Solar Phys (2010) 267: 463–491
DOI 10.1007/s11207-010-9649-6

AMR Simulations of Magnetohydrodynamic Problems by the CESE Method in Curvilinear Coordinates

Chaowei Jiang · Xueshang Feng · Jian Zhang · Dingkun Zhong

Received: 3 July 2010 / Accepted: 26 September 2010 / Published online: 27 October 2010

Abstract The objective of this paper is to present new extensions of the space – time conservation element and solution element (CESE) method for simulations of magnetohydrodynamic (MHD) problems in general curvilinear coordinates by using an adaptive mesh refinement (AMR) grid system. By transforming the governing MHD equations from the physical space (x, y, z) to the computational space (ξ, η, ζ) while retaining the form of conservation, the CESE method is established for MHD in the curvilinear coordinates. Utilizing the parallel AMR package PARAMESH, we present the first implementation of applying the AMR CESE method for MHD (AMR-CESE-MHD) in both Cartesian and curvilinear coordinates. To show the validity and capabilities of the AMR-CESE-MHD code, a suite of numerical tests in two and three dimensions including ideal MHD and resistive MHD are carried out, with two of them in both Cartesian and curvilinear coordinates. Numerical tests show that our results are highly consistent with those obtained previously by other authors, and the results under both coordinate systems confirm each other very well.

Keywords Adaptive mesh refinement · CESE method · Curvilinear coordinates · Magnetohydrodynamics

C. Jiang · X. Feng (✉) · D. Zhong
SIGMA Weather Group, State Key Laboratory for Space Weather, Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing 100190, China
e-mail: fengx@spaceweather.ac.cn

C. Jiang
College of Earth Sciences, Graduate University of Chinese Academy of Sciences, Beijing 100049, China

J. Zhang
Supercomputing Center, Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China

1. Introduction

The space – time conservation element and solution element (CESE) method, developed originally by Chang and his co-workers (Chang, 1995; Chang, Wang, and Chow, 1999; Wang and Chang, 1999; Yu *et al.*, 1998; Zhang, Yu, and Chang, 2002), is a new method in high-resolution computational fluid dynamics for solving linear and nonlinear convection-diffusion equations in one, two, and three dimensions. Many advantages have been seen from its applications. Unlike traditional numerical methods, the key principle of the CESE method is to treat space and time as one entity in calculating flux balance. In the CESE method, the first-order spatial derivatives are also considered as variables to be solved. By designing the solution element (SE) and conservation element (CE), the CESE method allows the physical parameters to have smooth profiles inside an SE, while between SEs or in CEs, they may be discontinuous. Thus, the CESE method can capture sharp discontinuity within a few grid points. In addition, the introduced numerical damping effect in this method is controllable.

The CESE method has been used to study flows with moving and steady shocks, acoustic waves, complex vertical flows, detonations, shock/acoustic/vortex interactions, dam-break flows, *etc.*; the reader is referred to the above-cited references. Recently, the CESE method has been extended to the field of magnetohydrodynamics (MHD) to study the shock tube MHD problem, the smooth Alfvén wave problem, and the well-known Orszag and Tang's MHD vortex problem (Zhang *et al.*, 2006). Feng, Hu, and Wei (2006) and Hu and Feng (2006) applied the CESE method to 2.5D resistive MHD equations in Cartesian coordinates, with the purpose of studying the magnetic reconnection process. They then successfully developed a solar wind model (the solar-interplanetary CESE model, also called the SIP-CESE model) using the CESE method on unstructured pentahedron mesh (Feng, Zhou, and Wu, 2007). In all these former studies, the numerical computations give favorable results, but the CESE method has a much simpler logic and operational technique. Furthermore, without any special treatment, the method can control the numerical error caused by $\nabla \cdot \mathbf{B}$ (divergence of magnetic field) within a tolerable level (Zhang *et al.*, 2006; Feng, Hu, and Wei, 2006; Feng, Zhou, and Wu, 2007), which is an attractive advantage over other finite volume methods for MHD.

Since the pioneering work by Berger and Colella (1989), the technique of adaptive mesh refinement (AMR) has been developing rapidly in computational fluid dynamics and is becoming a standard tool for treating problems with multi-orders of spatial or temporal scales. By automatically adapting the computational mesh to the solution of the governing partial differential equations (PDEs), methods based on AMR can assign more mesh points for regions demanding high resolution (*e.g.*, high gradient regions) and at the same time, give fewer mesh points to other less interesting regions (low gradient regions), thereby providing the required spatial resolution while minimizing memory requirements and CPU time. However, implementing a parallel-AMR approach in a computational code implies a considerable undertaking. To overcome this hurdle, a number of AMR software infrastructures have been developed that support the parallel implementation of PDEs. These include AmrLib/BoxLib (Rendleman *et al.*, 2000), Chombo (Colella *et al.*, 2007), DAGH (Parashar and Browne, 2007), GrACE (Parashar, 2007), PARAMESH (MacNeice *et al.*, 2000), and SAMRAI (Garaizar, Hornung, and Kohn, 1999).

In recent years, AMR techniques have been widely used in an increasing number of simulations of MHD problems in plasma physics and astrophysics, especially MHD problems with disparate spatial and temporal scales (Powell *et al.*, 1999; Ziegler, 1998; Linde, 2002; Stone *et al.*, 2008). The code FLASH (Linde, 2002) and BATSRUS (Powell *et al.*, 1999;

Tóth *et al.*, 2005, 2006) developed for astrophysics and space weather modeling (*e.g.*, computation of the solar wind, simulation of solar eruptions such as magnetic reconnection and coronal mass ejections (CMEs), and modeling of their effects on the Earth's magnetosphere) are two representative implementations of AMR techniques in MHD simulation. In the MHD simulations performed on AMR grids in recent years, the finite volume schemes of Godunov's type are the dominant numerical method of discretization, while the advantages of the CESE method on AMR grids have never been observed for the MHD simulations.

In this work, our primary goal is to realize the CESE method on AMR grids for MHD simulations with the help of the PARAMESH toolkit. PARAMESH, a package of Fortran 90 subroutines, is designed to provide an application developer with an easy route to extend an existing serial code which uses a logically Cartesian structured mesh into a parallel code with AMR (MacNeice *et al.*, 2000). The package builds a hierarchy of sub-grids to cover the computational domain, with spatial resolution varying to satisfy the demands of the application. These sub-grid blocks form the nodes of a tree data structure (quad-tree in two dimensions or oct-tree in three dimensions). Each grid block has a logically Cartesian mesh. The adaptation of the solution is realized by dividing (refining) and coarsening appropriate blocks. Due to the high performance and usability of this publicly available toolkit, it has been utilized for the management of the AMR grid system and parallelization. The same package is used by the code FLASH and ATHENA in the astrophysics community (MacNeice *et al.*, 2000; Olson and MacNeice, 2005; Olson, 2006), and is available on the website http://www.physics.drexel.edu/~olson/paramesh-doc/Users_manual/amr.html.

Our underlying motivations are as follows. In modeling solar-terrestrial physics problems, such as the solar wind, solar flares, coronal mass ejections, and the global evolution of magnetic structures, it is inevitable to use the spherical shell domain to describe the computational domain. In a global simulation such as the three-dimensional propagation of coronal mass ejections from the Sun to the Earth, one has to deal with the spherical boundary of the Sun. Evidently, the solar surface, described by a sphere of unit radius located inside the computational domain, cannot be consistent with any of the Cartesian or cylindrical coordinate surfaces. The inner (solar surface) boundary yields the question of how these boundary conditions are well realized on the grid, since it contains the surface from which the solar wind emanates. Intuitively, the Sun's geometry, obviously spherical shaped, would suggest the use of spherical coordinates (r, θ, ϕ), especially since the radial convergence of lines of constant (θ, ϕ) can entail the extra advantage of enhanced spatial resolution near the Sun's surface. Any grid partition of the spherical shell geometry will lead to polyhedron mesh grids in the case of spherical coordinates, and requires a spherical-surface body fitting produced by a cutting-cell technique in the case of Cartesian or cylindrical coordinates. As is well known, dealing with the boundary conditions properly is a critical aspect in such numerical modeling, as the Sun is the source of the activities in the corona and in the interplanetary space. The implementation of AMR in polyhedron grid systems is also a challenge. Even the block-AMR grid successfully used in Cartesian coordinates still has shortcomings when dealing with the spherical surface boundary conditions. Firstly, the blocks must be refined to a much higher level to approximate the spherical surface, which will be a great overhead on the management of the data structure, and even so, the spherical surface cannot be represented exactly and the corresponding numerical errors will be transported into the domain. Secondly, the boundary conditions become difficult to implement on the boundary blocks in the Cartesian condition, even impossible in the case of boundary coordinates using the characteristic method, where the finite differences must be computed on the surface (Wu *et al.*, 2001, 2006).

A natural choice to overcome these difficulties is the structured boundary-fitted grid in curvilinear coordinates, which can characterize the computational domain more excellently and precisely than other grid systems but with less memory requirement (Anderson, 1995; Hoffman and Chiang, 2000). These are all our considerations for the use of general curvilinear coordinates, which can handle the computational domain and the boundary conditions more excellently and precisely. At the same time, the AMR technique is efficient in solving problems with multi-orders of spatial and temporal scales like the solar wind flows and global evolution of CMEs from the Sun to the Earth, since the plasma flow and magnetic field admit a large disparity of magnitude in the solar-terrestrial space. To this end, the present work is devoted to the AMR implementation of the CESE-MHD method in general curvilinear coordinates, by transforming the governing PDEs from the physical space to the computational space, in anticipating the applications in solar-terrestrial physics. As the first step, several general benchmarks are validated to show the capabilities.

This paper is organized as follows. Section 2 illustrates the model equations. Section 3 presents the improved CESE method in both two and three dimensions, which is modified from the original method for the convenience of integrating the CESE method within the PARAMESH package. This method is extended to general curvilinear coordinates in Section 4. In Section 5 a brief description is given for the implementation of the CESE method on block-structured AMR grids by utilizing the PARAMESH package and the variable timestep algorithm. Section 6 provides the numerical tests, and the conclusions are given in Section 7.

2. Model Equations

The governing equations are three-dimensional time-dependent MHD equations written in Cartesian coordinates as follows:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}+\frac{\partial \mathbf{H}}{\partial z}-\frac{\partial \mathbf{F}_v}{\partial x}-\frac{\partial \mathbf{G}_v}{\partial y}-\frac{\partial \mathbf{H}_v}{\partial z}=\mathbf{S}, \tag{1}$$

where

$$\mathbf{U}\equiv(U_m)=(\rho,\rho\mathbf{v},p,\mathbf{B})=(\rho,\rho v_x,\rho v_y,\rho v_z,p,B_x,B_y,B_z)^{\mathrm{T}}, \tag{2}$$

$$\mathbf{F}\equiv(F_m)=\begin{pmatrix}\rho v_x\\ \rho v_x^2+p_0-B_x^2\\ \rho v_x v_y-B_xB_y\\ \rho v_x v_z-B_xB_z\\ \gamma p v_x\\ 0\\ v_xB_y-v_yB_x\\ v_xB_z-v_zB_x\end{pmatrix},\qquad \mathbf{G}\equiv(G_m)=\begin{pmatrix}\rho v_y\\ \rho v_y v_x-B_yB_x\\ \rho v_y^2+p_0-B_y^2\\ \rho v_y v_z-B_yB_z\\ \gamma p v_y\\ v_yB_x-v_xB_y\\ 0\\ v_yB_z-v_zB_y\end{pmatrix}, \tag{3}$$

$$\mathbf{H} \equiv (H_m) = \begin{pmatrix} \rho v_z \\ \rho v_z v_x - B_z B_x \\ \rho v_z v_y - B_z B_y \\ \rho v_z^2 + p_0 - B_z^2 \\ \gamma p v_z \\ v_z B_x - v_x B_z \\ v_z B_y - v_y B_z \\ 0 \end{pmatrix},$$

$$\mathbf{F}_\nu \equiv (F_{\nu m}) = \begin{pmatrix} 0 \\ \nu\rho \frac{\partial v_x}{\partial x} \\ \nu\rho \frac{\partial v_y}{\partial x} \\ \nu\rho \frac{\partial v_z}{\partial x} \\ 0 \\ \mu \nabla \cdot \mathbf{B} \\ \eta j_z \\ -\eta j_y \end{pmatrix}, \qquad \mathbf{G}_\nu \equiv (G_{\nu m}) = \begin{pmatrix} 0 \\ \nu\rho \frac{\partial v_x}{\partial y} \\ \nu\rho \frac{\partial v_y}{\partial y} \\ \nu\rho \frac{\partial v_z}{\partial y} \\ 0 \\ -\eta j_z \\ \mu \nabla \cdot \mathbf{B} \\ \eta j_x \end{pmatrix},$$

$$\mathbf{H}_\nu \equiv (H_{\nu m}) = \begin{pmatrix} 0 \\ \nu\rho \frac{\partial v_x}{\partial z} \\ \nu\rho \frac{\partial v_y}{\partial z} \\ \nu\rho \frac{\partial v_z}{\partial z} \\ 0 \\ \eta j_y \\ -\eta j_x \\ \mu \nabla \cdot \mathbf{B} \end{pmatrix}, \tag{4}$$

$$\begin{aligned} \mathbf{S} \equiv (S_m) = {} & [0, \mathbf{F}, (\gamma - 1)(\mathbf{v} \cdot \nabla p + \eta \mathbf{j} \cdot \mathbf{j} + \nu\rho \nabla \mathbf{v} \colon \nabla \mathbf{v}), \mathbf{0}]^{\mathrm{T}} \\ & - \nabla \cdot \mathbf{B}(0, \mathbf{B}, 0, \mathbf{v})^{\mathrm{T}}, \end{aligned} \tag{5}$$

with the total pressure $p_0 = p + (B_x^2 + B_y^2 + B_z^2)/2$. Here, $\rho, \mathbf{v}, p$ and $\mathbf{B}$ are the mass density, plasma velocity, gas pressure, and magnetic field, respectively. (j_x, j_y, j_z) are the components of the electric current $\mathbf{j} = \nabla \times \mathbf{B}$. $\mathbf{F}$ is the external force exerted on the plasma. ν is the kinematic plasma viscosity, and η is the electrical resistivity. γ is the ratio of the specific heats. $\mathbf{r}$ is the position vector, and t is the time. This model becomes ideal MHD for $\nu = 0, \eta = 0$, while it is resistive MHD when ν and η are finite. $U_m, F_m, G_m, H_m, F_{\nu m}, G_{\nu m}, H_{\nu m}$, and S_m $(m = 1, 2, \ldots, 8)$ are the components of the flux vectors $\mathbf{U}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{F}_\nu, \mathbf{G}_\nu, \mathbf{H}_\nu$, and $\mathbf{S}$, respectively. For completeness, the Jacobian matrices for the fluxes $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ are given in the Appendix.

