BATS-R-US Design

Code Version 7.4.4

K.C. Hansen, Gábor Tóth, Aaron Ridley, Darren DeZeeuw

January 31, 2026

Contents

1 Numerical Discretization **5**

1.1 Grid . . . 5

1.1.1 Block Structure . . . 5

1.1.2 Adaptive Mesh Refinement (AMR) . . . 8

1.2 Time stepping . . . 8

1.2.1 Local time stepping . . . 8

1.2.2 Time accurate mode . . . 9

1.2.3 Point-Implicit Source Terms . . . 9

1.3 Numerical Schemes . . . 10

1.3.1 Finite Volume Discretization . . . 10

1.3.2 Roe flux . . . 11

1.3.3 Rusanov flux . . . 12

1.3.4 Linde flux . . . 12

1.3.5 First and Second Order Accuracy . . . 12

1.3.6 Slope Limiters . . . 12

1.4 Divergence of B Control . . . 13

1.4.1 8-wave Scheme . . . 14

1.4.2 Diffusive Control . . . 14

1.4.3 Projection Scheme . . . 14

1.4.4 Constrained Transport Scheme . . . 15

1.4.5 Choosing a Divergence B Control Scheme . . . 15

Chapter 1

Numerical Discretization

1.1 Grid

1.1.1 Block Structure

BATS-R-US uses a block based, adaptive grid that allows the user to specify where he or she wants more resolution or to let the code determine where more resolution is needed. This is one of the most advantageous features of the BATS-R-US code. While other codes use non-Cartesian meshes, they are often tuned to a very specific instance of a specific problem. BATS-R-US uses a highly parallelizable, adaptable grid which is suited to a wide variety of problems.

Mesh refinement techniques that adapt the computational grid to the solution of the governing PDEs can be very effective in treating problems with disparate length scales. Methods of this type avoid under resolving the solution in regions deemed of interest (e.g., high-gradient regions) and, conversely, avoid over resolving the solution in other less interesting regions (low-gradient regions), thereby saving orders of magnitude in computing resources for many problems. For typical solar wind flows, length scales can range from tens of kilometers in the near Earth region to the Earth-Sun distance (1.5×10^{11} m), and time scales can range from a few seconds near the Sun to the expansion time of the solar wind from the Sun to the Earth ($\sim 10^5$ s). The use of a mesh containing varying size cells is extremely beneficial and almost a virtual necessity for solving problems with such disparate spatial and temporal scales.

Keeping in mind the desire for high performance on massively parallel computer architectures, a relatively simple yet effective block-based AMR technique has been developed. Here the governing equations are integrated to obtain volume-averaged solution quantities within blocks of rectangular Cartesian computational cells. The computational cells are embedded in regular structured blocks of equal sized cells. The blocks are geometrically self-similar with dimensions $\tilde{\ell}_x \times \tilde{\ell}_y \times \tilde{\ell}_z$ and consist of $N_x \times N_y \times N_z$ cells, where $\tilde{\ell}_x$, $\tilde{\ell}_y$, and $\tilde{\ell}_z$ are the nondimensional lengths of the sides of the rectangular blocks and N_x, N_y, and N_z are even, but not necessarily all equal, integers. Typically, blocks consisting of anywhere between $4 \times 4 \times 4 = 64$ and $12 \times 12 \times 12 = 1728$ cells are used. Solution data associated with each block are stored in standard indexed array data structures. It is therefore straightforward to obtain solution information from neighboring cells within a block.

Computational grids are then composed of many self-similar blocks. Although each block within a grid has the same data storage requirements, blocks may be of different sizes in terms of the volume of physical space that they occupy (see Figure 1.1). Starting with an initial mesh consisting of blocks of equal size (i.e., equal resolution), adaptation is accomplished by the dividing and coarsening of appropriate solution blocks (see Figure 1.2). In regions requiring increased cell resolution, a "parent" block is refined by dividing itself into eight "children" or "offspring." Each of the eight octants of a parent block becomes a new block having the same number of cells as the parent and thereby doubling the cell resolution in the region of interest. Conversely, in regions that are deemed overresolved, the refinement process is reversed, and eight children are coarsened and coalesced into a single parent block. In this way, the cell resolution is reduced by a factor of 2. Standard multigrid-type restriction and prolongation operators are used to evaluate the solution on all blocks created by the coarsening and division processes, respectively.

