BLOCK-STRUCTURED ADAPTIVE MESH REFINEMENT ON CURVILINEAR-ORTHOGONAL GRIDS

UDO ZIEGLER

Abstract. An adaptive mesh refinement scheme for the equations of magnetohydrodynamics on curvilinear-orthogonal grids is presented. A block-structured refinement approach is used sharing important properties with the hybrid central-upwind/constrained-transport single grid solver like angular momentum conservation on cylindrical and spherical grids and the preservation of magnetic field solenoidality. The adaptive mesh framework is part of the astrophysical fluid dynamics code NIRVANA which is continuously developed and maintained by the author. The code is parallelized on the basis of the MPI library, and dynamic load balancing in adaptive mesh simulations is achieved by space-filling curve domain decomposition techniques. For various two- and three-dimensional magnetohydrodynamic problems adaptive mesh functionality and efficiency are demonstrated.

Key words. adaptive mesh refinement, magnetohydrodynamics, numerics: central schemes

AMS subject classifications. 65M08, 65M50

DOI. 10.1137/110843940

1. Introduction. Adaptive mesh refinement (AMR) is a choice method for an efficient spatial discretization of multiscale computational problems. Among the varieties of existing refinement strategies a block-structured (or patch-structured) approach has shown to be particularly popular for the numerical solution of the equations of hydrodynamics and magnetohydrodynamics (MHD). In block-structured AMR numerical cells are not individually divided into finer cells, but finer cells are created in terms of logically rectangular patches which are overlaid the coarser cells. This way a hierarchy of refinement levels made up of patches can be established in order to meet resolution requirements [3], [5], [4]. In recent years a couple of MHD codes have been released which follow the block-structured refinement methodology. To name a few, these are the FLASH code [10], the SFUMATO code [22], the PLUTO code [23], the ENZO code [25], and the AMRVAC code [15], [13]. Some codes like ENZO and AMRVAC have their own AMR implementation. Others are built on external AMR packages. For instance, the FLASH code uses the PARAMESH tool [21], whereas AMR in PLUTO relies on the Chombo library [8]. The use of self-similar blocks having a fixed number of grid points and a fixed refinement ratio of 2 is a famous realization of block-structured AMR resulting in a block-tree data structure. Such a strategy is exploited by AMRVAC as well as in the astrophysical fluid dynamics code NIRVANA presented in this work.

Most block-structured AMR applications published in the literature are in Cartesian geometry using Cartesian grids. Applications based on the use of cylindrical grids or spherical grids are, by far, less frequently found especially for three-dimensional (3D) nonaxisymmetric problems. One reason might be because coding of algorithms suffers from an increased complexity associated with those geometries. A further complication represents the intrinsic grid anisotropy caused by the converging grid lines.
at the coordinate singularities degrading solution efficiency and accuracy already on an uniform grid.

Important properties of the MHD equations which are fulfilled in discrete form by the numerical scheme on a single grid must be guaranteed on an adaptive grid as well. Critical issues in this respect requiring extra care are the prolongation and restriction techniques for solution transfer from coarse-to-fine grids and fine-to-coarse grids, and concern mesh synchronization tasks at coarse-fine grid interfaces. In the numerics of MHD equations the solenoidality constraint for the magnetic field, in addition, needs special attention. Parallelism of the AMR scheme is a further issue involving enhanced algorithmic complexity.

In this paper a block-structured mesh refinement strategy for curvilinear-orthogonal grids is discussed as it has been recently realized in the NIRVANA code [24]. NIRVANA is a numerical code for the approximate solution of the time-dependent, compressible MHD equations. In orthogonal-curvilinear coordinates \((x, y, z)\) the metric tensor has diagonal form, \(H = \text{diag}(h_x^2, h_y^2, h_z^2)\). For cylindrical coordinates \((z, R, \phi)\) and spherical coordinates \((r, \theta, \phi)\) the metric scale factors are given by \((h_x, h_y, h_z) = (1, 1, R)\) and by \((h_x, h_y, h_z) = (1, r, r \sin \theta)\), respectively. In the trivial case of Cartesian coordinates the scale factors are \((h_x, h_y, h_z) = (1, 1, 1)\). The governing equations are

\[
\begin{align*}
\partial_t u &= -\frac{1}{h_y h_z} \left[ \partial_x (h_y h_z f_x) + \partial_y (h_z f_y) + \partial_z (h_y f_z) \right] + S, \\
\partial_t B_x &= -\frac{1}{h_y h_z} \left[ \partial_y (h_z E_z) - \partial_z (h_y E_y) \right], \\
\partial_t B_y &= -\frac{1}{h_z} \left[ \partial_z E_x - \partial_x (h_z E_z) \right], \\
\partial_t B_z &= -\frac{1}{h_y} \left[ \partial_x (h_y E_y) - \partial_y E_x \right] \\
\end{align*}
\]

with flux functions \(f_x, f_y, f_z\),

\[
\begin{align*}
f_x(u, B) &= \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p_t - B_x^2/\mu \\ \rho v_x v_y - B_x B_y/\mu \\ h_z (\rho v_x v_z - B_x B_z/\mu) \\ (e + p_t) v_x - B_x (v \cdot B)/\mu \end{pmatrix}, \\
f_y(u, B) &= \begin{pmatrix} \rho v_y \\ \rho v_y v_x - B_x B_y/\mu \\ \rho v_y^2 + p_t - B_y^2/\mu \\ h_z (\rho v_y v_z - B_y B_z/\mu) \\ (e + p_t) v_y - B_y (v \cdot B)/\mu \end{pmatrix}, \\
f_z(u, B) &= \begin{pmatrix} \rho v_z \\ \rho v_z v_x - B_x B_z/\mu \\ \rho v_z v_y - B_y B_z/\mu \\ h_z (\rho v_z^2 + p_t - B_z^2/\mu) \\ (e + p_t) v_z - B_z (v \cdot B)/\mu \end{pmatrix}
\end{align*}
\]
and geometric source term $\mathbf{S}$:

$$
\mathbf{S}(\mathbf{u}, \mathbf{B}) = \begin{pmatrix}
0 \\
\frac{1}{h_y} \frac{\partial h_y}{\partial x} \left[ q(v_y^2 + v_z^2) + 2p + B_z^2/\mu \right] \\
\frac{1}{h_z} \left[ \frac{\partial h_z}{\partial y} (q v_z^2 + p_t - B_z^2/\mu) - \frac{\partial h_y}{\partial x} (q v_z v_y - B_z B_y/\mu) \right] \\
0 \\
0
\end{pmatrix}.
$$

Here, $\mathbf{u} = (\rho, q v_x, q v_y, h_z q v_z, e)$ is the vector of primary hydrodynamical variables with $\rho$ the mass density, $\mathbf{v} = (v_x, v_y, v_z)$ the fluid velocity, and $e$ the total energy density. The magnetic field is denoted by $\mathbf{B} = (B_x, B_y, B_z)$, and the electric field $\mathbf{E}$ is defined by $\mathbf{E} = -\mathbf{v} \times \mathbf{B}$. The total pressure $p_t = p + B^2/2\mu$ is the sum of gas pressure $p$ and magnetic pressure with $\mu$ the magnetic permeability. The equations are complemented by an ideal gas law $p = (\gamma - 1) (e - q |\mathbf{v}|^2/2 - |\mathbf{B}|^2/2\mu)$ with adiabatic coefficient $\gamma$ and by the solenoidality property of the magnetic field, $\nabla \cdot \mathbf{B} = 0$.

