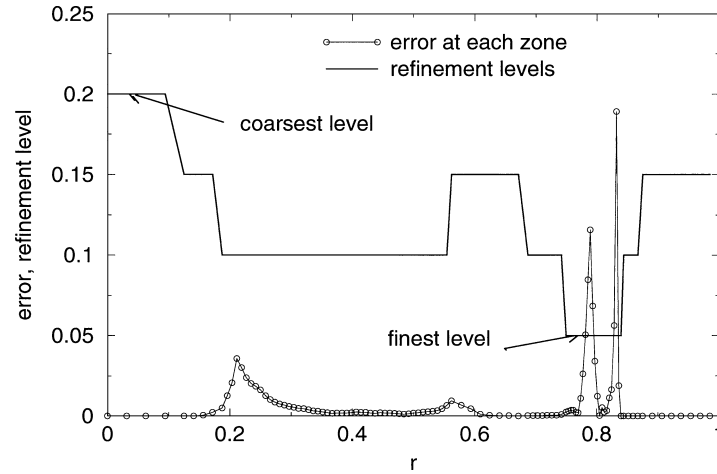


Figure 6. Plotting the error and refinement levels in each zone. Eq. (44) is used as the refinement function with parameters:  $ctore$  (refinement parameter) = 0.035,  $ctode$  (de-refinement parameter) = 0.007.

grid during the evolution. We have also looked at the movie for this problem to see the behavior of refine and de-refine grid during the calculations. The finest grid follows the shock wave and some de-refine regions are produced at contact discontinuity which is a constant state.

It is interesting to compare the computing time for AMR and uniform runs in 3D. Fig. 9 illustrates the time per processor using different resolution for AMR and uniform runs. The times for the uniform and AMR cases are the same for low resolutions,  $16^3$  and  $32^3$ , because the whole computational domain is refined to the maximum level at initial time. It means that there is no AMR in these cases, but the computing times for the resolutions  $64^3$  and  $128^3$  in AMR are less than for uniform grids. The computing time for lower resolution is affected by overhead in the AMR routines. Saving the computational cost in this test problem for  $128^3$  is approximately 30%. Increasing the size of computational domain, the number of refinement levels and the number of resolutions increase saving of the computational cost in the shock tube problem. Increasing these features allow us to see more lower level grids and more de-refine region when shock wave moves to left and rarefaction wave move to right.

Fig. 10 depicts the fine grid volume fraction in the AMR simulation with  $128^3$  zones. Initially, less than 15% of the computational volume was covered by the fine grid. The fine grid volume fraction increases as the shock wave propagates to the left and the rarefaction wave propagates to the right. Although de-refinement in the computational domain occurs around the origin at  $t = 0.25$ , the total number of grid cells continues to increase as the shock and rarefaction waves move to the