Two neighboring blocks, one of which has been refined and one of which has not, are shown in Figure 1.1. Any

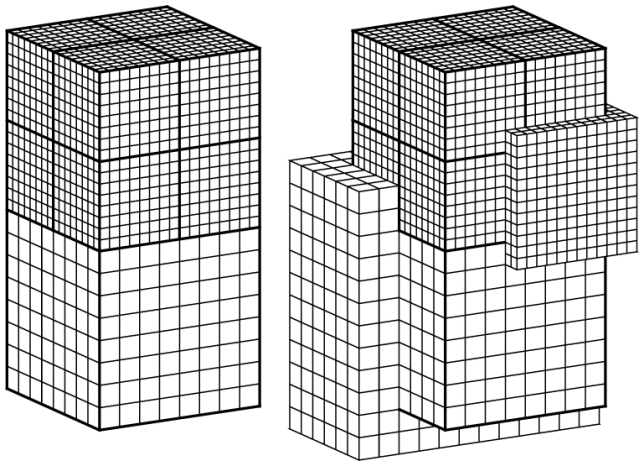

Figure 1.1: (left) Self-similar blocks used in parallel block-based AMR scheme. (right) Self-similar blocks illustrating the double layer of ghost cells for both coarse and fine blocks.

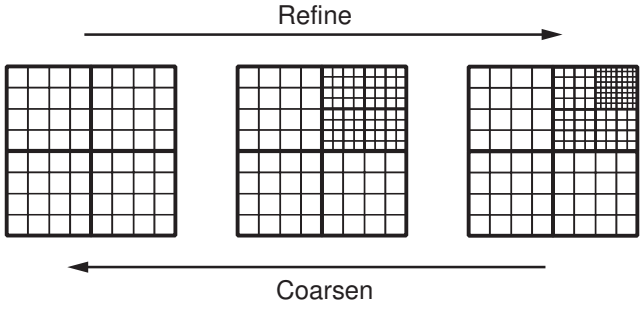

Figure 1.2: 2D example of the refining and coarsening of blocks that are 4x4 cells in size.

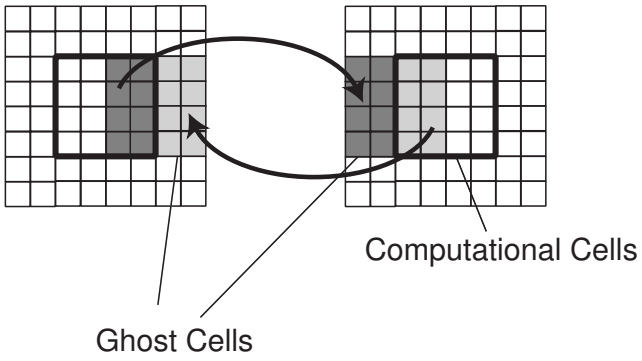

Figure 1.3: Computational cells and ghost cells for two neighboring blocks of the same resultion. Shaded cells indicate how ghost cells are filled from computational cells in the neighboring block.

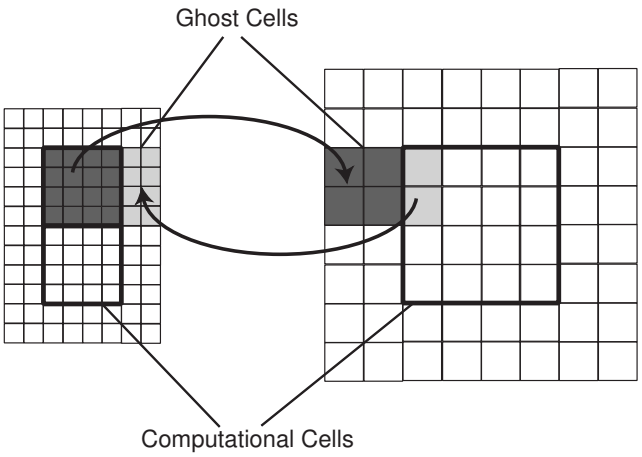

Figure 1.4: Computational cells and ghost cells for two neighboring blocks of the different resultion. Shaded cells indicate how ghost cells are filled from computational cells in the neighboring block.

of the blocks shown in Figure 1.1 can in turn be refined, and so on, leading to successively finer blocks. In the present method, mesh refinement is constrained such that the cell resolution changes by only a factor of 2 between adjacent blocks and such that the minimum resolution is not less than that of the initial mesh. In order that the update scheme for a given iteration or time step can be applied directly to all blocks in an independent manner, some additional solution information is shared between adjacent blocks having common interfaces. This information is stored in an additional two layers of overlapping "ghost" cells associated with each block as shown in Figures 1.1, 1.3 and 1.4. At interfaces between blocks of equal resolution (see Figure 1.3) these ghost cells are simply assigned the solution values associated with the appropriate interior cells of the adjacent blocks. At resolution changes (see Figure 1.4), restriction and prolongation operators, similar to those used in block coarsening and division, are employed to evaluate the ghost cell solution values. After each stage of the multistage time-stepping algorithm, ghost cell values are reevaluated to reflect the updated solution values of neighboring blocks. With the AMR approach, additional interblock communication is also required at interfaces with resolution changes to strictly enforce the flux conservation properties of the finite-volume scheme (see section 1.3.1).

