33rd EPS Conference on Plasma Phys. Rome, 19 - 23 June 2006 ECA Vol.**30I**, P-4.161 (2006)

A numerical code for studying the compressible MHD turbulence in a Reversed Field Pinch device

M. Onofri[1], L. Primavera[2], F. Malara[2], P. Veltri[2]

[1] *Aristotle University of Thessaloniki, Thessaloniki, 54006 Greece,*

[2] *Università della Calabria, Rende (CS), 87036 Italy,*

The numerical simulation of turbulence in a Reversed Field Pinch (RFP) device requires to address several nontrivial issues. The first one concerns the use of either cylindrical or toroidal geometry where, differently from the cartesian case, the turbulent cascade mainly evolves along the radial and axial directions. The compressibility also plays an important role, due to the fact that the plasma beta in such devices is rather low (typically $\beta < 10^{-2}$). Finally, boundary conditions may play an important role in the dynamical evolution of the system.

Cappello and Biskamp[1] and Merlin and Biskamp[2] tried to simulate the development of turbulence in an RFP device by using a numerical code which solved the Magnetohydrodynamics (MHD) equations in a three-dimensional, cylindrical configuration. They obtained the reversal of the axial magnetic field at the boundary and the dynamo effect predicted by the Taylor's [3] theory.

However, their code had some basic inconsistencies: to simplify the equations they supposed to have an incompressible plasma and, in order to avoid solving the Poisson equation for the pressure, which is a difficult numerical task, they neglected the pressure term in the momentum equation. This approach is clearly not correct, since the incompressibility condition requires the pressure to ensure the solenoidality of the velocity field.

Our approach is to solve the fully compressible, nonlinear, MHD equations in cylindrical geometry. In the compressible case, in principle, one has no specific information concerning the behaviour of some of the variables on the boundaries. We then use a characteristic decomposition method, suitably revised for the specific case of our variables and geometry, to impose the boundary conditions. We show below that the fully compressible description of the problem leads to results which are basically different from the ones obtained in the case of simplified models like in [1]-[2].

We integrate the following set of dimensionless equations, representing the evolution of a plasma in the MHD approximation:

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\nabla \cdot \mathbf{q}; \qquad \frac{\partial \mathbf{q}}{\partial t} = -\nabla \cdot \bar{\bar{\mathbf{F}}} + \mathbf{J} \times \mathbf{B} \\ \frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \mathbf{E}; \qquad\quad p = k\rho^{\gamma} \end{aligned} \tag{1}$$

where ρ is the density, $\mathbf{q} = \rho\mathbf{v}$ the momentum, $\mathbf{v}$ the velocity field, $\bar{\bar{\mathbf{F}}}$ is the flux tensor, defined as:

$$F_{ij} = q_i v_j + p\delta_{ij} - \mu\sigma_{ij} = q_i v_j + p\delta_{ij} - \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}(\nabla\cdot\mathbf{v})\delta_{ij}\right)$$

μ being the dimensionless viscosity coefficient, δ_{ij} the Kronecher delta and σ_{ij} the stress tensor. Finally, $\mathbf{B}$ is the magnetic field, $\mathbf{J} = \nabla\times\mathbf{B}$ the current density, $\mathbf{E} = -\mathbf{v}\times\mathbf{B} + \eta\mathbf{J}$ the electric field (η is the dimensionless electrical resisitivity of the plasma), k a constant parameter and γ the adiabatic index.

We use dimensionless quantities by normalizing the magnetic field and density to their typical values $\mathbf{B}_0$ and ρ_0, respectively. This allows us to measure the velocities in terms of the Alfvén speed: $c_A = \mathbf{B}_0/\sqrt{4\pi\rho_0}$. The lenghts are normalized to the radius of the cylinder a, therefore the time is measured in units of the Afvén time: $\tau_A = a/c_A$ and the pressure in units of $\rho_0 c_A^2$.

