Development of Unstructured-grid EM particle Code for Spacecraft Environment Analysis

M. Okada1, H. Usui2 , Y. Omura2, T. Sugiyama3, H. O. Ueda 4, T. Miyake5 and T. Murata6

1 *Research Organization of Information and Systems, National Institute of Polar Research, Tokyo, Japan*
2 *Research Institute for Sustainable Humanosphere, Kyoto University, Uji, Kyoto, Japan*
3 *Japan Agency of Marine-Earth Science and Technology, The Earth Simulator Center, Yokohama, Japan*
4 *Japan Aerospace Exploration Agency, Tsukuba, Japan*
5 *Toyama Prefectural University, Toyama, Jpan*
6 *Ehime University, Ehime, Japan*

Geospace environment simulator (GES) has started as one of the advanced computing research projects at the Earth Simulator Center in Japan Marine Science and Technology Center since 2002. By using this computing resource, a large scale simulation which reproduces a realistic physical model can be utilized not only for studying the geospace environment but also for various human activities in space. ES project aims to reproduce fully kinetic environment around a satellite by using the 3-dimensional full-particle electromagnetic simulation code which includes spacecraft model inside. Spacecraft can be modeled by the unstructured-grid 3D FPEM code. We will report current status of porting our simulation codes onto the ES and our concept of achieving the satellite environment in conjunction with the space weather.

1. Introduction

We have developed a 3-dimensional electromagnetic particle simulation code with an unstructured-grid system. This code solves Maxwell's equations which is discretized with tetrahedral elements in 3D simulation space. Plasma particles are also traced by solving the equations of motion with the Buneman-Boris method. The main advantage of this code is the adaptability of modeling more realistic shape of a spacecraft than the orthogonal grid code. Thus, this simulation code is suitable for analyzing the plasma environment in the vicinity of a spacecraft especially in the region within a Debye length from the surface of the spacecraft as well as the spacecraft charging phenomena. We will show the scheme of this
code and also show a couple of results from the test simulation runs taking into account of a realistic shape of a spacecraft..

NASA has developed a sophisticated spacecraft charging analyzing software package, NASCAP [1]. This software solves electrostatic potential at the surface of a spacecraft. The code has widely been used in spacecraft design for more the a decades and achieved certain reliability in our community. Though, one deficiency is that this code solves only static potential and not be able to solve time-dependent problem neither nor the electromagnetic problems, such as EM wave propagation analysis.

European research project, Spacecraft Plasma Interaction Network in Europe (SPINE), has been started since 2000, [2]. This project aims to construct an open source software of spacecraft-plasma interaction modelling tools. PicUp3D is one of the SPINE software and adopts unstructured-grid to model a spacecraft environment. This code also aims to solve spacecraft charging and not suitable for solving the EM environment at this moment.

2. Simulation Code

We have developed a 3-dimensional electromagnetic particle simulation code with an unstructured-grid system. This code solves Maxwell's equations which is discretized with tetrahedral elements in 3D simulation space. Plasma particles are also traced by solving the equations of motion with the Buneman-Boris method. The main advantage of this code is the adaptability of modeling more realistic shape of a spacecraft than the orthogonal grid code. Thus, this simulation code is suitable for analyzing the plasma environment in the vicinity of a spacecraft especially in the region within a Debye length from the surface of the spacecraft as well as the spacecraft charging phenomena.

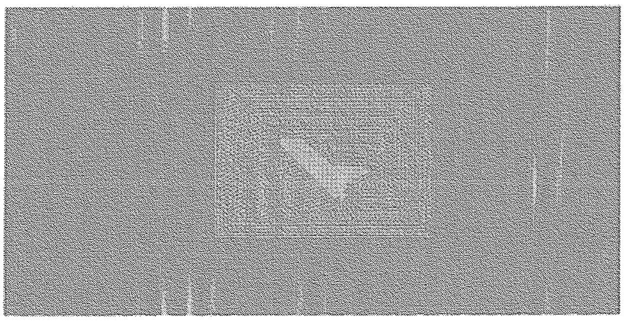

Fig. 1 Conceptual image of spacecraft environment modeled in 2D system.

Figure 1 shows a conceptual image of a spacecraft environment modeled in 2D simulation space. Unstructured-grid is used in the vicinity of a spacecraft. Wider area is solved with the orthogonal grid. By using this compound model, we can efficiently solve wider area with the high-performance supercomputers. Tracing plasma particles is indeed costly procedure still by using supercomputers. It becomes more and more reasonable way to understand spacecraft environment without launching a real spacecraft nowadays.

