Self-gravitational Magnetohydrodynamics with Adaptive Mesh Refinement for Protostellar Collapse

Tomoaki Matsumoto

Department of Humanity and Environment, Hosei University, Fujimi, Chiyoda-ku, Tokyo 102-8160, Japan

matsu@i.hosei.ac.jp

(Received 2006 September 6; accepted ———)

Abstract

A new numerical code, called SFUMATO, for solving self-gravitational magnetohydrodynamics (MHD) problems using adaptive mesh refinement (AMR) is presented. A block-structured grid is adopted as the grid of the AMR hierarchy. The total variation diminishing (TVD) cell-centered scheme is adopted as the MHD solver, with hyperbolic cleaning of divergence error of the magnetic field also implemented. The self-gravity is solved by a multigrid method composed of (1) full multigrid (FMG)-cycle on the AMR hierarchical grids, (2) V-cycle on these grids, and (3) FMG-cycle on the base grid. The multigrid method exhibits spatial second-order accuracy, fast convergence, and scalability. The numerical fluxes are conserved by using a refluxing procedure in both the MHD solver and the multigrid method. The several tests are performed indicating that the solutions are consistent with previously published results.

Key words: hydrodynamics — ISM: clouds — magnetohydrodynamics: MHD — methods: numerical — stars: formation

1. Introduction

Protostellar collapse is one of the most important processes in star formation, and exhibits a high dynamic range in density and spatial dimensions. Adaptive mesh refinement (AMR) is a powerful technique for performing numerical simulations with a high dynamic range in mesh schemes. Local high-resolution is realized by employing grids of differing resolutions. The finer grids are inserted and their location changed according to given refinement criteria. Following the introduction of the fundamental principles of AMR by Berger & Oliger (1984) and Berger & Colella (1989), this technique is becoming widely used in astrophysical simulations.

In star formation, as well as self-gravity, the magnetic field also plays an important role. Both self-gravity and magnetohydrodynamics (MHD) are therefore implemented in recent AMR codes (see the review of Klein et al. 2006).

The existing self-gravitational MHD AMR codes are generally based on Cartesian grids, and block-structured grids are adopted preferentially, although alternative approaches exist, e.g. RAMSES (Fromang et al. 2006). In block-structured AMR, numerical cells are refined in the unit of the block, and a block is itself an ordinal uniform grid. Block-structured grids are divided into two categories: patch-oriented grids and self-similar blocks. In the former approach, a block has variable number of cells, and the shape of the block is also changed in course of refinement. A large block containing many cells and a small block containing just a few cells can co-exist. This approach originated in the study of Berger & Colella (1989), and is adopted in many codes: e.g., the AMR code of Berkeley (Truelove et al. 1998), Enzo (Norman & Bryan 1999), and RIEMANN (Balsara 2001). In the latter approach, all blocks have the same number of cells, but the physical sizes of the blocks are different. This approach has also been adopted by many codes: e.g., FLASH (Fryxell et al. 2000; Banerjee & Pudritz 2006), and NIRVANA (Ziegler 2005). This approach offers advantages including relatively simple algorithms for refinement, parallelization, and vectorization. This paper also follows this approach. In particular, this approach enables the AMR code to implement a full multigrid (FMG)-cycle in the multigrid method for self-gravity. This scheme is an extension of the multigrid method for the nested grid approach (Matsumoto & Hanawa 2003a) to the AMR grid.

There are several ways to treat the MHD, particularly in the treatment of $\nabla \cdot B$. These approaches may be categorized as (1) the constrained transport method with a staggered grid, (2) the projection scheme with a Poisson solver, and (3) the eight-wave formulation (see Tóth 2000; Balsara & Kim 2004 for a comparison of these schemes).

Recently, Dedner et al. (2002) proposed another scheme for cleaning the $\nabla \cdot B$ term. This scheme has a formulation similar to the eight-wave formulation; two additional waves transfer the $\nabla \cdot B$ error isotropically at a given speed, and the waves decay at a given rate. As a result, the error is propagated independently of the gas motion, and the error is diluted. By contrast, using the eight-wave formulation, an additional wave transfers the error at the gas velocity. The error is always propagated downstream of the gas motion, and can become stagnated at the shock wave and the center of the collapsing cloud. The scheme of Dedner et al. (2002) is therefore adopted here.

The author has previously developed a self-gravitational MHD code with nested grid (Matsumoto & Hanawa 2003a). This approach enabled the AMR code to implement a full multigrid (FMG)-cycle in the multigrid method for self-gravity. This scheme is an extension of the multigrid method for the nested grid approach (Matsumoto & Hanawa 2003a) to the AMR grid.

The several tests are performed indicating that the solutions are consistent with previously published results.
2) The vectors \( F \) in equation (1) are expressed by
\[
\frac{\partial F_x}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} = S,
\]
(1) together with Poisson’s equation
\[
\nabla^2 \Phi = 4\pi G \rho,
\]
(2) where \( U \) is a vector of conservative variables, \( F_x, F_y, \) and \( F_z \) are numerical fluxes, and \( S \) is the source term vector. In equation (2), \( \Phi, G, \) and \( \rho \) denote gravitational potential, the gravitational constant, and density, respectively.

For ideal MHD with self-gravity, the vectors in equation (1) are expressed by
\[
U = (\rho, \rho v_x, \rho v_y, \rho v_z, B_x, B_y, B_z, \rho E)^T,
\]
(3)
\[
F_x = \begin{pmatrix}
\rho v_x \\
\rho v_x^2 + P + |B|^2 / 8\pi - B_z^2 / 4\pi \\
\rho v_x v_y - B_x B_y / 4\pi \\
\rho v_x v_z - B_x B_z / 4\pi \\
0 \\
v_z B_y - v_y B_z \\
-v_z B_x + v_x B_z \\
(\rho E + P + |B|^2 / 8\pi) v_x - B_x (B \cdot v) / 4\pi
\end{pmatrix},
\]
(4)
\[
S = (0, \rho g_x, \rho g_y, \rho g_z, 0, 0, 0, \rho g \cdot v)^T,
\]
(5)
where \( v = (v_x, v_y, v_z)^T \) represents velocity, \( B = (B_x, B_y, B_z)^T \) represents the magnetic field, \( g = (g_x, g_y, g_z)^T = -\nabla \Phi \) represents gravity, \( E = |v|^2 / 2 + (\gamma - 1) P/\rho + |B|^2 / 8\pi \rho \) is the total energy, and \( P \) represents pressure. The vectors \( F_y \) and \( F_z \) are obtained by rotating the components in \( F_x \) by the right-hand rule.

For ideal hydrodynamics with self-gravity, the governing equations are obtained by setting \( B = 0 \) and omitting the components of \( B \) (from the 5th to the 7th components) from equations (3)–(5).

Barotropic and isothermal equations of state are also implemented in the AMR code. For these equations of state, the component of \( E \) (the eighth component) is excluded from equations (3)–(5), and \( P \) is expressed as a function of \( \rho \).

3. Discretization

Equations (1) and (2) are solved by a difference scheme based on the finite-volume approach. The computational domain is divided into cells, each of size \( \Delta x \times \Delta y \times \Delta z \). A cell is labeled by \((i, j, k)\), the indices of the cell in the \( x, y, \) and \( z \)-directions, respectively. The location of the cell center is indicated by the position vector \( r_{i,j,k} \).

The conservative variables \( U \), the source term \( S \), and the gravitational potential \( \Phi \) are defined at the cell center, i.e., \( U_{i,j,k} := U(r_{i,j,k}), S_{i,j,k} := S(r_{i,j,k}), \) and \( \Phi_{i,j,k} := \Phi(r_{i,j,k}) \). The numerical fluxes \( F_x, F_y, \) and \( F_z \) are defined at the cell surfaces facing in the \( x, y, \) and \( z \)-directions, respectively. For convenience, the notation of \( F_{x,i,j,k} := F_x[r_{i,j,k} \pm (\Delta x/2) \hat{x}] \) is introduced, where \( \hat{x} \) denotes the unit vector in the \( x \)-direction. We introduce the following notation to describe the spatial differences,
\[
\partial_x Q_{i+1/2,j,k} = Q_{i+1,j,k} - Q_{i,j,k},
\]
(6)
\[
\partial_x Q_{i,j,k} = Q_{i+1/2,j,k} - Q_{i-1/2,j,k},
\]
(7)
\[
\partial_x^2 Q_{i,j,k} = Q_{i+1,j,k} - 2Q_{i,j,k} + Q_{i-1,j,k}/\Delta x^2.
\]
(8)
The differences in the \( y \) and \( z \)-directions are expressed in a similar manner.

