Application of Parallel Programming Method for 3D MHD Computer Simulations of Magnetic Reconnection in Plasma

D. P. Kostomarov, S. V. Bulanov, J.N. Inovenkov, E.Y. Echkina, A.V. Leonenko, O.A. Pavlova*

Moscow State University, Moscow, Russia.

Introduction

The term "magnetic field line reconnection" or simply "magnetic reconnection" refers to a broad range of problems that are of great interest for space and fusion plasmas. Magnetic reconnection is accompanied by an ultra-fast release of magnetic energy which transforms into different forms such as internal plasma energy, radiation and fast particles. For this reason re-connection processes are important for numerous applications and are used in order to explain different phenomena such as the disruptive instability in tokamak plasmas, solar flares and sub-storms in the earth's magnetosphere. The problem of magnetic field line reconnection is closely related to the problem of the structural stability of vector fields [1]. A system is structurally stable if, for any small variation, the resulting system is equivalent to the initial one.

The main goal of the present paper is to consider the regimes of driven magnetic field line reconnection in three-dimensional magnetic configurations with two null points using the parallel programming methods.

Mathematical model

We describe the magnetic reconnection processes by the system of the MHD equations for the plasma density *p,* velocity v, the temperature *T* and the vector-potential A [2]:

$$\begin{aligned}
&\partial_t \rho + \nabla(\rho \mathbf{v}) = 0,\\
&\rho(\partial_t \rho + (\mathbf{v}\nabla)\mathbf{v}) = -\frac{\beta}{2}\nabla p + \left[\left[\nabla\times\nabla\times\mathbf{A}\right]\times\left[\nabla\times\mathbf{A}\right]\right],\\
&\partial_t \mathbf{A} = \left[\mathbf{v}\times\left[\nabla\times\mathbf{A}\right]\right] + \nu_m \Delta\mathbf{A}, \qquad (1)\\
&\frac{\rho}{\gamma-1}(\partial_t T + (\mathbf{v}\nabla)T) + p\mathbf{v}\nabla = \nabla\cdot(k\nabla T) + 2\nu_m\frac{(\Delta\mathbf{A})^2}{\beta},\\
&p = \rho T
\end{aligned}$$

We also assume for simplicity that plasma transport coefficients: magnetic viscosity v_m, electric conductivity σ, thermal conductivity *k,* parameter *β,* adiabatic index 7 are constant. The magnetic field is defined as $\mathbf{B} = \nabla\times\mathbf{A}$. The system is completed by the boundary and initial conditions that correspond to the current sheet formation. To describe magnetohydrodynamics waves we choose the vector-potential at the boundaries $x = \pm 1$ and y — ±1 of the form

$$A_z\left(x,y,z,t\right) = A_z\left(x,y,z,t=0\right) + r^{1/2}\mathcal{F}\left(t-1/r+1\right), \qquad (2)$$

where $A_z(x,y,z,t)$ is the *z*-component of the vector-potential, $r^2 = x^2 + y^1$. The function $\mathcal{F}$ (ξ) is equal to

$$\mathcal{F}(\xi) = \begin{Bmatrix} -E_1(\xi)\ for\ \xi > 1 \\ 0 \quad for\ \xi < 1 \end{Bmatrix}, \qquad (3)$$

and is chosen so as to impose the electric field gradually. The boundary conditions for the remaining variables are defined in accordance with the MHD equations. In the region where the plasma enters the computational domain the density and the pressure are set equal to $\rho = 1$, $\rho = 1$, while the conditions of free outflow are imposed on the part of the boundary where the plasma leaves the computational domain.

* E-mails: kostomar@cs.msu.su, bulanov@gpi.fpl.ru, inov@cs.msu.us, ejane@cs.msu.su, lalex@cs.msu.su

All the results of the numerical simulations presented below are obtained for constant magnetic viscosity v_m = 0.006, pressure corresponding to β = 0.012, thermal conductivity k = 0.01, dimensionless electric field E_1 = 0.06.

The system of equations (1) has been solved numerically inside the computational box which has the form of the cube: —1 < x < 1, —1 < y < 1, —1 < z < 1. In this paper we used the explicit finite-difference scheme which is more convenient for the numerical realization than the semi-implicit and more effective Schnak algorithm [3].

Multiprocessor implementation of computational algorithm

The system of equations contains 8 independent functions (3 components of velocity and vector-potential, temperature and density) which we call primary functions. Also we introduce 9 other functions (pressure, 3 components of magnetic field and current density, module of current density and velocity divergence) to reduce number of calculation. We call them secondary functions.