At the same time, the Powell source terms $-\nabla \cdot \mathbf{B}(0, \mathbf{B}, 0, \mathbf{v})^{\mathrm{T}}$ are added to deal with the divergence of the magnetic field (Powell *et al.*, 1999), and we also use the diffusive control of $\nabla \cdot \mathbf{B}$ (Marder, 1987; Dedner *et al.*, 2002; Tóth *et al.*, 2006) by adding the term $\mu \nabla \cdot \mathbf{B}$ to viscous fluxes $\mathbf{F}_\nu, \mathbf{G}_\nu, \mathbf{H}_\nu$, which is equivalent to adding a diffusive source term $\nabla(\mu\nabla\cdot\mathbf{B})$ to the magnetic induction equation. In this way, the diffusive control term is a component of the viscous fluxes, which can be treated conveniently by the CESE method (Section 3). Here μ is a spatially varying coefficient properly chosen to maximize the diffusion without introducing a numerical instability.

The primitive variables ρ, $\mathbf{v}$, p, $\mathbf{B}$, current density $\mathbf{j}$, position vector $\mathbf{r}$ and time t in Equation (1) have been normalized by their corresponding characteristic values ρ_0, v_0, B_0^2/μ_0, B_0, j_0, L_0 and L_0/v_0, where ρ_0, B_0, L_0 are three properly chosen basic quantities used for nondimensionalization, μ_0 is the permeability of a vacuum, $v_0 = B_0/\sqrt{\mu_0\rho_0}$ is the Alfv́en speed, and $j_0 = B_0/(\mu_0 L_0)$ is the characteristic current density. In the same way, the viscosity ν and resistivity η are normalized by $L_0 v_0$ and $\mu_0 L_0 v_0$.

3. Improved CESE Method

The above MHD equation (1) can be rewritten as

$$\frac{\partial U_m}{\partial t} + \frac{\partial F_m}{\partial x} + \frac{\partial G_m}{\partial y} + \frac{\partial H_m}{\partial z} - \frac{\partial F_{\nu m}}{\partial x} - \frac{\partial G_{\nu m}}{\partial y} - \frac{\partial H_{\nu m}}{\partial z} = S_m. \tag{6}$$

Here, we can write $F_{\nu m}$, $G_{\nu m}$, $H_{\nu m}$ and S_m as functions of $\mathbf{U}$, its spatial gradients of $\mathbf{U}$ (*i.e.*, $\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z$) and the spatial vector $\mathbf{r}$, since

$$\nabla\cdot\mathbf{B} = U_{6x} + U_{7y} + U_{8z}, \tag{7}$$

$$\mathbf{j} = \nabla\times\mathbf{B} = (U_{8x} - U_{7z}, U_{6z} - U_{8x}, U_{7x} - U_{6y}). \tag{8}$$

For convenience, we first describe our improvement of the CESE scheme in the two-dimensional case and then extend it to the three-dimensional case directly. By using Gauss's divergence theorem in a three-dimensional Euclidean space E_3 (whose coordinates are defined as $x_1 = x$, $x_2 = y$, $x_3 = t$) in Equation (6), we have

$$\oint_{S(V)} \mathbf{h}_m \cdot \mathrm{d}\mathbf{s} = \int_V S_m \,\mathrm{d}V \tag{9}$$

where $\mathbf{h}_m = (F_m - F_{\nu m}, G_m - G_{\nu m}, U_m)$ is the space – time flux vector, $S(V)$ is the boundary of the space – time region V in E_3, and $\mathbf{h}_m\cdot\mathrm{d}\mathbf{s}$ is the space – time flux leaving through the surface element $\mathrm{d}\mathbf{s}$.

For Cartesian coordinates with a uniform mesh, a much simple CESE scheme can be used. In Figure 1, the spatial domain is divided into quadrilateral meshes. Let i, j and n be the indices for x, y and t, respectively. Let Ω denote the set of mesh points (i, j, n) in E_3 with $n = 0, \pm1/2, \pm1, \pm3/2, \ldots, i = n \pm 1/2, n \pm 3/2, n \pm 5/2, \ldots, j = n \pm 1/2,$ $n \pm 3/2, n \pm 5/2, \ldots$.

Figure 1 shows the projection of the mesh points onto the $x - y$ plane. The points marked by an open circle like G are at the time level n; and points marked by a filled circle such as A, B, C, and D are at the time level $n - 1/2$. The time interval is $\Delta t/2$.

For every mesh point in Ω, there exists a solution element SE(i, j, n). As shown in Figure 2, the solution element SE(i, j, n) associated with G′ is defined as the union of three planes, A′B′C′D′, W″E″EW, and S″N″NS, which intersect at G′ and are perpendicular to each other. Similarly, associated with the points A, B, C, and D, there are four solution elements: SE$(i + p/2, j + q/2, n - 1/2)$ for all $p, q = -1, 1$.

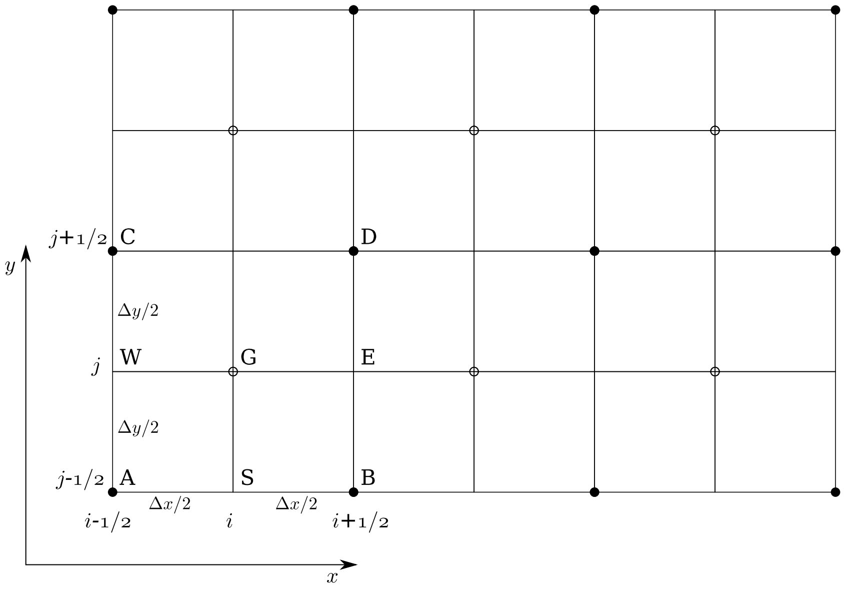
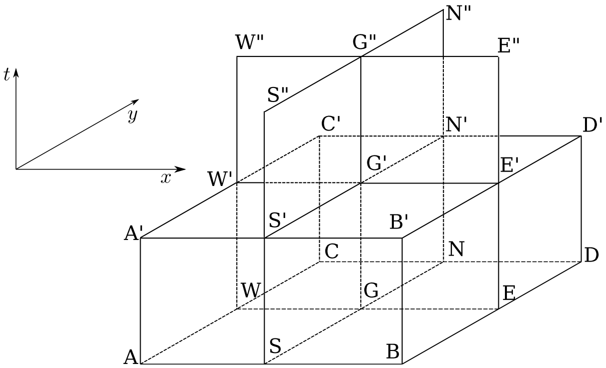


Figure 1 The CESE scheme in two dimensions; the projection of the mesh points in E_3 onto the $x - y$ plane.

Figure 2 The CESE scheme in two dimensions; definitions of CE and SE.



As usual in the CESE method (Zhang, Yu, and Chang, 2002; Zhang *et al.*, 2006; Feng, Hu, and Wei, 2006; Feng, Zhou, and Wu, 2007), the space – time conservation element (CE) of G′, ABCDA′B′C′D′, can be formed with surfaces associated with SE(i, j, n) and surfaces associated with the four SEs at the time level $n - 1/2$ (Figure 2).

Although similar to the definition of conservation elements (CEs) in Zhang *et al.* (2006), note that here the surfaces of the CE are parallel to the coordinate surfaces, which makes the integration of fluxes simpler and more convenient, especially in the three-dimensional case, because on any surface of the CE, the normal vector is along the coordinate axis perpendicular to the surface, and thus only one component of the total flux **h** is needed when calculating the space – time flux leaving through this surface.

Inside any SE(i, j, n), the distributions of the variables $U_m(x, y, t; i, j, n)$, $F_m(x, y, t;$ $i, j, n)$ and $G_m(x, y, t; i, j, n)$ are approximated by the first-order Taylor expansion at point

(i, j, n),

$$\begin{aligned} U_m(x, y, t; i, j, n) &= (U_m)^n_{i,j} + (U_{mx})^n_{i,j}\, \mathrm{d}x + (U_{my})^n_{i,j}\, \mathrm{d}y + (U_{mt})^n_{i,j}\, \mathrm{d}t, \\ F_m(x, y, t; i, j, n) &= (F_m)^n_{i,j} + (F_{mx})^n_{i,j}\, \mathrm{d}x + (F_{my})^n_{i,j}\, \mathrm{d}y + (F_{mt})^n_{i,j}\, \mathrm{d}t, \\ G_m(x, y, t; i, j, n) &= (G_m)^n_{i,j} + (G_{mx})^n_{i,j}\, \mathrm{d}x + (G_{my})^n_{i,j}\, \mathrm{d}y + (G_{mt})^n_{i,j}\, \mathrm{d}t, \end{aligned} \tag{10}$$

with

$$\begin{aligned} (\mathbf{F}_x)^n_{i,j} &= \frac{\partial \mathbf{F}}{\partial \mathbf{U}}(\mathbf{U}_x)^n_{i,j}, & (\mathbf{F}_y)^n_{i,j} &= \frac{\partial \mathbf{F}}{\partial \mathbf{U}}(\mathbf{U}_y)^n_{i,j}, & (\mathbf{F}_t)^n_{i,j} &= \frac{\partial \mathbf{F}}{\partial \mathbf{U}}(\mathbf{U}_t)^n_{i,j}, \\ (\mathbf{G}_x)^n_{i,j} &= \frac{\partial \mathbf{G}}{\partial \mathbf{U}}(\mathbf{U}_x)^n_{i,j}, & (\mathbf{G}_y)^n_{i,j} &= \frac{\partial \mathbf{G}}{\partial \mathbf{U}}(\mathbf{U}_y)^n_{i,j}, & (\mathbf{G}_t)^n_{i,j} &= \frac{\partial \mathbf{G}}{\partial \mathbf{U}}(\mathbf{U}_t)^n_{i,j}, \end{aligned} \tag{11}$$

where $\mathrm{d}x = x - x_i$, $\mathrm{d}y = y - y_j$, $\mathrm{d}t = t - t_n$. The viscous fluxes can be set as constant in the SE (Feng, Hu, and Wei, 2006) because they consist mainly of the first-order derivatives of the unknown variables (which are assumed to be linear in the SE), *i.e.*,

$$\begin{aligned} F_{vm}(x, y, t; i, j, n) &= (F_{vm})^n_{i,j}, \\ G_{vm}(x, y, t; i, j, n) &= (G_{vm})^n_{i,j}. \end{aligned} \tag{12}$$

Substituting Equations (10) and (12) into Equation (6), we can obtain

$$(U_{mt})^n_{i,j} = -(F_{mx})^n_{i,j} - (G_{my})^n_{i,j} + (S_m)^n_{i,j}. \tag{13}$$

Note that Equations (6) and (10) – (13) imply that the independent solution variables at (i, j, n) are merely $(U_m)^n_{i,j}$, $(U_{mx})^n_{i,j}$, $(U_{my})^n_{i,j}$, and $(U_{mz})^n_{i,j}$.

