Magnetic Field Propagator

Sandroos, Arto

May 2011

1 Introduction

The purpose of the test was to demonstrate that the field solver works, and to test the effects of grid refinement and increase of the order of accuracy of the solver on the results. The field solver is described in Londrillo and del Zanna (2004).

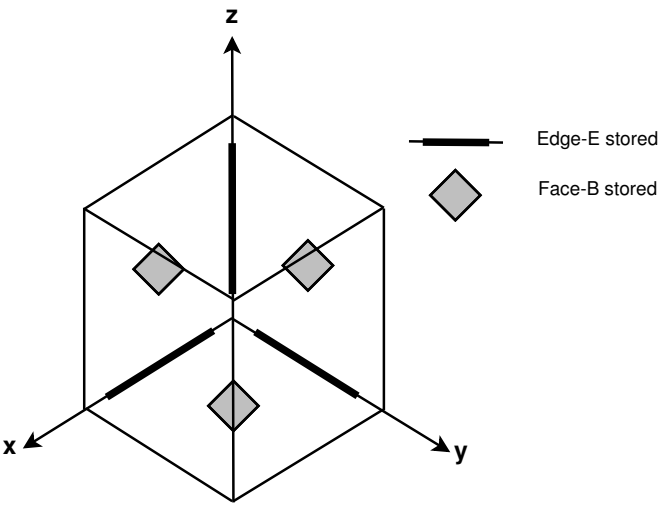

Figure 1: Edge-averaged values of projections of electric field $\Delta s^{-1} \int_{\text{edge}} \mathbf{E} \cdot d\mathbf{s}$ are stored on three edges per cell (thick lines). Face-averaged normal component of magnetic field $A^{-1} \oint_{\text{face}} \mathbf{B} \cdot d\mathbf{A}$ is stored on marked three cell faces (gray diamond). Here face normals $d\mathbf{A}$ and edge line elements $d\mathbf{S}$ always point to positive coordinate directions.

In Londrillo and del Zanna method face-averaged normal components of magnetic field are propagated forward in time using integral form of Faraday's law,

$$\frac{\partial}{\partial t} \oint_{\text{face}} \mathbf{B} \cdot d\mathbf{A} = -\int_C \mathbf{E} \cdot d\mathbf{S}, \tag{1}$$

where the integral on the left hand side is taken over a cell face, and the line integral is evaluated along the contour C of the cell face. If the magnetic field is averaged over the cell face, and electric field projected to cell edges and averaged, the Faraday's law for B_z-component takes the form

$$\begin{aligned} \frac{\partial}{\partial t}\langle B_z\rangle \Delta x \Delta y &= -\left(\langle E_x\rangle^{(i,j,k)} - \langle E_x\rangle^{(i,j+1,k)}\right)\Delta x \\ &\quad -\left(\langle E_y\rangle^{(i+1,j,k)} - \langle E_y\rangle^{(i,j,k)}\right)\Delta y, \end{aligned} \tag{2}$$

where $\langle \ldots \rangle$ indicates an averaged quantity and (i,j,k) etc. the indices of the cell in which the quantity is stored. Similar expressions are obtained for time derivatives of $\langle B_x\rangle$ and $\langle B_y\rangle$.

The main task in Londrillo and del Zanna algorithm is thus the calculation of the edge-averaged electric field components. Once those are known, it is straightforward to propagate $\mathbf{B}$ forward in time using equation (2).

As a side note, numerical algorithms in which the integral form of Faraday's law is used to propagate $\mathbf{B}$ are known as constrained transport methods. Londrillo and del Zanna method is a constrained transport upwinded method, i.e. out of the four possible candidates for each edge-averaged $\mathbf{E}$, the upwinded value is chosen in supersonic case. In subsonic cases the edge-averaged $\mathbf{E}$ is calculated with weighed averages of the four candidate $\mathbf{E}$'s, additionally a diffusive flux is added to stabilise the scheme. Constrained transport methods satisfy $\nabla \cdot \mathbf{B} = 0$ by construction, provided that the initial state and boundary conditions are divergence-free.

The switch to subsonic solver is done based on the value of characteristic speed vs. the speed of plasma. If values such as

$$\alpha_x^{\pm} = \max(0, \pm[V_x \pm c_{s,x}]), \tag{3}$$

where V_x and $c_{s,x}$ are the plasma and characteristic speeds to x-direction, is non-zero for both possibilities, diffusive terms are switched on. The characteristic speed is usually the speed of the fastest wave mode (speed of sound in hydrodynamics, fast magnetosonic speed in MHD). See a textbook on numerical solutions of hyperbolic equations for more detailed discussion. Fast magnetosonic speed is

$$V_{\mathrm{ms}}^2 = V_A^2 + c_s^2, \tag{4}$$

$$V_A^2 = B^2/\sqrt{\mu_0 \rho_m}, \tag{5}$$

$$c_s^2 = \gamma p/\rho_m. \tag{6}$$

Here (5) is the Alfvén speed and (6) the speed of sound in adiabatic ideal gas. γ is the polytropic index.

Part I
Reconstruction

2 Magnetic Field

Reconstruction of face-averaged magnetic field follows Balsara (2009). In practice the magnetic field needs to be reconstructed on the edges and averaged over the coordinate aligned with the edge, as edge-averaged electric field components are used to propagate **B**.

