§8. Development of PASMO Code

Ohtani, H., Horiuchi, R., Usami, S.

Magnetic reconnection is widely considered to play an important role in energetically active phenomena in high-temperature plasmas. In spite of intensive research, many basic questions on the details of mechanisms of reconnection still remain poorly understood. To clarify the relationship between particle kinetic effects and anomalous resistivity due to plasma instabilities in the reconnection phenomena, we have been developing a three-dimensional particle simulation code for an open system, called PASMO [1,2,3].

In the 2013 fiscal year, we developed the code PASMO in order to perform large-scale particle simulations on a distributed memory parallel computer system. In the last year, the code was parallelized on the domain decomposition method in one-dimensional direction (the direction is along the current direction, that is *z* direction, and the boundary condition is a periodic condition. In the decomposed domain, dynamics of particles and electromagnetic fields are controlled by one MPI process. See Fig.1.). However, the one-dimensional domain decomposition method should deal with the entire domain in *xy* plane, and cannot enlarge the simulation region in the plane when the memory size of one node is small. Because dynamics of the magnetic reconnection take place in *xy* plane, we need larger region in *xy* plane. We decompose the simulation box in three-dimension as shown in the bottom figure in Fig.1. In this decomposition, it is difficult to code the open boundary model, which was devised in the PASMO code [3]. As a first step, we developed the electromagnetic particle simulation code in periodic boundary condition, and checked the performance.

We checked the strong scaling of the developed code in the following numerical condition: The numbers of ions and electrons per cell are 50, respectively. The grid numbers are 258x258x514. The numbers of domain decomposition along xy directions are 2 and 4, respectively, and the number along z direction is changed as 2, 4, 8, 16, and 32. The number of SMP is fixed as 32. Figure 2 shows the performance of the code as a function of a number of MPI processes. A vertical axis shows inverses of calculation times normalized by the time in the 16 MPI processes case. Top of the Fig.2 is the case that the correction calculation is performed. The correction calculation modifies the microscopic inconsistencies between current and charge densities due to use of the mesh and weights by solving the Poisson equation [4]. The Poisson solver needs global calculation such as FFT, and then the performance becomes worth when the number of domain decomposition increases. Tentatively, we checked the performance without the correction calculation. The result is shown in the bottom of Fig.2. The performance keeps well when the number of the MPI processes increases. From this result, it is expected that the performance of the code is good if we adopt the algorithm, which does not need a global calculation, such as the charge conservation method [5].

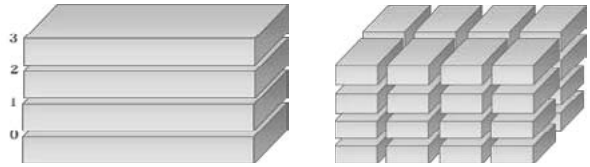

Fig.1. Concepts of the domain decomposition. Left is one-dimensional decomposition, and right is three-dimensional decomposition.

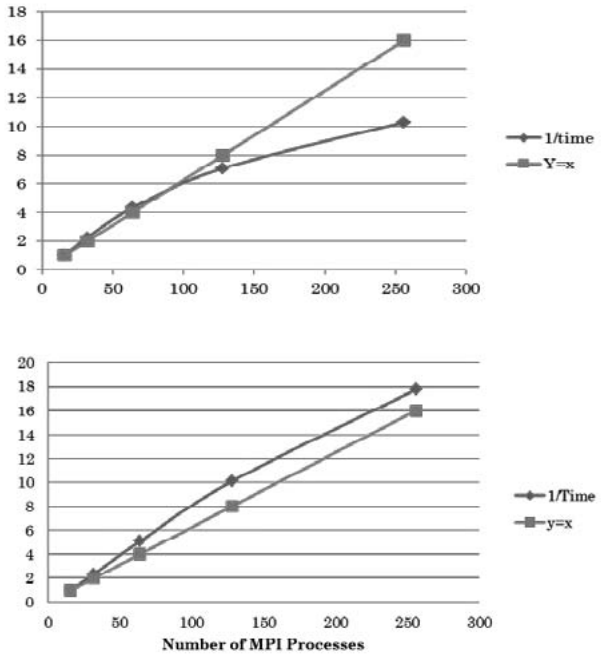

Fig.2. Performance of the code. Top and bottom are the calculations with and without correction. A vertical axis shows inverses of calculation times normalized by the time in the 16 MPI processes case.

1) Horiuchi, R. et al: Phys. Plasmas **6** (1999) 4565.
2) Ohtani, H. et al: LNCL **4759** (2008) 329.
3) Ohtani, H. and R. Horiuchi: PFR **4** (2009) 024.
4) Birdsall C.K. and Langdon, A.B.: Plasma Physics via Computer Simulation (IOP Publishing Ltd, 1991).
5) Esirkepov, T.Zh.: CPC **135** (2001) 144.