4. Grid Refinement

A self-similar block-structured grid is adopted. Each block consists of \( N_x \times N_y \times N_z \) cells, where \( N_x, N_y, \) and \( N_z \) denote the number of cells in the \( x, y, \) and \( z \)-directions respectively, with numbers of cells the same for all the blocks. The number of cells inside a block is fixed, but the cell width differs depending on the grid-level. The blocks are therefore self-similar. A schematic diagram of a block-structured grid is shown in Figure 1. The two-dimensional case is shown for simplicity in this figure, and the number of cells is fixed at \( N_x = N_y = 8 \).

If some cells satisfy a refinement criterion, the block in which these cells lie is divided into 8 child blocks, and every cell inside the parent block is also refined into 8 child cells (see Fig. 1). The cell width of the child cells is half of the parent cell width, that is the refinement ratio is fixed at two. The coarsest grid-level is labeled \( \ell = 0 \) (the base grid), and the finest grid-level is labeled \( \ell = \ell_{\text{max}} \). The \( \ell \)-th grid-level has \( 2^\ell \) times higher spatial resolution than the coarsest grid-level. The block-structured grid is managed by octree structure; the parent (coarse) block is linked with eight fine (child) blocks. Moreover, a block is linked with its neighboring blocks. These link lists are reconstructed every time the grids are refined.
In the construction of a child block, the conservative variables of the parent cells \( U_{I,J,K}^H \) are interpolated to obtain those of the child cells \( U_{i,j,k}^h \). For this purpose linear interpolation with a slope limiter is adopted,

\[
U_{i,j,k}^h = U_{I,J,K}^H + \nabla U^H \cdot (r_{i,j,k}^h - r_{I,J,K}^H),
\]

where \( r_{i,j,k}^h \) and \( r_{I,J,K}^H \) denote the position vectors indicating the centers of the child and parent cells respectively. The gradient inside the parent cell is slope-limited according to

\[
\nabla U^H = \minmod \left( \partial_x U_{I+1/2,J,K}^H, \partial_y U_{I,J+1/2,K}^H, \partial_z U_{I,J,K+1/2}^H \right).
\]

where \( \minmod(\cdot, \cdot) \) denotes the minmod function. Note that this interpolation conserves the conservative variables in the refinement procedure,

\[
\int_{\Omega_{I,J,K}^H} U^H(r) dr = \int_{\Omega_{I,J,K}^h} U^h(r) dr,
\]

where \( \Omega_{I,J,K}^H \) denotes the zone of a parent cell whose center is located at \( r_{I,J,K}^H \).

The refinement algorithm is based on that of Berger & Colella (1989), where grids of level \( \ell \) are refined using the following procedures to construct grids of level \( \ell + 1 \):

1. A blocks of grid-level \( \ell \) is marked if the cells inside the block satisfy a refinement criterion.
2. If blocks of grid-level \( \ell + 2 \) exist, then the corresponding blocks of \( \ell \) are also marked.
3. Blocks adjacent to marked blocks are also marked, so that the grids of level \( \ell + 1 \) properly nest the grids of level \( \ell + 2 \).
4. New blocks of level \( \ell + 1 \) are constructed as child blocks of the marked blocks.
5. Blocks of \( \ell + 1 \) are removed if their parent blocks are not marked.

These procedures are called in ascending order of grid-level, from \( \ell_{\text{max}} - 1 \) to 0.

5. Hydrodynamics and MHD

5.1. Basic Solvers

The governing MHD equations (1), which include as a special case the hydrodynamics equations, are solved on the block-structured grid described in §4. Methods used on ordinal uniform grids can be applied to the block-structured grid if boundary conditions are properly specified for each block. A MUSCL (Monotone Upstream-centered Scheme for Conservation Laws) approach and predictor-corrector method are adopted here for integration with respect to time in order to achieve second-order accuracy in space and time (e.g., Hirsch 1990), and an unsplit approach rather than a directional splitting approach is adopted.

The numerical flux is obtained using the linearized Riemann solver; the solvers are based on the schemes of Roe (1981) and Fukuoka & Hanawa (2000). For MHD, the hyperbolic divergence cleaning of Dedner et al. (2002) is adopted for reducing \( \nabla \cdot B \). According to Dedner et al. (2002), equations (3)–(5) are then modified as follows:

\[
U = (\rho, \rho v_x, \rho v_y, \rho v_z, B_x, B_y, B_z, \rho E, \psi)^T,
\]

where \( \psi \) is a scalar potential propagating a divergence error, \( c_h \) is wave speed, and \( c_p \) is the damping rate of the wave. Note that this modification requires additional components in the basic equation, and nine waves in total are solved for. This approach is similar to the eight-wave formulation. In the method of Dedner et al. (2002), two additional waves transfer the \( \nabla \cdot B \) error isotropically at the speed \( c_h \), and the waves decay at the rate of
In the predictor step, $U^{n+1/2}_{i,j,k}$ is updated to $U^{n+1}_{i,j,k}$ by a half time step, i.e.,

$$U^{n+1}_{i,j,k} = U^n_{i,j,k} - \Delta t \left( \frac{s}{2} \left( \partial_x F^n_{x,i,j,k} + \partial_y F^n_{y,i,j,k} + \partial_z F^n_{z,i,j,k} \right) - S^n_{i,j,k} \right),$$

where the superscript $n$ denotes the time level, and $\Delta t = t^{n+1} - t^n$. The numerical flux $F^n$, which is defined at the cell boundaries, has second-order spatial accuracy due to the MUSCL extrapolation on the primitive variables

$$Q = (\rho, v_x, v_y, v_z, B_x, B_y, B_z, P, \psi)^T.$$  

The source term $S^n_{i,j,k}$ is calculated by

$$S^n_{i,j,k} = \left( \begin{array}{c}
0 \\
\rho^n_{i,j,k} g_{x,i,j,k} - (\nabla \cdot B)_{i,j,k} B^n_{x,i,j,k} \\
\rho^n_{i,j,k} g_{y,i,j,k} - (\nabla \cdot B)_{i,j,k} B^n_{y,i,j,k} \\
\rho^n_{i,j,k} g_{z,i,j,k} - (\nabla \cdot B)_{i,j,k} B^n_{z,i,j,k} \\
0 \\
\rho^n_{i,j,k} (g_{i,j,k}^{n+1/2} \cdot \psi_{i,j,k}^n) - B^n_{i,j,k} \cdot (\nabla \psi)_i^n_{i,j,k} \\
0
\end{array} \right),$$

where $(\nabla \cdot B)_{i,j,k} = \partial_x B_x + \partial_y B_y + \partial_z B_z$, and the magnetic field defined at the cell surfaces $B_x, B_y, B_z$ is obtained from the ninth component of $F_{x,i+1/2,j,k}$. Similarly, $(\nabla \psi)_i^n_{i,j,k}$ is obtained from the fifth to seventh components of $F_{x,i+1/2,j,k}, F_{y,i+1/2,j,k}, F_{z,i+1/2,j,k}$. The ninth component of the source term (14) is evaluated separately by the operator splitting, in which the formal solution is used as follows,

$$\psi^{n+1/2} = \psi^{n+1/2,*} \exp \left[ -\frac{\Delta t}{2} \left( \frac{c_h}{c_p} \right)^2 \right],$$

where $\psi^{n+1/2,*}$ is the ninth component of $U^{n+1/2}$ solved by equation (15). The free parameters $c_h$ and $c_p$ are related to the time-marching, and described in § 5.2.