Divergence-free reconstruction of **B** begins by reconstructing face-averaged components of **B** along the cell faces. Reconstruction is done using Legendre polynomials

$$\begin{aligned} P_1(x) &= x, & (7)\\ P_2(x) &= x^2 - 1/12, & (8) \end{aligned}$$

which are orthogonal over the interval $[-1/2, +1/2]$. One also needs averages of Legendre polynomials, which often have simple forms:

$$\begin{aligned} \langle P_1(x)\rangle_x &= \int_{-1/2}^{+1/2} P_1(x)dx = 0, & (9)\\ \langle P_2(x)\rangle_x &= \int_{-1/2}^{+1/2} P_2(x)dx = 0, & (10)\\ \langle P_1(x)P_1(y)\rangle_{xy} &= 0, & (11)\\ \langle P_1(x)\rangle_y &= P_1(x)\Delta y, & (12) \end{aligned}$$

and so forth. In practice only quadratic terms, such as $\langle P_1^2(x)\rangle = 1/12$ or $\langle P_2^2(x)\rangle = 1/180$, have non-zero averages. Coefficients of Legendre polynomials are obtained by calculating slope-limited derivatives of $B_{\rm face}$ values. For example, reconstruction polynomial of B_x along x-faces is given by

$$\begin{aligned} B_x(x=\pm 1/2, y, z) &= B_0^{x\pm} + B_y^{x\pm}P_1(x) + B_z^{x\pm}P_1(z) && \longleftarrow 2^{\rm nd}\ \text{order}\\ &+ B_{yy}^{x\pm}P_2(y) + B_{yz}^{x\pm}P_1(y)P_1(z) + B_{zz}^{x\pm}P_2(z), && \longleftarrow 3^{\rm rd}\ \text{order} \end{aligned}$$

where $B_y^{x\pm}$ is the slope limited y-derivative of the B_x and so forth. Second-order accurate reconstruction of B_x in the cell interior is given by

$$\begin{aligned} B_x(x,y,z) &= a_0 + a_xP_1(x) + a_yP_1(y) + a_zP_1(z)\\ &+ a_{xx}P_2(x) + a_{xy}P_1(x)P_1(y) + a_{xz}P_1(x)P_1(z). & (13) \end{aligned}$$

Higher-order reconstruction formulas are given in Balsara (2009). Coefficients for reconstructing the $\mathbf{B}$ in the interior of the cell can be expressed using the derivatives along the cell faces. For example, the reconstruction coefficients for B_x are

$$
\begin{aligned}
a_0 &= \frac{1}{2}\left[B_0^{x+} + B_0^{x-}\right] - \frac{1}{6}a_{xx}, && (14)\\
a_x &= \left[B_0^{x+} - B_0^{x-}\right], && (15)\\
a_y &= \frac{1}{2}\left[B_y^{x+} + B_y^{x-}\right], && (16)\\
a_z &= \frac{1}{2}\left[B_z^{x+} + B_y^{x-}\right], && (17)\\
a_{xx} &= -\frac{1}{2}\left[b_{xy} + c_{xz}\right], && (18)\\
a_{xy} &= \left[B_y^{x+} - B_y^{x-}\right], && (19)\\
a_{xz} &= \left[B_z^{x+} - B_z^{x-}\right]. && (20)
\end{aligned}
$$

Similar expressions are obtained for B_y and B_z. Equation (18) follows from $\nabla \cdot \mathbf{B} = 0$ requirement, the rest of equations are obtained by requiring that the interior reconstruction agrees with the derivatives of face-averaged $\mathbf{B}$ at the cell faces.

In calculation of maximum characteristic speed, which in ideal MHD is the fast magnetosonic speed, reconstructed values of B^2 is also needed on the edges. Here the value at an edge aligned with z-coordinate is given:

$$
\begin{aligned}
\left\langle B_x^2 \right\rangle_z &= \left[B_0^{x\pm} + B_y^{x\pm} P_1(y)\right]^2 + \frac{1}{12}\left[B_z^{x\pm}\right]^2, && (21)\\
\left\langle B_y^2 \right\rangle_z &= \left[B_0^{y\pm} + B_x^{y\pm} P_1(x)\right]^2 + \frac{1}{12}\left[B_z^{y\pm}\right]^2, && (22)\\
\left\langle B_z^2 \right\rangle_z &= \left[c_0 + c_x P_1(x) + c_y P_1(y)\right]^2 \\
&+ \frac{1}{12}\left[c_z + c_{xz} P_1(x) + c_{yz} P_1(y)\right]^2 + \frac{1}{180}c_{zz}^2, && (23)\\
\left\langle B^2 \right\rangle_z &= \left\langle B_x^2 \right\rangle_z + \left\langle B_y^2 \right\rangle_z + \left\langle B_z^2 \right\rangle_z. && (24)
\end{aligned}
$$

In equations (21)-(24) $\langle B_x^2 \rangle$, $\langle B_y^2 \rangle$ were computed using the (simpler) reconstructions along cell faces, while $\langle B_z^2 \rangle$ was calculated using cell interior reconstruction (see Balsara, 2009 for equations). Equation (24) gives the second-order accurate value of B^2 averaged over the cell edge aligned with z-coordinate.