Note that the fourth component of the $\mathbf{u}$-equation (for $u = h_z q v_z$) has no source term expressing the conservation of angular momentum in cylindrical and spherical coordinates.

The organization of the paper is as follows. First, in section 2 NIRVANA’s MHD solver is summed up. This is followed by the description of NIRVANA’s AMR paradigm in section 3. A verification of the AMR scheme is then given in section 4. Several numerical examples in two-dimensional (2D) and 3D cylindrical and spherical geometry are presented. Finally, conclusions are drawn in section 5.

### 2. Single grid solver: Summary.

In block-structured AMR updating the numerical solution from a time level $t^n$ to a next time level $t^{n+1} = t^n + \delta t$ (where $\delta t$ is the discrete time-step) consists of applying a single grid solver to each grid patch in the mesh hierarchy separately and then matching those individual solutions in space and, eventually, in time in case different time-steps are used for patch updates.

In the following NIRVANA’s single grid MHD solver is reviewed insofar as necessary for an understanding of the applied AMR philosophy. The very details of the solver are not repeated here, and the reader is referred to Ziegler [33] for a more comprehensive discussion. In NIRVANA the set of equations (1.1) is solved by a hybrid method consisting of a finite-volume (FV) approximation of Godunov type for the $\mathbf{u}$-update (Euler subsystem including the Lorentz force) and a constrained-transport (CT) scheme for the $\mathbf{B}$-update (induction equation). By construction, the CT scheme guarantees that the time derivative of magnetic divergence vanishes precisely in a FV sense [30], [20], [2], i.e., $\frac{d}{dt} \nabla \cdot \mathbf{B} = 0$, where the overbar denotes volume averaging over a grid cell. The integral form of Maxwell’s solenoidality condition is therefore automatically fulfilled cellwise if so cellwise initially. Additional grid divergence cleaning procedures (see, e.g., [9], [6]) are redundant in CT schemes.

The FV form of (1.1) reads componentwise as

$$
\ddot{\mathbf{n}}_{i,j,k}(t) = -\frac{[\delta A_{x,i,j,k} \mathcal{F}_{x,i,j,k}^\nu |_{\mu} + [\delta A_{y,i,j,k} \mathcal{F}_{y,i,j,k}^\nu |_{\mu} + [\delta A_{z,i,j,k} \mathcal{F}_{z,i,j,k} |_{\mu} + \frac{1}{h_z} h_z \frac{\partial h_z}{\partial y} (q v_z^2 + p_t - B_z^2/\mu) - \frac{1}{h_y} h_y \frac{\partial h_y}{\partial x} (q v_z v_y - B_z B_y/\mu)]}{h_z} + \mathbf{S}_{i,j,k}
$$

(2.1)

and describes the time rate of change of volume-averages

$$
\ddot{\mathbf{n}}_{i,j,k} = \frac{1}{h_z} h_z \frac{\partial h_z}{\partial y} (q v_z^2 + p_t - B_z^2/\mu) - \frac{1}{h_y} h_y \frac{\partial h_y}{\partial x} (q v_z v_y - B_z B_y/\mu) + \mathbf{S}_{i,j,k}
$$


in cells \( C_{i,j,k} = [x_{im}, x_{ip}] \times [y_{jm}, y_{jp}] \times [z_{km}, z_{kp}] \) of a uniform grid represented by running indices \( i, j, \) and \( k \) and grid spacings \( \delta x = x_{ip} - x_{im}, \delta y = y_{jp} - y_{jm}, \) and \( \delta z = z_{kp} - z_{km} \). Half indices \( i_m = i - 1/2, i_p = i + 1/2 \) and analog indices for the other directions mark cell nodes with coordinates \((x_{im}, y_{jm}, z_{km})\). The geometric cell center has coordinates \((x_i, y_j, z_k)\). The cell volume is denoted by \( \delta V_{i,j,k} \). The areas of the cell bounding \( x \)-faces are denoted by \( \delta A_{xim,j,k} \) for the cell face at \( x_{im} \) and \( \delta A_{xip,j,k} \) for the cell face at \( x_{ip} \), respectively. Analog definitions apply for other cell faces. \( \mathcal{F}_x, \mathcal{F}_y, \) and \( \mathcal{F}_z \) are the face-averaged flux functions. A subscripted square bracket denotes an index difference; e.g., \( [\delta A_{xim,j,k} \mathcal{F}_{xim,j,k}]_{im}^p = \delta A_{xip,j,k} \mathcal{F}_{xip,j,k} - \delta A_{xim,j,k} \mathcal{F}_{xim,j,k} \) means the flux difference between the upper and lower \( x \)-faces. The numerical fluxes are approximated from the face-averaged flux functions by (i) computing the facial integrals with the help of a cubature formula of degree 2 and by (ii) utilizing central-upwind techniques for stability. Essentially, the method is an extension of the second-order semidiscrete scheme developed in [18] to curvilinear-orthogonal grids (see also the work in [14], [17]). The resulting flux formulae are equivalent to HLL fluxes (see equation (12) in [33]).

Point values of variables \( u \) needed in the flux computation process are obtained from reconstructing the solution based on the known set of averages \( \{ \overline{p}_{i,j,k} \} \). The reconstruction function \( \tilde{u}(x) \) in a cell \( C_{i,j,k} \) is given by

(2.2) \( \tilde{u}(x) = \overline{p}_{i,j,k} + (\partial u)_{i,j,k} (x - \overline{x}_i) + (\partial u)_{i,j,k} (y - \overline{y}_j) + (\partial u)_{i,j,k} (z - \overline{z}_k) \),

where \( \overline{x}_i, \overline{y}_j, \overline{z}_k \) are cell-volume-averaged coordinates. Explicit expressions for those in cylindrical and spherical coordinates are given in Table 1 in [33]. The numerical derivatives \( (\partial u)_{i,j,k}, (\partial u)_{i,j,k}, (\partial u)_{i,j,k} \) are approximations to the exact partial derivatives subject to a limiter function for oscillation control near flow discontinuities. The harmonic mean limiter of [29] has been used in all of the test problems in this paper. The reconstruction is conservative,

\[
\overline{p}_{i,j,k} \delta V_{i,j,k} = \int_{C_{i,j,k}} \tilde{u}(x) dV,
\]

and has the obvious property \( \overline{p}_{i,j,k} = \tilde{u}(\overline{x}_i, \overline{y}_j, \overline{z}_k) \); i.e., the cell-average is identical to the point value at coordinates \((\overline{x}_i, \overline{y}_j, \overline{z}_k)\).