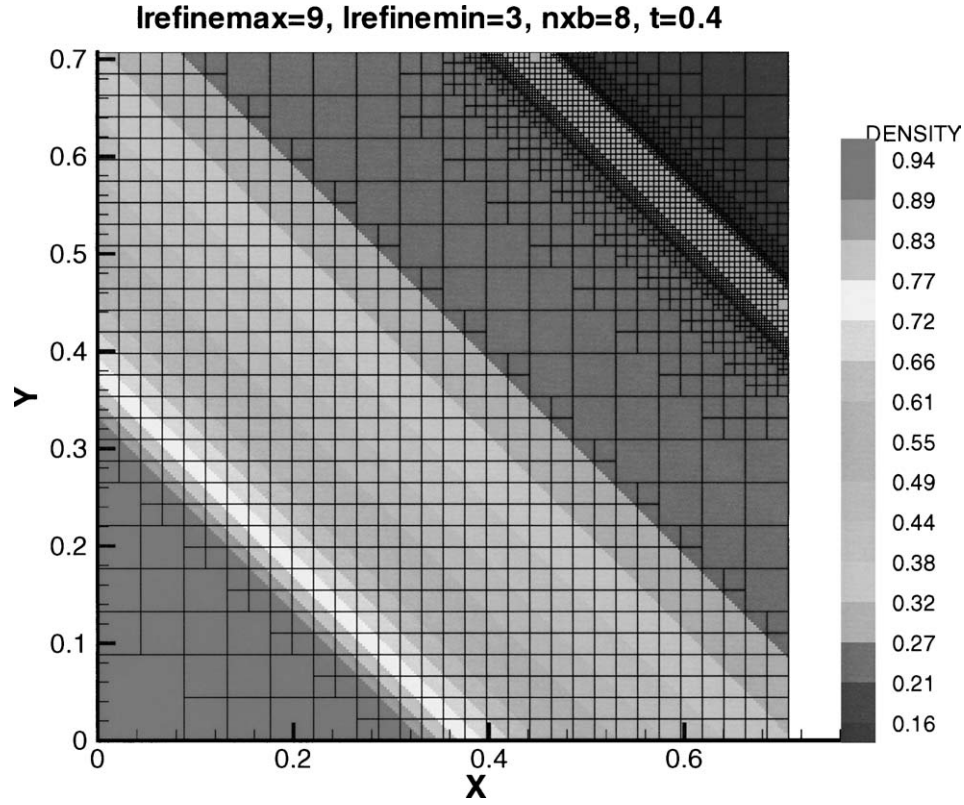


Figure 7. Density is plotted in the  $x$ - $y$  plane. Each square represents the block structure and each block has eight points in each direction. Eq. (44) is used as the refinement function with refinement parameters:  $ctore$  (refinement parameter) = 0.035,  $ctode$  (de-refinement parameter) = 0.007.

right and left, respectively, as shown in Fig. 10. The computational volume is more de-refined than refined around  $t = 0.3$ . The numbers of refined and de-refined zones depend on the refinement criteria and parameters used in numerical simulations.

In these kinds of test simulations, the saving in computational cost of the AMR run over a unigrid run are modest, since the wavelength of the wave,  $\lambda$  is comparable to the dimension  $L$  of the computational domain. In the simulation of the real astrophysical problem, it is expected that wavelength is much smaller than the dimension of computational domain, resulting in significant computational savings. For a full simulation of binary coalescence or accretion disk around the compact objects, one will need to handle multiple scales to evolve dynamics of the sources and propagation of the waves that develop. While the sources are expected to require high resolution, particularly near regions of shocks or shear layers, the waves can likely be handled with lower resolution.

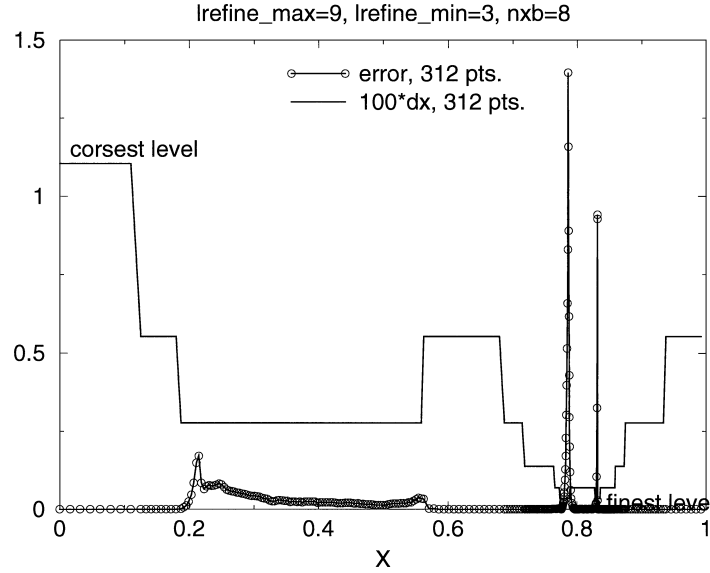


Figure 8. We plot the error and refinement levels in each zone to see the behavior of the refinement levels with errors. Eq. (44) is used as the refinement function with refinement parameters:  $c_{\text{core}}$  (refinement parameter) = 0.035,  $c_{\text{code}}$  (de-refinement parameter) = 0.007.

## 6. General Relativistic Hydrodynamical Test Problem

In this section, we first introduce the Schwarzschild geometry in spherical coordinates to define sources in the GRH equations. Then the accretion disk problems are solved, which have been analytically analyzed, to test the full GRH code in 2D on the equatorial plane.