The space – time flux conservation of Equation (9) can be approximated by its discrete counterpart (Figure 2),

$$\oint_{S(\mathrm{CE(G')})} \mathbf{h}_m \cdot \mathrm{d}\mathbf{s} = \oint_{S(\mathrm{ABCDA'B'C'D'})} \mathbf{h}_m \cdot \mathrm{d}\mathbf{s} = \int_{V(\mathrm{CE(G')})} S_m\, \mathrm{d}V \tag{14}$$

and with the aid of Equations (10) and (12), we have

$$\begin{aligned} &(U_m)^n_{i,j} - \frac{\Delta t}{2}(S_m)^n_{i,j} \\ &\quad = \frac{1}{\Delta x \Delta y} \sum_{p=-1,1;q=-1,1} \frac{\Delta x \Delta y}{4}\left[(U_m)^{n-1/2}_{ip,jq} + (U_{mx})^{n-1/2}_{ip,jq}\left(-p\frac{\Delta x}{4}\right)\right. \\ &\quad\quad \left. + (U_{my})^{n-1/2}_{ip,jq}\left(-q\frac{\Delta y}{4}\right)\right] \\ &\quad\quad - p\frac{\Delta y \Delta t}{4}\left[(F_m - F_{vm})^{n-1/2}_{ip,jq} + (F_{my})^{n-1/2}_{ip,jq}\left(-q\frac{\Delta y}{4}\right) + (F_{mt})^{n-1/2}_{ip,jq}\left(\frac{\Delta t}{4}\right)\right] \\ &\quad\quad - q\frac{\Delta x \Delta t}{4}\left[(G_m - G_{vm})^{n-1/2}_{ip,jq} + (G_{mx})^{n-1/2}_{ip,jq}\left(-p\frac{\Delta x}{4}\right)\right. \\ &\quad\quad \left. + (G_{mt})^{n-1/2}_{ip,jq}\left(\frac{\Delta t}{4}\right)\right] \end{aligned} \tag{15}$$

where $ip = i + p/2$, $jq = j + q/2$ for short. By solving the preceding nonlinear equations using Newton's method (Feng, Hu, and Wei, 2006; Feng, Zhou, and Wu, 2007), the values at time level n, $(U_m)^n_{i,j}$ can be obtained from the solution variables of $n - 1/2$.

Before calculation of the spatial gradients of flow variables $(U_{mx})^n_{i,j}$ and $(U_{my})^n_{i,j}$, we use a Taylor series expansion along the time axis from the time level $n - 1/2$ to obtain the variables at the points of A′, B′, C′ and D′. For example, at A′,

$$(U'_m)_{\mathrm{A}'} = (U_m)_{\mathrm{A}} + \frac{\Delta t}{2}(U_{mt})_{\mathrm{A}}. \tag{16}$$

Then, we divide the square plane A′B′C′D′ into four triangles: A′B′G′, B′D′G′, D′C′G′ and A′C′G′. In each triangle, we finite-difference U_m at the three vertices to obtain the flow variable gradient $(U_{mx})^n_{i,j}$ and $(U_{my})^n_{i,j}$ at point G′. After obtaining all four sets of flow variable gradients, an α type of reweighting procedure is applied to get the final variable gradients $(U_{mx})^n_{i,j}$, $(U_{my})^n_{i,j}$, with $\alpha = 1$ or 2 as a prescribed constant for shock capturing, and 0 for flows without shocks (see the details described in Zhang, Yu, and Chang, 2002; Zhang *et al.*, 2006; Feng, Hu, and Wei, 2006; and Feng, Zhou, and Wu, 2007).

Similarly, the three-dimensional case follows the two-dimensional case. Now, for instance, the SE(i, j, k, n) associated with G′ is defined as the union of four hyperplanes ($x-y-z$, $y-z-t$, $x-z-t$ and $x-y-t$ hyperplanes) which intersect at G′ in a four-dimensional Euclidean space (whose coordinates are set as $x_1 = x$, $x_2 = y$, $x_3 = z$, $x_4 = t$). Similarly, the CE of G′ is formed by hypersurfaces associated with SE(i, j, k, n) of G′ and eight SEs of $\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \ldots, \mathrm{A}_8$ at the time level $n - 1/2$, which are SE$(i + p/2, j + q/2, k + r/2, n - 1/2)$, for all $p, q, r = -1, 1$ (see Figure 3). Applying the space – time flux conservation law in CE (of G′) of the four-dimensional Euclidean space, we have

$$\begin{aligned}
(U_m)^n_{i,j,k} - \frac{\Delta t}{2}(S_m)^n_{i,j,k} = \frac{1}{\Delta x \Delta y \Delta z} \sum_{p=-1,1;q=-1,1;r=-1,1} & \frac{\Delta x \Delta y \Delta z}{8}\Bigg[(U_m)^{n-1/2}_{ip,jq,kr} \\
&+ (U_{mx})^{n-1/2}_{ip,jq,kr}\left(-p\frac{\Delta x}{4}\right) \\
&+ (U_{my})^{n-1/2}_{ip,jq,kr}\left(-q\frac{\Delta y}{4}\right) + (U_{mz})^{n-1/2}_{ip,jq,kr}\left(-r\frac{\Delta z}{4}\right)\Bigg] \\
&- p\frac{\Delta y \Delta z \Delta t}{8}\Bigg[(F_m - F_{vm})^{n-1/2}_{ip,jq,kr} + (F_{my})^{n-1/2}_{ip,jq,kr}\left(-q\frac{\Delta y}{4}\right) \\
&+ (F_{mz})^{n-1/2}_{ip,jq,kr}\left(-r\frac{\Delta z}{4}\right) + (F_{mt})^{n-1/2}_{ip,jq,kr}\left(\frac{\Delta t}{4}\right)\Bigg] \\
&- q\frac{\Delta x \Delta z \Delta t}{8}\Bigg[(G_m - G_{vm})^{n-1/2}_{ip,jq,kr} + (G_{mx})^{n-1/2}_{ip,jq,kr}\left(-p\frac{\Delta x}{4}\right) \\
&+ (G_{mz})^{n-1/2}_{ip,jq,kr}\left(-r\frac{\Delta z}{4}\right) + (G_{mt})^{n-1/2}_{ip,jq,kr}\left(\frac{\Delta t}{4}\right)\Bigg] \\
&- r\frac{\Delta x \Delta y \Delta t}{8}\Bigg[(H_m - H_{vm})^{n-1/2}_{ip,jq,kr} + (H_{mx})^{n-1/2}_{ip,jq,kr}\left(-p\frac{\Delta x}{4}\right) \\
&+ (H_{my})^{n-1/2}_{ip,jq,kr}\left(-q\frac{\Delta y}{4}\right) + (H_{mt})^{n-1/2}_{ip,jq,kr}\left(\frac{\Delta t}{4}\right)\Bigg]
\end{aligned} \tag{17}$$

where $ip = i + p/2$, $jq = j + q/2$ and $kr = k + r/2$ for short. When calculating the gradients of the variables, the values of eight vertices $\mathrm{A}'_1, \mathrm{A}'_2, \ldots, \mathrm{A}'_8$ are obtained similarly

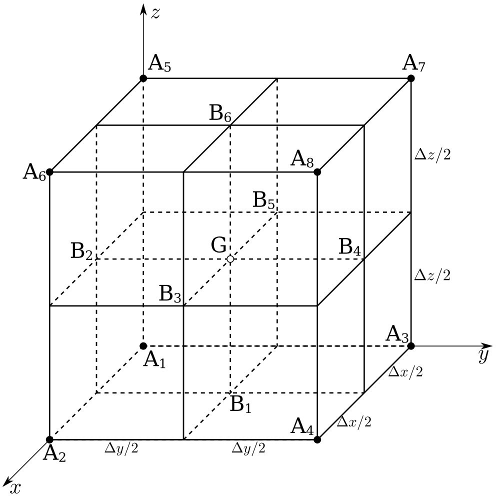


Figure 3 The CESE scheme in three dimensions; the projection of the mesh points in E_4 onto the Cartesian (x, y, z)-space.

by Taylor series expansion along the time axis from the time level $n - 1/2$. Then eight tetrahedrons can be determined by G and three points from the eight vertices surrounding G. By finite-difference of U_m at four vertices of each tetrahedron respectively, eight sets of U_{mx}, U_{my}, U_{mz} are obtained. Then the same procedure of reweighting is used to get the final U_{mx}, U_{my}, U_{mz} at point G′. For clarity, here we specifically denote the eight tetrahedrons $A_2A_3A_5G$, $A_4A_6A_1G$, $A_1A_4A_7G$, $A_2A_3A_8G$, $A_1A_6A_7G$, $A_2A_5A_8G$, $A_3A_5A_8G$, and $A_4A_6A_7G$ (Figure 3), where the differences are applied.

If using the present CESE scheme directly on the grid system of the AMR algorithm, it becomes difficult to transfer (prolong or restrict) the solution variables between meshes of different refinement levels, as these operations must be done on data of the same time level, while the present scheme is a space – time staggering stencil, which turns out to be a obstacle. Here we use an improved version of the present method for convenience: we only reserve the solution variables on the time level n of integer numbers, while the intermediate time level $n + 1/2$ can be seen as a prediction step as in the predictor-corrector type time integration and the solution variables on this prediction level are calculated temporarily; so a full time step here consists of two half steps which are advanced from n to $n + 1/2$ (from the points marked by filled circles to points marked by open circles in Figure 1) and from $n + 1/2$ to $n + 1$ (from the open circles to the filled circles), and as seen from Figure 1, the full time step from time n to $n + 1$ for point D is actually associated with eight points surrounding it at time n (marked by filled circle) and itself, which is a nine points stencil in two dimensions. Similarly, there are 27 points involved in one full time step in the three-dimensional case. In this way, all the mesh points in the scheme are at the same time level after one full-time step marching.

4. Modified CESE Method in General Curvilinear Coordinates

The MHD equation (Equation (1)) can be transformed from the physical space (x, y, z) to the reference space (ξ, η, ζ) by specifying the nonsingular mapping

$$x = x(\xi, \eta, \zeta); \qquad y = y(\xi, \eta, \zeta); \qquad z = z(\xi, \eta, \zeta), \tag{18}$$

and the Jacobian matrix

$$\begin{aligned}
\mathbf{J} &= \frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = \begin{pmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{pmatrix}, \\
\mathbf{J}^{-1} &= \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix}.
\end{aligned} \tag{19}$$

The reference space (ξ, η, ζ) is also called the computational space, and the mapping equation (Equation (18)) represents the transformation between the two spatial spaces. The elements of the Jacobian matrix are called metrics and may be computed analytically or numerically.

Now the corresponding transformed MHD equations in the reference space (ξ, η, ζ) read

$$\frac{\partial \hat{\mathbf{U}}}{\partial t} + \frac{\partial \hat{\mathbf{F}}}{\partial \xi} + \frac{\partial \hat{\mathbf{G}}}{\partial \eta} + \frac{\partial \hat{\mathbf{H}}}{\partial \zeta} - \frac{\partial \hat{\mathbf{F}}_\nu}{\partial \xi} - \frac{\partial \hat{\mathbf{G}}_\nu}{\partial \eta} - \frac{\partial \hat{\mathbf{H}}_\nu}{\partial \zeta} = \hat{\mathbf{S}}, \tag{20}$$

where

$$\begin{cases}
\hat{\mathbf{U}} = J\mathbf{U}, \\
\hat{\mathbf{F}} = J(\mathbf{F}\xi_x + \mathbf{G}\xi_y + \mathbf{H}\xi_z), \\
\hat{\mathbf{G}} = J(\mathbf{F}\eta_x + \mathbf{G}\eta_y + \mathbf{H}\eta_z), \\
\hat{\mathbf{H}} = J(\mathbf{F}\zeta_x + \mathbf{G}\zeta_y + \mathbf{H}\zeta_z), \\
\hat{\mathbf{F}}_\nu = J(\mathbf{F}_\nu\xi_x + \mathbf{G}_\nu\xi_y + \mathbf{H}_\nu\xi_z), \\
\hat{\mathbf{G}}_\nu = J(\mathbf{F}_\nu\eta_x + \mathbf{G}_\nu\eta_y + \mathbf{H}_\nu\eta_z), \\
\hat{\mathbf{H}}_\nu = J(\mathbf{F}_\nu\zeta_x + \mathbf{G}_\nu\zeta_y + \mathbf{H}_\nu\zeta_z), \\
\hat{\mathbf{S}} = J\mathbf{S}.
\end{cases} \tag{21}$$

Here J is the Jacobian determinant of the transformation $J = |\mathbf{J}|$, which is interpreted as the ratio of the cell volume in the reference space to that of the physical space. Note that Equation (20) is still written in conservation form, just like the original Equation (1). The transformed equations are also said to be of strong conservation form in the reference space (Viviand, 1975; Vinokur, 1974).