The CT scheme for the (staggered) magnetic field components reads as

(2.3) \[
\langle \dot{B}_x \rangle_{i,j,k} (t) = \frac{1}{\delta A_{xim,j,k}} \left( -\delta z [h_{zim}, \mathcal{E}_{zim,j,k}]_{j,m}^p + \delta y [h_{yim}, \mathcal{E}_{yim,j,k}]_{m}^p \right),
\]

\[
\langle \dot{B}_y \rangle_{i,j,k} (t) = \frac{1}{\delta A_{yim,j,k}} \left( -\delta x [h_{xim,j,k}]_{m}^p + \delta z [h_{zim,j,k}]_{m}^p \right),
\]

\[
\langle \dot{B}_z \rangle_{i,j,k} (t) = \frac{1}{\delta A_{zim,j,k}} \left( -\delta y [h_{yim,j,k}]_{m}^p + \delta x [h_{xim,j,k}]_{m}^p \right),
\]

and describes the time rate of change of face-averages

\[
\langle B_x \rangle_{i,j,k} (t) = \frac{1}{\delta A_{xim,j,k}} \int_{\delta A_{xim,j,k}} B_x (x, t) dA,
\]

\[
\langle B_y \rangle_{i,j,k} (t) = \frac{1}{\delta A_{yim,j,k}} \int_{\delta A_{yim,j,k}} B_y (x, t) dA,
\]

\[
\langle B_z \rangle_{i,j,k} (t) = \frac{1}{\delta A_{zim,j,k}} \int_{\delta A_{zim,j,k}} B_z (x, t) dA,
\]
where \( \langle \cdot \rangle^{x(y,z)} \) means face-averaging over \( \delta A_{x(y,z)} \). The divergence-free condition in discrete form becomes

\[
\frac{d}{dt} \left( \langle B_x \rangle^{x}_{x,j,k} \delta A_{x,y,k} \right)_{i,p} + \left[ \langle B_y \rangle^{y}_{y,i,k} \delta A_{y,z,k} \right]_{j,m} + \left[ \langle B_z \rangle^{z}_{z,i,j} \delta A_{z,x,k} \right]_{k,m} = 0.
\]

The edge-averaged electric field \( \mathcal{E} \) appearing in equations (2.3) is approximated with the help of a semidiscrete evolution-projection method originally developed for discretizing equations of Hamilton–Jacobi type [7], [18]. The resulting formulae are each equivalent to a composition of one-dimensional (1D) HLL Riemann solvers in the respective 2D subspaces reflecting the 2D nature of the individual components of the induction equation (see equations (20)–(22) in [33]). Also, this approach naturally reduces to a correct 1D upwinding for grid-aligned flows (see also [11], [12] for an alternative ansatz).

The reconstruction functions for the magnetic field components are given by

\[
\begin{align*}
\tilde{B}_x(x) &= \frac{1}{2} \left( \langle B_x \rangle^{x}_{x,j,k} + \langle B_x \rangle^{z}_{x,m,j,k} \right) + \left[ \langle B_x \rangle^{x}_{x,i,j,k} \right]_{i,m} \frac{x-x_i}{\Delta x} \\
&\quad + \left[ \left( \partial_x \tilde{B}_x \right)_{i,p,j,k} \frac{x-x_{i,m}}{\Delta x} + \left( \partial_y \tilde{B}_x \right)_{i,m,j,k} \frac{x_{i,p}-x}{\Delta x} \right] \left( y - \langle y \rangle \right) \\
&\quad + \left[ \left( \partial_z \tilde{B}_x \right)_{i,p,j,k} \frac{x-x_{i,m}}{\Delta x} + \left( \partial_z \tilde{B}_x \right)_{i,m,j,k} \frac{x_{i,p}-x}{\Delta x} \right] \left( z - z_k \right), \\
\tilde{B}_y(x) &= \left[ \left( \partial_x \tilde{B}_y \right)_{i,p,j,k} \frac{y-y_{i,m}}{\Delta y} + \left( \partial_x \tilde{B}_y \right)_{i,j,m,k} \frac{y_{j,p}-y}{\Delta y} \right] \left( x - \langle x \rangle \right) \\
&\quad + \frac{1}{2} \left( \langle B_y \rangle^{y}_{y,j,p,k} + \langle B_y \rangle^{y}_{y,i,m,k} \right) + \left[ \langle B_y \rangle^{y}_{y,i,j,k} \right]_{j,m} \frac{y-y_j}{\Delta y} \\
&\quad + \left[ \left( \partial_x \tilde{B}_y \right)_{i,j,p,k} \frac{y-y_{i,m}}{\Delta y} + \left( \partial_z \tilde{B}_y \right)_{i,j,m,k} \frac{y_{j,p}-y}{\Delta y} \right] \left( z - z_k \right), \\
\tilde{B}_z(x) &= \left[ \left( \partial_x \tilde{B}_z \right)_{i,j,k,p} \frac{z-z_{k,m}}{\Delta z} + \left( \partial_x \tilde{B}_z \right)_{i,j,k,m} \frac{z_{k,p}-z}{\Delta z} \right] \left( x - \langle x \rangle \right) \\
&\quad + \left[ \left( \partial_y \tilde{B}_z \right)_{i,j,k,p} \frac{z-z_{k,m}}{\Delta z} + \left( \partial_y \tilde{B}_z \right)_{i,j,k,m} \frac{z_{k,p}-z}{\Delta z} \right] \left( y - \langle y \rangle \right) \\
&\quad + \frac{1}{2} \left( \langle B_z \rangle^{z}_{z,i,j,k} + \langle B_z \rangle^{z}_{z,i,j,k} \right) + \left[ \langle B_z \rangle^{z}_{z,i,j,k} \right]_{i,m} \frac{z-z_k}{\Delta z},
\end{align*}
\]

where \( \partial_x \tilde{B}_z \) and other cross derivative terms are slope-limited approximations to the exact derivatives. \( \langle x \rangle \), \( \langle y \rangle \), \( \langle z \rangle \) is the face-averaged \( x(y,z) \)-coordinate where averaging is over the cell face \( \delta A_y \), \( \delta A_x \), \( \delta A_z \). As can be shown by explicit
calculation this $\mathbf{B}$-reconstruction is flux-conserving in the sense that

$$
\frac{1}{\delta A_{i,m,j,k}} \int_{\delta A_{i,m,j,k}} \tilde{B}_x(x) dA = \langle B_x \rangle_{i,m,j,k},
$$

$$
\frac{1}{\delta A_{y,i,m,k}} \int_{\delta A_{y,i,m,k}} \tilde{B}_y(x) dA = \langle B_y \rangle_{y,i,m,k},
$$

and is consistent with the solenoidality property of the CT scheme,

$$
\nabla \cdot \mathbf{B} = \nabla \cdot \langle \mathbf{B} \rangle = \left( \frac{\partial}{\partial x} \langle B_x \rangle \right)_{i,m,j,k} + \left( \frac{\partial}{\partial y} \langle B_y \rangle \right)_{y,i,m,j} + \left( \frac{\partial}{\partial z} \langle B_z \rangle \right)_{z,i,j,m,k}.
$$