### 6.1. SCHWARZSCHILD BLACK HOLE

The Schwarzschild solution determined by the mass  $M$  gives the geometry outside a spherical star or black hole. The Schwarzschild spacetime metric in spherical coordinates is

$$ds^2 = -\left(1 - \frac{2M}{r}\right)dt^2 + \left(1 - \frac{2M}{r}\right)^{-1}dr^2 + r^2d\theta^2 + r^2\sin^2\theta d\phi^2. \quad (45)$$

It behaves badly near  $r = 2M$ ; there the first term becomes zero and the second term becomes infinite in Eq. (45). That radius  $r = 2M$  is called the Schwarzschild radius or the Schwarzschild horizon.

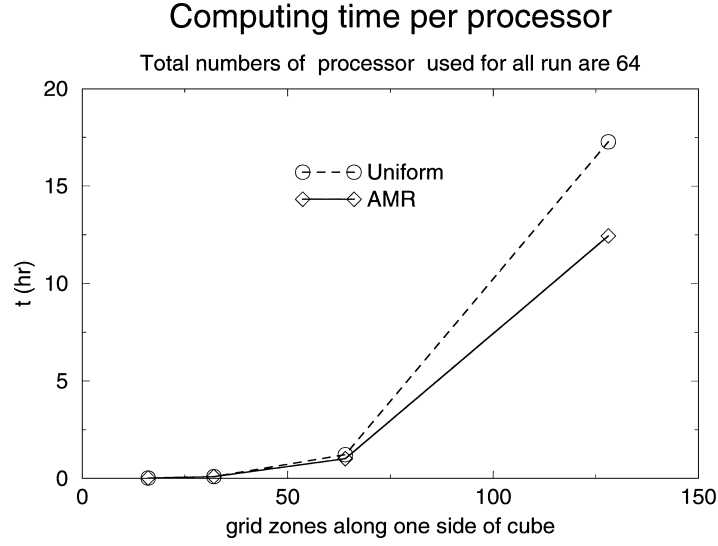


Figure 9. The computing time for AMR and uniform grid runs of the 3D shock tube problem. The red line is the time per processor for AMR and the black line is the time per processor for uniform grid runs.

The spacetime metric for this line element is as follows:

$$g_{\mu\nu} = \begin{pmatrix} -\left(1 - \frac{2M}{r}\right) & 0 & 0 & 0 \\ 0 & \left(1 - \frac{2M}{r}\right)^{-1} & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2 \sin^2 \theta \end{pmatrix}. \quad (46)$$

The lapse function and shift vector for this metric are given below

$$\beta^r = 0.0, \quad \beta^\theta = 0.0, \quad \beta^\phi = 0.0, \quad \alpha = \left(1 - \frac{2M}{r}\right)^{1/2} \quad (47)$$

## 6.2. THE SOURCE TERMS FOR SCHWARZSCHILD COORDINATES

The gravitational sources for the GRH equations are given by Eq. (13). In order to compute the sources in Schwarzschild coordinates for different conserved variables,