Note that the gravity $g^{n+1/2}_{i,j,k}$ lags by half a time step. This slight lagging is expected to have a negligible effect on the accuracy in the predictor step, and the gravity in the previous corrector step can be reused in this predictor step (Truelove et al. 1998). This avoids an additional call of the multigrid method to solve Poisson’s equation, significantly reducing the computational costs of the method.

In the corrector step, a spatially second-order numerical flux $F^{n+1/2}$ is obtained by applying MUSCL extrapolation on the primitive variables, which are converted from $U^{n+1/2}$. Using this flux, $U^n_{i,j,k}$ is updated to $U^{n+1}_{i,j,k}$ by a full time step,

$$U^{n+1}_{i,j,k} = U^n_{i,j,k} - \Delta t \left( \frac{s}{2} \left( \partial_x F^{n+1/2}_{x,i,j,k} + \partial_y F^{n+1/2}_{y,i,j,k} + \partial_z F^{n+1/2}_{z,i,j,k} \right) - S^{n+1/2}_{i,j,k} \right).$$

The source term in the corrector step is estimated at the time level of $t = t^{n+1/2},$

$$S^{n+1/2}_{i,j,k} = \left( \begin{array}{c}
0 \\
\rho^n_{i,j,k} g_{x,i,j,k} - (\nabla \cdot B)_{i,j,k} B^n_{x,i,j,k} \\
\rho^n_{i,j,k} g_{y,i,j,k} - (\nabla \cdot B)_{i,j,k} B^n_{y,i,j,k} \\
\rho^n_{i,j,k} g_{z,i,j,k} - (\nabla \cdot B)_{i,j,k} B^n_{z,i,j,k} \\
0 \\
\rho^n_{i,j,k} (g_{i,j,k}^{n+1/2} \cdot \psi_{i,j,k}^{n+1/2}) - B^n_{i,j,k} \cdot (\nabla \psi)^{n+1/2}_{i,j,k} \\
0
\end{array} \right),$$

where $\rho^{n+1/2}$ and $\psi^{n+1/2}$ are obtained from $U^{n+1/2}$, and $g^{n+1/2}$ is obtained by solving Poisson’s equation (2) at the half time step as follows,

$$\nabla^2 \phi^{n+1/2}_{i,j,k} = 4\pi G \rho^{n+1/2}_{i,j,k}. \tag{21}$$

Poisson’s equation is solved by the multigrid method as shown in § 6, and the multigrid method prepares not only the cell-centered $\Phi$ but also the cell-surfaced $g$. The gravity on the cell center $g_{i,j,k}^{n+1/2}$ is obtained by averaging the values of gravity at the cell surfaces,

$$g^{n+1/2}_{i,j,k} = \frac{1}{2} \left( g^{n+1/2}_{x,i+1/2,j,k} + g^{n+1/2}_{x,i-1/2,j,k} \\
g^{n+1/2}_{y,i,j+1/2,k} + g^{n+1/2}_{y,i,j-1/2,k} \\
g^{n+1/2}_{z,i,j,k+1/2} + g^{n+1/2}_{z,i,j,k-1/2} \right). \tag{22}$$

Note that $g$ at the cell surface is obtained by the difference of the cell-centered $\Psi$ (see equations [42]–[44]). Equation (22) therefore coincides with the central difference of $\Psi$ as far as the inside of a block is considered.

Similar to the predictor step, the ninth component of the source term is evaluated by the operator splitting,

$$\psi^{n+1} = \psi^{n+1,*} \exp \left[ -\Delta t \left( \frac{c_h}{c_p} \right)^2 \right], \tag{23}$$

where $\psi^{n+1,*}$ is the ninth component of $U^{n+1}$ obtained by equation (19).

5.2. Time Marching

The time-marching represented by equations (15) and (19) proceeds in units of the grid-level. The code is equipped with two modes of time-marching: an adaptive and a synchronous time-step mode. Usage of the modes depends on whether the gas is non-self-gravitational or self-gravitational. The adaptive time-step mode is appropriate for non-self-gravitational gases, and every grid-level has the same time step. This is because evolution on the fine grid affects the detached coarse grid immediately by the self-gravity, and so the same time step must be chosen for every grid-level (see
discussion of Truelove et al. 1998). Note that the adaptive
time-step mode could also be used for a self-gravitational
gas at the expense of the scheme becoming of first-order
temporal accuracy.

Figure 2 shows the order in which the grid-levels pro-
ced for the adaptive time-step mode schematically. The
numbers associated with the thick arrows denote the or-
der of the time-marching. Coarser grid-levels proceed be-
fore finer grid-levels. The fine grid-level undergoes several
sub-cycles until the time level of the fine grid-level is syn-
chronized with that of the coarse grid-level. The time step
of a finer grid-level, $\Delta t^\ell$, is given by

$$\Delta t^\ell = \Delta t^H 2^{-n},$$

$$n = \min \{ m \in \mathbb{N} | \Delta t^H 2^{-m} \leq \Delta t^\ell_{\text{CFL}} \}.$$  \hfill (24)

where $\Delta t^\ell_{\text{CFL}}$ denotes the time step calculated directly by
the CFL condition at the fine grid level, and $\Delta t^H$ denotes
the time step of the coarser grid level. Note that $\Delta t^h$
is fixed at $\Delta t^H$ in Berger & Colella (1989). On the
other hand, in the method presented here $\Delta t^h$ can be
equal to $\Delta t^H / 2^n$ for $n = 0, 1, 2, \cdots$, if $\Delta t^h$ satisfies
the CFL condition.

In the synchronous time-step mode, a common time step
$\Delta t$ is used at all the grid levels, and is given by

$$\Delta t = \min_{0 \leq \ell \leq \ell_{\text{max}}} \{ \Delta t^\ell_{\text{CFL}} \}.$$  \hfill (26)

where $\Delta t^\ell_{\text{CFL}}$ denotes the time step calculated by the CFL
condition at grid level $\ell$.

The solver described in § 5.1 includes two parameters,
$c_h$ and $c_p$, and these are related to the mode of the time-
marching. The wave speed $c_h$ is obtained as

$$c_h = \text{CFL}_{\Delta t=0} \frac{h}{\Delta H=0},$$  \hfill (27)

where CFL denotes the CFL number, and $\Delta t=0$ denotes
the time step at grid-level $\ell = 0$. For the adaptive time-
step mode,

$$h = \min_{\ell=\ell_{\text{max}}} \{ \Delta x^\ell, \Delta y^\ell, \Delta z^\ell \},$$  \hfill (28)

while for the synchronous time-step mode,

$$h = \min_{\ell=0} \{ \Delta x^\ell, \Delta y^\ell, \Delta z^\ell \}.$$  \hfill (29)

In both cases, $c_h$ is constant across all the grid levels, and
satisfies the CFL condition at every grid level.

The damping rate $c_p$ is obtained from $c_h$,

$$c_p^2 = 0.18 c_h,$$ \hfill (30)

where the coefficient of 0.18 is chosen according to Dedner
et al. (2002).

5.3. Boundary Condition for the Ghost Cells

Each block has ghost cells overlapping with the adjacent
blocks. Figure 3 shows the interface between the coarse and fine blocks in the $x$-direction. Two ghost cells
are prepared in each direction, as expressed by the gray
circles in the figure, because the MUSCL extrapolation
requires two cells in both the left and right sides of the
cell boundary where the numerical flux is obtained.

A boundary condition is imposed on the ghost cells
each time before the predictor and corrector steps pro-
ced. When a block is adjacent to another block of the same grid-level, data is exchanged between the two blocks by a simple copy. When a block is adjacent to a coarse
block, data is interpolated spatially. In addition, data is
interpolated temporally if the adaptive time-step mode is
being used. The procedure for the boundary conditions is
performed as follows:

1. Adjacent coarse blocks are identified.
2. Conservative variables on the coarse cells overlapping
the ghost cells of a fine block are interpolated temporally to coincide with the time level of the fine grid-level. This procedure is omitted if the syn-
chronous time-step mode is being used.
3. The temporally interpolated variables are converted
into the primitive variables. They are interpolated spatially to coincide with the grid points of the ghost
cells of the fine grid-level. The interpolated variables
are copied into the ghost cells.
4. Adjacent blocks in the same grid-level are identified.
5. The variables on cells overlapping the ghost cells are
simply copied into the ghost cells.