The semidiscrete scheme (2.1) and (2.3) is finally solved using a time-explicit Runge–Kutta integrator. NIRVANA has the choice between the standard second-order accurate Runge–Kutta method and the more stable third-order method of [27].

3. AMR on curvilinear-orthogonal grids. A major technical task any block-structured AMR implementation has to master concerns the management of the grid hierarchy which consists of a set of individual grid blocks with different resolutions. For efficiency reasons it is inevitable that such a block management system also provides information on the connectivity of a block with neighboring blocks, child blocks, and parent blocks. Further problems one has to deal with are associated with the FV framework for the MHD system of conservation laws. Important issues are (i) how to evolve the grid hierarchy one time-step within the Runge–Kutta multistage integration process, (ii) how to match solutions at coarse-fine grid interfaces in a manner consistent with the FV approach, and (iii) how to transfer the coarse grid solution to a generated finer grid. In two previous papers by Ziegler [31], [32] all such aspects have already been discussed for AMR on Cartesian grids. Because they build upon the same principles, major parts of the Cartesian AMR implementation carry over to the more general case of AMR on curvilinear-orthogonal grids. A summary of the basic ideas is given below. However, changes to the algorithm become necessary in the flux and electric field synchronization process at coarse-fine grid interfaces as well as in the restriction and prolongation operations of solution transfer. In fact, the prolongation algorithm for the magnetic field used in Cartesian geometry failed in preserving the solenoidality property on general curvilinear-orthogonal grids and, therefore, has been replaced by a new hierarchical prolongation approach. The failure is because the Cartesian algorithm relies on simpler interpolation violating flux conservation on general curvilinear-orthogonal grids. Such modifications are described in more detail.

3.1. General strategy. Block-structured AMR in NIRVANA is based on the use of generic blocks having a fixed size of four cells per coordinate direction. In the beginning a set of base level blocks spans the computational domain. Refinements are then achieved by a hierarchical nesting of blocks assuming a fixed refinement ratio of 2 between adjacent refinement levels in all coordinate directions. A 2D (3D) finer block always covers a quad (oct) of $2 \times 2$ ($2 \times 2 \times 2$) underlying coarser block cells. A system of nested blocks on each base level block is established building a block-tree structure. Block connectivity within such a tree as well as connectivity to spatially neighboring blocks is efficiently handled by pointers mediating instant data access. Load balancing in parallel simulations is achieved by an equal distribution of blocks among threads following space-filling curve techniques [26]. A space-filling curve of
Hilbert type serves for ordering base level blocks and an adapted curve of Morton type for the order of blocks within a base block’s tree.

A finer block is generated according to some mesh refinement criteria. The standard criterion in the NIRVANA code is a convex combination of criteria based on the relative strengths of first (respectively, second) derivatives of certain variables $u$. The expression for Cartesian AMR is detailed in [32]. It needs only slight adaptation for curvilinear AMR. According to this standard criterion, mesh refinement takes place if a prescribed threshold is exceeded. Those $u$-thresholds are usually somewhat problem-dependent parameters. In addition to the derivatives-based criterion, mesh refinement in NIRVANA may also be controlled by a Jeans-length-based criterion in case of self-gravitational flows or a Field-length-based criterion for flows containing thermally unstable gas. Here, in the test problems the standard criterion is used with the density, velocity, and magnetic field components being refinement triggering variables as the default.

Although the use of small-sized, self-similar blocks has clear advantages in terms of grid adaptivity, mesh organization, and load balancing issues, advancing the solution on a large number of small blocks reduces computational efficiency due to the increased synchronization overhead at block interfaces. In order to speed up integration, blocks of a given refinement level are therefore temporarily clustered to larger-sized patches. In parallel simulations the clustering is done separately for each domain partition. Ghost cells are added for patches in place of generic blocks. The discretized equations are then solved on these patches. The solution is copied back to generic blocks after a time-step is complete, and patches are then deallocated. Note that the additional overhead due to clustering is usually substantially overcompensated by the gain in computational speed. The adaptive grid is advanced with a single time-step given by the minimum of CFL-related time-steps computed over the full patch hierarchy; hence, there is no refinement in time. Within the multistage Runge-Kutta integration scheme the full patch hierarchy is updated stage-by-stage. Each patch is treated as a separate initial value problem with given boundary conditions obtained from grid neighbors (either copied from equal level grids or interpolated from coarser grids) or defined at physical boundaries. After each Runge-Kutta stage a synchronization of ghost cells takes place at intralevel patch interfaces to retrieve solution consistency. In particular, because of the staggering of magnetic field components those components at patch interfaces living on different patches, local or remote, must have unique values without ambiguity. In a restriction operation the less accurate coarse grid solution is replaced by the fine grid solution in overlapping regions. A flux and electric field correction step is necessary at fine-coarse grid interfaces to recover conservation and to maintain the magnetic field solenoidality constraint.

3.2. Prolongation operation. For cell-averaged quantities a conservative prolongation of the coarse grid solution onto the cells of a newly generated finer block can be constructed entirely on the basis of the reconstruction formula (2.2) for coarse cells. For a coarse cell $C_{i,j,k}^c$ covered by the oct (in three dimensions) of finer cells $C_{l,m,n}^f$ with running indices $l, m, n$, conservation is expressed by

$$\nabla_i \phi_c = \sum_{l,m,n} \nabla_i \phi_f ,$$

where a superscript $c$ ($f$) marks a coarse (fine) cell quantity and $\nabla_i \phi_c = \sum_{l,m,n} \nabla_i \phi_f$. Fine grid cell-averaged values $\nabla_i \phi_f$ obeying (3.1) are simply given by $\nabla_i \phi_f = \nabla_i \phi_c$. 
With the assumption that each of the two an intermediate (superscript $I$) 

The proof is straightforward:

$$
\sum_{l,m,n} \tau_l \partial_l V_{l,m,n}^c - \tau_l \partial_l V_{i,j,k}^c = \sum_{l,m,n} \tau_l \partial_l V_{l,m,n}^c - \int_{C_{i,j,k}} x dV
$$