3 Face-averaged Magnetic Field

In propagation of distribution function one needs the complete magnetic field vector, averaged over each cell face. These are easy to calculate using the averaging rules of Legendre polynomials (9)-(12) and expressions for the reconstructed

magnetic field in the cell interior (13). For example, $\langle \boldsymbol{B} \rangle_{yz}$ on x-faces is given by

$$\begin{aligned}
\langle B_x \rangle_{yz} &= B_0^{x\pm}, \\
\langle B_y \rangle_{yz} &= b_0 + b_x P_1(x), \\
\langle B_z \rangle_{yz} &= c_0 + c_x P_1(x),
\end{aligned}$$

where $B_0^{x\pm}$ is the face-averaged normal component of $\mathbf{B}$ stored to the face.

4 Volume-averaged Quantities

Volume-averaged quantities, such as the bulk velocity, have much simpler reconstructions. For example, x-component of bulk velocity is reconstructed with the following polynomial to the second order:

$$V_x(x,y,z) = V_{x0} + V_{xx}P_1(x) + V_{xy}P_1(y) + V_{xz}P_1(z). \tag{25}$$

In equation (25) V_{xx} is the slope-limited derivative to the x-direction and so forth.

5 Electric Field

Reconstruction of the electric field is a bit more complicated, because the electric field is obtained using an Ohm's law such as

$$\mathbf{E} = -\mathbf{V} \times \mathbf{B}, \tag{26}$$

and the vectors $\mathbf{V}$ and $\mathbf{B}$ have their own reconstructions. Electric field needs to be reconstructed for, at least, cell edges (propagation of B) and faces (propagation of distribution function). Electric field, reconstructed and averaged over cell x-face, is given by

$$\begin{aligned}
\langle E_x(x) \rangle_{yz} &= \langle B_y V_z - B_z V_y \rangle_{yz} \\
&= [b_0 + b_x P_1(x)]\,[V_{z0} + V_{zx} P_1(x)] \\
&+ \frac{1}{12}\,[b_y V_{zy} + b_z V_{zz} + b_{yx} V_{zy} P_1(x)] \\
&- [c_0 + c_x P_1(x)]\,[V_{y0} + V_{yx} P_1(x)] \\
&- \frac{1}{12}\,[c_y V_{yy} + c_z V_{yz} + c_{zx} V_{yz} P_1(x)]\,, \\
\langle E_y(x) \rangle_{yz} &= \langle B_z V_x - B_x V_z \rangle_{yz} \\
&= [c_0 + c_x P_1(x)]\,[V_{x0} + V_{xx} P_1(x)] \\
&+ \frac{1}{12}\,[c_y V_{xy} + c_z V_{xz} + c_{zx} V_{xz} P_1(x)]
\end{aligned}$$

$$
\begin{aligned}
&- \quad [a_0 + a_x P_1(x)]\,[V_{z0} + V_{zx} P_1(x)] \\
&- \quad \frac{1}{12}\,[a_y V_{zy} + a_z V_{zz} + (a_{xy} V_{zx} + a_{xz} V_{zz}) P_1(x)] \\
\langle E_z(x)\rangle_{yz} &= \langle B_x V_y - B_y V_x\rangle_{yz} \\
&= [a_0 + a_x P_1(x)]\,[V_{y0} + V_{yx} P_1(x)] \\
&+ \quad \frac{1}{12}\,[a_y V_{yy} + a_z V_{yz} + (a_{xy} V_{yy} + a_{yz} V_{zz}) P_1(x)] \\
&- \quad [b_0 + b_x P_1(x)]\,[V_{x0} + V_{xx} P_1(x)] \\
&- \quad \frac{1}{12}\,[b_y V_{xy} + b_z V_{xz} + b_{yx} V_{xy} P_1(x)]\,.
\end{aligned}
$$

Note that B_x can also be reconstructed using the simpler reconstruction formula on cell x-face, while B_y and B_z have to be reconstructed using the cell interior formulas. Although $P_1(x)$ appears in the formulas above, it should be replaced with value $\pm 1/2$ on the $\pm x$ faces.

Part II
Solver Accuracy and Correctness

6 Test Case

Field solver test was made by using a simple advection setup in a $(x, y, z) \in [-1, 1] \times [-1, 1] \times [-1, 1]$ grid and by using periodic boundary conditions. The initial condition was $B_x = 1$ in $(y, z) \in [-0.25, 0.25] \times [-0.25, 0.25]$ and zero elsewhere. The field was then propagated with a constant velocity $\mathbf{V} = (0, -1, -1)$.

Test case was run with two different cell sizes, 10×10 (sparse) and 20×20 (dense). Time step in sparse grid was $\Delta t = 0.05$ and in total 40 time steps were calculated. In dense grid the time step was halved, $\Delta t = 0.025$ and in total 80 time steps were calculated. The time step was chosen in such a way that at the end of simulation the advected field was back at its initial position.

7 Results

Figures 4-9 show selected results from field solver test in sparse and dense grid, and with 1^{st}- and 2^{nd}-order accurate solvers. In 2^{nd}-order accurate cases three different slope limiters were used (minmod, MC limiter, Van Leer).

