Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 214 (2006) 550–566

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

# Electromagnetic full particle code with adaptive mesh refinement technique: Application to the current sheet evolution

Keizo Fujimoto [*], Shinobu Machida

*Department of Geophysics, Graduate School of Science, Kyoto University, Kitashirakawa Oiwaketyo, Kyoto 606-8502, Japan*

Received 30 May 2005; received in revised form 4 October 2005; accepted 4 October 2005
Available online 21 November 2005

**Abstract**

We have developed a new two and a half dimensional electromagnetic particle code with adaptive mesh refinement (AMR) technique in an effort to give a self-consistent description of the dynamic change of the plasma sheet in association with magnetic reconnection, which includes multi-scale processes from the electron scale to the magnetohydrodynamic scale. The AMR technique subdivides and removes cells dynamically in accordance with a refinement criterion and it is quite effective to achieve high-resolution simulations of phenomena that locally include micro-scale processes. Since the number of particles per cell decreases in the subdivided cells and a numerical noise increases, we subdivide not only cells but also particles therein and control the number of particles per cell. Our code is checked against several well-known processes such as the Landau damping of the Langmuir waves and we show that the AMR technique and particle splitting algorithm are successfully applied to the conventional particle codes. We have also examined the nonlinear evolution of the Harris-type current sheet and realized basically the same properties as those in other full particle simulations. We show that the numbers of cells and particles are greatly reduced so that the time to complete the simulation is significantly shortened in our AMR code, which enables us to conduct large-scale simulations on the current sheet evolution.

*Keywords:* Electromagnetic PIC code; Adaptive mesh; Particle splitting; Magnetic reconnection

## 1. Introduction

Magnetic reconnection plays an important role in a fast conversion process of the magnetic energy to plasma kinetic and thermal energy. It occurs in the accretion disks of astrophysical bodies, solar flares, and the Earth magnetosphere, so that it is a ubiquitous phenomenon in the universe.

In the Earth magnetosphere, magnetic reconnection facilitates the entry of particles and energy from the solar wind into the magnetosphere at the day-side magnetopause. It is also believed to affect the dynamics of magnetospheric substorms by changing the current sheet structure and the configuration of magnetic field lines in the Earth magnetotail. In a reconnection process, the ideal magnetohydrodynamic (MHD) condition

---

[*] Corresponding author.
*E-mail addresses:* keizo@kugi.kyoto-u.ac.jp (K. Fujimoto), machida@kugi.kyoto-u.ac.jp (S. Machida).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.10.003

breaks down in the diffusion region arising near the X-line, where dissipation processes of the magnetic field are dominant. In a sufficiently collisional plasma, the resistive MHD theory (e.g. [24,25,33]) is valid for describing the diffusion region by parameterizing collisional effects. However in a region where the classical collision rate is very small such as in the magnetosphere, the Sweet–Parker diffusion region is elongated and the reconnection rate is impractically low [36]. Actually in such a collisionless plasma, the diffusion region develops a two-scale structure associated with the electron and ion scales, that is, the electron inertial length $\lambda_e$ and the ion inertial length $\lambda_i$, respectively. Within a distance from the X-line of the order of $\lambda_i$, the ions are easily unmagnetized because they have large gyroradii compared with the scale size $\lambda_i$. We call this region the ion diffusion region. On the other hand, even closer to the X-line the electrons also decouple from the ambient magnetic field. This region, which is called the electron diffusion region, has scale size of $\lambda_e$. Inside the electron diffusion region, where the electrons are unmagnetized and strongly accelerated away from the X-line, the nongyrotropic electron pressure can give rise to the reconnection electric field [15], which implies that kinetic effects of the electrons are essential to dissipation mechanisms of the magnetic field in the electron diffusion region. Outside the electron diffusion region but inside the ion diffusion region, the ion motion decouples from that of the still frozen-in electrons. This difference in motion produces currents, so-called Hall current system, in the vicinity of the X-line [31]. It is suggested that the inclusion of the Hall term realizes much faster reconnection rate than that in the case of a resistive MHD [7, and references therein]. Thus a full particle simulation, in which both electrons and ions are treated as particles, is very effective to describe the dynamics in the vicinity of the X-line.

Most full particle codes employ particle-in-cell (PIC) technique, in which the current and charge densities are accumulated on a spatial grid from the particle data and the forces on the particles are interpolated from the adjacent spatial grids. Though the full particle model using PIC technique is conceptually very simple and one does not introduce any approximation in the basic laws of mechanics and electromagnetism, the restrictions on the grid spacing are very severe and one must set it to be as small as the electron Debye length $\lambda_{De}$, which is the order of $10^3$ m in the central current sheet of the magnetotail. On the other hand, magnetic reconnection changes the current sheet structure dynamically in MHD scale of the order of $10^9$ m. Hence, for modelling of the magnetosphere including the nonlinear evolution of magnetic reconnection, the spatial dynamic range is required to be $10^6$. Furthermore, since the electromagnetic PIC code requires at least dozens of particles per cell to suppress a numerical noise, the number of particles for each species is over $10^{13}$ even in the two-dimensional system. However, a simulation with $10^{12}$ cells and $10^{13}$ particles in the two-dimensional system is unrealistic today due to the limitation of computer resources. Usual compromise is to use low mass ratio of the ion to the electron, although too low mass ratio allows a spurious wave mode to grow [8]. However, even when the mass ratio is 100 instead of 1834 that is realistic, the required numbers of cells and particles are $10^{10}$ and $10^{11}$, respectively. It is still difficult to conduct such a simulation with currently available computer resources.