Equations (1)-(6)are the basic equations solved by the EM particle simulation code. Maxwell's equations are descritized by unstructured-grid coordinate in the vicinity of a spacecraft. Poisson's equation (3) can be solved by Finite Element Method (FEM) over the tetrahedral mesh area.

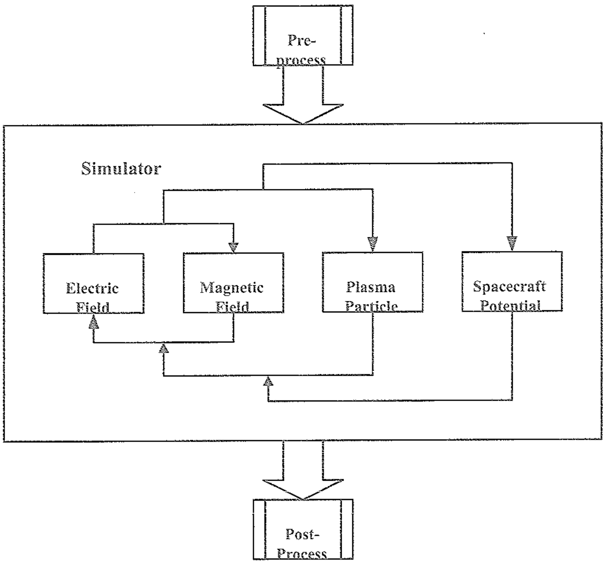

Fig. 2 Main flow of the FPEM simulator.

Figure 2 illustrates main flow of the full-particle electromagnetic spacecraft-plasma environment simulator. Electric and magnetic fields are selfconsistently solved by the leap-frog method. Plasma particles move by integrating equation of motion under the coulomb force and Lorenz force.

The motion of plasma particle generates current and the current affects the electromagnetic field. Plasma particles colliding the spacecraft surface determines the spacecraft potential and the spacecraft potential affects the plasma trajectories and the electromagnetic field. These multiple feedback system includes nonlinearity. Thus, the difficulty of understanding the plasma nature bases on this nonlinear feedback system.

In order to maximize computational resources, we tuned our simulation code for the Earth Simulator. Figure 3 shows the parallel efficiency achieved by the ES using domain decomposition method. Message passing interface (MPI) is used for the parallelization. The parallel efficiency of 91.39% is obtained at the condition of 1000CPUS in parallel execution. 97.115% of the code is vectorized. Final target of this project is to construct a coupled-simulation technique such as an unstructured-grid simulation interacts with orthogonal-grid simulation as shown in Figure 2.

$$\nabla \times \mathrm{E} = -\frac{\partial \mathrm{B}}{\partial t} \tag{1}$$

$$\nabla \times \mathrm{B} = \mu_o \mathrm{J} + \frac{1}{c^2}\frac{\partial \mathrm{E}}{\partial t} \tag{2}$$

$$\nabla \cdot \mathrm{E} = \frac{\rho}{\varepsilon_o} \tag{3}$$

$$\nabla \cdot \mathrm{B} = 0 \tag{4}$$

$$\frac{dv_i}{dt} = \frac{q_s}{m_s}(\mathrm{E} + v_i \times \mathrm{B}) \tag{5}$$

$$\frac{dx_i}{dt} = v_i \tag{6}$$

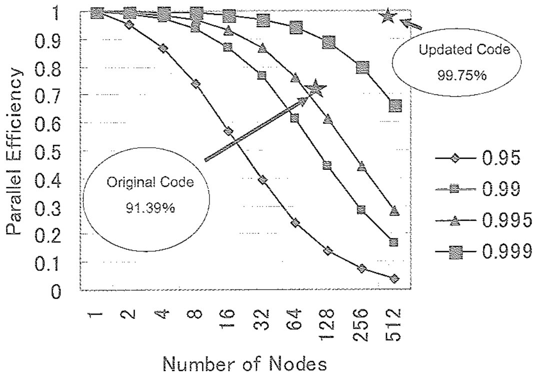

Fig.3 Parallel efficiency measured on ES and Amdar's Law.

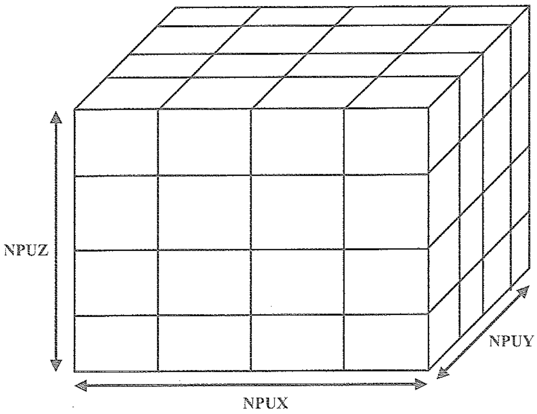

Fig.4 Domain decomposition method is used on the Earth Simulator for parallelization.