The division of blocks into children creates a very hierarchical relation for blocks in the grid. This structure, called an "octree" because each block is divided into eight children, can be used to easily find neighboring blocks.

With AMR, as with any numerical implemention, there are trade-offs to be considered. Larger blocks have a lower ghost cell to computational cell ratio than smaller blocks. A 12x12x12 block has a ratio of 1.37 while a 4x4x4 block has a ratio of 7. This means that larger blocks have less "wasted" memory. In addition, with larger blocks, message passing communication time may be less. However, when doing refinement larger blocks tend to resolve not only the desired feature but other nearby areas, thereby using more cells than necessary. See also the discussion in the section on "Setting Grid Structure" in the USER MANUAL.

1.1.2 Adaptive Mesh Refinement (AMR)

One of the advantages of this hierarchical data structure is that it is relatively easy to carry out local mesh refinement at anytime during a calculation. If, at some point in a computation, a particular region of the flow is deemed to be sufficiently interesting, better resolution of that region can be attained by refining the solution blocks in that region, without affecting the grid structure in other regions of the flow. Reducing the grid resolution in a region is equally easy (see Figure 1.2). BATS-R-US allows the user to specify multiple physics based criteria to direct the coarsening and division of blocks. In particular, decisions as to when to refine or coarsen blocks are made based on comparisons of the maximum values of various local flow quantities determined in each block to specified refinement threshold values. As an example, three typical flow quantities or refinement criteria, ϵ_k, have the forms

$$\epsilon_1 \propto |\nabla \cdot \mathbf{u}| \quad \epsilon_2 \propto |\nabla \times \mathbf{u}| \quad \epsilon_3 \propto |\nabla \times \mathbf{B}| \ . \tag{1.1}$$

These quantities represent local measures of the compressibility and vorticity of the plasma as well as the electric current density. Note that the refinement thresholds are dynamically adjusted so as to exercise control over the total numbers of blocks, and hence cells, used in a calculation.

Adaptive mesh refinement, or AMR, is one of main features of BATS-R-US. It allows the user to create a mesh which has the desired number of cells, but which also resolves the features of interest.

1.2 Time stepping

BATS-R-US provides several options for the temporal discretization of the system of MHD equations (**??**)–(**??**), which can be written in the compact form

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathbf{F} = S \tag{1.2}$$

where W, F, and S represent the set of conservative variables, the corresponding fluxes and source terms, respectively. First of all, one can choose between time accurate mode and local time stepping. The latter mode should be used to find a steady state solution, since it converges much (10 to 15 times) faster than the time accurate mode. Both modes can be combined with a 1-stage and a 2-stage scheme, although we strongly suggest the use of the 1-stage scheme for local time stepping. In addition to the above choices, some of the source terms can be treated explicitly as well as implicitly.

1.2.1 Local time stepping

Local time stepping means that each computational cell uses a time step which is based on the local numerical stability criterion. Having different time steps for each cell corresponds to a discretization of

$$\frac{1}{a}\frac{\partial W}{\partial t} + \nabla \cdot \mathbf{F} = S \tag{1.3}$$

where a is the acceleration factor, which is a function of W. Clearly, if a steady state is found for the system of equations (1.3), the first term must be zero irrespective of the value of the acceleration factor a. **Therefore a steady state solution of the accelerated system of equations (1.3) is a steady state solution of the original equations (1.2) as well.**

The same holds on the discrete level if local time stepping is combined with a one-stage time stepping scheme. An explicit 1-stage (forward Euler) time discretization can be written as

$$W^{n+1} = W^n + \Delta t \left(-\nabla \cdot \mathbf{F}^n + S^n\right) \tag{1.4}$$

where Δt is the **local time step**. If a numerical steady state is found then $W^{n+1} = W^n$, and thus the divergence of the numerical flux has to balance the source terms exactly, again, irrespective of the value of the local time step.

The numerical value of the time step is determined by the Courant-Friedrich-Lewy (CFL) stability condition

$$\Delta t = C \left(\frac{c_x + |u_x|}{\Delta x} + \frac{c_y + |u_y|}{\Delta y} + \frac{c_z + |u_z|}{\Delta z} \right)^{-1} \tag{1.5}$$

where c_x, c_y and c_z are the fast magnetosonic speeds in the x, y and z directions, and

$$C < 1 \tag{1.6}$$

is the **CFL number**. Stiff source terms might pose additional stability constraints, although this is not typical for the applications of BATS-R-US.

1.2.2 Time accurate mode

For time accurate mode the code must use the same time step in every computational cell. Numerical stability requires that this **global time step** is determined as the minimum of the local time steps given by (1.5). The global time step can become very small if $(c + |u|)/\Delta x$ is very large in some cells. The time step is typically determined by the smallest cells.