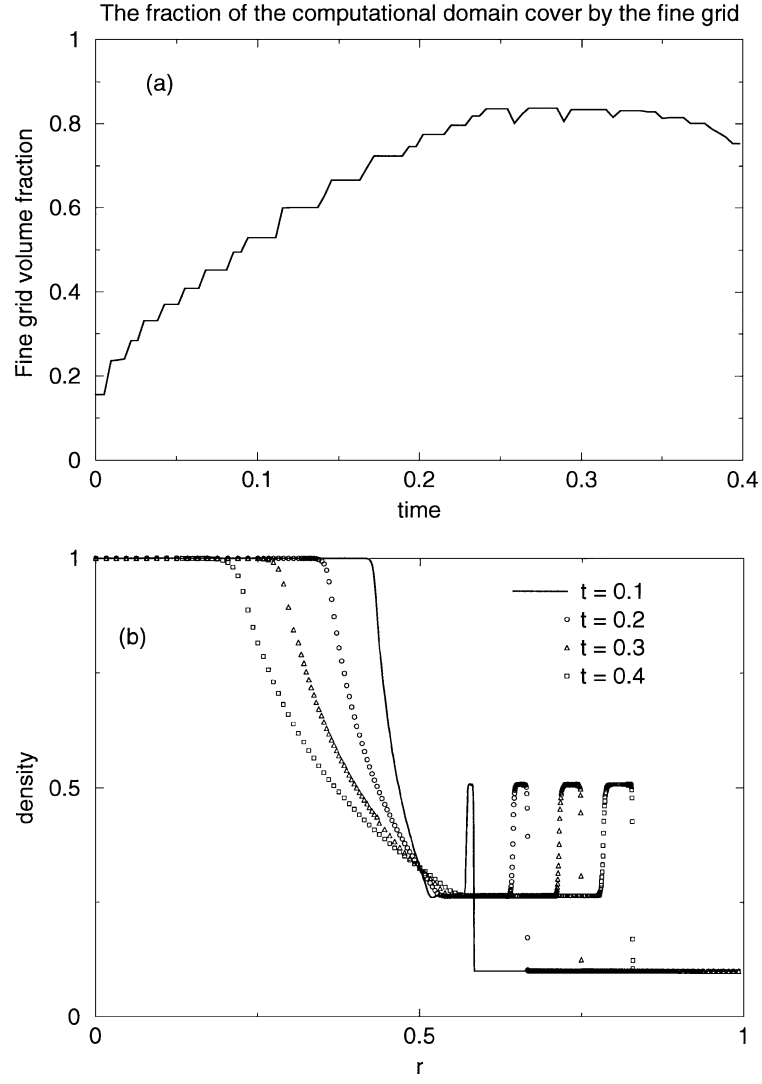


Figure 10. (a) The fraction of the volume covered by the fines grid for AMR run of the 3D shock tube problem. (b) The density of the shock tube problem at different times.

Eq. (13) can be rewritten as

$$\vec{S} = \begin{pmatrix} 0 \\ \alpha \sqrt{\gamma} T^{\mu\nu} g_{\nu\sigma} \Gamma_{\mu r}^{\sigma} \\ \alpha \sqrt{\gamma} T^{\mu\nu} g_{\nu\sigma} \Gamma_{\mu \theta}^{\sigma} \\ \alpha \sqrt{\gamma} T^{\mu\nu} g_{\nu\sigma} \Gamma_{\mu \phi}^{\sigma} \\ \alpha \sqrt{\gamma} (T^{\mu 0} \partial_{\mu} \alpha - \alpha T^{\mu\nu} \Gamma_{\mu\nu}^0) \end{pmatrix}. \quad (48)$$

It is seen in Eq. (13) that the source for conserved density,  $D$ , is zero but the other sources depend on the components of the stress-energy tensor, Christoffel symbols and 4-metric. After doing some straightforward calculations, the sources can be rewritten in Schwarzschild coordinates for each conserved variable in the following form

The source for the momentum equation in the radial direction is

$$\alpha\sqrt{\gamma}T^{\mu\nu}g_{\nu\sigma}\Gamma_{\mu r}^{\sigma} = \frac{1}{2}\alpha\sqrt{\gamma}(T^{tt}\partial_r g_{tt} + T^{rr}\partial_r g_{rr} + T^{\theta\theta}\partial_r g_{\theta\theta} + T^{\phi\phi}\partial_r g_{\phi\phi}). \quad (49)$$

The source for the momentum equation in the  $\theta$  direction is

$$\alpha\sqrt{\gamma}T^{\mu\nu}g_{\nu\sigma}\Gamma_{\mu\theta}^{\sigma} = \frac{1}{2}\alpha\sqrt{\gamma}T^{\phi\phi}\partial_{\theta}g_{\phi\phi}. \quad (50)$$

The source for the momentum equation in the  $\phi$  direction is

$$\alpha\sqrt{\gamma}T^{\mu\nu}g_{\nu\sigma}\Gamma_{\mu\phi}^{\sigma} = 0.0. \quad (51)$$

