Parallelized Plasma Particle Simulation with Dynamic Domain Decomposition

Yohei Yagi [1], Hideyuki Usui [1] and Masanori Nunami [2]

[1] Graduate School of System Informatics, Kobe University, Japan
[2] National Institute for Fusion Science, Japan

1 Space Plasma Simulation

Investigation of space plasma environment and its influence upon spacecraft is important in space utilization. Computer can calculate enormous task in short time by late improvement of calculation ability now. Because of this fact, space plasma simulation which simulates space plasma phenomena quantitatively on virtual space came to be known as a powerful tool.

In this study we use Particle-In-Cell (PIC) [1] as plasma particle simulation method. This method calculates interaction between plasma particles and electromagnetic fields without contradiction. Electromagnetic fields defined on mesh are obtained with Maxwell's equation and plasma particle dynamics with the equation of motion.

In the simulation, more than ten billion plasma particles are treated and they are updated every time step. Therefore computational cost is very high. To reduce the computational cost parallel computing by advanced supercomputer is necessary.

In this study, we are parallelizing own newly developed Adaptive Mesh Refinement PIC (AMR-PIC) [2] simulation code by incorporating Dynamic Domain Decomposition (DDD) scheme. By using this scheme we confirm the decrease of memory amount necessary in simulation and the increase of efficiency of parallel computing compared with conventional PIC simulation code.

2 Adaptive Mesh Refinement PIC

Full PIC simulation is a powerful simulation method to investigate microscopic phenomena because it can treat electron as well as ion kinetics with no fluid approximation. However spatial grid size and time step interval are basically determined by the Debye length which is a characteristic spatial length for the electrostatic shielding of a charged particle, and the Courant condition which gives the upper limit of the time step interval to avoid the numerical instability. We need to use the minimum spatial grid size and time step interval which correspond to the micro-scale phenomena. For example, when a simulation domain has high density region, the local Debye length at the highest density becomes very small and the corresponding grid size should be small enough. In such a case, full PIC simulation code requires huge number of grid points to model the region to guarantee numerical stability. In the conventional PIC code with uniform grids, the smallest special grid size which is used in the highest density region has to be assigned to the other low density regions. This method is not efficient when we use limited computer resources such as the amount of memory and calculation time.

To realize an efficient simulation with reasonable cost of computer resources, we introduce Adaptive Mesh Refinement PIC (AMR-PIC) method into the conventional PIC code. In AMR-PIC method, the spatial and temporal resolutions can be adjusted locally and dynamically on the local scales of phenomena. AMR-PIC method adopts hierarchical cell structure as shown in Figure 1 to implement local and dynamic adjustment of resolution. In hierarchical structure, the spatial and temporal resolutions are defined according to the hierarchy levels, where high and low levels correspond to the fine and coarse grid systems, respectively. As the simulation system evolves, some complex micro-scale phenomena can locally and intermittently occur in a hierarchical domain (Level L). If the grid size in Level L is too coarse to simulate the local complex phenomena, A higher hierarchical domain (Level L+1) is adaptively created in which the grid spacing size and the time step interval become half of those used in the domain of (Level L).

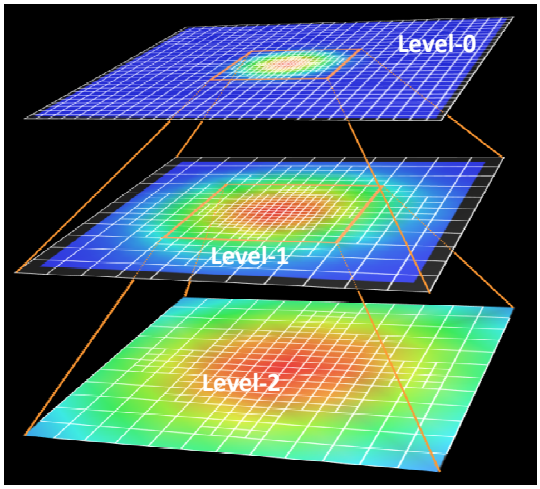

Fig.1. hierarchy of AMR-PIC

Arbitrary physical parameter can be used for a condition with which we can decide a region which needs hierarchy layer grids. For example if particle density is set for a condition, AMR-PIC creates a fine layer at a region where particle density is higher than other domain.

