§23. Development of a Multi-Scale Simulation Code with Adaptive Mesh Refinement and its Application to Conventional Codes

Usui, H., Kito, S. (Graduate School of System Informatics, Kobe Univ.),
Nunami, M.

Adaptive Mesh Refinement (AMR) technique can provide efficient numerical calculation by adapting fine cells to regions where higher numerical resolution is required. However, it is generally difficult for users to implement the AMR technique in their generic simulation programs which use uniform cells. Meanwhile, to investigate multi-scale phenomena in space plasma environment including plasma kinetic effects, we have been developing a new electromagnetic Particle-In-Cell (PIC) code called PARMER by incorporating the AMR technique [1,2]. In the present study, based on the numerical technique on AMR adopted in PARMER, we developed a computational framework for blocked-structured AMR simulation by which we can easily convert a generic uniform-cell program to the one with the AMR treatment [3].

In the block-structured AMR, regions required for the AMR treatment have a self-similar structure. The self-similar block-structured domains for AMR are managed in a fully threaded tree (FTT) data structure which allows recursive refinement on a block-by-block basis. Each block consists of a domain formed with the fixed number of cells with uniform cell size. A block in a different level of refinement in the FTT structure has different cell size keeping the same number of cells. In each block, we can incorporate the same uniform-cell program of our interest and independently perform the simulation. Since each block has a common domain with the same number of cells, what we need to consider is the cell size in each block depending on the refinement level which is given in the FTT structure.

As one of the application to the AMR framework, we tried to implement the one-dimensional plasma particle simulation called KEMPO1. To deal with a large number of plasma particles, we developed a new module by which particles located in one node region are managed in one common array defined in the corresponding node. The particles do not belong to each block which consists of the node and they are not affected by the generation of a hierarchical block. When some particles move from one node to the other, these particles are transferred to the corresponding node by MPI. In the particle simulation, the current density, which is defined at each grid point as a field component, is very important because it connects between the plasma dynamics and the fields. The current density is obtained by summing up the moments of particles located in each grid according to the continuity equation. We modified the routine of this current calculation applicable to the hierarchical grid system and implemented it to KEMPO1.

In order to examine the function of the AMR-framework KEMPO1, we performed test simulations of beam instability. In the simulations we have the background ions and electrons and an electron beam. As the initial condition, these plasma components are uniformly distributed in the simulation domain. To compare the results, we also performed the same simulation with the original KEMPO1. In the simulation using the AMR-framework version, refined blocks are generated when the intensity of the local electric field exceeds a threshold level.

Figure 1 shows the snapshots of v_x-x phase diagram of the particles and the x component of the electric field for the two simulations. In comparison with the two v_x-x phase diagrams, there seems no significant difference in terms of the number of the vortex and the overall structure. However, the E_x profiles are slightly different from each other. It may be caused by the different grid size at the boundaries between the hierarchical blocks. However, the locations of these boundaries change in time and the errors will be smeared out, which does not cause a big problem for this model. Although we need to decrease the errors at the boundaries between the hierarchical blocks, we basically could confirm that KEMPO1 was successfully implemented to the block-structured AMR framework.

1) Usui, H. et al.: Procedia Computer Science **4** (2011) 2337.
2) Usui, H. et al.: Plasma and Fusion Research **8** (2013) 2401149.
3) Usui, H. et al.: Procedia Computer Science **29C** (2014) 2351.

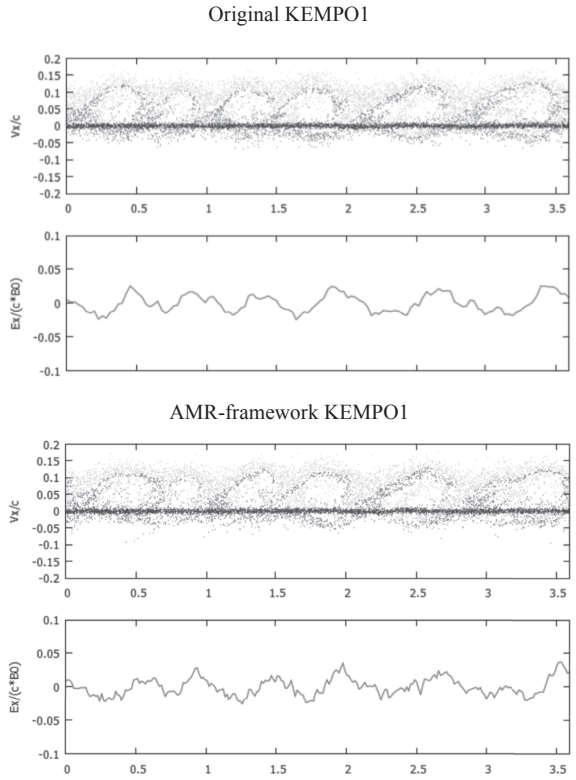

Fig. 1. Snapshots of v_x-x phase diagram and E_x