The source for the energy equation is

$$\alpha\sqrt{\gamma}\left(T^{\mu 0}\partial_{\mu}\alpha - \alpha T^{\mu\nu}\Gamma_{\mu\nu}^0\right) = \alpha\sqrt{\gamma}(T^{rt}\partial_r\alpha - \alpha T^{rt}g^{tt}\partial_r g_{tt}). \quad (52)$$

The nonzero components of the stress-energy tensor in Schwarzschild coordinates can be computed by Eq. (6); they are

$$\begin{aligned} T^{tt} &= \rho h \frac{W^2}{\alpha^2} + P g^{tt}, \\ T^{rr} &= \rho h W^2 (v^r)^2 + P g^{rr}, \\ T^{\theta\theta} &= \rho h W^2 (v^{\theta})^2 + P g^{\theta\theta}, \\ T^{\phi\phi} &= \rho h W^2 (v^{\phi})^2 + P g^{\phi\phi}, \\ T^{tr} &= \rho h \frac{W^2}{\alpha} v^r. \end{aligned} \quad (53)$$

### 6.3. GEODESICS FLOWS

As a general relativistic test problem, an accretion of dust particles onto a black hole is solved. The exact solution for pressureless dust is given in Hawley et al. (1984). This problem is evolved in 2D in spherical coordinates at constant  $\theta = \pi/2$ , which is the equatorial plane. Accordingly, the spatial numerical domain is the  $(r, \phi)$  plane. The  $2.4 \leq r \leq 20$  and  $0 \leq \phi \leq 2\pi$  are used for the computational domain. The initial conditions for all variables are chosen to have negligible values

except the outer boundary ( $r = 20M$ ), where gas is continuously injected radially with the analytic density and velocity. They are formulated as

$$v^r = \sqrt{\frac{2M}{r}} \sqrt{1 - \frac{2M}{r}} \quad (54)$$

and

$$\rho = \frac{1}{W} \frac{d}{r^2 (\frac{2M}{r})^{\frac{1}{2}} (1 - \frac{2M}{r})^{\frac{1}{2}}}. \quad (55)$$

where  $W$  is the Lorentz factor and is given by

$$W = \frac{1}{(1 - \frac{2M}{r})^{\frac{1}{2}}}. \quad (56)$$

Here,  $M$  is the mass of black hole

Throughout the calculation, whenever values at the outer boundary are needed, the analytic values are used. The code is run until a steady-state solution is reached using outflow boundary conditions at  $r = 2.4$ , inflow boundary conditions at  $r = 20$  and the periodic boundary in the  $\phi$  direction are used. No dependence on  $\phi$  is found throughout the entire simulation.

Fig. 11 depicts the rest-mass density  $\rho$ , absolute velocity  $v = (v^i v_i)^{1/2}$  and radial velocity  $v^r$  as a function of radial coordinate at a fixed angular position. The numerical solution agrees well with the analytic solution. Some convergence tests with the SR Hydro code are conducted. They confirmed that it is second-order convergent. The convergence test are also carried out on this problem to test the behavior of the source terms in the GR Hydro code. The analytic values of the accretion problem are compared with numerical results. The convergence results are given in Table III. It is noticed that code gives roughly second-order convergence.

The conservation form of the GRH equations are solved. From (9), the general form of a conservation law is expressed by stating that the variation per unit time

TABLE III  
 $L_1$ -norm error and convergence factors are given for different resolutions

No. of points	No. of time step	Convergence test	
		$L_1$ -norm error	Convergence factor
32	1	$1.675 \times 10^{-5}$	
64	2	$4.0811 \times 10^{-6}$	4.105
128	4	$9.6456 \times 10^{-7}$	4.23
256	8	$2.1372 \times 10^{-7}$	4.51