Now the CESE method can be built in the reference space (ξ, η, ζ) to solve the transformed MHD Equation (20): let $x_1 = \xi$, $x_2 = \eta$, $x_3 = \zeta$, $x_4 = t$ be the coordinates of the four-dimensional Euclidean space E_4; then the space – time flux vector is defined as $\hat{\mathbf{h}} = (\hat{\mathbf{F}} - \hat{\mathbf{F}}_\nu$, $\hat{\mathbf{G}} - \hat{\mathbf{G}}_\nu$, $\hat{\mathbf{H}} - \hat{\mathbf{H}}_\nu$, $\hat{\mathbf{U}})$, and by using Gauss's divergence theorem in this E_4 we still have

$$\oint_{S(V)} \hat{\mathbf{h}} \cdot \mathrm{d}\mathbf{s} = \int_V \hat{S}\,\mathrm{d}V \tag{22}$$

in the reference space because of the strong conservation form of Equation (20).

Similar to the case in the physical space (Section 3), a uniform and rectangular grid can be used in the reference space. Now $(\hat{\mathbf{U}}, \hat{\mathbf{U}}_\xi, \hat{\mathbf{U}}_\eta, \hat{\mathbf{U}}_\zeta)$ are introduced as the independent solution variables. In order to get the corresponding first-order Taylor expansions (Equation (10)), the derivatives of the flux vectors such as $\hat{\mathbf{F}}$, $\hat{\mathbf{G}}$, $\hat{\mathbf{H}}$ should be calculated, but this requires a little more work. For instance, let's write $\hat{\mathbf{F}}$ as

$$\hat{\mathbf{F}} = \alpha\mathbf{F} + \beta\mathbf{G} + \gamma\mathbf{H} \tag{23}$$

with $\alpha = J\xi_x$, $\beta = J\xi_y$ and $\gamma = J\xi_z$, all of which are known functions of (ξ, η, ζ). Then, by the chain rule, for $\hat{\mathbf{F}}_\xi$ and $\hat{\mathbf{F}}_t$ we have

$$\hat{\mathbf{F}}_\xi = \alpha_\xi\mathbf{F} + \beta_\xi\mathbf{G} + \gamma_\xi\mathbf{H} + \left(\alpha\frac{\partial\mathbf{F}}{\partial\mathbf{U}} + \beta\frac{\partial\mathbf{G}}{\partial\mathbf{U}} + \gamma\frac{\partial\mathbf{H}}{\partial\mathbf{U}}\right)\mathbf{U}_\xi, \tag{24}$$

$$\hat{\mathbf{F}}_t = \left(\alpha\frac{\partial\mathbf{F}}{\partial\mathbf{U}} + \beta\frac{\partial\mathbf{G}}{\partial\mathbf{U}} + \gamma\frac{\partial\mathbf{H}}{\partial\mathbf{U}}\right)\mathbf{U}_t, \tag{25}$$

where $\mathbf{U}_\xi = (\hat{\mathbf{U}}_\xi - J_\xi\mathbf{U})/J$, $\mathbf{U}_t = \hat{\mathbf{U}}_t/J$. In the same way, all the other derivatives of the flux vectors can be derived. With these preparations, the CESE method in curvilinear coordinates follows the same steps as those for the CESE method in Cartesian coordinates of the physical space in Section 3.

For convenience, we will call the solving variables $(\hat{\mathbf{U}}, \hat{\mathbf{U}}_\xi, \hat{\mathbf{U}}_\eta, \hat{\mathbf{U}}_\zeta)$ in the reference space *the reference solution variables,* and $(\mathbf{U}, \mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z)$ *the physical solution variables*. Since the conversions between the reference solution variables and the physical solution variables are frequently used, for instance in initial and boundary conditions, we address all the relations here. From the physical solution variables to the reference solution variables, we obtain

$$\begin{cases} \hat{\mathbf{U}} = J\mathbf{U}, \\ \mathbf{U}_\xi = \mathbf{U}_x x_\xi + \mathbf{U}_y y_\xi + \mathbf{U}_z z_\xi, \\ \mathbf{U}_\eta = \mathbf{U}_x x_\eta + \mathbf{U}_y y_\eta + \mathbf{U}_z z_\eta, \\ \mathbf{U}_\zeta = \mathbf{U}_x x_\zeta + \mathbf{U}_y y_\zeta + \mathbf{U}_z z_\zeta, \\ \hat{\mathbf{U}}_\xi = J_\xi\mathbf{U} + J\mathbf{U}_\xi, \\ \hat{\mathbf{U}}_\eta = J_\eta\mathbf{U} + J\mathbf{U}_\eta, \\ \hat{\mathbf{U}}_\zeta = J_\zeta\mathbf{U} + J\mathbf{U}_\zeta. \end{cases} \tag{26}$$

From the reference solution variables to the physical solution variables, we have

$$\begin{cases} \mathbf{U} = \hat{\mathbf{U}}/J, \\ \mathbf{U}_\xi = (\hat{\mathbf{U}}_\xi - J_\xi\mathbf{U})/J, \\ \mathbf{U}_\eta = (\hat{\mathbf{U}}_\eta - J_\eta\mathbf{U})/J, \\ \mathbf{U}_\zeta = (\hat{\mathbf{U}}_\zeta - J_\zeta\mathbf{U})/J, \\ \mathbf{U}_x = \mathbf{U}_\xi\xi_x + \mathbf{U}_\eta\eta_x + \mathbf{U}_\zeta\zeta_x, \\ \mathbf{U}_y = \mathbf{U}_\xi\xi_y + \mathbf{U}_\eta\eta_y + \mathbf{U}_\zeta\zeta_y, \\ \mathbf{U}_z = \mathbf{U}_\xi\xi_z + \mathbf{U}_\eta\eta_z + \mathbf{U}_\zeta\zeta_z. \end{cases} \tag{27}$$

5. AMR Implementation

In this section, we will describe the AMR implementation of the CESE method in curvilinear coordinates with the PARAMESH package, the refinement criteria, and the variable timestep algorithm.

5.1. CESE Method in PARAMESH

As the MHD equations are solved in the reference space (ξ, η, ζ), the grid is still in rectangular coordinates, which makes the implementation of AMR much more convenient and straightforward. Most of the default operations in the package like prolongation and restriction can be used without modification.

In detail, *i*) we specify all the solution variables $(\hat{U}, \hat{U}_\xi, \hat{U}_\eta, \hat{U}_\zeta)$ at cell center, with the number of variables $\texttt{nvar} = 4 \times 8 = 32$ defined in PARAMESH. The prolongation and restriction are also carried out on the gradients of the variables, which makes our scheme much different from the AMR implementations on the finite volume scheme. *ii*) The blocks are set to consist of $6 \times 6 \times 6$ or $8 \times 8 \times 8$ cells typically, and one layer of guard cells containing diagonal elements is specified because the CESE method associates three mesh points in all the three directions. *iii*) The linear interpolation is used when prolonging the data from parent blocks to their newborn child blocks, and the average is used when restricting data on child blocks back to their parents. The same interpolation is used when filling guard cells at block boundaries next to less refined neighbor blocks.

5.2. Refinement Criteria

Although several approaches are possible, in this work, the refining and coarsening of blocks are guided by using multiple physics-based refinement criteria (Powell, Roe, and Quirk, 1993; De Zeeuw, 1993; Powell *et al.*, 1999; Groth *et al.*, 2000). Similar to Powell *et al.* (1999) and Linde (1998), the refinement criteria are defined, according to a combination of curl and divergence of the velocity, and curl of the magnetic field

$$\begin{aligned} \chi_c &= \sqrt{V}\frac{|\nabla \cdot \mathbf{v}|}{|\mathbf{v}| + \epsilon a}, \\ \chi_r &= \sqrt{V}\frac{|\nabla \times \mathbf{v}|}{|\mathbf{v}| + \epsilon a}, \\ \chi_c &= \sqrt{V}\frac{|\nabla \times \mathbf{B}|}{|\mathbf{B}| + \epsilon\sqrt{p}}, \end{aligned} \tag{28}$$

where a is the plasma sound speed and p is its thermal pressure. The factor $\epsilon \ll 1$ is introduced into Equation (28) to eliminate the concern when either $\mathbf{v}$ or $\mathbf{B}$ is equal to zero. These criteria can detect all distinctive features in the plasma medium. Specifically, the curl of the velocity can be used to find shear layers, and the divergence of the velocity can be used to find shocks, while the curl of the magnetic field (the current density) is used to find electric current surfaces. The factor $\sqrt{V}$ is the length of the cell to the power of $3/2$, and a power greater than one is needed to weaken a possible divergence of the numerators in the criteria, which allows the scheme to resolve smooth regions of the flows as well as discontinuous ones (De Zeeuw, 1993).

If any of the maxima of these criteria in one block is greater than the threshold for refinement, this block is flagged to be refined, while if all of the maxima of these criteria in one block are less than the threshold for coarsening, this block is flagged to be coarsened. In order to set the thresholds, the standard deviation about zero is computed for χ (De Zeeuw, 1993):

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} \chi_i^2}{N}} \tag{29}$$

with i visiting all of the cells in the computation domain. Multiplying each σ by different, properly chosen factors gives the thresholds.

5.3. Variable Timestep

When there is significant variation in spatial resolution within the computational domain, the timestep computed using a fixed Courant number will probably also vary significantly (MacNeice *et al.*, 2000). By using a uniform finest timestep on all the grids (*i.e.*, the blocks in PARAMESH), the numerical diffusion effects on the much coarser blocks will become significant, especially in the case of the CESE method. At the same time, using the finest timestep globally will add a great overhead on the computation resource because of unnecessary fine timesteps of solution-advancing on these blocks.

The variable timestep is introduced here to overcome these problems. We use different timesteps on different blocks, and the timesteps are directly proportional to the physical scale irrespective of the refinement level, because the refinement level in the reference space does not represent the spatial resolution in the physical space. For instance, in spherical coordinates with the same refinement level, a block near the origin will be much smaller than a block far away from the origin.

In particular, we first define the timestep Δt of block b as a monotonic function of the physical scale

$$\Delta t = \mathrm{CFL}\frac{\Delta L_b}{v_{\max}} \tag{30}$$

where $v_{\max}$ is the maximal wave speed in the entire computation domain, and ΔL_b is the minimal physical size of cells in block b; and then the timestep is adjusted as

$$\Delta t = 2^{\alpha}\Delta t_{\min}, \quad \alpha = \left[\log_2(\Delta t/\Delta t_{\min})\right] \tag{31}$$

with $\Delta t_{\min}$ as the minimal timestep of all the blocks. Namely, on the two neighboring blocks, the timestep on one block is either the same or a factor of an integer number larger (or smaller) than that on the other block. At the interface of timestep jump, linear time interpolation from the larger-timestep block is used to set guard cell values when needed on the block with the finer timestep. When the blocks with the biggest timestep are advanced one step, all the blocks must be synchronized before moving on.

A short description of our time-stepping algorithm is presented here. We grouped the blocks by their timesteps with each group of blocks having the same timestep. In Figure 4, for instance, the blocks are divided into three groups with timesteps of Δt_1, Δt_2 and Δt_3, respectively, and $\Delta t_1 = 2\Delta t_2 = 4\Delta t_3$. Correspondingly, the groups are named `group 1`, `group 2` and `group 3`. The color arrows represent one step of solution advance of a given group, and the groups with the same color can be advanced simultaneously. The black arrows denote the necessity of time interpolation of solution when guard cell filling. The grid will not be modified until all blocks have been advanced through the biggest timestep. We show the basic algorithm as follows:

```
INITIAL the timestep: for all leaf blocks
  U2=unk; U1=unk
maxstep = dt(1)/dt(ndtlevel)
DO step = 1,maxstep
    DO n = 1,ndtlevel
      IF( time(n) == time(ndtlevel) )then
```

```
         for all leaf blocks in group n
           U1 = U2
           ADVANCE U2 using CESE method
         time(n) = time(n)+dt(n)
      END IF
    END DO

    DO n = 1,ndtlevel
      ke = (time(n)-time(ndtlevel))/dt(n)
      TIME INTERPOLATION: for all leaf blocks in group n
        unk = ke*U1+(1-ke)*U2
    END DO

    make all blocks SYNCHRONIZED here
    GUARDCELL FILLING globally
    DO n = 1,ndtlevel
       IF( time(n) == time(ndtlevel) )then
          for all leaf blocks in group n
            U2 = unk
       END IF
    END DO
END DO
END the timestep: for all leaf blocks
  unk = U2
```

where `ndtlevel` is the number of timestep levels or the number of groups; *e.g.*, `ndtlevel=3` in Figure 4. `time(n)` is the timer of group `n`, `dt(n)` the timestep Δt of group `n`, and `ke` is a factor used for the linear interpolation of time; `unk` is the solution data-structure in PARAMESH while the copy of `unk`: `U1, U2` can be allocated and deallocated temporarily.