For time accurate mode the forward Euler scheme (1.4) gives first order accuracy with respect to Δt. The temporal accuracy can be made second order with the two-stage time stepping scheme. The first stage uses $\Delta t/2$ as the time step, and the second stage calculates fluxes and sources based on the first stage results:

$$\begin{aligned} W^{n+1/2} &= W^n + \frac{\Delta t}{2} \left(-\nabla \cdot \mathbf{F}^n + S^n\right) \\ W^{n+1} &= W^n + \Delta t \left(-\nabla \cdot \mathbf{F}^{n+1/2} + S^{n+1/2}\right) \end{aligned} \tag{1.7}$$

It is easy to see that the two-stage scheme is twice as expensive as the one stage scheme (1.4). It depends on the application whether it is worthwhile to use the two-stage scheme. If the time steps are very small due to the stability requirement, the first order scheme may be accurate enough, or in other words, the truncation errors of the spatial discretization may be much larger than the temporal errors due to the one-stage scheme. In such a case it makes sense to use the less expensive one-stage scheme. For other problems the second order accuracy of the two-stage scheme may become important.

1.2.3 Point-Implicit Source Terms

Stiff source terms can be handled with the point implicit scheme, so they do not impose stability constraint on the time step. The starting point is the semi-implicit discretization

$$W^{n+1} = W^n + \Delta t \left(-\nabla \cdot \mathbf{F}^n + S^{n+1}\right) \tag{1.8}$$

Note that the source is evaluated at time level $n+1$ in contrast with the explicit discretization (1.4). This is a non-linear equation for W^{n+1}. It is sufficient to solve a linearized form using a Taylor expansion of S around S^n, which leads to the linear equation

$$\left(\frac{I}{\Delta t} - \frac{\partial S}{\partial W} \right) \left(W^{n+1} - W^n\right) = -\nabla \cdot \mathbf{F}^n + S^n \tag{1.9}$$

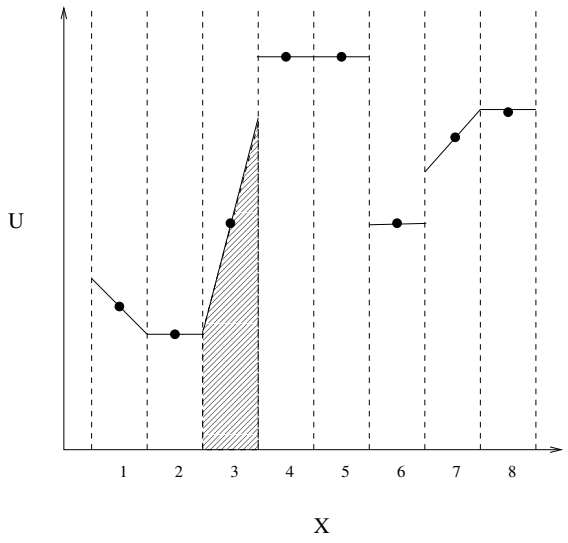

Figure 1.5: Finite volume representation of 1D data. The grid cells are bounded by cell interfaces (dashed lines). The dots represent the cell averages of U. The solid lines show possible extrapolations of U to the cell interfaces.

for the unknown $(W^{n+1} - W^n)$, where I is the identity matrix, and $\partial S/\partial W$ is the Jacobian matrix at time level n. As long as S depends on the local value of W only, the linear equations can be solved for each cell separately, which explains why this is called the **point-implicit scheme**. Even if S contains some non-local terms, one can evaluate the non-local part at time level n and take the partial derivatives with respect to the local part of the source term only. For example

$$\frac{\partial(\mathbf{B}\cdot\mathbf{v}\nabla\cdot\mathbf{B})}{\partial\mathbf{B}} \approx \mathbf{v}\nabla\cdot\mathbf{B} \tag{1.10}$$

can be used. A partially implicit discretization of the source term is still more stable than an explicit discretization.

The point implicit scheme has been written for some specific source terms, and in most applications it is not needed and it should not be used.

1.3 Numerical Schemes

BATS-R-US provides second order accurate, conservative, oscillation free numerical schemes to solve the MHD equations. Second order accuracy is required to obtain accurate solutions at reasonable grid resolutions. Numerical conservation is a necessary condition to obtain correct jump conditions through discontinuities like shock waves. The oscillation free property is very important to avoid over shoots and under shoots near discontinuities and sharp gradients, because these oscillations can lead to negative densities and pressures which would crash the code.