The prolongation procedure for the staggered magnetic field components is a bit more complex. Here, in order to ensure magnetic field solenoidality in each new fine cell of the oct it is not sufficient to solely take point values from the coarse grid reconstruction at the proper fine cell positions. However, the formulae (2.4) are used to compute fine cell values at a coarse cell’s faces wherever there is no neighboring fine grid from which values are naturally copied. Accordingly, flux-conserving new fine values at the coarse cell faces are given by the point values of B-reconstruction at proper positions, i.e.,

$$
\langle B^f_{x} \rangle_{l,m,n} = \tilde{B}^c_{x}(x_{l,m}, y_{m,n}, z_{n}) ,
\langle B^f_{y} \rangle_{l,m,n} = \tilde{B}^c_{y}(x_{l,m}, y_{m,n}, z_{n}) ,
\langle B^f_{z} \rangle_{l,m,n} = \tilde{B}^c_{z}(x_{l,m}, y_{m,n}, z_{n}) .
$$

The situation is illustrated in Figure 3.1(a). It remains to compute the 12 magnetic field values at the inner fine cell faces. This is done using a hierarchical prolongation procedure as follows. In a first step the coarse cell is virtually divided into two subcells with the following plane $x = x_i$ through the geometrical cell center (Figure 3.1(b)). With the assumption that each of the two $x$-halves is divergence-free in an FV sense an intermediate (superscript $I$) face-averaged value $\langle B^f_{I} \rangle_{i,j,k}$ at $x_i$ can be calculated from the divergence condition applied to any half. Taking, for instance, the $x$-lower half the condition reads as

$$
\langle B^f_{x} \rangle_{l,m,n} = \tilde{B}^c_{x}(x_{l,m}, y_{m,n}, z_{n}) ,
\langle B^f_{y} \rangle_{l,m,n} = \tilde{B}^c_{y}(x_{l,m}, y_{m,n}, z_{n}) ,
\langle B^f_{z} \rangle_{l,m,n} = \tilde{B}^c_{z}(x_{l,m}, y_{m,n}, z_{n}) .
$$

where the brackets contain the previously computed fine cell values at the coarse cell
Fig. 3.1. Magnetic field prolongation strategy: (a) fine cell values at coarse cell faces, (b) prolongation of interior fine cell $x$-component, (c) prolongation of interior fine cell $y$-component, (d) prolongation of fine cell $z$-component.

Then, the four fine cell $x$-components of the magnetic field at $x_i$ are given by

$$\langle B^f_{x,i,m,n} \rangle_x = \tilde{B}^i_x(\langle y \rangle_{x,i}, z_n),$$

where $\tilde{B}^i_x(y, z)$ is a flux-conservative 2D $B_x$-reconstruction in the plane $x = x_i$. It is determined from the set $\{ \langle B^f_{x,i,j,k} \rangle_x \}$ and, by construction, is equipped with the property $\langle B^f_{x,i,j,k} \rangle_x = \tilde{B}^i_x(\langle y \rangle_{x,j}, z_k)$. Indeed, it can be shown that $\sum_{m,n} \langle B^f_{x,i,m,n} \rangle_x \delta A^f_{x,m,n} =$
approaches may differ in design details. A completely different ansatz for a divergence-transfer of the coarse grid magnetic field solution onto the fine grid. Note that the complete prolongation operation which, by construction, is a divergence-free dimension) of fine cell faces covering a coarse cell face:

In the next step the coarse cell is further divided with the cutting plane this time at \( y = y_j \), resulting in four subcells (Figure 3.1(c)). Again, applying the divergence condition to any quarter of the previous halves gives the two intermediate values \( \langle B^x_i \rangle_{i\pm\frac{1}{2},j,k} \). Taking, for instance, the \( y \)-lower subcell in the \( x \)-lower half the condition reads as

\[
\sum_i \langle B^x_i \rangle_{i,m,n} \delta A^f_{x,m,n} + \langle B^y_{y_i} \rangle_{i-\frac{1}{2},j,k} \delta A^f_{y,-i-\frac{1}{2},j,k} - \sum_n \langle B^y_{y_i} \rangle_{i,m,n} \delta A^f_{y,m,n} + \langle B^z_{z_i} \rangle_{i,m,n} \delta A^f_{z,m,n} = 0 ,
\]

where in \( \langle B^y_{y_i} \rangle_{i-\frac{1}{2},j,k} \) the average is taken over the area \( \delta A^f_{y,-i-\frac{1}{2},j,k} \) with half \( x \)-range \([x_{im}, x_{i}]. \) Note that this step needs the previously computed fine values \( \langle B^x_i \rangle_{i,m,n} \). With the help of the computed sets \{\( \langle B^y_{y_i} \rangle_{i-\frac{1}{2},j,k} \}\}, flux-conservative 1D reconstruction functions \( B^f_{y_i}(z) \) along the \( z \)-direction and through points \( (x_i, y_j) \) can be constructed equipped with the property \( \langle B^y_{y_i} \rangle_{i,\pm\frac{1}{2},j,k} = \tilde{B}^f_{y}(z_k) \). Then, the four fine cell \( y \)-components of the magnetic field at \( y_j \) are simply given by taking point values of the respective reconstruction function:

\( \langle B^y_{y_i} \rangle_{i,m,n} = \tilde{B}^f_{y}(z_n) \).

Finally, cutting the coarse cell additionally at \( z = z_k \) the oct of fine cells results (Figure 3.1(d)). The remaining four fine cell \( z \)-components of the magnetic field at \( z_k \) are obtained directly by an application of the divergence condition in each of the four octants in either the \( z \)-lower half or \( z \)-upper half of the coarse cell. This completes the prolongation operation which, by construction, is a divergence-free transfer of the coarse grid magnetic field solution onto the fine grid. Note that the presented prolongation strategy is similar to the one described in [19], although both approaches may differ in design details. A completely different ansatz for a divergence-free reconstruction is used in the polynomial methods of [28], [1].