Formal demonstration of solver accuracy would be made by comparing the solutions against the initial state after a full period and by calculating the L^1 error norms. Here the peak value of B_x at the end of simulation is used as a proxy for solver accuracy instead. The L^1 error norms are not diffucult to calculate in this case, however.

The results indicate that moving into 2^{nd}-order accuracy has a larger impact on the quality of the solution than doubling of the grid size (which is what adap-

Solver	sparse grid	dense grid
1^{st}	8.33%	16.1%
2^{nd} (minmod)	21.2%	50.6%
2^{nd} (MC limiter)	38.7%	93.1%
2^{nd} (Van Leer)	15.7%	34.4%

Table 1: Peak value of B_x, as percentage of initial value, with different solver versions in sparse and dense grid.

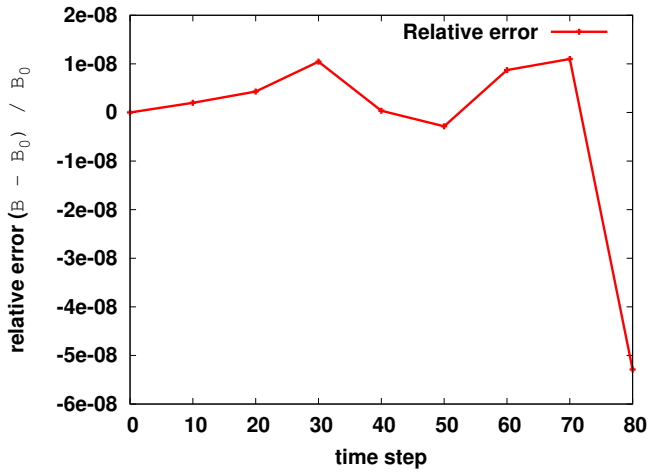

Figure 2: Relative error in conservation of magnetic field in the test case.

tive mesh refinement effectively does). Obviously the best results were obtained with a 2^{nd}-order accurate solver in denser grid. The results are summarized in Table 1.

Figure 2 shows the relative error in conservation of magnetic field in the test case. The solver is conservative to the seventh decimal over the simulated 80 time steps. The test case was computed using single-precision floating point numbers.

Part III
Boundary Conditions

8 Introduction

Magnetic field propagator in general is discussed in detail in Londrillo and del Zanna (2004). The behaviour of the solver at the boundaries is usually not

discussed on publications as it heavily depends on what is simulated. Here the boundaries are discussed, and a rectangular cuboid simulation domain is used as an example. A two-dimensional slice of the domain near the boundaries is illustrated in Fig. 3.

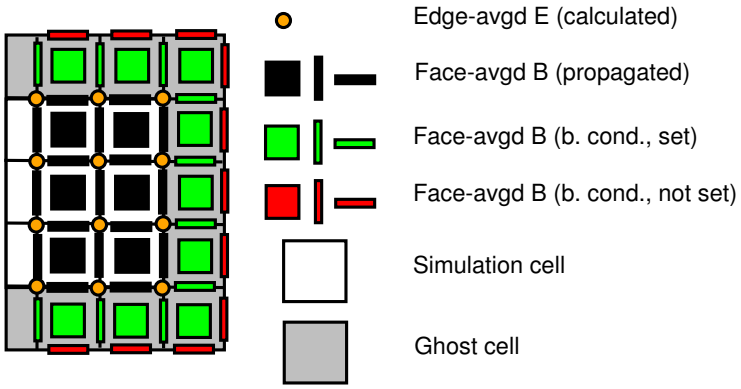

Figure 3: Illustration of simulation volume (white cells) and the faces on which face-averaged magnetic field is propagated (black), set via boundary conditions (green), and set by $\nabla \cdot \mathbf{B} = 0$ condition (red). Here the surface of the simulation domain has been defined to be included in the simulation domain, i.e. $\mathbf{B}$ is propagated on the surface. Edge-averaged electric field has to be calculated on the marked edges (orange). Three-dimensional generalisation follows in the obvious way.

The outmost layer of the mesh is taken to be a "ghost layer" (gray cells), which here mostly means that the distribution functions are set on these cells via boundary conditions and not propageted with the Vlasov solver. Velocity moments $(n, n\mathbf{V}, \mathbf{p})$ of the distribution function are then calculated in the same manner as for the simulation cells (white). If magnetic field is propagated on the surface of the simulation domain, components of the electric field must be calculated on all cell edges within and on the surface of the domain, as illustrated by the orange circles in Fig. 3.

Magnetic field must be set on some of the faces (green) in the ghost layer with boundary conditions - these values are used when the electric field is calculated on the cell edges on the domain's surface. Divergence-free condition of the magnetic field must of course be fulfilled in the ghost layer as well, in practice the easiest way to reach this is to leave at least one of the face values undefined by boundary conditions (red faces). These values are calculated from $\nabla \cdot \mathbf{B} = 0$ condition instead. One can, of course, make another choise on which which cell face values are set with boundary conditions.

9 Test Case

The test case used to verify that boundary conditions work is the same as used for solver correctness (Sect. 6). Here we are propagating B_x to $(-y, +z)$ direction with a constant velocity. When the field is hits the boundary, it should flow through. Also the values of face-averaged B on "dummy faces" must not effect the solution inside the simulation domain - here B on dummy faces was set to value 666.