1.3.1 Finite Volume Discretization

BATS-R-US uses a finite volume discretization of the MHD equations. Space is divided into **grid cells**, and the discrete conservative variables are defined as the cell-averages

$$W_j(t) = \frac{1}{V_j}\int_{V_j} W(\mathbf{x},t)d\mathbf{x} \tag{1.11}$$

where V_j is the volume of the cell indexed by j (see Fig. 1.5). The differential equations (1.2) are discretized in their integral form

$$\frac{\partial W_j}{\partial t} = \frac{1}{V_j}\int_{V_j} (-\nabla \cdot \mathbf{F} + S)\, d\mathbf{x} \tag{1.12}$$

i.e. the equations are integrated for the cell volume. Using the Gauss theorem, the volume integral of fluxes can be replaced by a surface integral. In the discrete form we obtain

$$W_j^{n+1} = W_j^n - \frac{\Delta t}{V_j}\sum_{l=1}^{6} \mathbf{F}_l^n \cdot \mathbf{n}_l + \Delta t S_j^n \tag{1.13}$$

where $\mathbf{F}_l$ and $\mathbf{n}_l$ are the flux and the normal surface vector through the l-th face of cell j. The source term is simply evaluated for W_j^n, which is a second order approximation to the volume averaged source term. Second order time accuracy can be achieved by the two-stage scheme (1.7).

The finite volume method automatically leads to a conservative discretization since the fluxes are defined at the cell interfaces and they are added to and subtracted from the neighboring cell-averages such that the total change remains zero. This argument holds when the time step Δt is the same for all the cells. For local time stepping, the conservation is only ensured for the final steady state, when $W^{n+1} = W^n$ and thus

$$\frac{1}{V_j}\sum_{l=1}^{6} \mathbf{F}_l^n \cdot \mathbf{n}_l = S_j^n \tag{1.14}$$

for each cell independent of the time step. An additional complication arises for the AMR grid at resolution changes, where the face of a coarse cell is shared with 4 fine cell neighbors. BATS-R-US ensures conservation by setting the flux for the coarse cell equal to the sum of the 4 fine fluxes.

The remaining question is how to calculate the fluxes at the cell faces. BATS-R-US provides several numerical flux functions. These differ in their diffusiveness and robustness. In general the more diffusive the scheme, the more robust it is. It is problem dependent which flux function should be used. In principle one should select the least diffusive scheme which is still robust enough to handle the problem. In practice it is difficult to tell in advance whether a scheme will work or not for the full run, so some experimentation may be required. In the following sections we describe the available flux functions and slope limiters. For sake of concreteness we concentrate on the flux through the face orthogonal to the x direction.

1.3.2 Roe flux

The Roe flux is the most accurate, costly, and least robust of the three available flux functions. It is based on an approximate solution of the Riemann problem. The Riemann problem arises at the cell interface: given the state vectors W^L and W^R on the left and right sides of the interface, what will be the flux through the cell interface?

The Roe solver uses a characteristic decomposition of the discontinuity $W^R - W^L$ to obtain an answer. The amplitude of the k-th characteristic wave can be determined by multiplying the jump with the k-th left eigenvector l^k of the $\partial F/\partial W$ matrix. The k-th characteristic wave moves with a speed λ^k. Depending on the sign of λ^k, we should use the left or the right state to evaluate the flux. After some algebra, the Roe flux is obtained

$$F_l^{\rm Roe} = \frac{F^R + F^L}{2} - \frac{1}{2}\sum_k r^k |\lambda^k| l^k (W^R - W^L) \tag{1.15}$$

where r^k is the k-th right eigenvector of $\partial F/\partial W$. The eigenvectors and the eigenvalues are evaluated at some average state of W^R and W^L. We use the arithmetic average of the primitive variables $(U^R + U^L)/2$ for this purpose.

The Roe solver requires the calculation of the left and right eigenvectors and all the eigenvalues. This is quite expensive for the MHD equations, although BATS-R-US uses a highly optimized subroutine for the flux calculation. The Roe solver is not implemented for the Boris corrected MHD equations, because the eigenvectors are too complicated.

We mention that an entropy fix is implemented for the Roe flux, which prevents the formation of rarefaction shock waves. The entropy fix slightly increases $|\lambda^k|$ when it is very close to zero.

1.3.3 Rusanov flux

The Rusanov (also called TVD Lax-Friedrich or TVDLF) flux function greatly simplifies the Roe solver by replacing $|\lambda^k|$ in (1.15) with

$$\lambda^{\max} = \max_k |\lambda^k| = c_x + |u_x| \tag{1.16}$$

where c_x is the fast magnetosonic speed. Since now $\lambda^{\max}$ can be taken out of the sum and the product of left and right eigenvectors give a Kronecker delta in (1.15), the Rusanov flux function becomes

$$F_l^{\text{Rusanov}} = \frac{F^R + F^L}{2} - \frac{1}{2}\lambda^{\max}(W^R - W^L) \tag{1.17}$$

Again, $\lambda^{\max}$ is evaluated for the average of primitive variables. Note that the same $\lambda^{\max}$ is used for the time step calculation in (1.5). For cell j we actually take the larger of $\lambda^{\max}_{j-1/2}$ and $\lambda^{\max}_{j+1/2}$.