In the temporal interpolation, the quadratic interpola-
tion is performed on $U^{H,n}$, $U^{H,n+1/2}$, and $U^{H,n+1}$. Note
that $U^{H,n}$ and $U^{H,n+1}$ are of second-order temporal ac-
curacy while $U^{H,n+1/2}$ is of first-order temporal accuracy.

Using these variables, $U^H(t)$ in $t_n \leq t \leq t_{n+1}$ is interpola-
ted as,

$$U^H(t) = \left( U^{H,n+1} - U^{(1),H,n+1} \right) \left( \frac{t - t_n}{t_{n+1} - t_n} \right)^2 + \frac{(t_{n+1} - t) U^{H,n+1} + (t - t_n) U^{(1),H,n+1}}{t_{n+1} - t_n},$$ \hfill (31)

where $U^{(1),H,n+1}$ denotes the conservative variables of first-order accuracy at the time level $t_{n+1}$, defined as,

$$U^{(1),H,n+1} = 2U^{H,n+1/2} - U^{H,n}.$$  \hfill (32)
For the spatial interpolation, a slope-limited gradient is adopted. In Figure 3, the gradient of the primitive variables inside the cell \((I,J,K)\) is evaluated by

\[
\nabla Q^H = \begin{pmatrix}
\frac{1}{\Delta V^H} \left( \minmod \left( \partial_y Q^H_{I+1/2,J,K}, \partial_y Q^H_{I-1/2,J,K} \right) \right) \\
\frac{1}{\Delta S^H} \left( \minmod \left( \partial_y Q^H_{I,J+1/2,K}, \partial_y Q^H_{I,J-1/2,K} \right) \right)
\end{pmatrix},
\]

(33)

It may be noted that Equation (33) describes a simple interpolation in the normal direction, with slope-limited interpolation in the transverse directions.

5.4. Refluxing at the Interfaces

Refluxing was introduced by Berger & Colella (1989), and has often been used in standard implementations of AMRs. This technique maintains consistency between the fine and coarse grid levels during the time-marching, and ensures that conservation laws, e.g., mass conservation, are satisfied.

As shown in Figure 3, \(U^H_{I,J,K}\) is updated using \(F^H_{x,i-1/2,j,k}\), while \(U^h_{i,j,k}\) is updated using \(F^{h,n}_{x,i-1/2,j,k}\) through several sub-cycles until the fine grid-level catches up with the coarse grid-level, where \(j \in [j_0, j_1], k \in \{k_0, k_1\}\), and \(n\) denotes the index of the sub-cycles of the fine grid-level. Obviously, \(F^H_{x,i+1/2,j,k}\Delta S^H \Delta t^h,n\) should be equal to \(\Sigma F^{h,n}_{x,i-1/2,j,k} \Delta S^h \Delta t^h,n\) to ensure that the conservation laws are satisfied, where \(\Delta S^H = \Delta y^H \Delta z^H\) and \(\Delta S^h = \Delta y^h \Delta z^h\). In refluxing, \(U^H_{I,J,K}\) is re-calculated, taking account of the difference between \(F^H_{x,i+1/2,j,k}\Delta S^H \Delta t^h\) and \(\Sigma F^{h,n}_{x,i-1/2,j,k} \Delta S^h \Delta t^h,n\). More explicitly, as indicated in Figure 4, the refluxing procedure updates \(U^{H,*}\) to \(U^H\) according to

\[
U^H = U^{H,*} - \frac{1}{\Delta V^H} \left( \sum_n \sum_{\text{surface}} F^{h,n}_{x,i-1/2,j,k} \Delta S^h \Delta t^h,n \right),
\]

(35)

where

\[
\sum_n \Delta t^h,n = \Delta t^H,
\]

\[
\sum_{\text{surface}} \Delta S^h = \Delta S^H,
\]

and \(\Delta V^H\) denotes the volume of the coarse cell.

6. Multigrid Method for Self-gravity

6.1. Multigrid Cycles

The multigrid method is widely used in many AMRs for solving Poisson’s equation. In many AMRs, a solution converges by means of the V-cycle on the grids of the AMR hierarchy, and the FMG-cycle on a uniform base grid. By contrast, the multigrid method presented here uses not only V-cycles, but also FMG-cycles on the grids of the AMR hierarchy. An FMG-cycle on the hierarchical grids is also implemented in the approach of Matsumoto & Hanawa (2003a), and the same strategy is adopted here.

Figure 5 shows the grids used in the multigrid cycles for the case \(N_x = N_y = N_z = 8\). For convenience, a grid is labeled by

\[
\Omega^m = \{ \Omega_0, \Omega_1, \Omega_2, \ldots, \Omega_{m_{\text{max}}} \},
\]

(37)

where \(\ell = [0, \ell_{\text{max}}] \) and \(m = [0, m_{\text{max}}] \), respectively. For example, the hatched grids in Figure 5c is labeled as \(\Omega^2\), where the grid of \(\ell = 1\) is coarsened by a factor 2. The coarsest level of all the grids shown in Figure 5a is \(m = 0\), while that of the grids shown in Figure 5b is \(m = 1\), irrespective of grid-level \(\ell\). In addition a composite grid is defined as

\[
\hat{\Omega}^m := \Omega_0^m \cup \cdots \cup \Omega_{\ell_{\text{max}}}^m.
\]

(38)

For example, the composite grids shown in Figures 5a, 5b, and 5c are expressed as \(\hat{\Omega}^0\), \(\hat{\Omega}^1\), and \(\hat{\Omega}^2\), respectively.

An overview of the cycles used by the present numerical method is now presented. First, the solution on \(\Omega_1^m\) for \(0 \leq m \leq m_{\text{max}}\) converges under FMG-cycles, as shown in Figures 5a-5c and 6a. The maximum coarsening level in the FMG-cycle is given by \(m_{\text{max}} = \log_2(N_x, N_y, N_z) - 1\). In other words, the grid is coarsened until the number of the cells per block is decreased up to two at least in one direction. At the bottom of the FMG-cycle, which is marked by \(V\) in Figure 6a, the solution then converges on \(\Omega_0^\ell\) for \(0 \leq \ell \leq \ell_{\text{max}}\) and \(m = m_{\text{max}}\) under V-cycles (the hatched grids in Figs. 6a-6f). Typically, only one iteration of the V-cycle is sufficient, although, for reference, two cycles of the V-cycle are illustrated in Figure 6b. At the bottom of the V-cycle, which is marked by \(B\) in Figure 6b, the solution on the coarsened base grid \(\Omega_0^m\) for \(m \geq m_{\text{max}}\) converges under FMG-cycles, as shown in Figures 5b-5f and 6c. Finally, at the bottom of the FMG-cycle on the base grid, which is marked by \(E\) in Figure 6c, an exact solution is given according to the boundary conditions, because there is only one cell in the computational domain.

Note that computation at \(\hat{\Omega}^0\) of the FMG-cycle accounts for most of the computational time of the multigrid method, because the number of cells in \(\hat{\Omega}^0\) dominates all others. The number of cells decreases by a factor of 1/8 every time the grids are coarsened. This indicates that the computation on \(\hat{\Omega}^0\) dominates the overall computational cost, and this scheme is scalable to the number of cells in a similar way to the hydrodynamics scheme (see also Matsumoto & Hanawa 2003a).
6.2. Smoothing

As a smoothing procedure, the red-black Gauss-Seidel iteration is adopted. The application of the ordinal red-black Gauss-Seidel iteration is restricted to a uniform grid. This iteration scheme is therefore modified so as to apply to composite grids, in which grids with different resolutions co-exist.