Figure 10 shows the numerical solution after 10, 20, and 80 time steps. Evidently the dummy faces do not affect the solution as there are no visible artifacts. The initial (physical) magnetic field also propagates neatly through the corner of the simulation domain. Figure 11 shows the numerical solutions at the same time steps for a 2^{nd}-order accurate solver version. Again, no visible artifacts are present indicating that the boundaries are working as expected. The solution is also less diffusive than the one calculated with 1^{st}-order accurate solver (compate the rightmost panels on Figs. 10 and 11).

Performing the same test but with B_y or B_z component, and using different combinations of the propagation velocity, i.e. $(+y, -z)$ instead of $(-y, +z)$, produced the same results for 2^{nd}-order accurate solver (with and without MPI).

References

[1] Balsara, J. Comp. Phys., 228, 2009.

[2] Londrillo and del Zanna, J. Comp. Phys., 195, 2004.

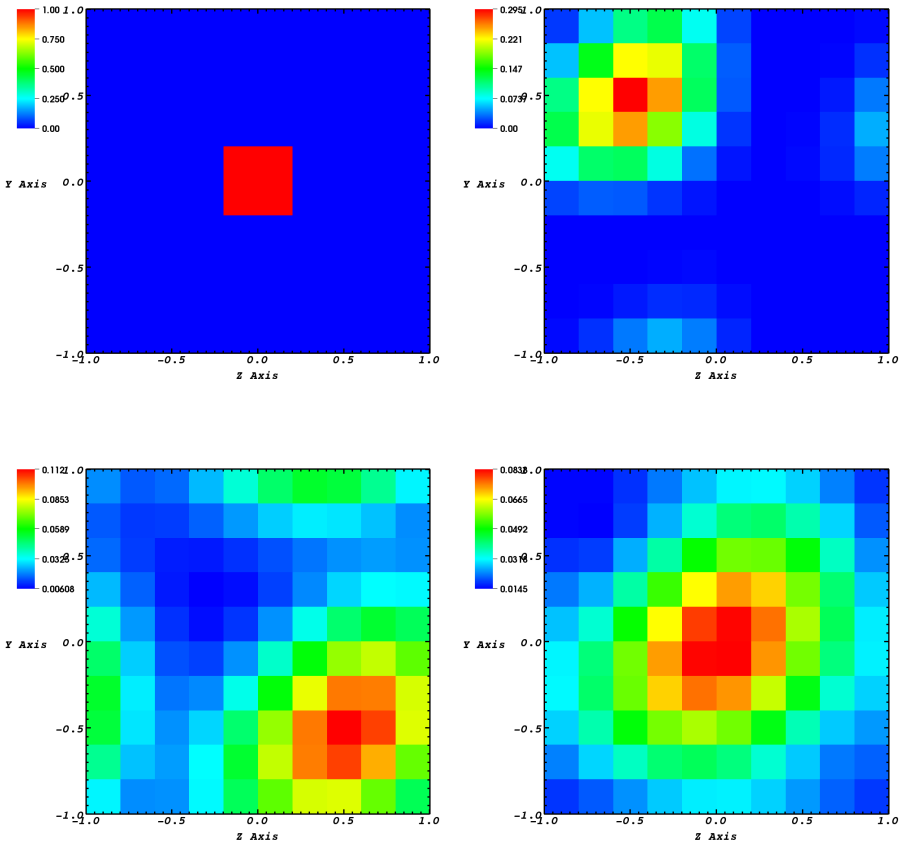

Figure 4: Propagation of B_x with a 1^{st}-order accurate field solver in a sparse grid. The pictures show the initial state, and solutions after 10, 30, and 40 time steps with periodic boundary conditions. Maximum value at 40 time steps is 8.33% of initial value.