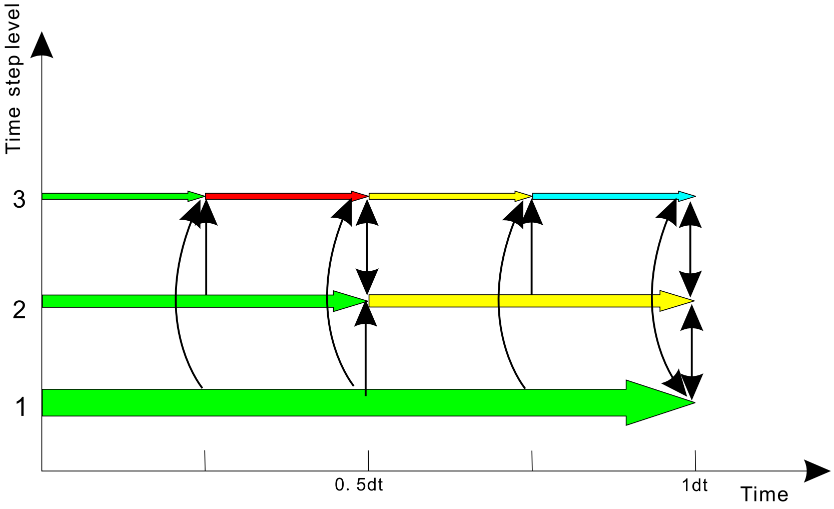


Figure 4 The variable timestep scheme: an example.

6. Numerical Tests

As far as we know, the CESE method has never yet been realized on the grid system of the AMR algorithm to solve the MHD equations or even the hydrodynamic (HD) equations. To test the feasibility of this totally new implementation, some benchmarks are validated here by reporting the numerical results using the AMR-CESE method in general curvilinear coordinates.

Section 6.1 shows the solution of a two-dimensional MHD vortex problem on AMR grids in Cartesian coordinates for validation of the CESE method with AMR. In Section 6.2, a fast magnetic reconnection problem is calculated to test the capability of modeling resistive MHD. Section 6.3 presents the results of a two-dimensional MHD blast wave problem in both Cartesian and polar coordinates to verify the CESE method in curvilinear coordinates. In Section 6.4, the three-dimensional version of the MHD blast wave problem is calculated in both Cartesian and spherical coordinates, and these problems are also carried out on an AMR grid. In these tests, we choose the adjustable constant $\alpha = 2$, and the coefficient μ in the diffusive control terms of $\nabla \cdot \mathbf{B}$ is set as

$$\mu = 0.2\frac{{L_{\mathrm{cell}}}^2}{\Delta t} \tag{32}$$

with variable values according to the physical size of the cell L_{cell} and the local timestep Δt. The factor 0.2 is chosen considering the stability constraint $\Delta t < \frac{L_{\mathrm{cell}}^2}{2\mu}$. The ratio of the specific heats $\gamma = 5/3$ and the kinematic plasma viscosity $\nu = 0$ are used in all these tests. For the vortex problem and the blast wave problems, the ideal MHD is modeled by setting $\eta = 0$.

6.1. MHD Vortex Problem

The MHD vortex problem, proposed by Orszag and Tang (1979), has been used in many papers as a two-dimensional numerical test for MHD codes (Jiang and Wu, 1999; Tóth, 2000; Zhang *et al.*, 2006). It is a particularly good test for capability examination of AMR implementations due to its own significant features of MHD turbulence; as time evolves, the structure of the field will become very complicated with shocks and discontinuous surfaces (Ziegler, 1998, 2003; Collins and Norman, 2004; Stone *et al.*, 2008).

As usual, the computational domain is set as $[0, 2\pi] \times [0, 2\pi]$, and the initial conditions are given as

$$\begin{aligned} \rho(x, y, 0) &= \gamma^2, \\ p(x, y, 0) &= \gamma, \\ \mathbf{v}(x, y, 0) &= (-\sin y, \sin x, 0), \\ \mathbf{B}(x, y, 0) &= (-\sin y, \sin 2x, 0). \end{aligned} \tag{33}$$

Periodic boundary conditions are imposed on boundaries in both the x and y directions. Initially, the domain is uniformly divided into 16×16 blocks with each block consisting of 6×6 cells, and three levels of refinements are used in this test such that the mesh size of the middle level is the same as that used by Jiang and Wu (1999) and Zhang *et al.* (2006), which makes our results comparable with theirs.

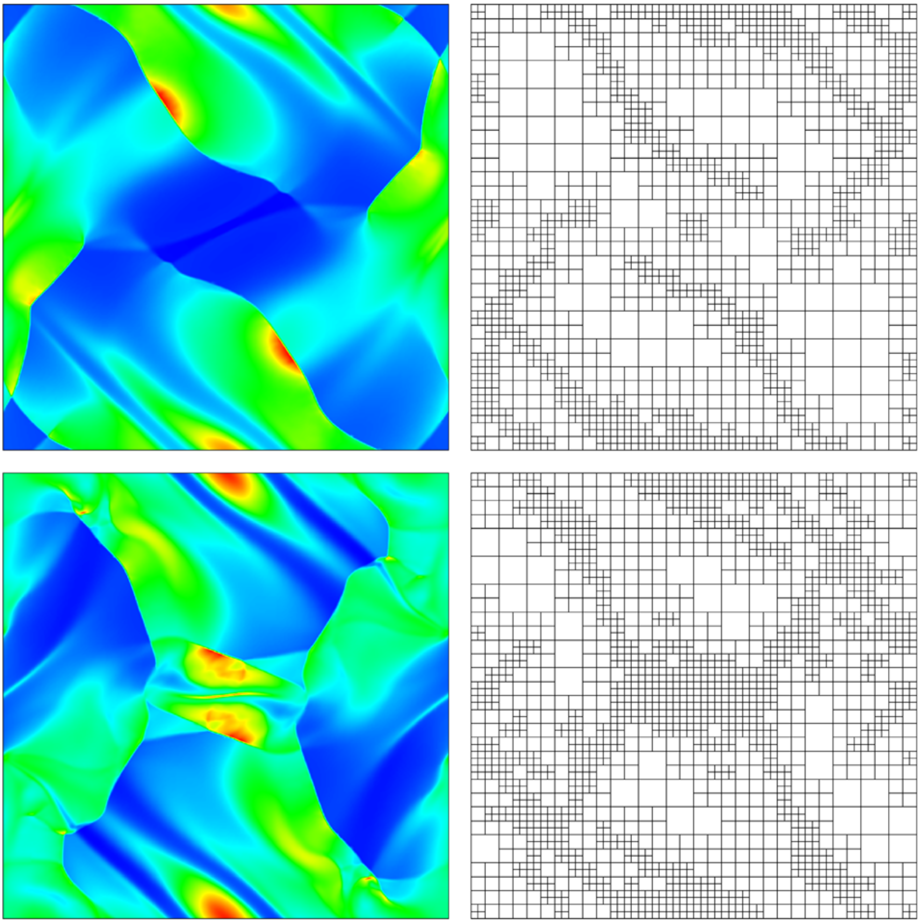


Figure 5 MHD vortex problem: color-coded images of pressure at $t = 2$ and 3 (left column) and the grid blocks at the corresponding time (right column, each cell here represents one grid block).

The left column in Figure 5 shows the pressure contours at $t = 2$ and 3, respectively. For the purpose of detailed comparison, the pressure profiles along the line of $y = 0.625\pi$ plotted in Zhang *et al.* (2006) is also presented in Figure 6 at the same time. These figures show that our results are almost the same as the former results (Jiang and Wu, 1999; Zhang *et al.*, 2006), and the shock profiles in our results are much sharper than the results obtained by Zhang *et al.* (2006), especially near $x = 0.4$ and 3.5 at $t = 2$ and near $x = 0.4$ and 4.4 at $t = 3$ because the finest meshes occur there. The right column in Figure 5 shows the structures of the blocks at the corresponding time. We can see immediately that the blocks of higher refinement levels are clustering near the shocks and discontinuities, which means that these features are well captured by the adaptive meshes.

Although the simple linear prolongation and restriction in the implementation of the AMR grid is considered here, the AMR-CESE method gives satisfactory results.

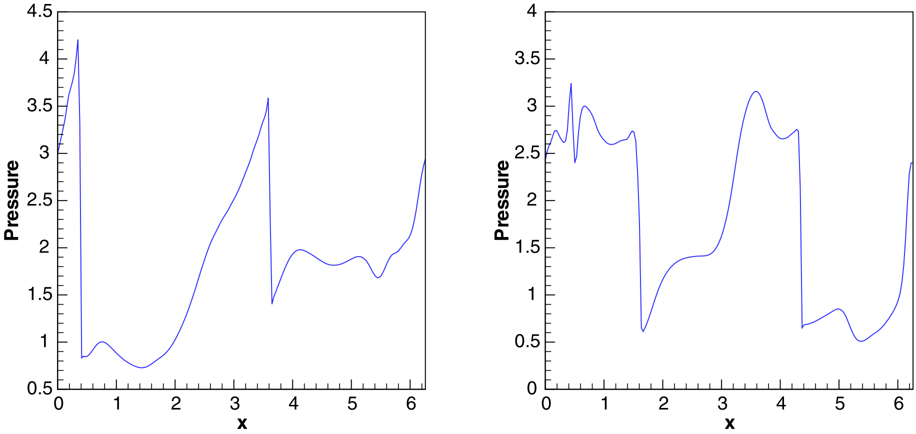


Figure 6 MHD vortex problem: pressure profile along line $y = 0.625\pi$ at $t = 2$ (left) and 3 (right).

6.2. Spontaneous Fast Reconnection Problem

Spontaneous fast reconnection is a new type of nonlinear instability in a long current sheet system (Ugai, 1999) representing the basic physical mechanism of magnetic reconnection responsible for catastrophic events in space plasmas such as solar flares, CMEs, and geomagnetic substorms. Here this important nonlinear instability problem is calculated by using the AMR-CESE method. As done by Ugai (1999) and Feng, Hu, and Wei (2006), we take the initial static equilibrium condition

$$\mathbf{B} = (B_x, 0, 0), \qquad p = \frac{(1+\beta) - B_x^2}{2}, \qquad \rho = \frac{2p}{1+\beta}, \qquad \mathbf{v} = \mathbf{0}, \tag{34}$$

$$B_x(x, y) = \begin{cases} \sin(\pi y/2) & \text{if } 0 < y < 1, \\ 1 & \text{if } 1 < y < Y_1, \\ \cos[(y - Y_1)\pi/1.2] & \text{if } Y_1 < y < Y_m (= Y_1 + 0.6), \\ 0 & \text{if } y > Y_m, \\ -B_x(x, -y) & \text{if } y < 0, \end{cases} \tag{35}$$

where the width of the magnetized region is chosen as $Y_1 = 3.6$ and the plasma β is 0.15. An initial disturbance is assumed by imposing a localized resistivity at the origin $(x, y) = (0, 0)$ in the form

$$\eta(\mathbf{r}) = 0.02\mathrm{e}^{-(x/1.1)^2-(y/1.1)^2} \tag{36}$$

for the initial time range $0 < t < 4$. After $t = 4$, consider a current-driven anomalous resistivity model as

$$\eta(\mathbf{r}, t) = \begin{cases} k_\mathrm{R}[V_\mathrm{d}(\mathbf{r}, t) - V_\mathrm{C}] & \text{if } V_\mathrm{d} > V_\mathrm{C}, \\ 0 & \text{if } V_\mathrm{d} < V_\mathrm{C}, \end{cases} \tag{37}$$

$$V_\mathrm{d}(\mathbf{r}, t) = \left|\mathbf{J}(\mathbf{r}, t)/\rho(\mathbf{r}, t)\right|, \qquad V_\mathrm{C}(\mathbf{r}, t) = V_\mathrm{C0}\left[T(\mathbf{r}, t)/T_0\right]^\alpha, \tag{38}$$

where $k_\mathrm{R} = 0.003$, $V_\mathrm{C0} = 4$, $\alpha = 0.5$ and $T_0 = (1+\beta)/2$.

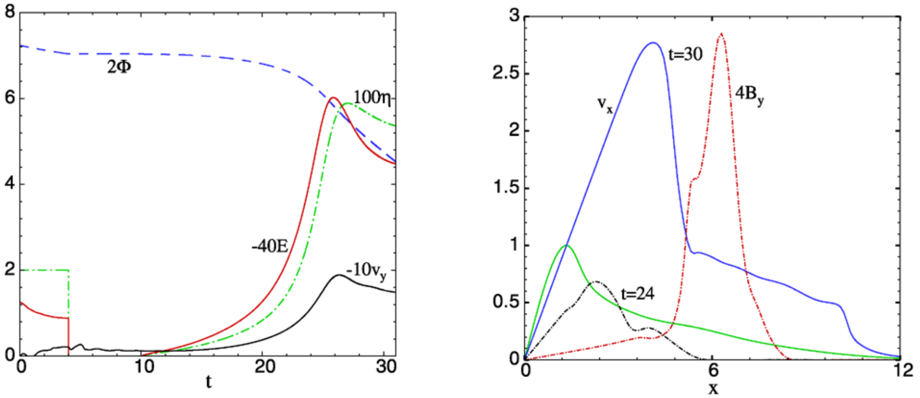


Figure 7 Spontaneous fast reconnection problem: (left) temporal variations of $E(\mathbf{r}=0)$, $\eta(\mathbf{r}=0)$, $v_y(x=0,y=0.9)$ and $\Phi(t)$ and (right) profiles of the outflow velocity $v_x(x,y=0)$ (solid lines) and the reconnected field component $B_y(x,y=0)$ (dotted lines) along the x-axis at $t=24$ (left) and at $t=30$ (right).