The discretization of Poisson’s equation on the composite grids is now introduced. Poisson’s equation can be expressed as a set of two equations,

\[ \nabla \cdot g = -4\pi G\rho, \]  
\[ g = -\nabla \Phi. \]  

These equations are discretized as

\[
\begin{align*}
\partial_x g_{x,i,j,k} + \partial_y g_{y,i,j,k} + \partial_z g_{z,i,j,k} &= -4\pi G\rho_{i,j,k}, \\
g_{x,i+1/2,j,k} &= -\partial_x \Phi_{i+1/2,j,k}, \\
g_{y,i,j+1/2,k} &= -\partial_y \Phi_{i,j+1/2,k}, \\
g_{z,i,j,k+1/2} &= -\partial_z \Phi_{i,j,k+1/2},
\end{align*}
\]

where \( g_{x,i,j,k}, g_{y,i,j,k}, \) and \( g_{z,i,j,k} \) are the components of gravity defined on the cell surfaces.

Considering the fine cells adjacent to the interface between the fine and coarse cells in Figure 7, the potential on the ghost cell, \( \Phi^B \) is required in order to obtain \( g_{x,i-1/2,j,k}^h \). The potential on the ghost cell is given by,

\[
\begin{align*}
\Phi^B &= \frac{10\Phi_{i,j,k}^h + 8\Phi_{i-1,j,k}^h - 3\Phi_{i+1,j,k}^h}{15}, \\
\Phi^* &= \frac{9\Phi_{i,j,K-1}^H + 3(\Phi_{i,j,K-1}^H + \Phi_{i,j-1,K-1}^H) + \Phi_{i,j-1,K-1}^H}{16}.
\end{align*}
\]

where \( \Phi^B \) is obtained by quadratic interpolation in the \( x \)-direction, and \( \Phi^* \) is obtained by bilinear interpolation in the \( y - z \) plane. The quadratic interpolation in the direction normal to the interface satisfies the necessary conditions for so-called conservative interpolation (see Trotttenberg, Oosterlee, & Schüller 2001). By this procedure, the components of gravity of the fine cells adjacent to coarse cells are obtained.

For the coarse cells adjacent to fine cells, the gravity at the interface is given by summing the gravity on the corresponding cell surfaces of the fine cells. For the example of Figure 7, such a coarse gravity is given by,

\[
g_{x,i+1/2,j,k}^H = \frac{1}{4} \sum_{j,k} g_{x,i-1/2,j,k}^h + \frac{1}{4} \sum_{j,k} g_{x,i+1/2,j,k}^h \]  

instead of \( g_{x,i+1/2,j,k}^H = \partial_x \Phi_{i+1/2,j,k}^H \). This is a similar strategy to the refluxing described in § 5.4, and ensures that the solution satisfies Gauss’s theorem: when the normal component of gravity is summed up over the surfaces of any cell, the sum equals the mass contained by the cell multiplied by \( 4\pi G \),

\[
\sum_{\text{surface}} g \cdot \Delta S = -4\pi G \rho \Delta V,
\]

where \( \Delta V \) denotes the volume of the cell, and \( \Delta S \) denotes the vector of the cross-section of the cell in the direction normal to the cell surface. We also confirmed that the refluxing of the gravity is necessary for second-order accuracy by a convergence test (see § 8.3).

This refluxing of the gravity is adopted only for the FMG-cycle on the composite grids shown in Figure 6a, in
Fig. 5. Schematic diagram of the coarsening of grids in the multigrid method. The cell-boundaries and block-boundaries are displayed by thin and thick lines respectively. (a–c) Coarsening of grids in the FMG-cycle on the AMR hierarchy. The number of the cells per block decreases up to $2^3$. (d–f) V-cycle on the AMR hierarchy. The solution converges sequentially on the hatched blocks sequentially. (g–i) Coarsening of grids in the FMG-cycle on the uniform base grid.
Fig. 6. Schematic diagram of the multigrid cycles. The lines pointing diagonally downwards from left to right denote the restriction operators, while the lines pointing diagonally upwards from left to right denote the prolongation operators. The $S$ symbols denote the smoothing operators. The $V$ symbols in panel a denote the $V$-cycles on the AMR hierarchy shown in panel b. The $B$ symbols in panel b denote the FMG-cycles on the uniform base grid shown in panel c. At the point denoted by $E$, an exact solution is obtained according to the boundary conditions.
which the solution converges simultaneously on $\Omega_m^\ell$ over the grid-levels $\ell$. On the other hand, the refluxing of gravity is not adopted in the V-cycle, because the solution converges sequentially over the grid levels.

According to the discretization of equation (41), the Gauss-Seidel iteration is expressed as,

$$
\Phi_{i,j,k}^{\text{new}} = \Phi_{i,j,k} - \frac{h^2}{6} R_{i,j,k},
$$

(49)

$$
R_{i,j,k} := \partial_x g_{x,i,j,k} + \partial_y g_{y,i,j,k} + \partial_z g_{z,i,j,k} + 4\pi G \rho_{z,i,j,k},
$$

(50)

where $h$ denotes the cell width, $R_{i,j,k}$ denotes the residual, and $L$ denotes the Laplacian operator. Equation (49) updates $\Phi_{i,j,k}$ to $\Phi_{i,j,k}^{\text{new}}$ at every iteration. Typically only two iterations are sufficient for the FMG-cycle, and one iteration for the V-cycle. Hereafter, $\Phi_{i,j,k}^{\text{new}} = L^{-1}GS(\Phi_{i,j,k}, \rho_{i,j,k})$ refers the Gauss-Seidel iteration given by equations (49) and (50).

6.3. Prolongation and Restriction

The full-weight prolongation is adopted here, and is given by

$$
\Phi_{I,j,K}^H = \mathcal{T}_h \Phi^H
$$

$$
= \frac{1}{64} \left[ 27 \Phi_{I,j,K}^H + \Phi_{I,j\pm 1,K,K\pm 1}^H + 9 \left( \Phi_{I,j\pm 1,K,K\pm 1}^H + \Phi_{I,j,K,K\pm 1}^H \right) \right. 
$$

$$
\left. + 3 \left( \Phi_{I,j\pm 1,K\pm 1}^H + \Phi_{I,j\pm 1,K,K\pm 1}^H + \Phi_{I\pm 1,j,K,K\pm 1}^H \right) \right],
$$

(51)

where $(I, J, K)$ indicates a coarse cell overlapping with a fine cell $(i, j, k)$, and the sign of $\pm$ depends on the parity of $i$, $j$, and $k$.

The restriction procedure is performed by averaging the values of the fine cells $(i, j, k)$ which overlap the corresponding coarse cell $(I, J, K)$,

$$
\Phi_{I,j,K}^H = \mathcal{T}_h \Phi^H = \frac{1}{8} \sum_{i=1}^{i+1} \sum_{j=1}^{j+1} \sum_{k=1}^{k+1} \Phi_{i,j,k}^h.
$$

(52)

The prolongation and restriction introduced above is used in common with all the multigrid cycles.

6.4. FMG-Cycle on AMR hierarchy

An FMG-cycle based on the standard algorithm of the FMG-cycle for linear equations (see e.g., Press & Teukolsky 1991) is implemented on the composite grids of the AMR hierarchy, because the basic operators are prepared on the composite grids. The smoothing, prolongation, and restriction procedures are given by equations (49), (51), and (52), respectively. In the smoothing procedure, the refluxing of the gravity is performed according to equation (47). In the prolongation, the variables on $\Omega^\ell_m$ are transferred onto $\Omega^{\ell+1}_m$. Similarly, the restriction procedure transfers variables on $\Omega^\ell_m$ onto $\Omega^{\ell+1}_m$. The overlap of coarse grids on fine grids, $\Omega^\ell_m \cap \Omega^{\ell+1}_m$, is therefore not taken into account in the FMG-cycle.

