Numerical notes for a magneto-frictional implementation

Federico A. Nuevo

November 3, 2014

Abstract

These notes describe the magnetofrictional implementation developed by Gherardo Valori. We follow the description of the magneto-frictional method discussed in Valori et al. (2005) and Valori et al. (2007).

1 Basic Equations

The magneto-frictional method assumes that in the momentum balance equation the pressure gradient is neglected and the Lorentz force is balanced by a frictional force.

$$\nu \boldsymbol{u} = \boldsymbol{J} \times \boldsymbol{B}\,, \tag{1}$$

where $\nu = B^2/\mu$ is a numeric viscosity with μ a numerical value that changes in the time.

The current density is related to the magnetic field as

$$\boldsymbol{J} = \boldsymbol{\nabla}\times\boldsymbol{B}\,. \tag{2}$$

The evolution of the magnetic field is given by the induction equation

$$\frac{\partial \boldsymbol{B}}{\partial t} = \boldsymbol{\nabla}\times(\boldsymbol{u}\times\boldsymbol{B}) + C_L \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{B}) = \boldsymbol{F}(\boldsymbol{B}, t)\,, \tag{3}$$

where $\boldsymbol{F}(\boldsymbol{B}, t)$ is an oparator that involves spatial derivatives of $\boldsymbol{B}$ and the time. It is defined as:

$$\boldsymbol{F}(\boldsymbol{B}, t) \equiv \mu\, \boldsymbol{\nabla}\times \left(\frac{[(\boldsymbol{\nabla}\times\boldsymbol{B})\times\boldsymbol{B}]\times\boldsymbol{B}}{B^2} \right) + C_L \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{B})\,. \tag{4}$$

The term $C_L \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{B})$ diffuses the numerical divergences.

Given an initial magnetic field and boundary conditions, the current density is calculated using Equation 2 with some Finite Difference (FD) scheme. Then, a fictitious plasma velocity is calculated using Equation 1, after that the magnetic field is evolved introducing $\boldsymbol{J}$ and $\boldsymbol{u}$ in Equation 3 and using some time integration scheme

An important point that is not clear in the papers is: what are the boundary conditions at the top and the sides of the integration box?

2 Finite Difference Scheme

We discretize the box having a volume L^3 using a 3D grid with $x_i = i\Delta x$, $y_j = j\Delta y$ and $z_k = k\Delta z$ where the indexes range from 0 to N and $\Delta x = \Delta y = \Delta z = L/N$. We assume that the box is a cube, the spatial resolution is the same in all directions and the grid is uniform for simplicity. The bottom boundary ($z = 0$) corresponds to $k = 0$, the top boundary corresponds to $k = N$, and the side boundaries correspond to i or j equal to 0 or N. The ghost layers correspond to some of the spatial indexes equal to -1 or -2 or $N+1$ or $N+2$. The time is discretized as $t_n = n\Delta t$.

If $f(x, y, z, t)$ is some component of $\boldsymbol{B}$, $\boldsymbol{J}$ or $\boldsymbol{u}$ or a combination of components as $(u_x B_y - u_y B_x)$, the discretized version of f is

$$f^{(n)}_{i,j,k} = f(x_j, y_j, z_k, t_n) \tag{5}$$

and its spatial derivatives can be calculated using a fourth order central scheme:

$$\left(\frac{\partial f}{\partial x}\right)^{(n)}_{i,j,k} = \frac{f^{(n)}_{i-2,j,k} - 8f^{(n)}_{i-1,j,k} + 8f^{(n)}_{i+1,j,k} - f^{(n)}_{i+2,j,k}}{12\Delta x} \tag{6}$$

$$\left(\frac{\partial f}{\partial y}\right)^{(n)}_{i,j,k} = \frac{f^{(n)}_{i,j-2,k} - 8f^{(n)}_{i,j-1,k} + 8f^{(n)}_{i,j+1,k} - f^{(n)}_{i,j+2,k}}{12\Delta y} \tag{7}$$

$$\left(\frac{\partial f}{\partial z}\right)^{(n)}_{i,j,k} = \frac{f^{(n)}_{i,j,k-2} - 8f^{(n)}_{i,j,k-1} + 8f^{(n)}_{i,j,k+1} - f^{(n)}_{i,j,k+2}}{12\Delta z} \tag{8}$$

To use these expressions close to the boundaries (and in the boundaries), we need the ghost layers. The component values in the ghost layers are obtained using a fourth order polynomial extrapolation. **Isn't it better to use a one-sided FD scheme in this grid and not ghost layers with extrapolated values?**

The time integration is obtained using a forward Euler scheme

$$\boldsymbol{B}^{(n+1)} = \boldsymbol{B}^{(n)} + \Delta t\, \boldsymbol{F}(\boldsymbol{B}^{(n)}, t_n) \tag{9}$$

To evaluate the spatial derivatives included in the operator $\boldsymbol{F}$ (see Equation 4), the Equations 6 to 8 are used. I believe that Valori et al. (2011) used a more sophisticated temporal scheme, but we can start to work with a forward Euler scheme and improve it later.

3 Questions for future implementations

We discuss with Cristina and Pascal about some new implementations in the magneto-frictional method. We are interested in your opinion. The implementations are modular we can implement some of those or all.

1. We can use a pseudo-spectral (PS) scheme with periodic boundary conditions (i.e. use Fast Fourier Transform (FFT)), instead of using a FD scheme.

2. We can include the dynamics of the plasma velocity and use a more realistic viscosity (frictional) term.

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \frac{1}{\rho}\boldsymbol{J} \times \boldsymbol{B} + \nu \nabla^2 \boldsymbol{u}\,, \tag{10}$$

   where ρ is a numerical density (we can choose $\rho \propto B^2$).

3. We can use the potential vector $\boldsymbol{A}$ that satisfy $\boldsymbol{B} = \boldsymbol{\nabla}\times\boldsymbol{A}$ to eliminate the numerical divergences. The induction equation for $\boldsymbol{A}$ is

$$\frac{\partial \boldsymbol{A}}{\partial t} = \boldsymbol{u} \times (\boldsymbol{\nabla}\times\boldsymbol{A})\,. \tag{11}$$

   If we work with the potential vector it will be very useful to use a PS scheme with FFT because, in this case, we can calculate the $\boldsymbol{B}$ field in a very easy way as: $\boldsymbol{B}_{\boldsymbol{k}} = i\boldsymbol{k} \times \boldsymbol{A}_{\boldsymbol{k}}$.

4. We can decrease the number of equations by imposing $A_z = 0$. This is permitted by the gauge freedom on $\boldsymbol{A}$. This is also more accurate mumericaly than solving $\boldsymbol{\nabla}\cdot\boldsymbol{A} = 0$. It simplifies also the transformation of $\boldsymbol{A}$ to $\boldsymbol{B}$, and the reverse way.

5. The boundary conditions at $z = 0$ is the vector magnetogram. What about the boundary conditions on $\boldsymbol{u}$? $u_z = 0$ looks fine. Do we set $\partial u_{x,y}/\partial z = 0$ to let some freedom on $u_{x,y}$ evolution? What about lateral and top boundaries? Are open boundary coditions $\partial u_l/\partial n = 0$, $\partial B_l/\partial n = 0$ ($l = x, y, z$) fine (with $\partial A_l/\partial n = 0$)? (In the case of the spectral code, the periodicity imposes the boundary conditions). Are there not too many boundary conditions? (e.g. for an MHD code at $z = 0$, B_z, v_x, v_y and v_z are imposed, not B_x and B_y).