According to the symmetry boundary conditions with respect to the x- and y-axes, the computational region is restricted to the first quadrant only and taken to be a rectangular box, $(x,y)\in[0,6Y_1]\times[0,2Y_1]$. On the other boundaries, the free boundary conditions are assumed. Three levels of refinements are used, with the coarsest cell size $\Delta y=\Delta x/3=0.05625$ and the finest cell size $\Delta y=\Delta x/3=0.0140625$, so that the finest mesh size is nearly the same as that used by Ugai (1999) and Feng, Hu, and Wei (2006).

We run our simulation until $t=30$, when the reconnection evolution has accomplished the initial evolutionary phase $(10<t<20)$ and the explosive phase $(20<t<27)$; the results are presented in Figures 7 and 8. Figure 7 shows the temporal variations of four typical quantities: the resistivity $\eta(\mathbf{r}=0,t)$, the reconnection rate $E(\mathbf{r}=0,t)=\eta(\mathbf{r}=0,t)J(\mathbf{r}=0,t)$, the initially residing magnetic flux $\Phi(t)$ (defined by Ugai, 1999),

$$\Phi(t)=\int_{y=0}^{y=L_y}B_x(x=0,y,t)\,\mathrm{d}y=\Phi(0)+\int_0^{\tau=t}E(\mathbf{r}=0,\tau)\,\mathrm{d}\tau \qquad (39)$$

and the plasma inflow velocity $v_y(x=0,y=0.9,t)$, as well as the temporal behaviors of B_y and v_x at $t=24$ and $t=30$. This figure corresponds to Figures 1 and 2 obtained by Ugai (1999), or Figure 2 in Feng, Hu, and Wei (2006). Figure 8 gives the magnetic field configurations, pressure p and current density $|\mathbf{J}|$ at the typical times $t=18$ and $t=30$, just before and after the explosive phase. Our results are almost identical to those in Ugai (1999) and Feng, Hu, and Wei (2006); however, much less computation resource is needed due to the implementation of the AMR grid.

6.3. Two-dimensional MHD Blast Wave Problem

The MHD spherical blast wave problem (Zachary, Malagoli, and Colella, 1994; Gardiner and Stone, 2008) is initiated by an over-pressured region in the center of a strongly magnetized medium with low plasma β. The blast will drive fast shocks moving outward, compressing the plasma and magnetic field ahead and leaving rarefied plasma behind.

In this section, the two-dimensional MHD blast wave problem is calculated in both Cartesian coordinates and polar coordinates to validate our AMR-CESE method in curvilinear

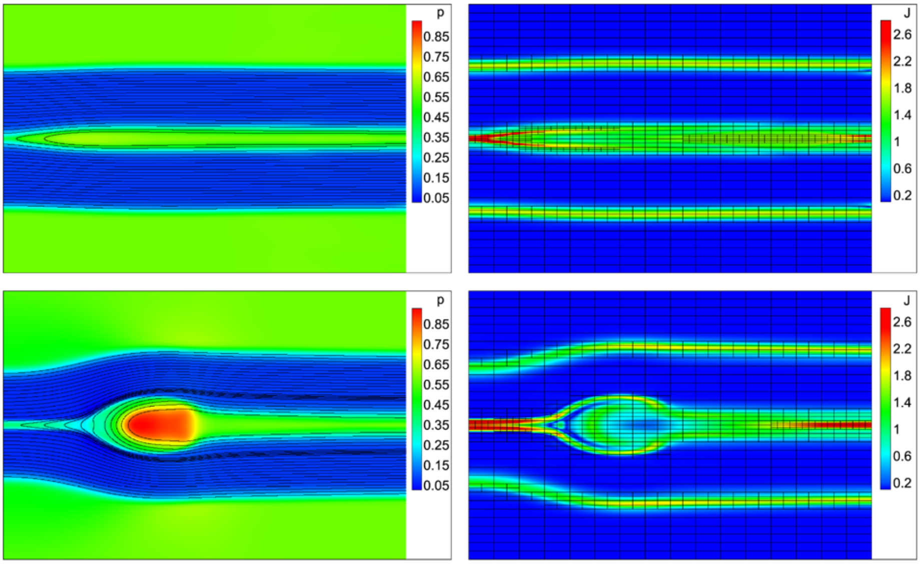


Figure 8 Spontaneous fast reconnection problem: magnetic field lines and the color-coded images of plasma pressure (left column) and color-coded images of current density and the corresponding grid blocks (right column). The first row shows snapshots at $t = 18$, the second row at $t = 30$.

coordinates. We first present the computation under a uniform mesh in both coordinate systems, and then we show the results of AMR implementation in the polar coordinates. In these tests (and in the three-dimensional cases in the next section), we followed the test suite of Skinner and Ostriker (2010), which was designed to verify their code *Athena* in cylindrical geometry.

Here, the computational domain in Cartesian coordinates is $(x, y) \in [1, 2] \times [-0.5, 0.5]$, and in polar coordinates is $(r, \theta) \in [1, 2] \times [-1/3, 1/3]$, so that the domains in the physical space are roughly similar in both cases with the center of the over-pressured region at $(x, y) = (1.5, 0)$. For polar coordinates, we set the reference space as (ξ, η), which is related to the physical space (x, y) by

$$x = r\cos\theta, \qquad y = r\sin\theta; \qquad r = \mathrm{e}^{\xi}, \qquad \theta = \eta. \tag{40}$$

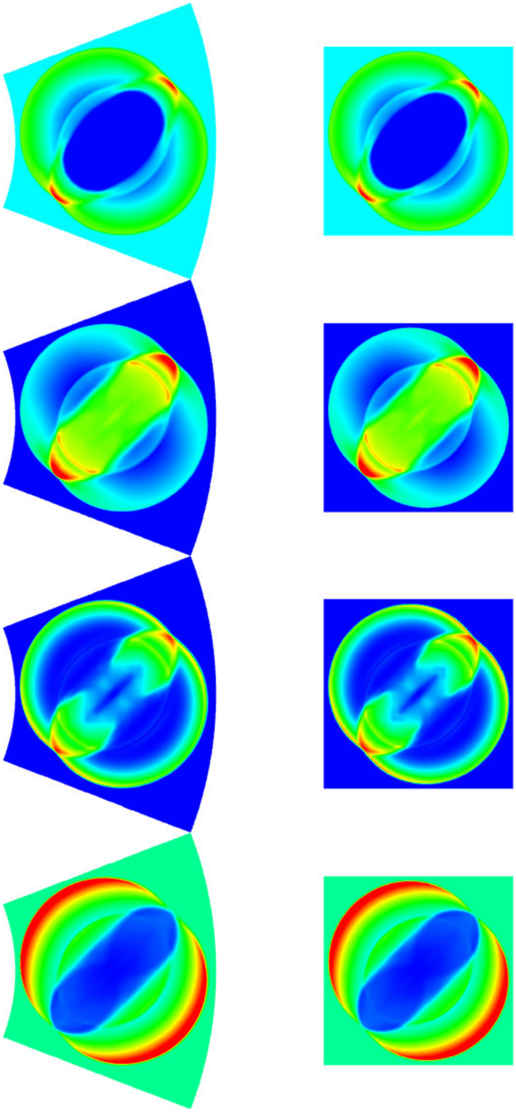
Thus, by using a uniform mesh in the reference space with cell size $\Delta\xi = \Delta\eta$, the cells in the physical space will have cell size $\Delta r = \mathrm{e}^{\xi}(\mathrm{e}^{\Delta\xi} - 1) \approx r\Delta\theta$, which means that the cells are close to regular squares not only in the reference space but also in the physical space. With the transformation given by Equation (40), all the other relations needed in Equation (21) can be derived analytically.

The initial conditions for this problem are set as

$$\rho = 1, \qquad p = \begin{cases} 10 & \text{if } \sqrt{(x-1.5)^2 + y^2} < 0.1, \\ 0.1 & \text{if } \sqrt{(x-1.5)^2 + y^2} \geq 0.1, \end{cases} \tag{41}$$
$$\mathbf{v} = \mathbf{0}, \qquad \mathbf{B} = (1/\sqrt{2}, 1/\sqrt{2}, 0)$$

while the boundary variables are fixed (as numerical computation is terminated before the outermost shock reaches any boundary faces).

Figure 9 Color-coded images of the density, pressure, specific kinetic energy, and magnetic energy at time $t = 0.2$ for the two-dimensional MHD blast wave problems computed in polar coordinates (left column) and Cartesian coordinates (right column).



The initial conditions are evolved until time $t = 0.2$ using 256×256 computational grids in both coordinate systems. Figure 9 shows the color-coded images of the density, pressure, specific kinetic energy, and magnetic energy calculated in both cases at the end time ($t = 0.2$). We also plot these variables along a horizontal line (*i.e.*, the x-axis) through the center of the blast (Figure 10).

As shown in these figures, our results are highly in agreement with those calculated in Skinner and Ostriker (2010). The outermost surface seen in Figure 9 is a fast-mode shock that is only weakly compressive and energetically is dominated by the magnetic field. Interior to this, one finds two dense shells of gas which propagate parallel to the magnetic field. These shells are bounded by a slow-mode shock and contact surface (separating the

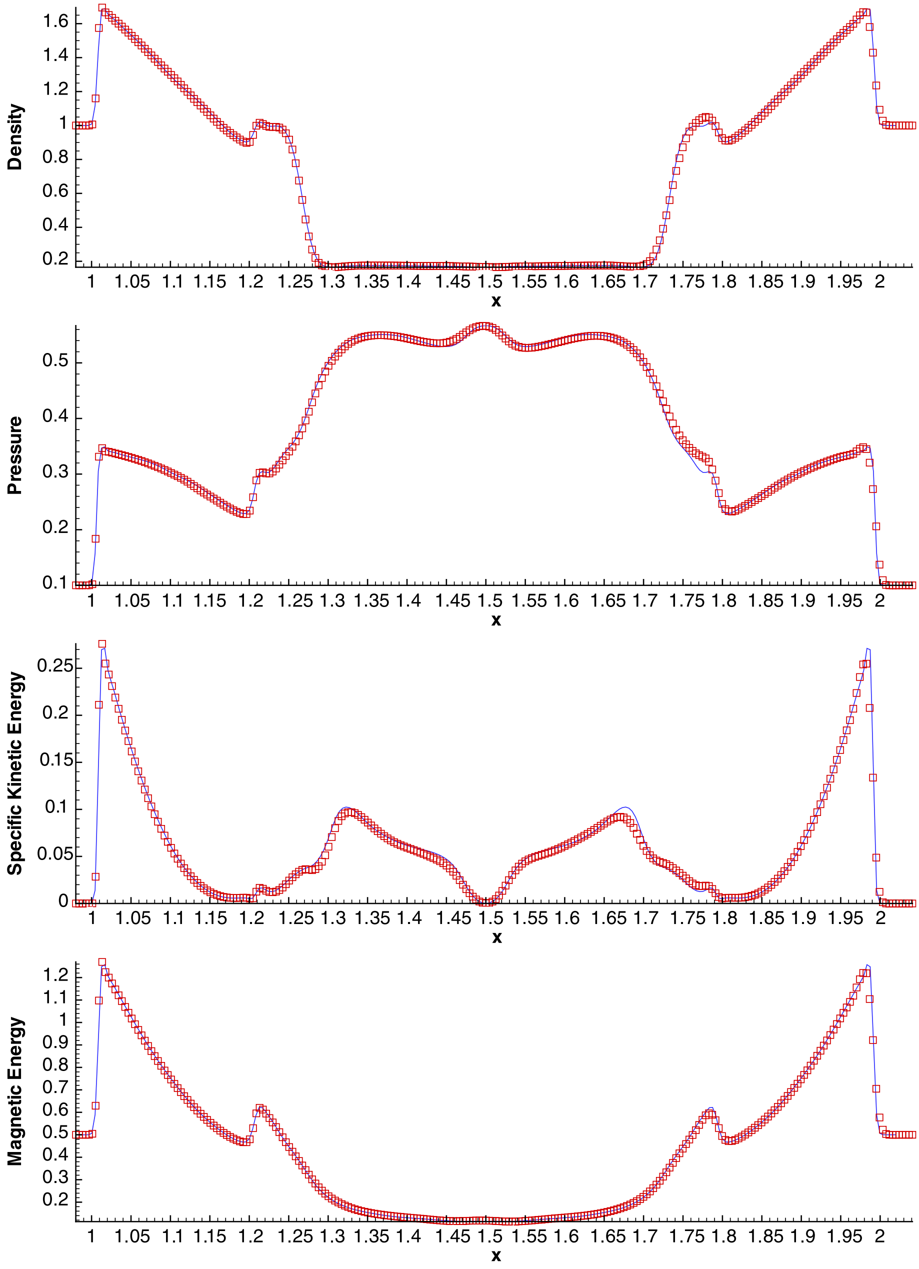


Figure 10 Plots of the same variables in Figure 9 along a horizontal line through the center of the blast. The solid lines denote the results in Cartesian coordinates, and the squares denote the results in polar coordinates.

initially hot, interior gas from the surrounding cool ambient medium) on the outer and inner surfaces, respectively (Gardiner and Stone, 2008). Note that the results for polar coordinates are almost the same as those for Cartesian coordinates, as seen from the side-by-side comparison in Figure 9 and the plots in Figure 10, and the symmetry is well preserved in both

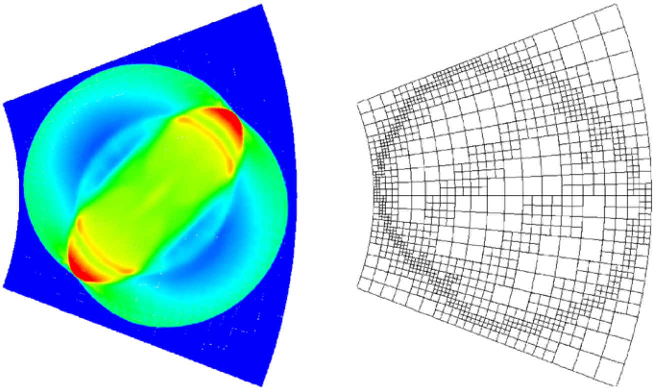


Figure 11 Color-coded image of pressure at $t = 0.2$ for the two-dimensional MHD blast wave problem computed in polar coordinates on the AMR grid system (left), and the corresponding grid blocks (right).

cases. In Figure 10, there is a slight deviation between the two results. This may be caused by the varying mesh sizes in physical space in the case of polar coordinates.