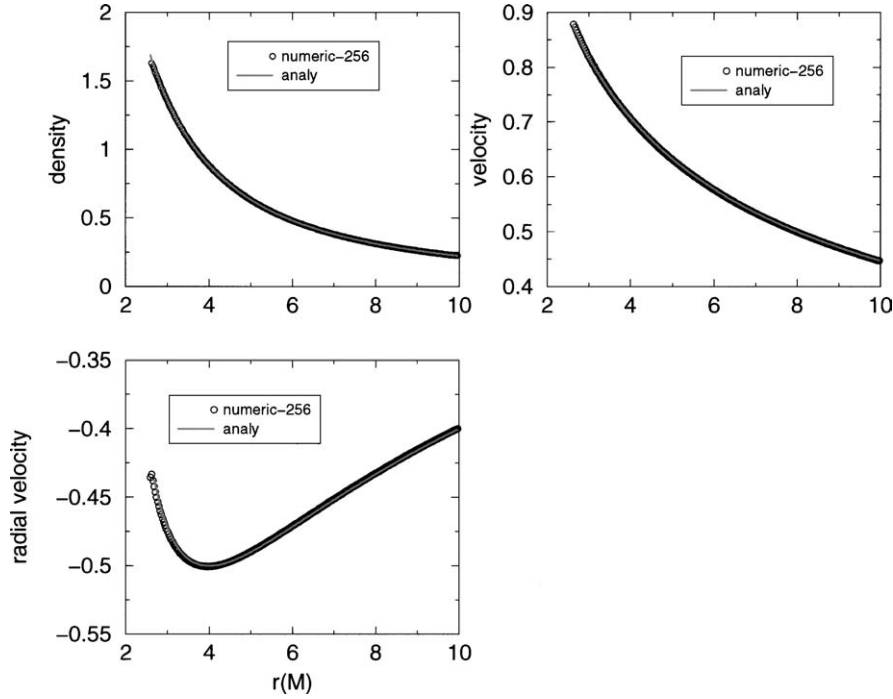


Figure 11. The analytic solutions (red solid lines) with the numerical solutions (black circles) using 256 zones in the radial direction for thermodynamical variables,  $\rho$ ,  $v$  and  $v^r$ .

of this conserved equation within the volume is zero. Then it is expected that conserved variables must be conserved to machine accuracy,  $\sim 10^{-16}$ , at each time step. The results of conservation variables in the numerical test problem show that these variables are conserved to machine accuracy.

#### 6.4. CIRCULAR MOTION OF TEST PARTICLES

The circular motion of a fluid on the equatorial plane will be simulated. To do this, the circular flow is set up with negligible pressure in the equatorial plane, in which angular velocity at each radial direction  $r$  is given by

$$v^\phi = \frac{1}{\sqrt{1 - \frac{2M}{r}}} \sqrt{\frac{M}{r^3}}. \quad (57)$$

It is called a Keplerian velocity.

The analytic expectation of circular motion is that the last stable circular orbit for a particle moving around a Schwarzschild black hole in a Keplerian disk is at  $r = 6M$  ( $M$  is the mass of black hole). Therefore, the gas or particles fall into the black hole, if their radial position is less than  $6M$ . When their radial position is bigger

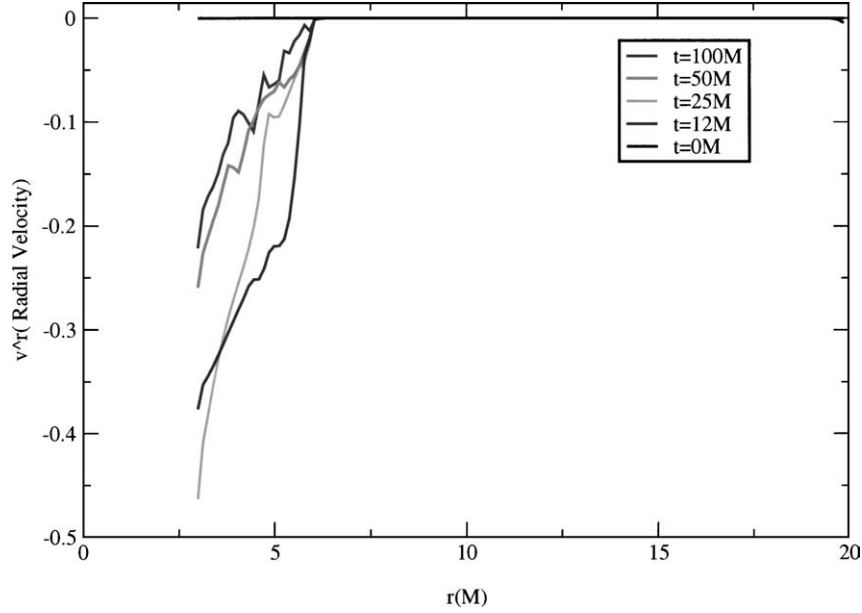


Figure 12. Radial velocity versus  $r$  is plotted. The radial velocity of fluid in the disk remains zero for  $r > 6M$ , during the evolution.  $r = 6M$  is called the last stable circular orbit in a Keplerian disk.