### 3.3. Flux and electric field correction

A conservative restriction operation is applied when transferring the more accurate fine grid solution in a hierarchical manner onto the coarser grid. For cell-averaged quantities the formula is given by (3.1). As a consequence the computed coarse grid flux and electric field components lying directly at coarse-fine grid interfaces are no longer consistent with the copied solution. In order to ensure consistency and, thus, to recover conservation and magnetic field solenoidality a correction step is necessary. Obviously, the coarse grid fluxes at those interfaces must be replaced by the sum of fine grid fluxes for the quad (in three dimensions) of fine cell faces covering a coarse cell face:

\[
F^c_{x_i,m,j,k} \rightarrow F^c_{x_i,m,j,k}^{corr} \quad \delta A^f_{x_i,m,j,k} \rightarrow \sum_{m,n} \delta A^f_{x_l,m,n} F^f_{x_l,m,n} ,
\]

\[
F^c_{y_i,m,j,k} \rightarrow F^c_{y_i,m,j,k}^{corr} \quad \delta A^f_{y_i,m,j,k} \rightarrow \sum_{i,n} \delta A^f_{y_l,m,n} F^f_{y_l,m,n} ,
\]

\[
F^c_{z_i,j,k_m} \rightarrow F^c_{z_i,j,k_m}^{corr} \quad \delta A^f_{z_i,j,k_m} \rightarrow \sum_{l,m} \delta A^f_{z_l,m,n} F^f_{z_l,m,n} .
\]

In analogy, the edge-centered coarse grid electric field components must be replaced by the sum of fine grid values for the double of fine cell edges covering a coarse cell.
edge:

\[ E_c^{c, corr}_{x_1,j,m,k} \rightarrow E_c^{corr}_{x_1,j,m,k}, \quad \delta E_c E_c^{corr}_{x_1,j,m,k} = \sum_l \delta E_l E_{x_1,m,n,m}, \]

\[ E_c^{corr}_{y_1,m,j,k} \rightarrow E_c^{corr}_{y_1,m,j,k}, \quad \delta y E_c E_{y_1,m,j,k} = \sum_m \delta y_m E_{y_1,m,n,m}, \]

\[ E_c^{corr}_{z_1,m,j,k} \rightarrow E_c^{corr}_{z_1,m,j,k}, \quad \delta z E_c E_{z_1,m,j,k} = \sum_n \delta z_n E_{z_1,m,n,m}. \]


4.1. Shock-cloud interaction. In a first example the interaction of a planar shock wave with a cloud is simulated in order to demonstrate the functionality of the AMR implementation for cylindrical geometry in axisymmetric mode. For that purpose a computational region \((z,R) \in [-2 \cdot 10^{17}, 6 \cdot 10^{17}] \times [0, 2 \cdot 10^{17}] \text{m}^2\) is discretized by an \(N_z \times N_R = 256 \times 64\) base grid. At most two mesh refinement levels are added leading locally to a four times higher resolution compared to the base grid. In this test problem AMR is triggered solely by scaled gradients and second derivatives of a passive tracer variable \(C\) which obeys the additional equation

\[ \partial_t C + \mathbf{v} \cdot \nabla C = 0, \quad C(t = 0) = \begin{cases} 1 & \text{inside the cloud,} \\ 0 & \text{otherwise,} \end{cases} \]

hence identifying cloud material. A uniform grid run with \(N_z \times N_R = 1024 \times 256\) has also been performed for comparison. Both uniform grid and adaptive grid simulations are done in parallel on 64 processors. The AMR run is about 5.5 times faster than the uniform grid run. Its total overhead including overhead due to communication between processors and intrinsic AMR overhead is roughly 30%. Owing to the fact that a relatively small problem is solved expressed by a low computational load per MPI thread, the observed amount of overhead is more than acceptable.

The physical setting of the problem is as follows: a planar Mach 30 shock propagating in the \((+z)\)-direction and initially located at \(z = -10^{17} \text{m}\) approaches a spherical gas clump with radius \(r_{cl} = 6 \cdot 10^{16} \text{m}\). The cloud with center at \(z = 0\) has a mass density of \(\rho_{cl} = 1.84 \cdot 10^{-28} \text{kg/m}^3\) and a temperature of \(T_{cl} = 116 \text{K}\) and is at rest. The cloud is embedded in a medium with density \(\rho_0 = 2.14 \cdot 10^{-22} \text{kg/m}^3\) and temperature \(T_0 = 10^4 \text{K}\). The cloud and ambient medium are in pressure equilibrium. Given the shock Mach number and the ambient gas parameters the conditions behind the shock \((z < -10^{17} \text{m})\) can be computed from the Rankine–Hugoniot formulae assuming an adiabatic index \(\gamma = 5/3\). In addition, the system is penetrated by a constant magnetic field in the \(z\)-direction with a field strength of \(10^{-10} \text{T}\).

Figure 4.1 shows a zoom-in of the full cylindrical domain. The first plot (counting from left to right) illustrates the initial conditions where the shock is just at the lower \(z\)-boundary. The next two plots show the density structure in logarithmic scale after the final time \(t = 2 \cdot 10^8 \text{s}\). The block distribution of the adaptive mesh with a total of \(\approx 1700\) blocks belonging to that time is shown in the fourth plot. Recall that, here, a block is a rectangular collection of \(4 \times 4\) numerical cells. There is an overall excellent qualitative agreement of the AMR run (third plot) with the single grid run (second plot) demonstrating the functionality of the AMR algorithm for 2D cylindrical geometry. As expected, the incident shock has crushed the cloud while accelerating it in the \(z\)-direction. The passage of the shock has also provoked a compression of the longitudinal magnetic field behind the cloud around the axis. There is an associated
confinement of cloud material originating from the cloud surface but torn out by the action of instabilities. This confinement of cloud matter causes mesh refinement at the same place, as can be clearly seen in the block distribution at $z \gtrsim 1.7 \cdot 10^{17} \text{m}$. The reflected bow shock located in the upstream direction and the secondary oblique shock in the downstream direction are not covered by refinements because mesh refinement criteria have been intentionally restricted to spatial changes in the tracer variable. Hence, those features are less sharp compared with the high-resolution single grid run.

### 4.2. Rising hot bubble.

In the rising bubble problem a hot, rarefied gas bubble is placed in a spherically symmetric, isothermal atmosphere. The atmosphere is exponentially stratified in radial direction $r$. In dimensionless units density and pressure of the atmosphere are given by $\varrho = p = \exp(-r)$, where a sound speed $c = 1$ and external gravity $g = -e_r$ is assumed. The gravity term is added as a source term to the momentum equation. The density inside the bubble is 10% of the atmospheric density, and its temperature is a factor 10 higher than the atmospheric temperature, hence, assuming the system is in initial pressure equilibrium. The bubble is assumed to have spherical shape with a radius of $0.5$. Driven by buoyancy forces the bubble moves radially outwards towards lower density. The fate of the bubble depends on the strength of an imposed magnetic field. A weak magnetic field may not be able to suppress the Rayleigh–Taylor instability, whereas a strong magnetic field may have a stabilizing effect preventing the development of small-scale structures at the bubble's surface. First, 2D axisymmetric simulations in both cylindrical and spherical geometry and for both a weak and a strong magnetic field initially oriented along the
geometric axis are performed. Second, a 3D nonaxisymmetric model in a spherical wedge is considered with a magnetic field imposed in the r-direction. All simulations make use of an adiabatic equation of state with $\gamma = 5/3$. The value of the magnetic permeability is set to the vacuum permeability $\mu = \mu_0 = 4\pi \cdot 10^{-7}$.