We assume to have a cylindrical domain, obtained either by straightening the original toroidal vessel or by cutting a small piece of it, such that the curvature effects can be neglected. The differential operators in Eq.s (1) are written in cylindrical coordinates, described by: $r\in[0,1]$ (we chose the radius of the cylinder as the unit length), $\theta\in[0,2\pi]$ and $z\in[0,2\pi R]$ (R being the aspect ratio, namely the ratio between the axial length and the radius, of the cylinder). We discretize the coordinates by choosing a uniformly spaced grid of points along the three directions:

$$\begin{aligned} r_j &= (j+1/2)\Delta_r, \quad j = 0,1,\ldots,N_r, \quad \Delta_r = 1/(N_r+1/2)\\ \theta_k &= k\Delta_\theta, \quad k = 0,1,\ldots,N_\theta-1, \quad \Delta_\theta = 2\pi/N_\theta\\ z_l &= l\Delta_z, \quad l = 0,1,\ldots,N_z-1, \quad \Delta_z = 2\pi R/N_z. \end{aligned}$$

We suppose to have periodic boundary conditions along both the θ and z directions, thus we can compute derivatives along those directions by simply using discrete Fast Fourier Transform (FFT) algorithms. Boundary conditions along the r direction, on the axis of the cylinder, are authomatically determined by the geometry of the system. The conditions, obtained by imposing the regularity of the cylindrical differential operators at $r = 0$ [4], are written in the form of parity conditions for each azimuthal Fourier wavenumbers:

$$\begin{aligned} v_r^m(-r) &= \mp v_r^m(r); \quad v_\theta^m(-r) = \mp v_\theta^m(r); \quad v_z^m(-r) = \pm v_z^m(r)\\ B_r^m(-r) &= \mp B_r^m(r); \quad B_\theta^m(-r) = \mp B_\theta^m(r); \quad B_z^m(-r) = \pm B_z^m(r)\\ \rho^m(-r) &= \pm\rho^m(r); \quad p^m(-r) = \pm p^m(r) \end{aligned}$$

where the upper sign holds for *even* values of m and the bottom sign for *odd* values of m. The conditions above hold for $r\to 0$.

For the boundary conditions at the edge of the cylindrical vessel $r = 1$, we suppose to have a viscous, impenetrable wall, acting as a perfect conductor, which leads to the conditions: $\mathbf{v}(r = 1) = 0$ and $B_r(r = 1) = 0$. For the other quantities, we use a characteristic decomposition method [5]. In this approach, Eq.s (1) are put in hyperbolic form by neglecting the dissipative terms and the derivatives along the θ and z directions. In such a way, by introducing for a generic quantity $f(r,t)$ a dependence of the kind: $f(r-ct)$, one gets an eigenvalue system of equations, where the eigenvectors describe the waves propagating to the left or right direction at a given radial point r with the phase velocities given by the eigenvalues. On the boundary ($r = 1$), the waves are of two types: the ones entering the computational domain, and the ones going out of it. The latter can be computed by using the internal points. Using the conditions for the velocity field and B_r at the boundary, one gets some relations which allow to express the amplitude of the incoming waves, in terms of the amplitudes of the outgoing waves. Finally, the dissipative terms and the derivatives along θ and z directions (which do not contribute to the wave propagation and can be computed by using the internal points) are added to get the final value of the quantity on the boundary. The details of the calculation and the final formulae are too bulky to be inserted here. Please, refer to [6] for further details.

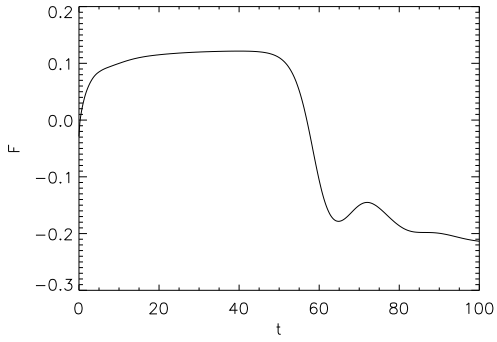

Figure 1: Time evolution of the reversal parameter obtained by integrating the set of Eq.s (1) with ρ=const. and $p = 0$ as in [2].