than  $6M$ , they should rotate in a circular orbit. Here, this problem is simulated and the numerical solution is compared to the analytic expectations. It tests the code with sources in the  $\phi$  direction.

In order to simulate this problem, the computational domain is chosen to be  $3M \leq r \leq 20M$  and  $0 \leq \phi \leq 2\pi$ . The computational domain is filled with constant density and pressureless gas, rotating in circular orbits with the Keplerian velocity and zero radial velocity. In Fig. 12, the radial velocities of the gas versus radial coordinate at different times are plotted. It is numerically observed that the gas inside the last stable orbit,  $r = 6M$ , falls into the black hole, while gas outside the last stable orbit follows circular motion with the Keplerian velocity as it is expected analytically. Fig. 13 shows the density of the fluid versus radial coordinate at different times to see the behavior of the disk. It is seen that gas falls into the black hole for  $r < 6M$ . Accordingly, the numerical results from our code are consistent with the analytic expectations.

## 7. Conclusion

In this paper, the Eulerian-type numerical solution of GRH equations constructed for relativistic astrophysics is presented. This code is constructed for general spacetime metric with lapse function and shift vector. It is capable of solving special relativistic problem and general relativistic problem coupling with the Einstein and

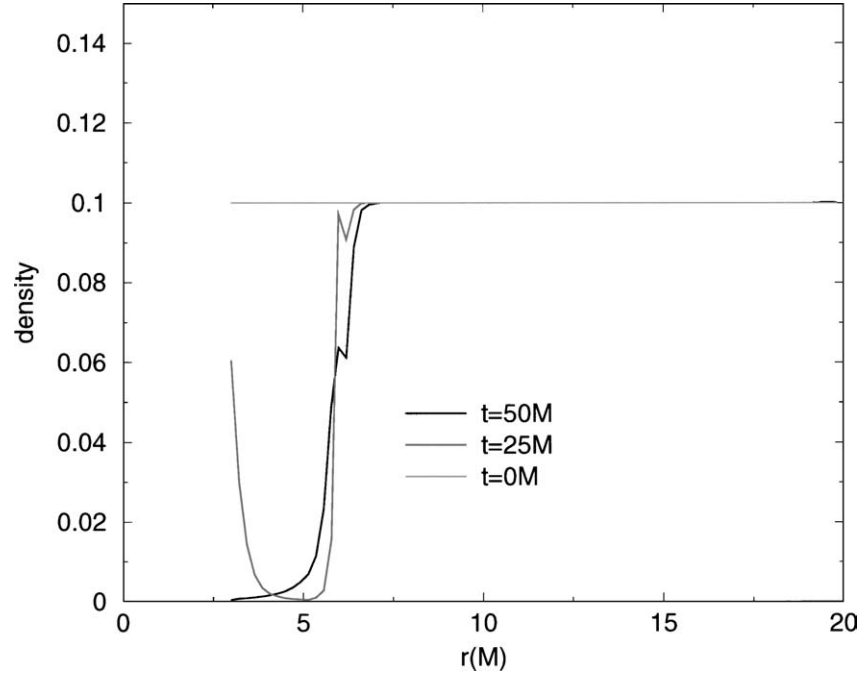


Figure 13. Density versus  $r$  is plotted. Density of the fluid is plotted at different times using different colors. The matter falls into the black hole when  $r$  is less than  $6M$ .

hydrodynamic equations. This paper explains how GRH equation can be solved with AMR.

In the first part of the paper, the GRH equations and the numerical method are given. GRH equations are written in the conservative form and the numerical method and Paramesh package is applied to them with some appropriate way to get working code with AMR.