The Rusanov flux function is very efficient and robust, but more diffusive than the Roe flux. Especially contact discontinuities diffuse fast for the Rusanov flux function.

1.3.4 Linde flux

The Linde flux is in between the Roe and Rusanov fluxes in every sense. It is more accurate but more expensive and less robust than the Rusanov flux, on the other hand, it is less accurate but less expensive and more robust than the Roe-flux.

1.3.5 First and Second Order Accuracy

We still need to define how the left and right states W^L and W^R are obtained. For a first order scheme

$$\begin{aligned} W^L_{j+1/2} &= W_j \\ W^R_{j+1/2} &= W_{j+1} \end{aligned} \tag{1.18}$$

Spatially first order schemes are extremely diffusive, even for a Roe flux. They should only be used at the very beginning of a steady state calculation, since the first order schemes are fast and robust, so they can overcome the initial transients. For the final steady state solution as well as for a time accurate simulation, however, a spatially second order scheme is required.

Second order spatial accuracy is achieved by a linear reconstruction of the left and right states. To guarantee positivity, BATS-R-US uses the primitive variables $U = \{\rho, \mathbf{u}, \mathbf{B}, p\}$ for the linear reconstruction

$$\begin{aligned} U^L_{j+1/2} &= U_j + \frac{\Delta x}{2}\left(\frac{\partial U}{\partial x}\right)_j \\ U^R_{j+1/2} &= U_{j+1} - \frac{\Delta x}{2}\left(\frac{\partial U}{\partial x}\right)_j \end{aligned} \tag{1.19}$$

The gradient $\partial V/\partial x$ must be carefully defined so that no spurious oscillations are produced by the numerical scheme.

1.3.6 Slope Limiters

There are several possibilities to define oscillation free gradients. We implemented two variants in BATS-R-US: the **minmod** limiter and the **monotonized central** (MC) limiter. The minmod limiter is more diffusive and more robust, while the MC limiter is less diffusive and less robust. The computational cost is very comparable, although the minmod limiter is slightly less expensive.

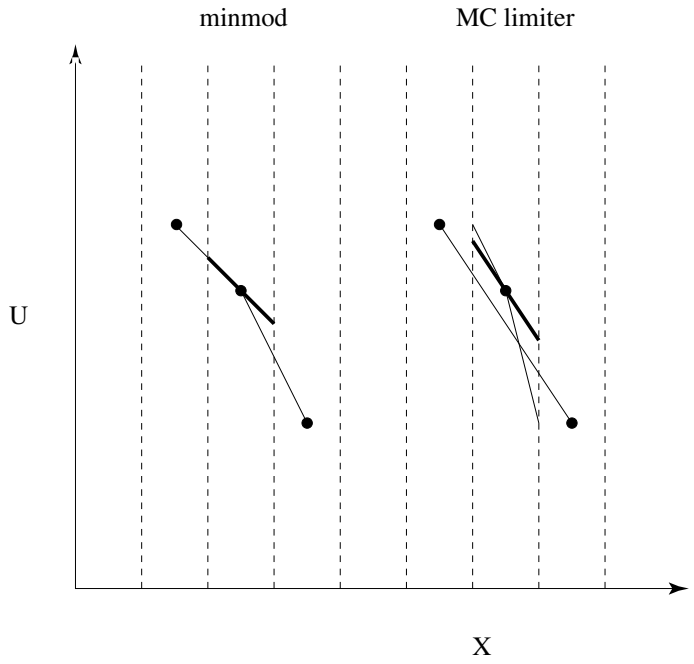

Figure 1.6: Three different slope limiters applied to the same configuration. Each limiter considers a few different approximate slopes (thin lines) and selects one of them (thick line).

The minmod and MC limited slopes are defined as

$$\left(\frac{\partial U}{\partial x}\right)_j^{\rm minmod} = \frac{1}{\Delta x}{\rm minmod}\left(\Delta U_{j-1/2}, \Delta U_{j+1/2}\right) \tag{1.20}$$
$$\left(\frac{\partial U}{\partial x}\right)_j^{\rm MC} = \frac{2}{\Delta x}{\rm minmod}\left(\Delta U_{j-1/2}, \Delta U_{j+1/2}, \frac{1}{4}(U_{j+1}-U_{j-1})\right)$$

where $\Delta U_{j+1/2} = U_{j+1} - U_j$ and the **minmod** function is defined as the argument with the smallest modulus when all arguments have the same signs and otherwise it is zero. The action of the slope limiters is depicted in Fig. 1.6.

