§61. Development of Simulation Code on Open System in Parallel Computing Method

Ohtani, H., Ishiguro, S., Horiuchi, R.

Magnetic reconnection plays an important role in high temperature plasmas, and leads to the fast energy release from magnetic field to plasmas and the change of magnetic field topology. In order to investigate the behavior of magnetic reconnection both from the microscale viewpoint for electron and ion dynamics and from the macroscale one for dynamical change of field, a full open boundary model with a particle simulation is required. Moreover, we need to perform three-dimensional simulation in order to study relationship between excitation of instability and mechanism of magnetic reconnection. Because such a simulation becomes large scale, we develop a simulation code with distributed parallel algorithm for the distributed memory and multi-processors computer system.

For the distributed parallel algorithm, there are some programming languages, for example, Massive Parallel Interface (MPI) and High Performance Fortran (HPF). HPF is easier compared with MPI, because it is enough to only add the instruction statements for parallel calculation to the Fortran programming code. We adopt HPF for the distributed parallel computing method in this paper.

It is very important to decide what array is distributed in the distributed parallel algorithm. There are two cases. One case is to distribute space. For example, the field quantity is defined by three coordinates, and we distribute it along z-direction. In this case, the array of particle's information are also distributed to have the same index of distribution as that of the space where it exists. However, the communication between HPF processes is needed when the particle moves to the neighboring distributed space and the global calculation (for example, Fourier transformation) is performed. Another case is not to distribute the space, but to distribute only information of particles. In this case, we do not need the communication frequently because every HPF process has the same information of space and performs the field solver. We adopt the latter way in this paper.

In Fig. 1, we show the schematic illustration of distributed array of particle position, velocity and so on. Every HPF process controls its particle's array. The information of electron and ion are recorded in the first and latter half of the array, respectively. In each electron's and ion's part of the array, the information of active and waiting particles are recorded in the first and latter half, respectively. 'Active' means a array of the particles subject to the calculations of charge density and current density, and 'waiting' means a empty array for the particle which will come into the system in future. Using this type of distributed array, we can perform the parallel calculation as BLOCK distribution and attain a proper load balance between HPF processes. It is easy to change

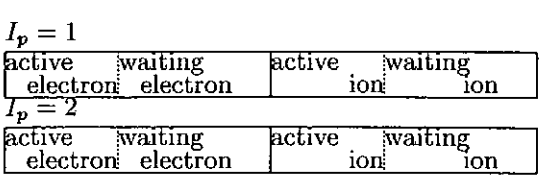

Fig. 1. Schematic illustration of distributed array of particle position, velocity and so on.

the number of parallel calculation.

The free boundary condition for particles is developed in the Ref.1 and 2 (Fig.2). We show how to calculate the number of incoming particles across the open downstream boundary in the distributed parallel algorithm. In every HPF process (I_p), we first calculate the average particle velocity $\overline{v_x}(I_p)$ and number density $n(I_p)$ in region I. Second, we obtain the net number of outgoing particles passing across the boundary during one time step Δt in the system as

$$N^{net} = -\sum_{I_p=1}^{N_p} n(I_p)\overline{v_x}(I_p)\Delta t x_b, \qquad (1)$$

where x_b is width of region I and N_p is the number of HPF process. According to the charge neutrality condition, the net numbers of electrons and ions are the same. So the numbers of incoming electrons and ions to the system are given as

$$N_e^{in} = N_e^{out} - N^{net}, N_i^{in} = N_i^{out} - N^{net}. \qquad (2)$$

Then, we distribute N_e^{in} and N_i^{in} to every HPF process as

$$n_e^{in}(I_p) = N_e^{in}/N_p, n_i^{in}(I_p) = N_i^{in}/N_p. \qquad (3)$$

We assume that the physical state outside is the same as that in region I. Then, the positions and velocities of the incoming particles can be defined by using the information of particles crossing surface 2 from left to right. In this distributed parallel algorithm, the information of particles crossing surface 2 is kept in every HPF process, and we use it for $n_e^{in}(I_p)$ and $n_i^{in}(I_p)$.

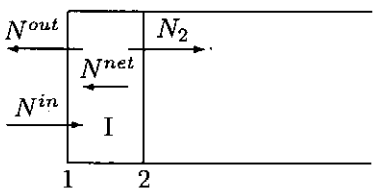

Fig. 2. Illustration of the free boundary condition for particles.

Reference
1) Horiuchi, R. et al.: Earth, Planets and Space, **53**, 439 (2001).
2) Pei, W. et al.: Phys. Plasmas, **8**, 3251 (2001).