Then, we have presented and tested solution of 3D relativistic hydrodynamic equations using MUSCL scheme with Marquina fluxes. The results are obtained for problems involving relativistic flows and strong shocks. These problems are solved using uniform and adaptive grid. AMR is one of important techniques in numerical relativity to solve the real astrophysical problems, such as accretion disks around the compact objects, and coalescing of the compact binaries. Adaptive-mesh simulations pose many complex design problems, and a variety of different techniques that exist to solve these problems. AMR is used to solve relativistic equations and to present some results from our numerical calculations. We solved the shock tube problem using different refinement levels to see the behavior of adaptive grid and also to compare AMR and uniform results with analytic solution. Results are converging to analytic solution when the number

of resolutions are increased. Computing time used to solve AMR and uniform grid to get same accuracy are also compared. Results show that when the number of resolutions are increased AMR uses much less computing time than uniform grid.

Finally, the full general relativistic hydrodynamical equation is tested using 2D test problems which are called free falling gas onto black hole and the circular motion of particles around the black hole. The numerical results from these test problems agree with analytic expectation and converge to analytic solution when the number of resolutions are increased.

As a conclusion, AMR has permitted us to address problems of astrophysical interest which are really difficult to solve with uniform grid. Because these problems are dominated by local physics, this code has been designed to handle highly relativistic flows. Hence, it is well suited for 2D and 3D simulations of astrophysical problems. First astrophysical results will be presented in a forthcoming paper for 2D accretion disk problem around the black hole.

### Acknowledgments

I would like to thank Joan M. Centrella for his support and Peter MacNeice and Bruce Fryxell for a useful discussion. This project is supported by NSF PHY9722109. This project was carried out at NASA/GSFC, Laboratory of High Energy Astrophysics. It is also supported by NASA/GSFC IR&D. It has been performed using NASA super computers/T3E clusters.

### References

- Aloy, M.A., Ibanez, J.M., Marti, J.M. and Muller, E.: 2000, *Astrophys. J. Lett.* **510**, 119–122.
- Banyuls, F., Font, J.A., Ibanez, J.M., Marti, J.M. and Miralles, J.A.: 1997, *Astrophys. J.* **476**, 221.
- Bona, C., Masso, J., Seidel, E. and Walker, P.: 1998, gr-qc/9804052.
- Colella, P. and Woodward, P.R.: 1984, *J. Comput. Phys.* **54**, 174–201.
- Donat, R., Font, J.A., Ibanez, J.M. and Marquina, A.: 1998, *J. Comput. Phys.* **146**, 58.
- Font, J.A., Miller, M., Suen, W.-M. and Tobias, M.: 2000, *Phys. Rev. D*, **61**, 044011.
- Hawley, J.F., Smarr, L.L. and Wilson, J.R.: 1984, *ApJ* **227**, 296.
- Hirsch, C.: 1992, in: R. Gallagher and O. Zienkiewicz (eds.), *Numerical Computation of Internal and External Flows*, vol. 2, Wiley-Interscience, New York.
- Ibanez, J.M. and Marti, J.: 1999, *JCAM* **109**, 173.
- MacNeice, P., Olson, K.M., Mobarry, C., deFainchtein, R. and Packer, C.: 2000, *Comput. Phys. Commun.* **126**, 330–354.
- Misner, C.W., Thorne, K.S. and Wheeler, J.A.: 1973, *Gravitation* (W.H. Freeman Co., San Francisco).
- Norman, M.L. and Winkler, K.-H.A.: 1986, in: M.L. Norman and K.-H. A. Winkler (eds.), *Astrophysical Radiation Hydrodynamics*, Reidel, Dordrecht, Holland.

Quirk, J.: 1994, *Int. J. Numer. Met. Fluids.* **18**, 555–574.

Schutz, B.F.: 1985, in: B.F. Schutz (ed.), *A First Course in General Relativity*, Cambridge University Press, Cambridge.

Shu, C.W., and Osher, S.J.: 1989, *J. Comput. Phys.* **83**, 32–78.

Toro, E.F.: 1999, *Riemann Solver and Numerical Methods for Fluid Dynamics*.