This document is provided by JAXA.

In order to maximize simulation region, we must use full computational resource of the Earth Simulator. We estimated two way of parallelization to achieve maximum optimization on the ES. One is the domain decomposition method and the other is the particle decomposition method. ES system adopts SMP (Symmetric Multi-Processor) system within a node and distributed memory multi-processor system between the nodes. Thus, we adopts both domain decomposition method and the particle decomposition method. FPEM uses particle decomposition method within a node and domain decomposition method is used for parallelization between the nodes.

Figure 4 shows how we decompose 3D simulation region on the ES. Each x, y and z directions are divided into 8 sub-domains. Each sub-domain is parallelized with particle decomposition method. Hole simulation region is parallelized into 8x8x8=512 sub-domains by domain decomposition method.

Main concern of the highly parallel computing is known as Amdar's Law.

$$T_{all} = T_{init} + n(T_{calc} + T_{comm}) \tag{7}$$

$$T_{calc} = k_1 \frac{N_g^3}{p}, T_{comm} = k_2 N_g^2 p^{\gamma} \tag{8}$$

Total computation time, T_{all}, can be estimated as a sum of initialization time, T_{init}, and calculation time, T_{calc}, and communication time, T_{comm}, multiplied by the number of loop, n. Calculation time decreases as increase the number of nodes, p, used for simulation as indicated equation (7). Though, communication time, T_{comm}, can increase as increase of the number of nodes, p. If the ES system performs as if $\gamma<1$, T_{comm} will become small relative to the T_{calc} even 512 parallel computation. Constant k_1 and k_2 are determined by the simulation code. These constants are well-known as the granuality parameters. We measured these parameters by test simulations and obtained $k_1=1.3$ and $k_2=0.0013$. By using these numbers, k_1 and k_2, we can estimate total simulation time as Figure 5.

Figure 5 indicates total computation time in seconds as a function of number of nodes for different system size in MGrid. Total computation time decreases as the number of nodes increases until 64 nodes. If we use more than 128 nodes, communication time takes more than the calculation time. Thus, parallelization with 64 nodes provides best performance for relatively small system size model.

On the other hand, realistic simulation model consumes huge system resource, such as 1000x1000x1000Grid=1000MGrid. If we model $1km^3$ simulation box, 1000MGrid model is the minimum scale of a simulation. If we inject 64particles/grid, it consumes 3TB of memory in total. The size of main memory in each node is 16GB for ES, the simulation system in each node becomes 125x125x125Grids.

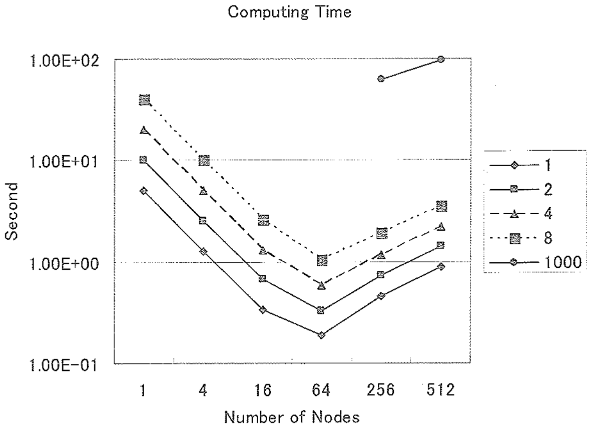

Fig. 5 Estimated time of simulation for various number of nodes.

Memory consumption is another concern for both computational point of view and system size point of view. Figure 6 shows how large system size we can model on ES. We have adopted list-vector method in original code. This method consumes huge amount of memory for faster computation on vector processors as a work array. This work array consumed about 50% of the total memory size. We installed a new work-area-free algorithm into the code and optimized for ES. New algorithm consumes about 5% of total memory size. Thus, we successfully saved 45% of memory by the optimization. After optimization granuality of the code also increases and the parallel efficiency became 99.75%.