6.5. V-Cycle on AMR hierarchy

In the V-cycle, the multilevel adaptive technique (MLAT) is adopted by using the Full Approximation Scheme (FAS) (see e.g., Trottenberg, Oosterlee, & Schüller 2001). The solution is iterated towards convergence on $\Omega^\ell_m$, where the boundary condition at $\partial\Omega^\ell_m$ is obtained from $\Omega^{\ell-1}_m$ according to equations (45) and (46). The prolongation procedure transfers variables from $\Omega^{\ell+1}_m$ to $\Omega^\ell_m$, while the restriction procedure transfers variables...
from $\Omega^n_m$ to $\Omega^{n+1}_m$.

We now describe the method of the V-cycle in terms of the the fine and coarse grids, $\Omega^n_m$ and $\Omega^{n+1}_m$. As a pre-smoothing procedure, the smoothing procedure is applied to the initial guess of $\Phi^h$ on $\Omega^n_m$,

$$\Phi^h = L^{-1}_{\text{FMG}}(\Phi^h, \rho^h).$$

(53)

The density on $\Omega^{n+1}_m$ is then obtained as

$$\rho^{H,s} := \{ \begin{array}{ll} \mathcal{I}_h^H \rho^h + \tau & \text{on } \Omega^n_{m-1} \cap \Omega^n_m \\ \rho^h & \text{on the remaining part of } \Omega^{n+1}_m \end{array} , \right.$$  (54)

where

$$\tau = L^H \mathcal{I}_h^H \Phi^h - \mathcal{I}_h^H L^H \Phi^h,$$

is the so-called $\tau$-correction. The initial guess on $\Omega^n_{m-1}$ is given by,

$$\Phi^{H,s} := \{ \begin{array}{ll} \mathcal{I}_h^H \Phi^h & \text{on } \Omega^n_{m-1} \cap \Omega^n_m \\ \Phi^h & \text{on the remaining part of } \Omega^{n+1}_m \end{array} . \right.$$  (55)

Using $\rho^{H,s}$ and $\Phi^{H,s}$, the approximate solution $\hat{\Phi}^H$ is obtained from the coarser grids. Then, $\hat{\Phi}^h$ is updated to $\Phi^h$ on $\Omega^n_m$ according to,

$$\hat{\Phi}^h = \bar{\Phi}^h + \mathcal{I}_h^H \left( \hat{\Phi}^h - \mathcal{I}_h^H \bar{\Phi}^h \right).$$

(56)

Finally, post-smoothing is applied to $\hat{\Phi}^h$,

$$\Phi^{h,\text{new}} = L^{-1}_{\text{GS}}(\hat{\Phi}^h, \rho^h).$$

(57)

Thus, by this algorithm, the initial guess $\Phi^h$ is updated to $\Phi^{h,\text{new}}$.

6.6. Utilization of Multigrid Method

The front-end of the multigrid method described here is the FMG-cycle on the AMR hierarchy. Given density $\rho$, the boundary condition for $\Phi$, and the initial guess $\Phi^0$, the FMG-cycle is called iteratively, and $\Phi^n$ is updated to $\Phi^{n+1}$ by means of the following procedure,

$$R_{i,j,k} = 4\pi G \rho_{i,j,k} - \mathcal{L}_{\Phi}^n_{i,j,k},$$

(58)

$$\Phi^+_{i,j,k} = \Phi^n_{i,j,k} + L^{-1}_{\text{FMG}}(0, R_{i,j,k}),$$

(59)

where $\mathcal{L}_{\text{FMG}}(\Phi, \rho)$ denotes the Poisson solver of the FMG-cycle on the AMR hierarchy with initial guess $\Phi$ and the right-hand side of the Poisson equation $\rho$. The reflowing of the gravity is implemented in both the $L$ and $L^{-1}_{\text{FMG}}$ operators. The FMG-cycle $L^{-1}_{\text{FMG}}$ is always called for the fixed boundary condition of zero, because the boundary condition is transmitted to the residual $R$ according to equation (59).

The solution converges under the iterative cycle of equations (59) and (60), reducing the absolute value of the residual $|R|$. In the problem of cloud collapse, the solution of $\Phi$ in the previous time step is adopted as the initial guess, and even a single cycle reduces the residual typically by $|R|^2 \lesssim 10^{-4}$.

7. Parallelization and Vectorization

The code is fully written in Fortran90, and parallelized by the MPI library. The data is partitioned into the nodes used for the calculation in unit of block; all cells within a given block are assigned to the same node. For a given grid-level, all the blocks are ordered by means of the Peano-Hilbert space filling curve, and the blocks are then assigned to the nodes according to this order.

The vectorization in block units is computationally intensive. Because a block has $N_x \times N_y \times N_z$ cells, the vector-length is of the order $N_x \times N_y \times N_z$. In the typical case of $N_x = N_y = N_z = 512$, the vector-length is around 512. This vector-length is acceptable for the vector processor used in the calculations.

8. Numerical Tests

8.1. Double Mach Reflection

We consider the double Mach reflection problem as a hydrodynamics test problem. This test problem was initially proposed by Woodward & Colella (1984), and has since been widely used for tests of high resolution schemes. A planar shock of Mach 10 travels in a medium of $\rho = 1.4$, $P = 1$, and $\gamma = 1.4$ with incident angle of 60° against a rigid wall. The incident shock interacts with the wall, and a complicated structure develops featuring a strong and weak reflected shock, contact discontinuities, and a small jet at the wall (lower boundary; $1/6 \leq x \leq 4$ and $y = 0$). The computational domain given by $x \in [0,4]$ and $y \in [0,1]$ is covered by $8 \times 32$ blocks at $\ell = 0$, and the maximum grid level is set at $\ell_{\text{max}} = 4$. Each grid has $8^2$ cells ($N_x = N_y = 8$), and so the minimum and maximum resolutions are $h = 1/64$ and $1/1024$ respectively, where $h = \Delta x = \Delta y$. Three-dimensional code is applied to this problem even though it has a two-dimensional symmetry, with all the variables uniform in the $z$-direction. In order to maintain this two-dimensional symmetry, the condition $\rho = 0$ is imposed. The Roe scheme is modified here based on Kim et al. (2003) in order to avoid the carbuncle instability at shock waves. The following refinement criterion is adopted,

$$\max |E(\rho_{i,j,k}), E(P_{i,j,k})| \geq 10^{-2},$$

(61)

$$E(q_{i,j,k}) = \frac{1}{q_{i,j,k}} \left( \partial_i^2 q_{i,j,k} + \partial_k^2 q_{i,j,k} \right) h^2.$$  (62)

This criterion captures the curvature of the density and pressure distributions.

Figure 8a shows the density distribution at $t = 0.2$. The shock waves are resolved by finest grids as shown in Figure 8b. Figure 8c shows the magnification of figure 8a around the double Mach stems. An eddy is resolved sharply at the end of the jet, and it is exhibited more clearly in the distribution of the entropy $P/\rho^n$ (Fig. 8d).

8.2. MHD Rotor Problem

The MHD rotor problem was first proposed by Balsara & Spicer (1999), and it tests the propagation of non-linear
Alfvén waves. The model set-up is same as that of Tóth (2000) and Crockett et al. (2005). The computational domain is a unit square of \(x, y \in [0, 1]\). At the initial stage, a uniform cylinder with \(\rho = 10, \, P = 1\), and radius 0.1 rotates at an angular velocity of 20. The ambient medium is at rest with \(\rho = 1, \, P = 1\), and \(v = 0\). At the boundary between the cylinder and ambient medium, a taper region is used in order to reduce the initial transition (see Tóth 2000). The computational domain is subject to a uniform magnetic field, \((B_x, B_y, B_z) = (5, 0, 0)\). The adiabatic index of the gas is \(\gamma = 1.4\). The computational domain is covered by 16 × 16 blocks at \(\ell = 0\), and each block consists of 8 × 8 cells \((N_x = N_y = 8)\). The maximum grid level is set at \(\ell_{\text{max}} = 2\). The coarsest and finest grids therefore exhibit an effective resolution of \(h = 1/128\) and \(1/512\) respectively. The following refinement criterion is adopted,

\[
\max |\mathcal{E}(\rho_{i,j,k}), \mathcal{E}(P_{i,j,k})| \geq 1,
\]

\[
\mathcal{E}(\rho_{i,j,k}) = \left[ \mathcal{E}_x^2(q_{i,j,k}) + \mathcal{E}_y^2(q_{i,j,k}) + \mathcal{E}_z^2(q_{i,j,k}) \right]^{1/2}
\]

\[
\mathcal{E}_x(q_{i,j,k}) = h \left[ \frac{\partial q_{i+1/2,j}}{\partial x} + \frac{\partial q_{i-1/2,j}}{\partial x} + \epsilon h^2 \partial_x^2 q_{i,j,k} \right]
\]

and \(\mathcal{E}_y\) and \(\mathcal{E}_z\) are defined in a similar manner to \(\mathcal{E}_x\), where \(\epsilon = 10^{-2}\) (e.g., Fryxell et al. 2000).