Figure 11 presents the result of the same problem in polar coordinates but computed on the AMR grid system. Three levels of refinements are used in the test, and it shows that the fast- and slow-mode shocks are identified by the grid resolution.

6.4. Three-Dimensional MHD Blast Wave Problem

Now we test our implementation in the three-dimensional case by computing the MHD blast wave problem using the same procedure as in Section 6.3. Specifically, we first calculate the blast wave problem using uniform mesh in both three-dimensional Cartesian coordinates and spherical coordinates to validate the CESE method in curvilinear coordinates, and then we compute the same problem in spherical coordinates on an AMR grid system to examine the capability of AMR implementation in curvilinear coordinates.

For convenience of comparison, we set the domain in the Cartesian coordinates as $(x, y, z) \in [1, 2] \times [-0.5, 0.5] \times [-0.5, 0.5]$ and in the spherical coordinates as $(r, \theta, \phi) \in [1, 2] \times [\pi/2 - 1/3, \pi/2 + 1/3] \times [-1/3, 1/3]$, and the center of the over-pressured region at $(x, y, z) = (1.5, 0, 0)$ in both cases. Similarly, for the spherical coordinates, the mapping from the reference space (ξ, η, ζ) to the physical space (x, y, z) is given by

$$\begin{cases} x = r\sin\theta\cos\phi, \\ y = r\sin\theta\sin\phi, \\ z = r\cos\theta, \end{cases} \qquad r = \mathrm{e}^{\xi}, \qquad \theta = \eta, \qquad \phi = \zeta. \tag{42}$$

The initial conditions for this problem are given as

$$\rho = 1, \qquad p = \begin{cases} 10 & \text{if } \sqrt{(x-1.5)^2 + y^2 + z^2} < 0.1, \\ 0.1 & \text{if } \sqrt{(x-1.5)^2 + y^2 + z^2} \geq 0.1, \end{cases} \qquad \mathbf{v} = \mathbf{0}, \qquad \mathbf{B} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}) \tag{43}$$

and for the uniform mesh, 128 × 128 × 128 cells are used.

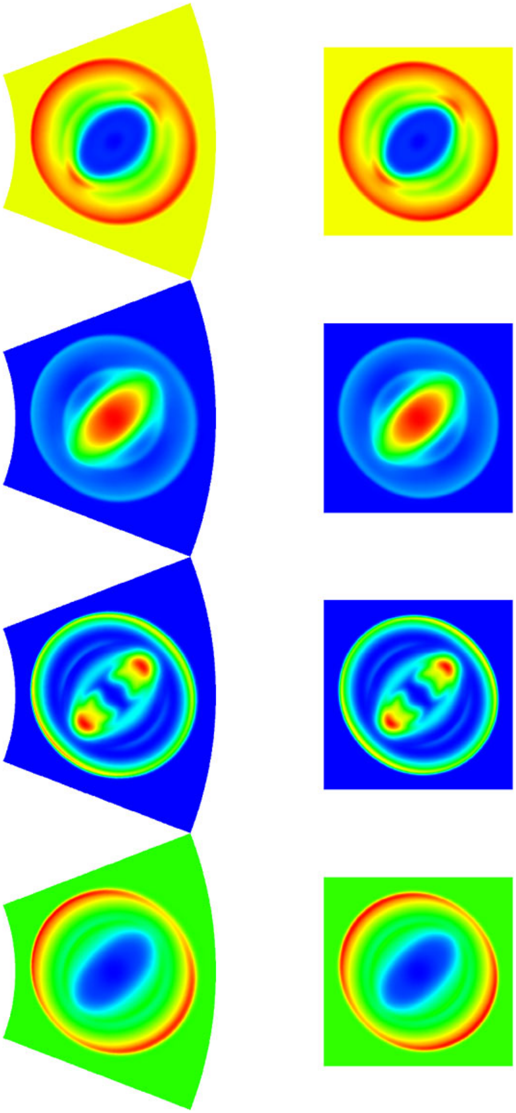


Figure 12 Color-coded images of density, pressure, specific kinetic energy, and magnetic energy at $t = 0.2$ for the three-dimensional MHD blast wave problems computed in spherical coordinates (left column) and Cartesian coordinates (right column).

Figure 12 shows images of the density, pressure, kinetic energy, and magnetic energy sliced along the $z = 0$ plane at time $t = 0.2$ in both coordinate systems. The general structure of the solution is the same as in the two-dimensional calculation (Section 6.3). Plots of these variables along the x-axis through the center of the blast in Figure 13 show agreement between the results of Cartesian coordinates and those of spherical coordinates.

Figure 14 presents the results of the same problem in spherical coordinates but computed on the AMR grid system. Three levels of refinements are used in the test, and it shows the capturing of the fast- and slow-mode shocks by the grid system.

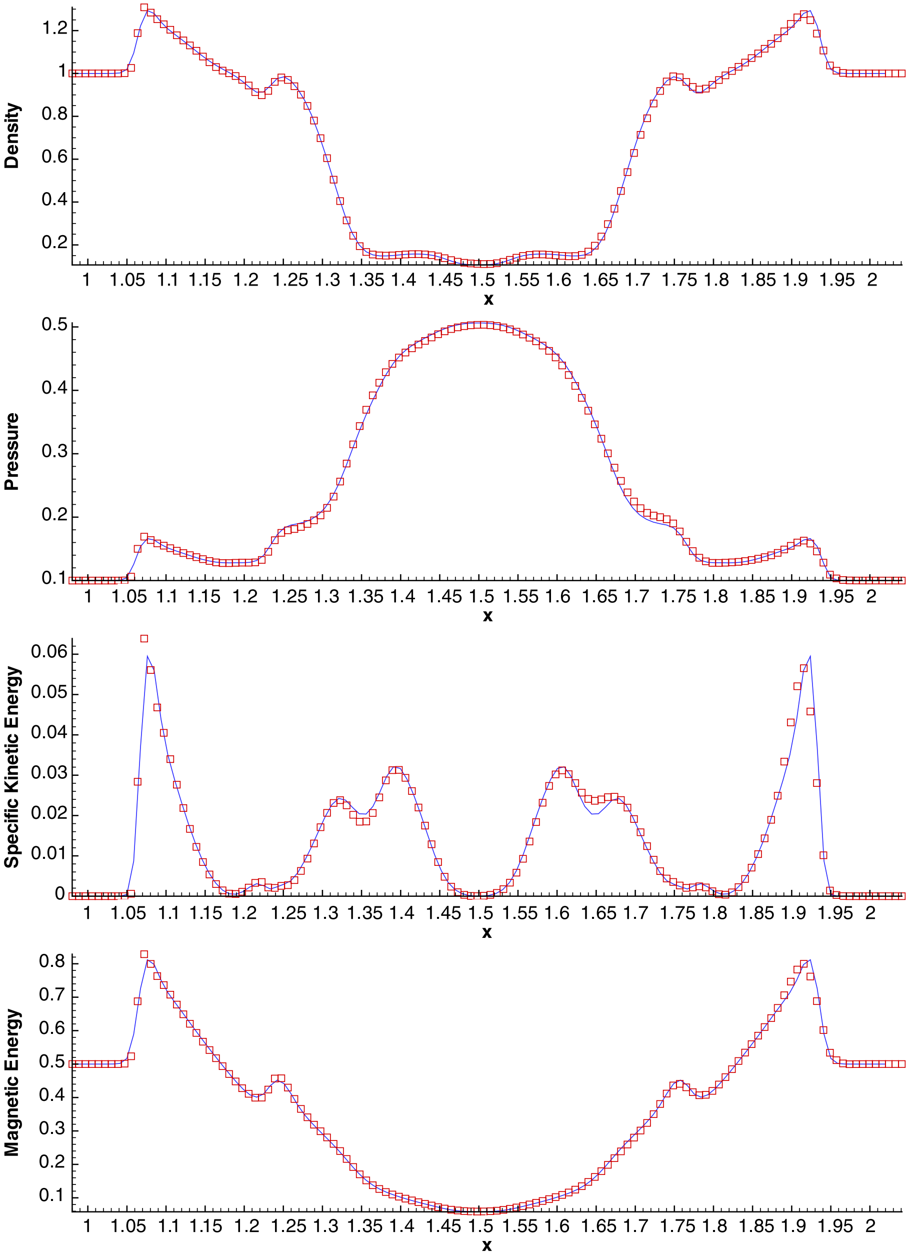


Figure 13 Plots of the same variables in Figure 12 along a horizontal line through the center of the blast. The solid lines denote the results in Cartesian coordinates, and the squares denote the results in spherical coordinates.

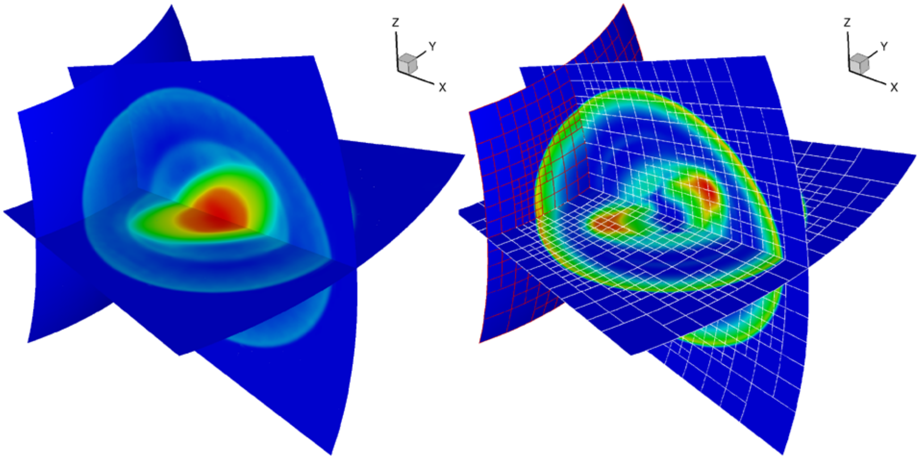


Figure 14 Color-coded pressure (left) and specific kinetic energy (right) sliced along the (r, θ, ϕ) planes at $t = 0.2$ for the three-dimensional MHD blast wave problem computed in spherical coordinates on the AMR grid system and the corresponding grid blocks (right).

7. Conclusions

In this paper, we have presented a new implementation of the AMR-CESE method for MHD problems by using the improved CESE method in Cartesian and general curvilinear coordinates. To the best of our knowledge, this is the first application of the CESE method on an adaptive mesh refinement grid and the first attempt to use the CESE method in curvilinear coordinates for modeling MHD problems.

First, we rearrange the space – time points of the original CESE method in a uniform mesh and avoid the space – time staggering nature by integrating two half timesteps into one full timestep; then the CESE method in Cartesian coordinates of the physical space is extended to the reference space, where the MHD equations are solved in the general curvilinear coordinates. In this way, we give the improved version of this numerical scheme in both two and three dimensions for MHD equations. Besides all the salient advantages of the original CESE method such as *i*) a unified treatment of flow evolution in space and time, *ii*) enforcement of local and global space – time flux conservation in a coherent and efficient manner, and *iii*) efficient evaluation of the fluxes at the interface of any pair of conservation elements (CEs) by means of staggered space – time grids without using Riemann solvers or other flux models), our improved method has the following merits.

i) The solution points are explicitly given on the mesh nodes, while in the original method, these points must be calculated after setting the grids.

ii) In the original method for the three-dimensional case, the projection of the CE onto the spatial space is a 24-faced polyhedron (Zhang, Yu, and Chang, 2002), while it is simplified to a rectangular cuboid in the improved one.

iii) As described in Section 3, the computation of the flux leaving through the surface of the CE is greatly simplified, as the normal vector of each face is along the corresponding axis and only one component of the space – time flux vector is needed on each face. This will reduce the CPU time significantly, as the majority of the computation time of the CESE method is consumed by the flux integration.

iv) By avoiding the space – time staggering nature, the improved method can be more easily coded, the boundary conditions will be more easily implemented, and the method can be readily used on the block-AMR grid.

v) The versatility of the general curvilinear coordinates allows our method to be applied in calculations of problems with simple curved boundaries or problems with very complex boundaries if combined with the method of overlapping grids (Henshaw and Schwendeman, 2008).