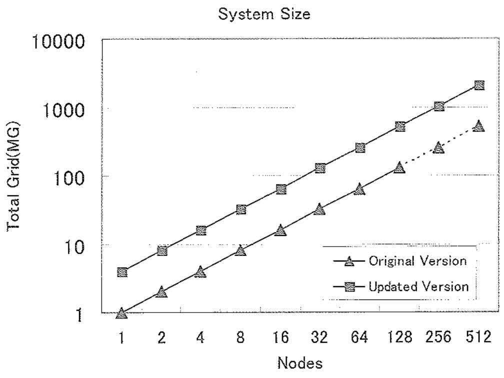

Fig. 6 Total system size achieved by ES before and after optimization of the FPEM code.

Figure 6 shows the system size before and after the work-area optimization. Before the optimization, maximum size of the simulation system available on ES was about 1000x1000x500Grid in total. We achieved 1000x1000x2000Grid=2000MGrid model after optimization.

3. Pre- and Post-process

Pre- and Post-process may be the biggest concern other than the simulator engine itself. In order to build realistic model, we must incorporate CAD system and visualization system. Small satellite such like, INDEX and Sphere-Probe can be model by CAD system installed on PC. This method can be applied up to 0.1MGrid model because of the grid generation. But once the configuration becomes realistic total number of grid will easily become more than 1MGrid. We have tested several types of CAD systems and grid generation systems.

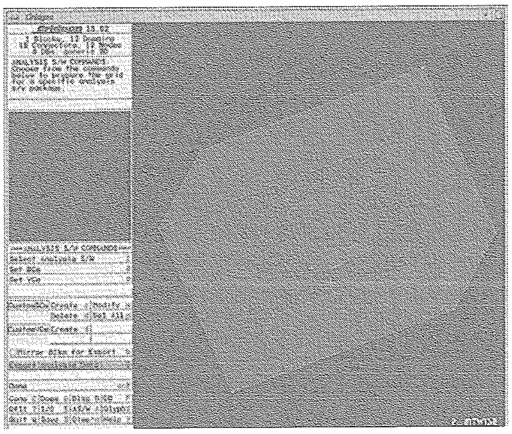

Fig. 7 GEOTAIL like structure modeled inside the simulation box. The size of the outer edge is 10m.

Figure 7 shows GEOTAIL like structure modeled inside the simulation box. A grid generation software, GridGen® is used. The total number of grid is more than 2MGrid. Minimum size of a grid becomes 1cm and the scale of simulation box is 10m. It also consumes 4-6 months of engineering time for modeling. This process take more time than the simulation time itself. After simulation run we need to visualize the results obtained by the simulation. We are testing several type of visualization software. Figure 8 is a sample output visualized with AVS

3. Conclusion

We transplanted our full-particle electromagnetic plasma simulation code to the Earth Simulator. Fundamental performance tests have been performed by the full ES nodes. Maximum parallel efficiency of 99.75% was achieved in 512 nodes. Total system size of 2000x1000x1000Grid model can be modeled on the ES by using full ES nodes.

Pre- and post-process of unstructured-grid model has been tested for several types of models. Current plans of future simulation are the following:

1. Piggy-bag satellite: INDEX model
2. Sphere-probe model
3. GEOTAIL model
4. Solar-sail model
5. Ion-thruster engine model
6. Reconnection model

In order to confirm the results of our code, we need to solve simple physical model, like sphere-probe model. This will give the adaptability of the simulation. INDEX model is expected to give a realistic model to understand satellite environment in the aurora region. This model is also thought to be feasible due to the size of the satellite. GEOTAIL model and Ion-thruster models are expected to be understood current satellite mission. Solar-sail and reconnection models are one of our goals to give a feasibility study for the future satellite missions. Reconnection is a first interest in the plasma physics in the earth's magnetosphere.

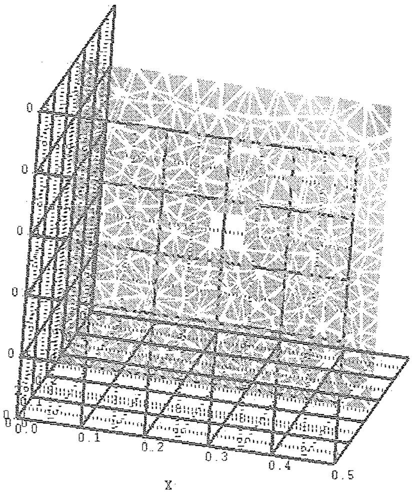

Fig.8 Electrostatic potential visualized around INDEX satellite by AVS.

References

NASA Charging Analyzer Program, http://pwerweb.grc.nasa.gov/pvsee/software/NASCAP.html, NSA Glenn Research Center

Spacecraft-Plasma Interaction Network in Europe, http://www.estec.esa.nl/wmwww//wma/R_and_D/Spine/Events.html, 2000

This document is provided by JAXA.