Suppose we have n processors which we can enumerate in the sequential order from 0 to *n-1* and the grid size is N_1 * N_2 * N_3. We can assume for simplicity that N_1 = kn where $k \in N$. Usually N_1 is much greater than n. Our calculations are equally distributed among η processors. We also distribute the primary arrays among n processors and each part of the whole array we consider as a circular buffer.

Let us consider the array structure at the initial time step. The first two cells are empty, the others *N* slices store the initial values of function. The value at a new time step can be calculated as follows. All the cells shift circularly by two to the left. So at the first time step function values located in cells O... N_1 — 1 at the second time step are placed into cells N_1, N_1 + 1, *o*.. N_1 — 3 and so on as a result we can get the set of formulae:

$\mathbf{f}^t$ (cur_new) = $\mathbf{G}(\mathbf{H}\,\mathbf{f}^n$ (prev, cur, next))

where $\mathbf{f}^t = \left\{f_1^t, f_2^t, \ldots, f_{17}^t\right\}$ consists of 8 primary functions and 9 secondary functions at a new time step, $\mathbf{f}^n$ is an analogous vector at the preceding time step, n = t — 1 for the primary function, n = t for the secondary function, **H** is a diagonal matrix which components h_{jj} = 1 if the calculation of f_i^t requires the information about components f_j^n and h_{jj} = 0 otherwise. In our case h_{jj} = 1 if f_i^t is the primary function and $i = j$, h_{jj} = 0 if f_i^t is the secondary function and $i = j$. G is an operator which calculates functions f_i^t i = 1..17, j = 1..17,

$$cur\_new = (first + k) mod(N_1 + 2),$$

$$prev = (first + k + l) mod(N_1 + 2),$$

$$cur = (first + k + 2) mod(N_1 + 2),$$

$$next = (first + k + 3) mod(n_1 + 2),$$

where *first* is an index of *zero* two-dimensional slice at a new time step, k ranges from 1 to *HI* — 2 (values t at the boundary two-dimensional slices are calculated with the use of boundary conditions).

For calculation of the boundary two-dimensional slice value we must use slice values from the adjacent processors. So process 0 send the last two dimensional slice, the process with the number ranging from 1 to n — 2 send two boundary slices, process n — 1 send the first slice. As a result we get the number of messages at each time step which is equal to 2(n — 2) + 2 = *2n* — 2. The size of array that stores the primary function values is $(N_1/n$ + 3) * N_2 * N_3 for process 0 and n — 1 and $(N_1/n$ + 4) * N_2 * N_3 for the other processes. Additionally we use 3 two-dimensional slices to store pressure and 8 separate variables to store other secondary functions. The multiprocessor implementation algorithm which we described in the section allows us to expect that the calculations will be really concurrent with a small latency for transfer.

Result of the computer simulation

We consider configurations with two regular null points connected by a separatrix. This system is structurally unstable. This configuration is the analogy of the "spontaneous" solar flares. Here we consider a structurally unstable current-free magnetic configuration with two connected null points. An example of such a magnetic configuration is given by a magnetic field of the form

$$B(x,y,z) = \left(-2xy - z^2 + x^2 + \in\right)e_x + \left(-2yz - x^2 + y^2 + \in\right)e_y + \left(-2zx - y^2 + z^2 + \in\right)e_z \qquad (4)$$

with $\in \neq 0$.

In the simulations we take ϵ = 0.16.

We consider initial magnetic configuration with two null points connected by the separatrix line directed the diagonal.

As the result of the evolution of the perturbation exited from the boundary, this configuration transforms into the structurally stable one. The system with two null points connected by separatrix is destroyed and transform into configuration with separate null points. The electric current is localised near the null point. As the result of this transformation the part of the energy is rested.

The multiprocessor implementation algorithm which we describe allows us to expect that the calculations will be really concurrent with a small latency for transfer. The calculation is carried out on the cluster in Science Research Computer Center. The configuration of cluster as follows: 36 processors - Pentium HI/550 MHz/ RAM 500 MB.

References

1. V. I.Arnol'd, Chapitres supplementaires de la theorie des Equations differentielles ordinaires, (Editions Mir, Moscou, 1980)
2. S. V. Bulanov, E. Yu. Echkina, I. N.Inovenkov, F. Pegoraro, V. V. Pichoushkin, Plasma Phys. Rep. 7, (2001)
3. S. V. Bulanov, E. Yu. Echkina, I.N.Inovenkov, F. Pegoraro, Phys. Plasmas 6, (2002)