With the aid of the PARAMESH package, we extend the CESE method on AMR grids. The variable timestep algorithm, allowing the timesteps to differ from block to block, is also adopted to save computational resources and reduce the numerical viscosity.

Our method can be applied to model both ideal MHD and resistive MHD problems. To demonstrate the simulation capability, four problems from both ideal MHD and resistive MHD are calculated and compared with previously published results, including the MHD vortex problem, the fast magnetic reconnection problem, and the two- and three-dimensional MHD blast wave problems. All the numerical results for these problems show good consistency with the previous results. In particular, the last two problems are calculated in Cartesian coordinates and curvilinear coordinates, and the results for both coordinate systems confirm each other very well.

Unlike the uniform mesh, note that on AMR grids for conservation laws, additional consideration is needed for the space – time flux conservation at the interfaces with spatial or time resolution changes (*i.e.*, the interfaces of grid blocks at different spatial refinement levels or with different timesteps) to fulfill the global conservation properties of the CESE method in principle. Specifically, fluxes entering or leaving a grid cell through a common cell face shared with four cells of a more refined neighbor should be equal to the sum of the fluxes across the appropriate faces of the four smaller cells. In this work, we did not consider this constraint in the code because of the requirement of additional inter-block communication at these interfaces, which is a nontrivial work. Even so, all the numerical experiments give satisfactory results. As this work is devoted to the preliminary study of the CESE method on an AMR grid, enforcing the global conservation on AMR grids and the detailed comparison of results with and without this conservation constraint are left for our future work.

Our next plan is the implementation of the AMR-CESE method on solar wind modeling and global simulation of solar explosive phenomena (the initiation and evolution of CMEs), by extending our SIP-CESE MHD model (Feng, Zhou, and Wu, 2007; Hu *et al.*, 2008; Zhou, Feng, and Wu, 2008; Zhou and Feng, 2008; Feng *et al.*, 2009) on an AMR grid system, to discuss the initiation of solar transient events. It is expected that our SIP-AMR-CESE MHD model in curvilinear coordinates will facilitate the implementation of time-dependent boundary input by using the continuously available data of the Solar Dynamics Observatory mission.

Acknowledgements The work is jointly supported by the National Natural Science Foundation of China (41031066, 40921063, 40874091, 40890162, 40904050, 40874077, and 40536029), the 973 project under grant 2006CB806304, and the Specialized Research Fund for State Key Laboratories. The numerical calculation has been completed on our SIGMA Cluster computing system. Special thanks go to our anonymous reviewer for helpful suggestions for the improvement of the paper.

Appendix: Jacobian Matrices of the Fluxes

The Jacobian matrices for the fluxes **F**, **G** and **H** in Equation (1) are given here by

$$A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -{v_x}^2 & 2v_x & 0 & 0 & 1 & -B_x & B_y & B_z \\ -v_x v_y & v_y & v_x & 0 & 0 & -B_y & -B_x & 0 \\ -v_x v_z & v_z & 0 & v_x & 0 & -B_z & 0 & -B_x \\ -\frac{\gamma p v_x}{\rho} & \frac{\gamma p}{\rho} & 0 & 0 & \gamma v_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-v_x B_y + v_y B_x}{\rho} & \frac{B_y}{\rho} & -\frac{B_x}{\rho} & 0 & 0 & -v_y & v_x & 0 \\ \frac{-v_x B_z + v_z B_x}{\rho} & \frac{B_z}{\rho} & 0 & -\frac{B_x}{\rho} & 0 & -v_z & 0 & v_x \end{bmatrix}, \tag{44}$$

$$B = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -v_x v_y & v_y & v_x & 0 & 0 & -B_y & -B_x & 0 \\ -{v_y}^2 & 0 & 2v_y & 0 & 1 & B_x & -B_y & B_z \\ -v_y v_z & 0 & v_z & v_y & 0 & 0 & -B_z & -B_y \\ -\frac{\gamma p v_y}{\rho} & 0 & \frac{\gamma p}{\rho} & 0 & \gamma v_y & 0 & 0 & 0 \\ \frac{v_x B_y - v_y B_x}{\rho} & -\frac{B_y}{\rho} & \frac{B_x}{\rho} & 0 & 0 & v_y & -v_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-v_y B_z + v_z B_y}{\rho} & 0 & \frac{B_z}{\rho} & -\frac{B_y}{\rho} & 0 & 0 & -v_z & v_y \end{bmatrix}, \tag{45}$$

$$C = \frac{\partial \mathbf{H}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -v_x v_z & v_z & 0 & v_x & 0 & -B_z & 0 & -B_x \\ -v_y v_z & 0 & v_z & v_y & 0 & 0 & -B_z & -B_y \\ -{v_z}^2 & 0 & 0 & 2v_z & 1 & B_x & B_y & -B_z \\ -\frac{\gamma p v_z}{\rho} & 0 & 0 & \frac{\gamma p}{\rho} & \gamma v_z & 0 & 0 & 0 \\ \frac{v_x B_z - v_z B_x}{\rho} & -\frac{B_z}{\rho} & 0 & \frac{B_x}{\rho} & 0 & v_z & 0 & -v_x \\ \frac{v_y B_z - v_z B_y}{\rho} & 0 & -\frac{B_z}{\rho} & \frac{B_y}{\rho} & 0 & 0 & v_z & -v_y \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{46}$$

References

Anderson, J.D.: 1995, *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, New York, 168.

Berger, M.J., Colella, P.: 1989, *J. Comput. Phys.* **82**, 64.

Chang, S.: 1995, *J. Comput. Phys.* **119**, 295.

Chang, S., Wang, X., Chow, C.: 1999, *J. Comput. Phys.* **156**, 89.

Colella, P., Graves, D.T., Keen, N.D., Ligocki, T.J., Martin, D.F., McCorquodale, P.W., Modiano, D., Schwartz, P.O., Sternberg, T.D., Van Straalen, B.: 2007, *Chombo, Software Package for AMR Applications*, Lawrence Berkeley National Laboratory, http://seesar.lbl.gov/anag/chombo.

Collins, D.C., Norman, M.L.: 2004, *Bull. Am. Astron. Soc.* **36**, 1605.

Dedner, A., Kemm, F., Kröner, D., Munz, C., Schnitzer, T., Wesenberg, M.: 2002, *J. Comput. Phys.* **175**, 645.

De Zeeuw, D.L.: 1993, A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations. Ph.D. thesis, The University of Michigan.

Feng, X., Hu, Y., Wei, F.: 2006, *Solar Phys.* **235**, 235.

Feng, X., Zhou, Y., Wu, S.T.: 2007, *Astrophys. J.* **655**, 1110.

Feng, X.S., Zhang, Y., Yang, L.P., Wu, S.T., Dryer, M.: 2009, *J. Geophys. Res.* **114**, A10103.

Garaizar, X., Hornung, R., Kohn, S.: 1999, *Structured Adaptive Mesh Refinement Applications Infracture*, Lawrence Livermore National Laboratory, http://www.llnl.gov/casc/SAMRAI.

Gardiner, T.A., Stone, J.M.: 2008, *J. Comput. Phys.* **227**, 4123.

Groth, C.P.T., De Zeeuw, D.L., Gombosi, T.I., Powell, K.G.: 2000, *J. Geophys. Res.* **105**, 25053.

Henshaw, W.D., Schwendeman, D.W.: 2008, *J. Comput. Phys.* **227**, 7469.

Hoffman, K.A., Chiang, S.T.: 2000, *Computational Fluid Dynamics, II*, 4th edn. Engineering Education System, Wichita, 21.

Hu, Y., Feng, X.: 2006, *Solar Phys.* **238**, 329.

Hu, Y.Q., Feng, X.S., Wu, S.T., Song, W.B.: 2008, *J. Geophys. Res.* **113**, A03106.

Jiang, G., Wu, C.: 1999, *J. Comput. Phys.* **150**, 561.

Linde, T.J.: 2002, *Am. Phys. Soc. Meeting*, Abstract F3005.

Linde, T.J.: 1998, A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere. Ph.D. thesis, The University of Michigan.

MacNeice, P., Olson, K.M., Mobarry, C., de Fainchtein, R., Packer, C.: 2000, *Comput. Phys. Commun.* **126**, 330.

Marder, B.: 1987, *J. Comput. Phys.* **68**, 48.

Olson, K.: 2006, In: Deane, A., Brenner, G., Emerson, D., McDonough, J., Tromeur-Dervout, D., Satofuka, N., Ecer, A., Periaux, J. (eds.) *Parallel Computational Fluid Dynamics*, Elsevier, Amsterdam, 341.

Olson, K., MacNeice, P.: 2005, In: Plewa, T., Linde, T., Weirs, V.G. (eds.) *Adaptive Mesh Refinement-Theory and Applications*, Springer, Berlin, 315

Orszag, S.A., Tang, C.: 1979, *J. Fluid Mech.* **90**, 129.

Parashar, M.: 2007, *Grid Adaptive Computational Engine (GrACE)*, Rutgers University, http://www.caip.rutgers.edu/tassl/projects/grace/.

Parashar, M., Browne, J.: 2007, *Distributed Adaptive Grid Hierarchy*, Rutgers University, http://userweb.cs.utexas.edu/users/dagh/.

Powell, K.G., Roe, P.L., Quirk, J.: 1993, In: Hussaini, M.Y., Kumar, A., Salas, M.D. (eds.) *Algorithmic Trends in Computational Fluid Dynamics*, Springer, New York, 303.

Powell, K.G., Roe, P.L., Linde, T.J., Gombosi, T.I., De Zeeuw, D.L.: 1999, *J. Comput. Phys.* **154**, 284.

Rendleman, C., Beckner, V., Lijewski, M., Crutchfield, W., Bell, J.: 2000, *Comput. Vis. Sci.* **3**, 147.

Skinner, M.A., Ostriker, E.C.: 2010, *Astrophys. J. Suppl.* **188**, 290.

Stone, J.M., Gardiner, T.A., Teuben, P., Hawley, J.F., Simon, J.B.: 2008, *Astrophys. J. Suppl.* **178**, 137.

Tóth, G.: 2000, *J. Comput. Phys.* **161**, 605.

Tóth, G., Sokolov, I.V., Gombosi, T.I., Chesney, D.R., Clauer, C.R., De Zeeuw, D.L., *et al.*: 2005, *J. Geophys. Res.* **110**, A12226.

Tóth, G., De Zeeuw, D.L., Gombosi, T.I., Powell, K.G.: 2006, *J. Comput. Phys.* **217**, 722.

Ugai, M.: 1999, *Phys. Plasmas* **6**, 1522.

Vinokur, M.: 1974, *J. Comput. Phys.* **14**, 105.

Viviand, H.: 1975, *European Space Research Organization Technical Translation ESRO-TT-144*, 153 (English translation).

Wang, X., Chang, S.: 1999, *Comput. Fluid Dyn. J.* **8**, 309.

Wu, S., Zheng, H., Wang, S., Thompson, B., Plunkett, S., Zhao, X., Dryer, M.: 2001, *J. Geophys. Res.* **106**, 25.

Wu, S., Wang, A., Liu, Y., Hoeksema, J.: 2006, *Astrophys. J.* **652**, 800.

Yu, S., Chang, S., Jorgenson, P.C.E., Park, S., Lai, M.: 1998, In: Bruneau, C.-H. (ed.) *Proc. 16th Int. Conf. on Numerical Methods in Fluid Dynamics*, *Lecture Notes in Physics* **515**, Springer, Berlin, 433.

Zachary, A.L., Malagoli, A., Colella, P.: 1994, *SIAM J. Sci. Comput.* **15**(2), 263.

Zhang, M., John Yu, S., Henry Lin, S., Chang, S., Blankson, I.: 2006, *J. Comput. Phys.* **214**, 599.

Zhang, Z., Yu, S.T.J., Chang, S.: 2002, *J. Comput. Phys.* **175**, 168.

Zhou, Y.F., Feng, X.S.: 2008, *Sci. China E* **51**, 1600.

Zhou, Y.F., Feng, X.S., Wu, S.T.: 2008, *Chinese Phys. Lett.* **25**, 790.

Ziegler, U.: 1998, *Comput. Phys. Commun.* **109**, 111.

Ziegler, U.: 2003, In: Falgarone, E., Passot, T. (eds.) *Turbulence and Magnetic Fields in Astrophysics*, *Lecture Notes in Physics*, **614**, Springer, Berlin, 127.