Development of a Scalable PIC Simulator and Its Application to Spacecraft–Plasma Interaction Problems

Yohei Miyake[1], Hiroshi Nakashima[2], and Hideyuki Usui[1]

[1]Graduate School of System Informatics, Kobe University, Japan
[2]Academic Center for Computing and Media Studies, Kyoto University, Japan

1 Introduction

The particle-in-cell (PIC) method is widely used for various problems in space plasma physics. Spacecraft-plasma interaction problem is one of such extensive topics. With increasing space development and exploration, a strong demand should arise regarding better understanding of plasma disturbance around a spacecraft. For the study, we have developed an electromagnetic PIC code EMSES [1] for the study of spacecraft-plasma interactions. EMSES is assumed to run on modern supercomputers, and thus various optimizations are made for distributed-memory systems.

Among the optimizations, load balancing is a very common issue for parallelization of the PIC simulations. In the PIC method, a huge number of particles are mapped onto a simulation domain that is segmented with a mesh, and the particle population potentially takes arbitrary distributions. Hence, it is generally difficult to distribute computation loads equally in terms of both the number of particles and domain size to many computation nodes. We have proposed a dynamic load-balancing algorithm OhHelp [2] for the PIC simulations, and successfully implemented EMSES with it.

2 EMSES: Electro-Magnetic Spacecraft Environment Simulator

In EMSES, we solve Maxwell's equations and Newton's equations of particle motion as a time-domain problem. EMSES handles many macro-particles as Lagrangian variables distributed over a simulation domain. The domain is segmented with a mesh and electromagnetic field vectors are defined on the grid points as Eulerian variables. An interaction between these particles and fields is then processed by means of a standard PIC scheme [3].

EMSES has a capability of including solid bodies of a spacecraft. Specifically, we process a charge accumulation on spacecraft surfaces, which is a source of the spacecraft charging, and redistribute the surface charge in order to maintain an equi-potential over the conducting surface. In addition, we eliminate a remained transverse electric field that is tangential to the surfaces. By the treatments, we can handle both electrostatic and electromagnetic problems emerging near a spacecraft.

3 Optimizations for Distributed-Memory Parallel Computers

EMSES is parallelized based on static domain decomposition. The simulation domain is partitioned into equal-size subdomains, and each subdomain is assigned to a certain computation node. This simple decomposition, however, poses a severe load imbalance when most particles are concentrated on a certain subdomain.

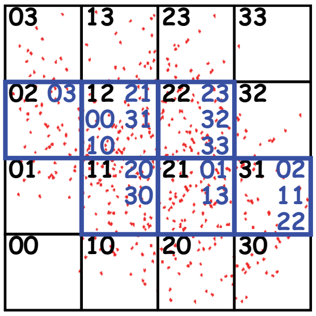

Fig. 1. Space domain partitioning used in OhHelp. The red dots represent the distribution of particles mapped onto the domain. The black-color digit in each subdomain is a computation node that is always responsible for the subdomain. The blue-color digits are computation nodes temporarily assigned to the subdomain as helpers. For example, node 11 is helped by nodes 20 and 30 by deputing parts of its own particles and replicated field data to them.

We resolve this imbalance by applying a dynamic load-balancing algorithm called OhHelp [2]. In OhHelp, load balancing in terms of the number of particles is accomplished by making each node help another heavily loaded node. The loaded node deputes a part of its own particles and replicated subdomain field data to its helpers. This basic concept is shown schematically in Figure 1. OhHelp monitors particle movements through subdomain boundaries to check whether the helper assignment at that moment can keep good load balancing. When unacceptable imbalance found, OhHelp dynamically reconfigures the helper assignment to regain perfect balancing. The detailed algorithms for the helper assignment and load-balancing check are described in our previous papers [2].

Since the load balance check and particle transfer between subdomains potentially cause a considerable overhead, we should take care of reducing the frequency of these operations in implementation of EMSES with OhHelp. Such optimization is achieved by introducing an overlap region with a certain thickness around the interface between adjacent subdomains. Since a maximum particle velocity in a system is generally much smaller than the light speed except for relativistic problems, an overlap region with a thickness of a few grids can reduce the frequency of the particle transfer drastically. Furthermore, by exploiting the fact that some particles (e.g., gyrating

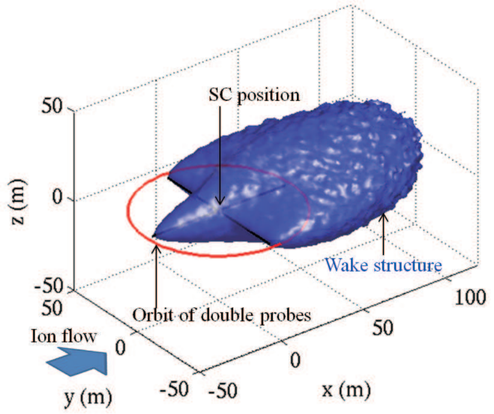

Fig. 2. Isosurface plot of proton density in case with deployment of thin wire booms.