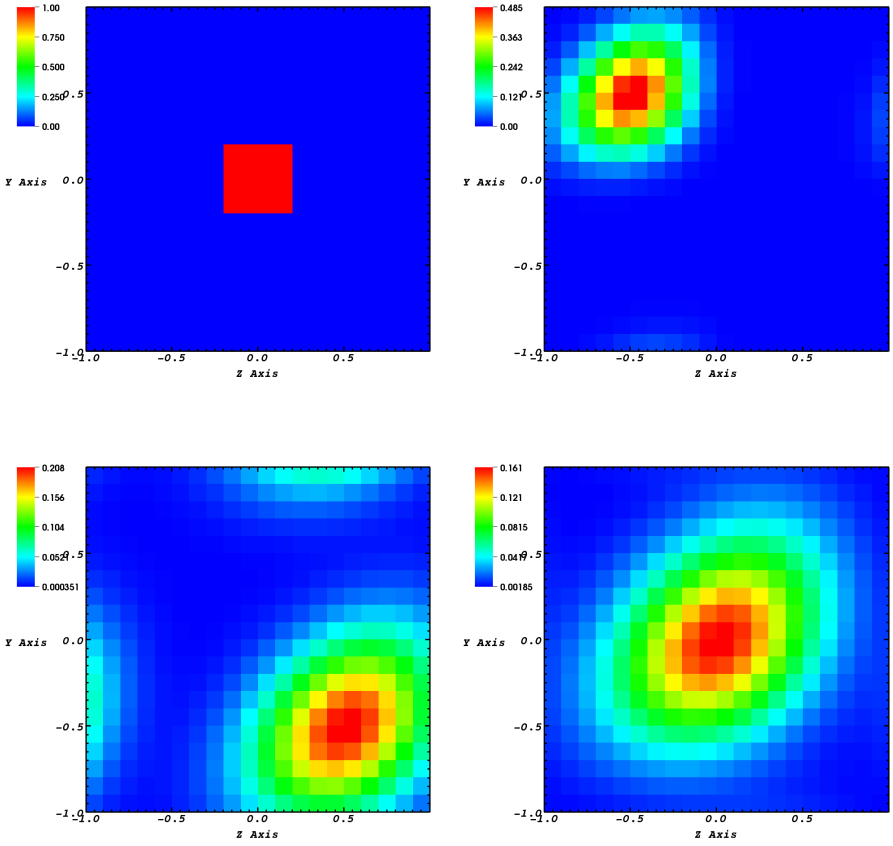

Figure 5: Propagation of B_x with a 1$^{\text{st}}$-order accurate field solver in a dense grid. The pictures show the initial state, and solutions after 10, 30, and 40 time steps with periodic boundary conditions. Maximum value at 80 time steps is 16.1% of initial value.

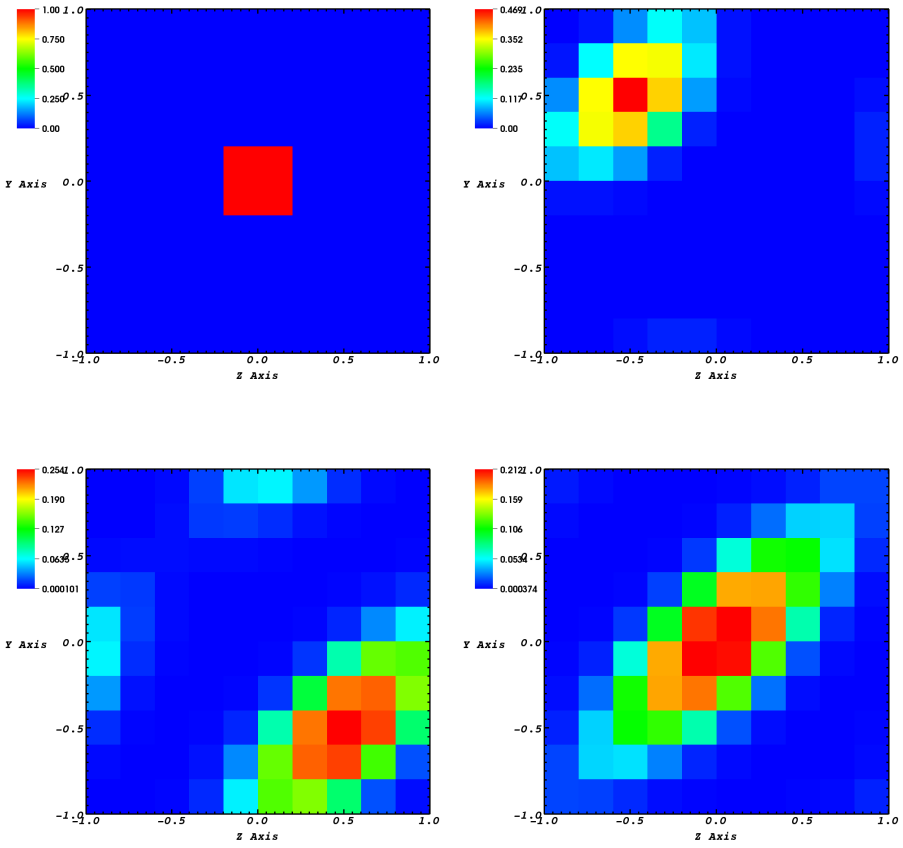

Figure 6: Propagation of B_x with a 2^{nd}-order accurate field solver in a sparse grid. The pictures show the initial state, and solutions after 10, 30, and 40 time steps with periodic boundary conditions. Maximum value at 40 time steps is 21.2% of initial value. Minmod limiter was used.