1.4 Divergence of B Control

There is a big difference between the view of theorists, who would generally insist that $\nabla \cdot \mathbf{B}$ should be exactly zero, and practitioners of numerical MHD, who usually take a more pragmatic approach, and are satisfied with $\nabla \cdot \mathbf{B}$ converging to zero as the grid resolution Δx and the time step Δt approach zero. The justification for the latter approach is simple: none of the numerical values agree to the analytical solution exactly, so why should one insist that a specific combination of them, namely some numerical representation of $\nabla \cdot \mathbf{B}$ should be equal to the analytic value, i.e. zero.

One way of ensuring a small numerical value for $\nabla \cdot \mathbf{B}$ is to demand that some particular discretization is exactly zero. The projection and the constrained transport schemes fall into this class. Another possibility is to set the numerical value of $\nabla \cdot \mathbf{B}$ to zero in the initial and boundary conditions, and to trust the scheme to maintain this condition until the end of the simulation to the accuracy of the truncation error. The 8-wave scheme and the diffusive divergence B control belong to this class.

1.4.1 8-wave Scheme

The default scheme used in the BATS-R-US code is called 8-wave scheme, because the MHD equations can be rewritten in a form, where in addition to the entropy, 2 slow, 2 Alfvén and 2 fast waves, there is an 8th wave associated with finite $\nabla \cdot \mathbf{B}$. The momentum, energy , and induction equations (**??**)- (**??**) and (**??**)) are modified to

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot \left(\mathbf{u}\rho\mathbf{u} - \frac{\mathbf{B}\mathbf{B}}{\mu_0} \right) + \nabla \left(p + \frac{\mathbf{B}^2}{2\mu_0} \right) = -\frac{1}{\mu_0}(\nabla \cdot \mathbf{B})\mathbf{B} \tag{1.21}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{B} - \mathbf{B}\mathbf{u}) = -(\nabla \cdot \mathbf{B})\mathbf{u} \tag{1.22}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left[\mathbf{u} \left(\frac{1}{2}\rho v^2 + \frac{\gamma}{\gamma - 1}p + \frac{B^2}{\mu_0} \right) - \frac{(\mathbf{u} \cdot \mathbf{B})\,\mathbf{B}}{\mu_0} \right] = -\frac{1}{\mu_0}(\nabla \cdot \mathbf{B})\mathbf{B} \cdot \mathbf{u} \tag{1.23}$$

The source terms on the right hand side are zero analytically, but can be non-zero numerically due to truncation errors. The source terms are added at every time step with some simple discretization. The 8-wave scheme is very robust, and it seems to produce correct results for the vast majority of MHD problems. A potential drawback, however is, that it may not give correct jump conditions, since it is not conservative.

1.4.2 Diffusive Control

Diffusive control (Linde and Malagoli, 2000) adds terms that diffuse away the numerically generated $\nabla \cdot \mathbf{B}$. The new source terms occur in the induction and energy conservation equations

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{B} - \mathbf{B}\mathbf{u}) = D\nabla(\nabla \cdot \mathbf{B}) \tag{1.24}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left[\mathbf{u} \left(\frac{1}{2}\rho v^2 + \frac{\gamma}{\gamma - 1}p + \frac{B^2}{\mu_0} \right) - \frac{(\mathbf{u} \cdot \mathbf{B})\,\mathbf{B}}{\mu_0} \right] = \frac{1}{\mu_0} D\mathbf{B} \cdot \nabla(\nabla \cdot \mathbf{B}) \tag{1.25}$$

Taking the divergence of the modified induction equation, we obtain

$$\frac{\partial \nabla \cdot \mathbf{B}}{\partial t} = \nabla \cdot D\nabla(\nabla \cdot \mathbf{B}) \tag{1.26}$$

which is a simple diffusion equation for $\nabla \cdot \mathbf{B}$ with a diffusion coefficient D. The diffusion coefficient is set to a value that provides maximum diffusion of $\nabla \cdot \mathbf{B}$ but still keeps the scheme numerically stable. This requirement is met if

$$D = \delta \frac{(\Delta x)^2}{\Delta t} \tag{1.27}$$

with $\delta \leq 4/6$. This value is marginally stable, so the value used should be smaller than this. We have found that numbers around 0.4 work well. More testing remains to be done to find the optimal parameter.

The diffusive source terms can be added to the 8-wave source terms, or used by themselves (Sokolov). In the above form the diffusive source terms are only conservative for the induction equation and for a uniform grid. However we have some ideas on how to make the diffusive scheme fully conservative and these ideas will be implemented and tested soon.