To compute the radial derivatives and including the boundary conditions specified above, we use fourth order accurate compact finite difference schemes [7]: $[f']_{j-1} + 4[f']_j + [f']_{j+1} = 3(f_{j+1} - f_{j-1})/\Delta_r$ at the internal points $j = 0,1,\ldots,N_r - 1$, and: $2[f']_{j-1} + [f']_j = (5f_j - 4f_{j-1} - f_{j-2})/(2\Delta_r)$ at $j = N_r$. At the point $j = 0$ the first formula can still be used due to the parity boundary conditions.

Finally, the time stepping is implemented by using a third order, explicit, Runge-Kutta scheme. The code is parallelized by dividing the computational domain along the z direction and assigning each processor a single subdomain. Communications among the processors is realized by using the Message-Passing-Interface (MPI) library. We tested the code by ensuring that it conserves the energy: we compared the energy dissipated by the numerical scheme with the energy lost through physical dissipation and we checked the former keeps at least three orders of magnitude smaller than the latter.

As a further test, we modified Eq.s (1) to integrate the same system of equations as in [2]. We start with a force-free magnetic configuration, perturbed by small amplitude velocity fluctuations. In all the runs we use a spatial resolution of $128 \times 16 \times 32$ gridpoints. In this case, we obtained a value of the temporal evolution for the reversal parameter $F(t) = B_z^{00}(r=1)/ < B_z^{00}(r) >_S$, shown in Fig. 1, qualitatively very similar to that obtained in [2] (notice the different temporal interval). A quantitative direct comparison is not possible due to the difference in the boundary conditions, but the behaviour of the reversal parameter is reproduced very well in our simulation.

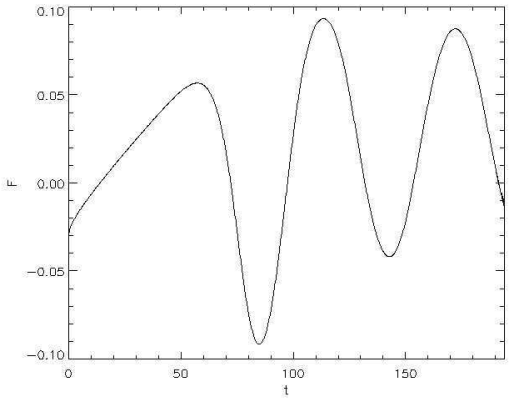

Figure 2: Time evolution of the reversal parameter obtained by integrating the original set of Eq.s (1).

Finally, to examine the differences with the fully compressible code, we re-run the simulation with the original set of Eq.s (1). The behaviour of the reversal parameter in this case is shown in Fig. 2, and it turns out to be very different from the one shown in Fig. 1. In particular, in the case of the complete system of equations, the reversal parameter changes sign several times and does not remain negative as in the previous case.

These results are very preliminar and further investigations are still in progress. However, they seem to indicate that the inclusion of compressibility plays a fundamental role in determining the temporal behaviour of the system.

References

[1] S. Cappello, D. Biskamp, Nuclear Fusion **36**, 5 (1996).

[2] D. Merlin, D. Biskamp, Rep. IPP 6/276, Max-Planck-Institut für Plasma-Physik Garching (1988).

[3] J. B. Taylor, Physical Review Letters **30**, 139 (1974).

[4] V. G. Prymak, Journal of Computational Physics **118**, 366 (1995).

[5] T. J. Poinsot, S. K. Lele, Journal of Computational Physics **101**, 104 (1992).

[6] M. Onofri, L. Primavera, F. Malara, P. Veltri, in preparation.

[7] S. K. Lele, Journal of Computational Physics **103**, 16 (1992).