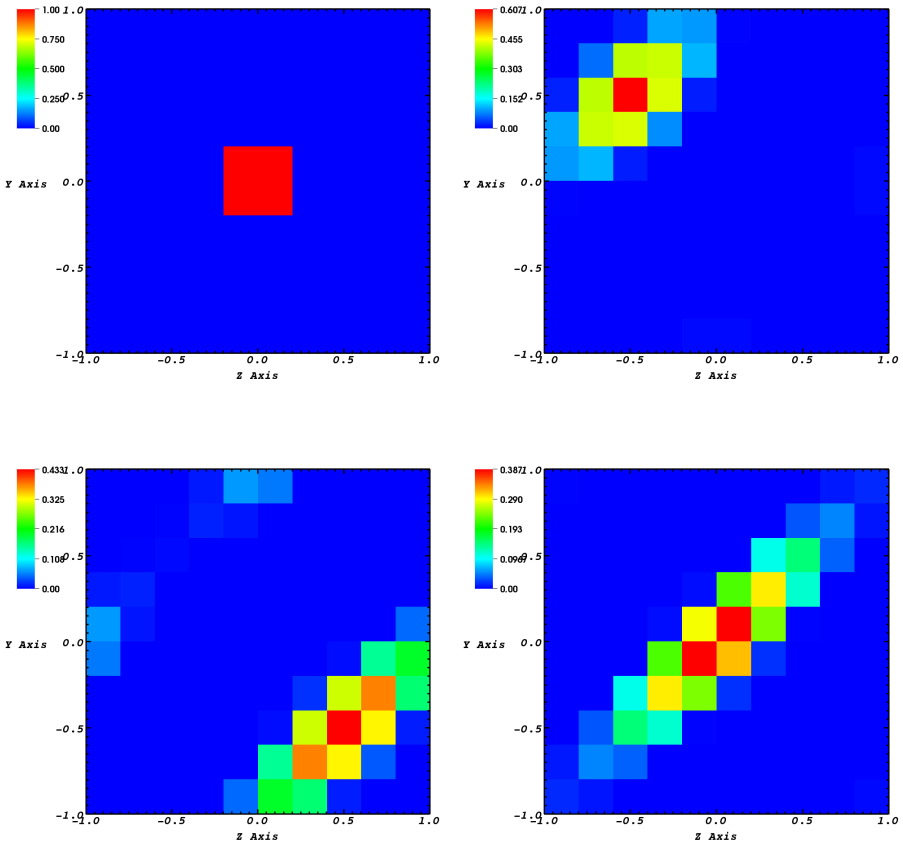

Figure 7: Propagation of B_x with a 2^{nd}-order accurate field solver in a sparse grid. The pictures show the initial state, and solutions after 10, 30, and 40 time steps with periodic boundary conditions. Maximum value at 40 time steps is 38.7% of initial value. MC limiter was used.

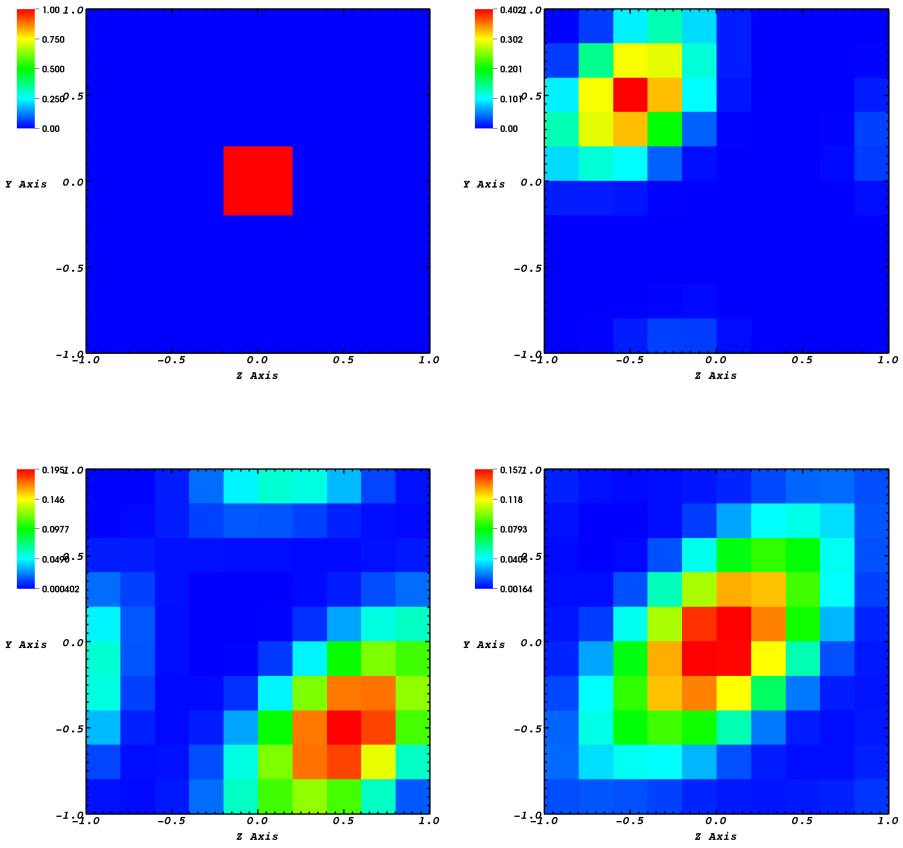

Figure 8: Propagation of B_x with a 2^{nd}-order accurate field solver in a sparse grid. The pictures show the initial state, and solutions after 10, 30, and 40 time steps with periodic boundary conditions. Maximum value at 40 time steps is 15.7% of initial value. Van Leer limiter was used.