Figures 9a–9d show the density, thermal pressure, Mach number, magnetic pressure at \(t = 0.15\). The finest grids capture the outer shock fronts and inner complex structures, as shown in Figure 9f. The distribution of the physical variables plotted here exhibit excellent agreement with Figure 18 of Tóth (2000) and Figure 12 of Crockett et al. (2005). Figure 9e shows the amplitude of the magnetic divergence error normalized by the cell width and the local magnetic field strength, \(h|\nabla \cdot \mathbf{B}|/|\mathbf{B}|\). The divergence error is calculated as follows (see the ninth component of equations [12]–[14]):

\[
(\nabla \cdot \mathbf{B})_{i,j,k}^{n+1/2} = \frac{\psi_{i,j,k}^n + \psi_{i,j,k}^{n+1} \Delta t \epsilon / c_s^2}{\epsilon h^2 |\nabla |}
\]

The divergence error reaches a maximum value of \(1.5 \times 10^{-2}\) at the inner discontinuity, and is \(2 \times 10^{-3}\) at the outer shock fronts. The divergence cleaning of Dedner et al. (2002) keeps the divergence error small.

### 8.3. Accuracy of Multigrid Method

The accuracy of the multigrid method is examined using the approach of Matsumoto & Hanawa (2003a). Two uniform spheres of masses 1 and 2, and both of radius 6/1024, are located at \((x, y, z) = (12/1024, 0, 0)\) and \((-12/1024, 0, 0)\) in the computational domain \(x, y, z \in [-0.5, 0.5]\). The imposed physical boundary condition is given by a Dirichlet boundary condition on \(\Phi\), and is evaluated by the multipole expansion of the density distribution at \(\ell = 0\), where the monopole, dipole, and quadruple moments are taken into account.

The computation was started from the initial guess \(\Phi = 0\). The residual decreased by more than a factor of several hundred with each FMG iteration given by equations (59) and (60). After 10 iterations, the residual was of the order of the round-off error. The numerical solution for the gravity obtained by this computation is compared to the analytic exact solution.

Figure 10 shows the relative error of the numerically calculated gravity, \(|g - g_{\text{ex}}| / |g_{\text{ex}}|\), on a logarithmic scale, where \(g_{\text{ex}}\) denotes the exact gravity obtained analytically. In this calculation, \(N_x = N_y = N_z = 16\) and \(\ell_{\text{max}} = 4\) are adopted. The block distribution is also shown in
Fig. 9. MHD rotor problem solved by the AMR code. (a) Density, (b) thermal pressure, (c) Mach number, (d) magnetic pressure, (e) magnetic monopole, and (f) grid distribution are shown at $t = 0.15$. In (a)-(d) 30 contour lines are shown in the ranges of $0.483 \leq \rho \leq 12.95$, $0.0202 \leq P \leq 2.008$, $0 \leq |v|/c_s \leq 8.18$, and $0.0177 \leq B^2/\pi \leq 2.042$ respectively.
Figure 10. The maximum error occurs at the edges of the spheres due to the discretization error that arises from the sharp density contrast of the spheres. In the other domains, the error is less than \( \sim 10^{-3} \). It is noteworthy that no significant error appears at the interfaces between the coarse and fine grids. This is attributed to the refluxing of the gravity described in § 6.2.

Figure 11 shows dependence of the error on resolution. The error of the gravity is measured by changing the number of cells inside a block, \( N_x, N_y, \) and \( N_z \), but fixing the block distribution. The ordinates of Figure 11a, 11b, and 11c denote the errors measured by the average, root mean square, and maximum values, corresponding to the \( L_1, L_2, \) and \( L_\infty \) norms, respectively. The abscissa denotes the cell width of the coarsest grid \( \ell = 0 \). The five lines show the errors at the grid levels of \( \ell = 0, \ldots, 4 \), and the five points in the lines denote the errors for \( N_x = N_y = N_z = 2, 4, 8, 16, \) and 32. All the lines, except for \( \ell = 4 \) in Figure 11c, exhibit the second-order accuracy of the multigrid method presented here. The line of \( \ell = 4 \) in Figure 11c shows instead a first-order accuracy. This is attributed to the discretization error of the density put on this grid level.

8.4. Convergence of Multigrid Method

The decrease of the residual defined by equation (50), is measured to evaluate the efficiency of the iteration for the same problem as described in § 8.3. Figure 12 shows the maximum residual in each grid level as a function of the number of the FMG-cycles defined by equations (59) and (60) for the cases \( N_x = N_y = N_z = 8, 16, \) and 32. The residuals plotted here are multiplied by \( h^2 \) so that they have a dimension of \( \Psi \). For the case \( N_x = N_y = N_z = 8 \) (Fig. 12a), the residuals in all the grid levels decrease in proportion to \( \exp(-6n) \). After 7 iterations of the FMG-cycles, the residuals at all grid levels reach the round-off error. On the other hand, for the cases \( N_x = N_y = N_z = 16 \) and 32 (Fig. 12b and 12c), the rate at which the residuals decrease on the coarse grid levels are slower than those of the fine grid levels. The slower convergence is due to the boundary condition imposed on the coarsest grid.

8.5. Collapse and Fragmentation of An Isothermal Cloud

The present numerical technique is applied to the problem of the collapse and fragmentation of an isothermal cloud as a gravitational hydrodynamics test problem. While isothermal collapse has been calculated by many authors, the particular model of Bate & Burkert (1997) and Truelove et al. (1998) is followed here. The initial cloud has an uniform and spherical density distribution. It rotates rigidly around the z-axis. The mass of the cloud is \( 1M_\odot \), and the radius is \( R_c = 5 \times 10^{16} \) cm. The density of the cloud is therefore \( \rho_0 = 3.79 \times 10^{-18} \) g cm\(^{-3}\). Traditionally a cloud is characterized by two global quantities, \( \alpha = E_{\text{th}}/|E_{\text{grav}}| \) and \( \beta = E_{\text{rot}}/|E_{\text{grav}}| \), where \( E_{\text{th}}, E_{\text{rot}}, \) and \( E_{\text{grav}} \) denote the thermal, rotation, and gravitational energies. The cloud here has \( \alpha = 0.26 \) and \( \beta = 0.16 \). The speed of sound and the angular velocity are obtained as \( c_s = 0.166 \) km s\(^{-1}\) and \( \Omega = 7.14 \times 10^{-13} \) s\(^{-1}\) respectively. The cloud is perturbed by a bar perturbation with amplitude of 10%: \( \rho = \rho_0 (1 + 0.1 \cos 2\phi) \). The cloud is embedded in an ambient gas, whose density is 0.01\( \rho_0 \).

The computational domain is \( x, y, z \in [-2R_c, 2R_c] \times [-2R_c, 2R_c] \times [0, 2R_c] \), and a mirror boundary condition is imposed on the \( z = 0 \) plane. Blocks of \( N_x = N_y = N_z = 8 \) are adopted. At the initial stage, \( 16 \times 16 \times 8 \) blocks are distributed at the grid level of \( \ell = 0 \). The cloud radius \( R_c \) is therefore resolved by 32 cells. This initial resolution is same as that of Truelove et al. (1998). The Jeans condition is employed as a refinement criterion; the blocks are refined when the Jeans length is shorter than 8 times of cell width; \( (\pi c_s^2/G\rho)^{1/2} < 8h \). This refinement criterion is twice as severe as that of Truelove et al. (1997).