3 Dynamic Domain Decomposition

3.1 Load Imbalance

Parallel computing is a method which uses multiple computers to calculate divided tasks concurrently. But a certain problem occurs when difference of size of divided tasks exists in parallelized computers. The problem occurs in synchronization which is necessary to synchronize state of progress of calculation among computers. In this process, computer which has comparatively small task and takes short computation time waits for a computer which has large task and takes long

computation time. In this situation computer which has small task becomes idle. Therefore due to this problem, computer resource of parallelized computers will be wasted. This difference of tasks called load imbalance. To avoid load imbalance, it is desirable that the amounts of divided tasks are equivalent among computers used for parallel computing.

Load Imbalance is caused by particles in parallelized PIC simulation and hierarchical structure in parallelized AMR-PIC. In PIC simulation, particles move from a grid to another one in simulation space. In case of parallelized simulation which divides simulation space, uniformity simulation space division is not efficient for solving load imbalance because of ununiformed number of particles located in each divided region. Meanwhile in AMR-PIC simulation, AMR scheme creates fine grids locally and dynamically with hierarchical structure. Fine grids in (Level L+1) have half grid size and time step interval of grids in (Level L). Thus the number of grid is not equal between the processor which has fine grids and the processor which has no fine grids. And the cost of simulating is not equal between a grid in (Level L+1) and a grid in (Level L) because of difference of time step interval.

For efficiency of parallel computing, we should solve load imbalance problem and developed a new domain decomposition method.

3.2 Details of Dynamic Domain Decomposition

The new method is called Dynamic Domain Decomposition (DDD) scheme. With DDD, we can divide a simulation space to equalize the number of particles and cost of fine grid created by AMR of each divided region during the simulation run.

In this study, we introduce particle loops as a unit of simulating cost for computer. Particle loop means the cost of each cell in simulation space. We obtain particle loops to count the number of particles in one cell basically. However, in a grid in (Level L+1), time step interval of that grid is half of a grid in (Level L). Thus (Level L+1) grid requires twice calculating cost of (Level L). And the computational cost of particles becomes double in (Level L+1) grid. Therefore we count particles with weight in hierarchy grids. That weight of particle doubled as hierarchy level is increase by one. For example, we count a particle in (Level 0) which is base level as one particle loop. A particle in (Level 1) is counted as two particle loops. And a particle in (Level 2) is counted as four particle loops.

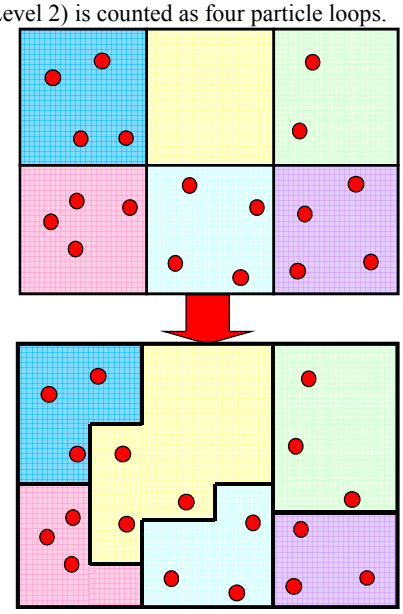

Fig.2. Process of Dynamic Domain Decomposition

DDD scheme calculates the summation of particle loops of whole simulation domain and we obtain average number of particle loops of simulation space. This average is used for target value to solve load imbalance problem. DDD modifies each divided sub-domain so that the summation of particle loops of each divided sub-domain becomes almost equal to the average particle loops.

Fig.2 shows the image of DDD. The whole domain is divided into six sub-domains. Each sub-domain is taken care by each process of computers. Six colour rectangles shown in the upper part of Fig.2 means whole simulation domain. And particles are indicated in red. As a result of DDD, each sub-domain is modified so that summation of particle loops of each divided region become almost equal.

4 Conclusion

We introduced AMR-PIC scheme which can increase simulation precision locally and decrease used amount of memory and Dynamic Domain Decomposition scheme which can increase efficiency of parallel computing into plasma particle simulation code with PIC which needs enormous computation memory and computation time. Using this simulation code we increase computation time 1.7 times as much as computation time of conventional PIC simulation code by solving the model which has concentration of particle locally in simulation space.

References

[1] C. K. Birdsall and A. B. Langdon, “Plasma Physics Via Computer Simulation,” New York: *McGraw-Hill*, 1985.

[2] A. M. Khokhlov,”Fully Threaded Tree Algorithms for Adaptive Refinement Fluid Dynamics Simulations”, J. Comput. Phys. 143, 519, 1998.