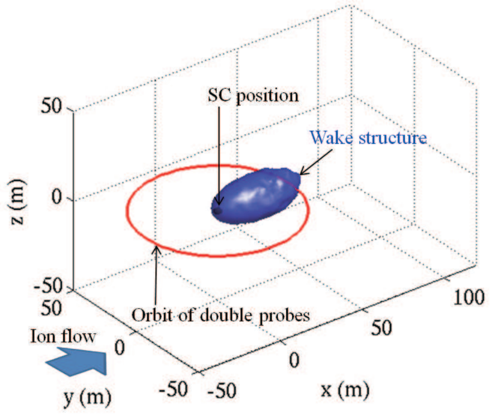

Fig. 3. Isosurface plot of proton density in case without deployment of thin wire booms.

particles) go back and forth between adjacent subdomains crossing their interface, the optimization promises a certain level of reduction in the total amount of inter-node communication.

As the lowest-level optimization, we sort particles based on their coordinates. In the current and charge computation phases, the moments of each particle should be deposited onto grid points adjacent to the particle position. Hence, the particle sorting can improve the locality of reference to the current and charge arrays, and thus promote effective use of cache memories.

We have conducted a performance evaluation of the optimized code by means of a weak scaling on the T2K Open Supercomputer in Kyoto University. In order to confirm a dynamic load-balancing effect provided by OhHelp, we examine both uniform and highly-localized particle distributions in the evaluation. In the latter case, a majority of particles reside only in a small volume corresponding to the size of one subdomain. As a result, we found a good scalability for parallel executions using up to $\sim 10^3$ cores for both cases. We also confirmed that the performance degradation due to the particle localization is $\sim 15\,\%$. Conceptually, without no capability of load balancing such as provided by OhHelp, the localized case should show little speedup even when using 10^3 cores. (In practice, there is no potential of executing such a simulation due to the memory shortage of the heavily loaded node.) By considering that point, the $15\,\%$ performance degradation is concluded to be within an acceptable level.

4 3-dimensional Simulations on Spacecraft Interactions with a Streaming Plasma

We have applied the optimized EMSES code to an ion wake problem around a spacecraft immersed in a streaming plasma. Ion wakes are often identified as a potential source of errors in electric field measurements based on the double-probe technique. Particularly in tenuous plasmas, a positively-charged spacecraft due to photoemission causes enhancement of a wake potential, resulting in serious spurious electric fields observed by double-probe instruments. To examine such wake enhancement, we performed 3-dimensional PIC simulations. In the simulations, we processed 3×10^9 particles and $512 \times 512 \times 128$ grid points by 1024-process executions.

Figures 2 and 3 show isosurface plots of the proton density obtained in cases with and without deployment of thin wire booms from the spacecraft body. The isosurfaces represent points of a local density being half of the unperturbed level, and thus can capture the overall wake structures. Generally, a wake is formed behind a solid object, only when its size is sufficiently larger than the local Debye length. However, Figures 2 and 3 indicate that not only a spacecraft body but also even thin wire booms contribute to wake formation substantially, despite the extremely small radius of the booms ($\sim 1\,\mathrm{mm}$) compared to the local Debye length ($\sim 10\,\mathrm{m}$). This effect is caused by a high positive potential of the booms. Such a potential enhances an effective radius of the boom conductors seen by streaming ions over their physical size. The results provide important knowledge for the assessment of spurious electric fields caused by enhanced ion wakes.

5 Conclusion

We have developed the 3-dimensional PIC code for the study of spacecraft-plasma interactions. The code is parallelized based on the simple domain decomposition, and the scalability in terms of the number of particles are ensured by applying a dynamic load-balancing algorithm OhHelp. With a number of further optimizations such as frequency reduction of the load balance check, we confirmed a good scalability for parallel executions using up to $\sim 10^3$ cores. The developed code is applied successfully to practical problems of spacecraft-plasma interactions.

Acknowledgement

The computations in the present study were performed using the Fujitsu HX600 Cluster of Academic Center for Computing and Media Studies at Kyoto University.

References

[1] Miyake, Y. and H. Usui, New electromagnetic particle simulation code for the analysis of spacecraft-plasma interactions, Phys. Plasmas **16** 062904:1–11, 2009.

[2] Nakashima, H., Y. Miyake, Y. Omura, H. Usui, OhHelp: a scalable domain-decomposing dynamic load balancing for particle-in-cell simulations, Proc. of the 23rd Intl. Conf. on Supercomputing, New York, 90–99, 2009.

[3] Birdsall, C. K., and A. B. Langdon, Plasma Physics via Computer Simulation, McGraw-Hill, New York, 1985.