Figure 13 shows the maximum density as a function of time. The cloud collapses increasing the central density gradually in the early phase. A thin disk forms due to
Fig. 11. The relative error of the numerically computed gravity $|g - g_{\text{ex}}| / |g_{\text{ex}}|$ as a function of $h_{\text{max}}$ (cell width of the coarsest grids). The error is measured by (a) average, (b) root mean square, and (c) maximum on each grid-level separately. The open diamonds, open circles, filled diamonds, filled circles, and filled square denote the errors on grids of $\ell = 0, 1, 2, 3, 4,$ and 5, respectively. The dashed lines display the relationship of (a-c) errors in proportion to $h_{\text{max}}^2$, and (c) that in proportion to $h_{\text{max}}$.

Fig. 12. The maximum residual ($|h^2 R|_{\text{max}}$) as a function of the number of the FMG-cycles for the cases (a) $N_x = N_y = N_z = 8$, (b) $N_x = N_y = N_z = 16$, and (c) $N_x = N_y = N_z = 32$. The open diamonds, open circles, filled diamonds, filled circles, and filled square denote the residuals measured on grids of $\ell = 0, 1, 2, 3,$ and 4, respectively. The dashed line displays the relationships $|h^2 R|_{\text{max}} \propto \exp(-4n)$ and $\exp(-6n)$, where $n$ denotes the number of the FMG-cycles.
the fast rotation of the cloud at the stage of $\rho_{\text{max}} = 2 \times 10^{-13} \text{g cm}^{-3}$, and $t = 1.25 \times 10^{12} \text{s} (= 1.16 t_{\text{ff}})$, where $t_{\text{ff}} = (3\pi/32G\rho_0)^{1/2} = 5.60 \times 10^{12} \text{s}$, and this disk bounces in the $z$-direction. After the bounce, the density increases more rapidly, and it approaches a singularity at $t = 1.35 \times 10^{12} \text{s} (= 1.25 t_{\text{ff}})$.

Figure 14a shows the density distribution in the $z = 0$ plane at the stage of $\rho_{\text{max}} = 1.02 \times 10^5 \text{g cm}^{-3}$. The disk has two density peaks, which are resolved by the fine blocks. Each density peak has a filamentary structure. This density structure was also found by Truelove et al. (1998) (see their Fig. 13).

Figure 14b shows the upper left bar at the stage $\rho_{\text{max}} = 1.01 \times 10^8 \rho_0$. The bar collapses to form a very narrow filament. This is also consistent with Truelove et al. (1998) (see their Fig. 13).

8.6. Collapse and Outflow Formation of A Cloud Core with Slow Rotation and Oblique Magnetic Field

The collapse of a cloud with magnetic field parallel to the rotation axis has been simulated by Machida et al. (2004); Machida et al. (2005a); Machida et al. (2005b); Ziegler (2005); Banerjee & Pudritz (2006); Fromang et al. (2006), while the collapse of an oblique magnetic field has been simulated by Matsumoto & Tomisaka (2004); Machida et al. (2006). In the case of the oblique magnetic field, the rotation axis changes its direction due to the anisotropic magnetic braking during the collapse, and an outflow is ejected after an adiabatic core formation.

In this paper, model MF45 of Matsumoto & Tomisaka (2004) is calculated by the AMR presented here. The initial cloud has the density profile of the critical Bonnor-Ebert sphere (Ebert 1955; Bonnor 1956), but the density is increased by a factor of 1.68. The central density is $\rho_0 = 1 \times 10^{-19} \text{g cm}^{-3}$, and the radius of the cloud is $R_c = 5.49 \times 10^{17} \text{cm}$. The cloud rotates slowly at the angular velocity of $\Omega_0 = 7.11 \times 10^{-7} \text{yr}^{-1} (= 0.15 t_{\text{ff}}^{-1})$, where the freefall time is $t_{\text{ff}} = 2.10 \times 10^{7} \text{yr}$. The initial magnetic field is inclined at an angle of $45^\circ$ with respective to the $z$-axis (rotation axis), and is of strength $18.6 \mu \text{G}$. The cloud has parameters of $\alpha = 0.5$, $\beta = 0.02$, and $E_{\text{mag}}/|E_{\text{grav}}| = 0.721$, where $E_{\text{mag}}$ denotes the total magnetic energy inside the cloud.

In order to mimic the formation of the first stellar core, the equation of stage is changed according to the density: an isothermal gas of $T = 10 \text{K} (c_s = 0.19 \text{km s}^{-1})$ is assumed when density is less than $\rho_{cr} = 2 \times 10^{-13} \text{g cm}^{-3}$, while a polytrope gas of $\gamma = 5/3$ is assumed when density is higher than $\rho_{cr}$.

Figure 15a shows the central region of $(287 \text{ AU})^3$ at $t = 4.81 \times 10^5 \text{yr}$. As calculated by Matsumoto & Tomisaka (2004), the outflow is ejected from the region near the first core. The outflow speed reaches $10 c_s$, which is also consistent with the previous calculation.

Figure 15b shows the block distribution, where grid lev-
No. | Self-gravitational MHD-AMR | 17

Fig. 15. (a) Three-dimensional view of protostellar collapse and outflow formation at $t = 4.81 \times 10^5$ yr. The 6 disk-shaped isosurfaces are shown for $4.7 \leq \log \rho/\rho_0 \leq 7.0$, and the 4 bi-polar isosurfaces are shown for $2.0 \leq v_r/c_s \leq 9.0$. The tubes display the magnetic field lines. The coordinates are shown in the units of $c_s/(4\pi G\rho_0)^{1/2}$. (b) Same as panel a but block distribution is shown. The blocks of grid-levels 11 to 13 are shown.

Levels of $\ell = 11 - 13$ are shown. Each grid level has $8^3$ blocks. This grid distribution reproduces effectively the same cell distribution as the nested grid including $64^3$ cells in each grid level.

9. Summary

A self-gravitational MHD code, called SFUMATO, applying the AMR technique is presented. The grid is configured in a block structure.

The MHD scheme is implemented so that it is of second-order spatial accuracy by means of the TVD approach. The upwind numerical flux is obtained by the linearized Riemann solver. The scheme is fully cell-centered, and the divergence error of the magnetic fields is cleaned.

The self-gravity is solved by a multigrid method composed of: (1) FMG-cycle on the AMR hierarchical grids, (2) V-cycle on these grids, and (3) FMG-cycle on the base grid. The FMG-cycle on the AMR hierarchical grids enables a scalable dependence on the number of cells. The multigrid method ensures that the solution converges rapidly; the residual is reduced by a factor $10^{-3} - 10^{-2}$ every iteration. The multigrid method exhibits a second-order spatial accuracy. Moreover, no spurious features appear at the interfaces between fine and coarse grid levels, due to the flux conservation at the interface.

The MHD scheme and multigrid method are combined so as to have second-order temporal accuracy. The time-marching has two modes: adaptive and synchronous time-step modes. The former mode is adopted for a problems including self-gravity, while the latter mode is used for all other problems.

The AMR code was tested by considering test problems given by the double Mach reflection, the MHD rotor, fragmentation of an isothermal cloud, and outflow formation in a collapsing cloud. The present results are in good agreement with those of previous studies conducted by the present author as well as other authors.

The author would like to thank K. Tomisaka and T. Hanawa for valuable discussions. Numerical computations were carried out on the VPP5000 supercomputer at the Astronomical Data Analysis Center (ADAC) of the National Astronomical Observatory, Japan. This research was supported in part by Grants-in-Aid for Young Scientists (B) 16740115, for Scientific Research (C) 17540212, and for Scientific Research (B) 17340059 awarded by the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Berger, M. J., & Colella, P. 1984, Journal of Computational Physics, 82, 64
Dedner, A., Kemm, F., Kröner, D., Munz, C.-D., Schnitzer, T., & Wesenberg, M. 2002, Journal of Computational Physics, 175, 645