1.4.3 Projection Scheme

The projection scheme eliminates the numerically generated $\nabla \cdot \mathbf{B}$ in every time step. The solution $\mathbf{B}^*$ provided by the base scheme can be written as the sum of a curl and a gradient

$$\mathbf{B}^* = \nabla \times \mathbf{A} + \nabla\varphi \tag{1.28}$$

where the physically meaningful and divergence free part is the first $\nabla \times \mathbf{A}$ term. By taking the divergence of the above equation, we get a Poisson equation for the unknown scalar field φ:

$$\nabla \cdot \mathbf{B}^* = \nabla^2 \varphi \tag{1.29}$$

This equation can be solved with some iterative scheme. In BATS-R-US, the BiCGSTAB scheme is used, which is a parallel Krylov-type method. Once the solution is found (with some accuracy) the divergence free solution can be easily obtained

$$\mathbf{B}^{n+1} = \mathbf{B}^* - \nabla \varphi \tag{1.30}$$

It is interesting to note that the correction term above is very similar to the diffusive source term in the modified induction equation (1.24) of the diffusive control scheme if we take $\varphi = D \nabla \cdot \mathbf{B}$. This is a very crude solution, but for certain iterative schemes it would be the first iterate of the Poisson equation.

1.4.4 Constrained Transport Scheme

The Constrained Transport (CT) scheme relies on a specific staggered discretization of the magnetic field and the electric field which maintains the divergence free property. **It is important to note that THE CT SCHEME CAN ONLY BE USED IN THE TIME ACCURATE MODE**. It is also important to distinguish the fully conservative CT scheme (which follows Balsara and Spicer 1999 and was generalized to AMR by Tóth and Roe, 2000) which is used in BATS-R-US, from the original non-conservative Hawley and Evans (1988) scheme.

A main feature of the CT scheme is the introduction of variables that are stored on face centers and cell edges. The previously described schemes all have variables stored at the cell center. For CT, the magnetic field is stored at face centers (we use lower case $\mathbf{b}$ to distinguish from the above cell centered $\mathbf{B}$) and we introduce the electric field $\mathbf{E}$ which is stored on cell edges. The main steps of the algorithm are the following

- Discretize b^x, b^y, b^z at the x, y, z face centers of the cells, respectively
- Calculate E^x, E^y, E^z at cell edges by interpolation of fluxes provided by the base scheme
- Message pass the electric field $\mathbf{E}$ for consistency at refinement changes
- Update b^x, b^y, b^z with finite differencing $\nabla \times \mathbf{E}$
- Interpolate $\mathbf{b}$ to the cell centered $\mathbf{B}$

As long as $\nabla \cdot \mathbf{b}^n = 0$, this algorithm ensures that $\nabla \cdot \mathbf{b}^{n+1} = 0$ as well. BATS-R-US uses the projection scheme to ensure that the initial condition has exactly zero $\nabla \cdot \mathbf{b}$. When ever a new run is begun, the code is restarted, or the $\nabla \cdot \mathbf{B}$ control method is changed to CT from some other scheme, it is recommended that a projection be computed rather accurately to ensure that the initial condition for the CT scheme is divergence free. With this in mind, it is possible to switch to the CT scheme at any time during a run.

In a CT scheme the boundary conditions must be applied on the electric field. Currently the code uses a simple $\mathbf{E} = 0$ condition at the inner boundary. This should be improved in the future by using the electric field provided by the ionosphere model.

Refinement and coarsening maintains $\nabla \cdot \mathbf{b} = 0$ using specially designed restriction and prolongation operators. However, the blocks intersecting the inner boundary should not be refined or coarsened while the CT scheme is used, because the current algorithm cannot deal with the newly created or removed cells while maintaining $\nabla \cdot \mathbf{b} = 0$. This might be improved in the future.

1.4.5 Choosing a Divergence B Control Scheme

For steady state problems local time stepping is strongly advised. This excludes using the constrained transport scheme for $\nabla \cdot \mathbf{B}$ control. The projection scheme is quite expensive for the AMR grid when the code is running on many processors. The current implementation of the diffusive divergence control seems to produce results which seem to be less accurate than the results obtained with the other schemes. Therefore we recommend the 8-wave scheme as

the most efficient method for obtaining a steady state solution. With an improved implementation of the diffusive control it is possible that the diffusive control in itself or combined with the 8-wave scheme will work better.

If the steady state solution shows an excessive amount of $\nabla \cdot \mathbf{B}$ (as compared with the current or the gradients of $\mathbf{B}$) then adding diffusive control is likely to reduce $\nabla \cdot \mathbf{B}$ by a factor of about 10. The projection scheme can reduce the error even more, but it is costly.

For time accurate problems the constrained transport, the 8-wave and the diffusive control need about the same CPU time, while the projection scheme is still very expensive if BATS-R-US is run with many blocks on many processors. For most time accurate problems we find the 8-wave scheme to be the most robust discretization. If keeping $\nabla \cdot \mathbf{B} = 0$ has a high priority, one can experiment with the constrained transport scheme. Note, however, that the boundary conditions are not fully developed, and they are mostly appropriate for an Earth magnetosphere run without the ionosphere model.