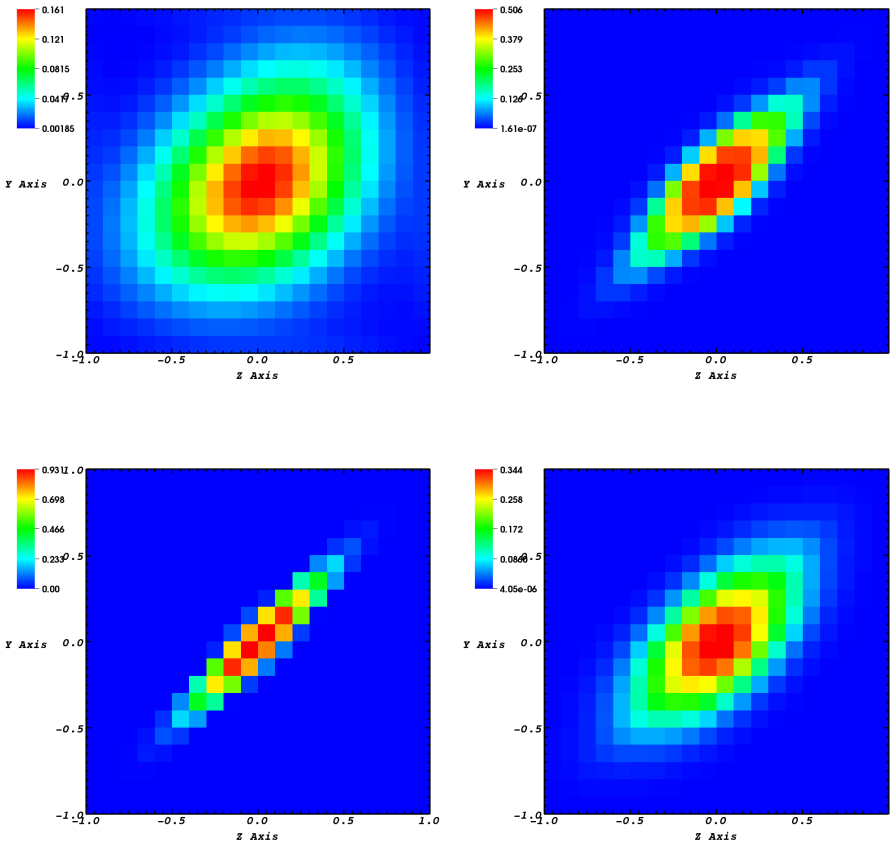

Figure 9: Numerical results of B_x propagation after 80 time steps in dense grid with (top left) 1$^{\text{st}}$-order accurate solver ($B_{x,\max} = 16.1\%$ of initial), and with 2$^{\text{nd}}$-order accurate solver with (top right) minmod limiter ($B_{x,\max} = 50.6\%$ of initial), (bottom left) MC limiter ($B_{x,\max} = 93.1\%$ of initial), (bottom right) Van Leer limiter ($B_{x,\max} = 34.4\%$ of initial).

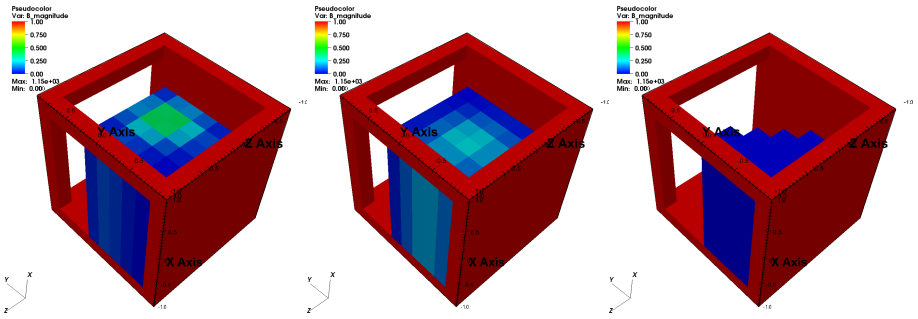

Figure 10: Propagation of B_x through the boundary of simulation domain with a constant velocity, using 1^{st}-order accurate solver. The panels show the numerical solution after 10 (left), 20 (middle), and 80 (right) time steps. Face-averaged B was set to value 666.0 on faces external to the simulation domain that should not affect the solution (cells drawn in red).

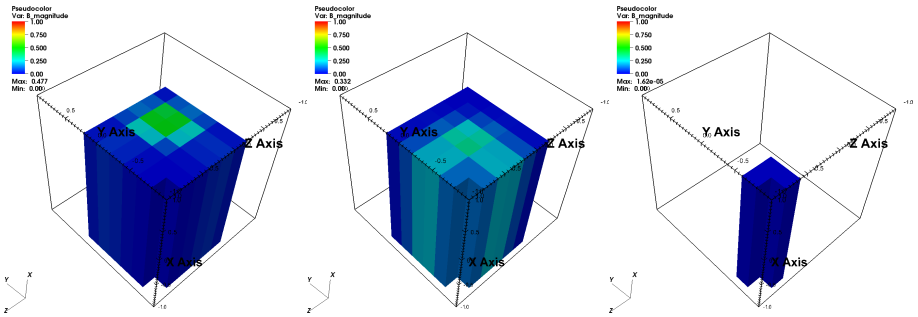

Figure 11: Propagation of B_x through the boundary of simulation domain with a constant velocity, using 2^{nd}-order accurate solver. The panels show the numerical solution after 10 (left), 20 (middle), and 80 (right) time steps. Here face-averaged B was set to zero value faces external to the simulation domain that should not affect the solution (cells drawn in red).