**4.2.1. 2D axisymmetric models.** In the cylindrical grid case the domain $(z,R) \in [0.2,3.8] \times [0,1.44]$ is discretized by a $N_z \times N_R = 128 \times 48$ base grid. AMR is restricted to having at most three additional refinement levels leading to a maximum effective resolution equivalent to a $1024 \times 384$ grid. The bubble is initially located at coordinates $(1,0)$. The magnetic field is along the $z$-direction, $B = B_0 e_z$, with strength $B_0 = 4 \cdot 10^{-4}$ in the weak field case and $B_0 = 4 \cdot 10^{-1}$ in the strong field case.

In the spherical grid case the domain $(r,\theta) \in [0.2,3.8] \times [0,\pi/3]$ is discretized by an $N_r \times N_\theta = 128 \times 64$ base grid. Again, a maximum of three additional refinement levels have been allowed. The initial bubble position is $(1,0)$. In order to ensure a divergence-free representation on the spherical grid, the axis-aligned magnetic field, $(B_r, B_\theta, B_\phi) = (B_0 \cos \theta, -B_0 \sin \theta, 0)$, is discretized according to

$$\langle B_r \rangle_{r,m,j} = -\frac{B_0}{2} \sin^2 \theta_j \frac{\cos \theta_{jm} - \cos \theta_{jm}}{\sin \theta_j}$$

$$\langle B_\theta \rangle_{\theta,r,m} = -B_0 \sin \theta_{jm},$$

where $B_0$ adopts the same values as in the cylindrical model.

The resulting density structure for both geometries is shown in Figure 4.2. The plots correspond to an evolution time $t = 1.5$. The density contours have been overlaid by the block distribution of the adaptive grid. During its rise through the atmosphere the spherical bubble becomes elongated. As expected, in the weak field case the action of Rayleigh–Taylor instability leads to corrugation of the bubble surface and leads to vortex generation, whereas the presence of a stronger field suppresses the instability. In the strong field case the rarefied, hot bubble takes on the shape of a bullet. The overall bubble evolution proceeds similarly in both grid geometries. In the weak field case differences are seen in the details of surface corrugation. In the strong field case the curvature of the tail of the bubble is slightly larger on the spherical grid. Obviously, details in the structure are influenced by the grid anisotropy. Generally, the bubble surface is captured by the highest refinement level. The relative divergence error measured in a 1-norm is less than $10^{-13}$ for the cylindrical grid models and less than $10^{-11}$ for the spherical grid models.

As already demonstrated in [33] the numerical solutions in Cartesian, cylindrical, and spherical geometry for this bubble problem agree very well on a qualitative level. This is likewise true here for adaptive cylindrical and spherical grids. In order to be more quantitative, the convergence behavior of the magnetic bubble problem in spherical geometry has been analyzed. For that purpose the relative errors of the (positive) mass density and radial magnetic field in the 1-norm,

$$ERR(u) = \frac{1}{\# grid points} \sum_{grid} \frac{\left| \frac{u - u_{ref}}{u_{ref}} \right|}{\left| \frac{u - u_{ref}}{u_{ref}} \right|}, \quad u = \{\varrho, B_r\}$$

have been computed on uniform grids $256 \times 128$, $512 \times 256$ and $1024 \times 512$ as well as on an adaptive grid with $128 \times 64$ base resolution plus two or three additional refinement levels. A $2048 \times 1024$ uniform grid simulation provides the reference solution $u_{ref}$. The
result is summarized in Table 4.1. The measured order of (self-)convergence on the uniform grid is below second order, which might be expected because of the presence of the contact discontinuity. The errors on the adaptive grid are worse but not too far from the values of their equivalent uniform grids. This might also be expected because the contact discontinuity is fully covered by the adaptive mesh but not the magnetoacoustic waves propagating away from the bubble.
Table 4.1
Relative error in mass density and radial magnetic field.

<table>
<thead>
<tr>
<th>Grid</th>
<th>256 \times 128</th>
<th>512 \times 256</th>
<th>1024 \times 512</th>
<th>128 \times 64 + 2</th>
<th>128 \times 64 + 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR(\rho)</td>
<td>4.1(−2)</td>
<td>1.93(−2)</td>
<td>6.5(−3)</td>
<td>2.30(−2)</td>
<td>1.09(−2)</td>
</tr>
<tr>
<td>ERR(B_r)</td>
<td>7.6(−3)</td>
<td>3.6(−3)</td>
<td>1.3(−3)</td>
<td>6.2(−3)</td>
<td>5.1(−3)</td>
</tr>
<tr>
<td>Order</td>
<td>1.2/1.1</td>
<td>1.6/1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2.2. 3D nonaxisymmetric model. Here, the rising bubble problem is solved on a spherical wedge \((r, \theta, \phi) = [0.2, 3.8] \times [\pi/20, 9\pi/20] \times [-\pi/4, \pi/4]\) discretized by an \(N_r \times N_\theta \times N_\phi = 96 \times 48^2\) base grid. Three additional levels of refinement allow one to achieve an effective resolution equivalent to a \(768 \times 384^2\) grid. The bubble is located initially at coordinates \((1, \pi/4, 0)\). The atmosphere is threaded by a purely radial magnetic field given by

\[
\langle B_r \rangle_{im,j,k}^r = \frac{0.016}{r^2_{im}}.
\]

Note that the field strength varies from 0.4 (strong field) at \(r = 0.2\) to a value of \(\approx 10^{-3}\) (weak field) at \(r = 3.8\). The solution after a time \(t = 1.5\) is illustrated in Figure 4.3. The bubble has deformed into a cap-like shape, and its contact interface has developed irregularities due to the action of surface instabilities. Again, the relative divergence error measured in the 1-norm is considerably below the typical discretization error but is of the same order as in the 2D model. The contact interface is completely covered by the highest level of refinement. The total number of blocks at the given evolutionary time is about \(10^5\), corresponding to a total number of \(\approx 6\) million grid points.

For the present 3D problem the parallel efficiency of the AMR algorithm has been evaluated using a PC cluster with AMD Opteron 16-core CPUs and Infiniband connection. At a fixed problem size a series of simulations with an increasing number of CPU cores from 16 up to 512 have been done (strong scaling analysis). The obtained
scaling behavior is catalogued in Table 4.2. Despite a relatively low load per MPI thread the simulation on 256 cores reaches a respectable efficiency of 74% compared to 16 cores. On 512 cores the communication overhead has increased substantially and the efficiency drops to 39%. From the timings the turnover where computational gain is no longer expected lies somewhat above but close to 512 cores.

4.3. Alfvén wings. The following test problem has been adapted from [16]. The initial state consists of a constant density ($\rho = 1$) flow in the Cartesian $z$-direction overlaid by a constant magnetic field in the Cartesian $x$-direction. Hence, flow direction and magnetic field direction are perpendicular. An isothermal equation of state is used. The flow speed $v_0$, Alfvén speed $c_A$, and sound speed $c_s$ are given by $v_0 = c_A = 1$ and $c_s = 10$, respectively. Using a magnetic permeability $\mu = 1$ the magnetic field strength is $B_0 = 1$. The flow velocity is then subject to a perturbation

$$\mathbf{v} \rightarrow \alpha \mathbf{v},$$

where $\alpha$ is a time- and space-dependent scaling function given by

$$\alpha = 1 - \min(10t, 1) \left(1 - \tanh\left(4 \max(2|x - x_0| - 1, 0)\right)\right).$$

Technically, each velocity component is scaled by this function after each time-step. The perturbation with origin at Cartesian coordinates $x_0 = (18, 0, -5)$ acts as an obstacle launching Alfvén waves which propagate along two rays in the $xz$-plane and at angles $\pm \pi/4$ relative to the $z$-direction.

The problem is solved on a spherical domain $(r, \theta, \phi) \in [5, 30] \times [\pi/4, 3\pi/4] \times [-\pi/4, \pi/4]$ spanned by a base grid of $N_r \times N_\theta \times N_\phi = 64^3$. A maximum of two additional refinement levels has been allowed. At all boundaries variables are left at their initial values. In spherical coordinates the flow is given by $(v_r, v_\theta, v_\phi) = (\cos \theta, -\sin \theta, 0)$. The magnetic field is discretized according to

$$\langle B_r \rangle_{r,m,j,k} = -B_0 \sin^2 \theta_j \frac{\Delta \theta}{\cos \theta_j \cos \theta_j - \cos \theta_j m} \sin \phi_k - \sin \phi_k m,$$

$$\langle B_\theta \rangle_{i,j,m,k} = B_0 \cos \theta_j m \sin \phi_j - \sin \phi_j m,$$

$$\langle B_\phi \rangle_{i,j,k,m} = -B_0 \sin \phi_j \left(2 \sin^2 \theta_j + \frac{\sin \theta_j \cos \theta_j m - \sin \theta_j m \cos \theta_j m}{\Delta \theta}\right),$$

which is a solenoidal representation of the constant $B_x$-field on the spherical grid. The simulation is stopped at $t = 7.75$ before the waves leave the computational domain. The final Alfvén wave pattern is illustrated in Figure 4.4. It shows the expected
Fig. 4.4. Left: Alfvén wings in 3D space represented by an isocontour $|v| = 0.9$. Right: Zoom-in showing the contour lines of $|v|$ and the block distribution in the $\phi = 0$ plane. The plots belong to a time $t = 7.75$.

A wing-like structure which is symmetric with respect to the $\phi = 0$ plane and with the two arms building an angle of 90 degrees. The block distribution closely follows this structure, as can be seen in the zoom-in plot of Figure 4.4. However, in order to enable this optimal grid adaptivity a fine tuning of refinement parameters was necessary. Here, mesh refinement was triggered by normalized values of gradients and second derivatives in the gas density, momentum density, and magnetic field.

4.4. MHD blast wave. As a last test the propagation of a spherical blast wave in a constant magnetic field is considered. The problem is solved in a ring using both cylindrical and spherical coordinates. In spherical geometry the computational domain is given by $(r, \theta, \phi) \in [2.5, 4.5] \times [0.43\pi, 0.57\pi] \times [0, 2\pi]$ with the explosion site at coordinates $(3.5, \pi/2, 0)$. The domain is discretized by $N_r \times N_\theta \times N_\phi = 32 \times 48 \times 256$ base grid points. In cylindrical geometry the computational domain is given by $(z, R, \phi) \in [-1, 1] \times [2.5, 4.5] \times [0, 2\pi]$ with the explosion site at coordinates $(0, 3.5, 0)$. The resulting remnant structure is resolved with up to three additional refinement levels. The blast wave is initiated by the deposit of a large amount of thermal energy (respectively, an overpressure) in a small volume around the explosion site. In a medium with constant density $\rho = 1$ the pressure is set to

$$p = \begin{cases} 5 \cdot 10^4 & \text{within sphere of radius } r_{\text{init}} \text{ around explosion site,} \\ 1 & \text{otherwise,} \end{cases}$$

where $r_{\text{init}} = 0.15$. The equation of state is the ideal gas law with $\gamma = 1.67$. The magnetic field is oriented in the Cartesian $z$-direction. In cylindrical geometry the magnetic field is simply $\mathbf{B} = B_0 \mathbf{e}_z$. In spherical geometry the magnetic field components are given by equations (4.1) supplemented by a zero $B_\phi$. A magnetic field strength of $B_0 = 100$ is adopted corresponding to a low plasma-$\beta$ ambient medium. The simulations were stopped at evolution time $t = 5 \cdot 10^{-3}$.

Figure 4.5 illustrates the solution. Very similar results are obtained for both types of geometries. The hot interior bubble becomes stretched along the geometric axis. The shell is surrounded by cap-like shocks which are resolved with the highest refinement level. The shell becomes much broader towards the midplane, where the
Fig. 4.5. Density structure and block distribution in the ring midplane and in the $\phi = 0$ plane with zoom-in around the remnant (top: cylindrical geometry, bottom: spherical geometry).
shocks degenerate to a magnetoacoustic wave. Since mesh refinement is triggered by gradients the broader shell is not resolved with the highest possible refinement level.

5. Conclusions. I have presented a block-based adaptive mesh refinement strategy for curvilinear-orthogonal systems with a focus on cylindrical and spherical grids. This new feature has become part of the NIRVANA software—an MPI-parallel code for compressible MHD—extending the primal Cartesian AMR code version. The AMR framework has been imposed on a hybrid Godunov-type central-upwind, CT scheme. A major challenge in the use of CT for AMR turned out to be the preservation of the solenoidality property of the magnetic field. This is because CT is based on an evolution of the staggered, face-averaged magnetic field components instead of cell-centered quantities. The problem of a divergence-free prolongation of the coarse grid solution onto generated finer grids has been solved by a hierarchical approach where a stepwise subdivision of the underlying coarse cell provides fine grid values utilizing the divergence-free condition in each stage of subdivision. By means of a couple of test simulations the AMR algorithm has been shown to work well in two and three dimensions. However, the high complexity of the algorithm currently prohibits nonaxisymmetric AMR simulations which include the geometric axis. This is due to the nonlocal character of boundary conditions at the axis and the presence of coordinate singularities. The latter imply severe problems in the mathematical treatment of those magnetic field components defined at degenerate grid cell faces. A future effort tends to tackle this issue.

REFERENCES