In the Earth magnetotail, the number density of plasma is confirmed to be much higher in the central plasma sheet than in the lobe region according to the observations with satellites (e.g. [2,3]), so that the electron Debye length is the smallest in the central plasma sheet. Therefore, the spatial resolution required for the numerical stability is much higher in the central plasma sheet than in the lobe region. Since the diffusion region appears also in the central plasma sheet, the spatial resolution should be high there also in terms of the physical perspective. In conventional full particle simulations of magnetic reconnection (e.g. [15,16,29,27,34]), the spatial resolution has been uniform in the whole simulation area in accordance with that required in the central plasma sheet. Thus, in the lobe region which occupies most of the simulation area, the spatial resolution has been unnecessarily high. The adaptive mesh refinement (AMR) technique is beneficial to reducing the number of cells and saving computer resources by introducing finer cells hierarchically in only the regions, such as the central plasma sheet and the diffusion region, where higher spatial resolution is required (see Section 2.1).

A block structured adaptive method was extensively developed by Berger and the co-workers [4,5] to solve partial differential equations on a hierarchy of nested cells covering high-resolution regions. An introduction of the AMR technique to a particle code has been examined for the application to the N-body simulations describing cosmological matter such as the formation and structure of galaxies [12,18,20,21,32,35,39,40]. In the N-body codes, the mass density is assigned to the neighboring grids using the cloud-in-cell (CIC) algorithm, which is equivalent to the PIC algorithm for orthogonal coordinates [6]. In the early N-body codes with the AMR technique [12,18,32,39], they adopted nested cells as like as Berger and Oliger [5] to increase force resolution, so that the refined regions were give in rectangular shape. Though the rectangular refinements facilitate the data structure, they are not suitable for models including the complicated structures that are difficult to cover efficiently with rectangular regions. On the other hand, a tree structured cells have been also used in order to increase the spatial resolution locally and achieve a high dynamical range simulation [1,9,13]. Kravtsov et al. [21] developed the adaptive refinement tree (ART) code, in which the tree structured cells are connected with each other on all hierarchical levels. The ART code subdivides all cells that satisfy a refinement criterion regardless of the shape of the refined regions, so that it enables us to achieve an efficient refinement to cover a complicate structure. Although their code is adaptive only in space, recently developed codes [20,35,40] are adaptive not only in space but also in time. Plasma simulations using an electrostatic PIC code with the AMR have been recently examined in order to describe the ion beam transport in a heavy ion fusion [37].

In this paper, we present a newly developed two and a half dimensional electromagnetic full particle code with the AMR technique. In this code we deal with two species of charged particles, that is, ions (protons) and electrons. The current and charge densities are assigned onto the hierarchical cell corners using the PIC algorithm, which process is similar to the mass density assignment by the CIC algorithm used in the N-body codes. The electromagnetic field is calculated via Maxwell's equations, unlike the Poisson equation used to derive the gravitational field in the N-body codes. The particles (superparticles) are also subdivided in the refined regions in order to suppress the numerical noise that arises due to the decrease in the number of the particles per cell. In the following sections, we describe the details of our code (Section 2), and discuss the results of test simulations and application to the current sheet evolution (Section 3). Finally, we summarize and conclude this work in Section 4.

## 2. Description of the code

### 2.1. Data structure and cell generator

In order to increase a spatial resolution efficiently, the AMR technique subdivides only cells that satisfy a refinement criterion and add data sets for finer cells hierarchically onto the uniform base cells that cover the entire simulation area. If a base cell is refined, it has four child cells that have half the size of the base cell. These child cells can be also refined in turn and finer cells are generated, and so on. In our code, a refinement level $L$ in the hierarchy is defined by using the cell size of the level ($\Delta_L$) as $L \equiv \log_2(l_y/\Delta_L)$, where $l_y$ is the vertical size of the two-dimensional simulation area. We use only cells with integer level.

The data structure in our approach is completely different from that used in the conventional electromagnetic particle codes. Cells are treated as independent units organized in refinement trees rather than elements of arrays, so that one can build a very flexible cell hierarchy that can be easily modified. The hierarchical cell structure in our code is basically the same as the fully threaded tree structure developed by Khokhlov [19], and supported by a set of pointers as shown in Fig. 1. Each cell is needed to have information of the parent, child, and neighboring cells. Since a cell has four child cells, if any, we group these four cells together in order to save memory for pointers and facilitate parallel computing. We call this group an oct (after an octet in the case of a cubic cell). Each cell has a pointer to the child oct (*OctCh*). Each oct has pointers to the parent cell (*iPr*), the parent cells of the neighboring octs (*iNb*), and a representative particle belonging to the oct (*inp*). Note that each particle belongs to the finest oct in which the particle is located. Each particle has a pointer to the neighboring particle in the same oct (*pnb*), which leads to a beaded structure of particles as described in Fig. 1 [35]. In addition to these pointers, each oct has information of the level of cells included therein, the integral position at a corner of the oct, and the number of particles in the oct. The level of a cell is very useful information to efficiently sweep through a level. The position of a oct and the number of particles included in the oct are helpful to construct the beaded structure of particles. Each cell also has physical data at a corner of the cell; charge density, current density, and electromagnetic fields.

A refinement process subdivides a cell that satisfies a refinement criterion and generates four child cells. In electromagnetic particle codes, one of the refinement criteria should be defined by the local electron Debye length in terms of the numerical stability. However, additional criteria can be introduced in accordance with

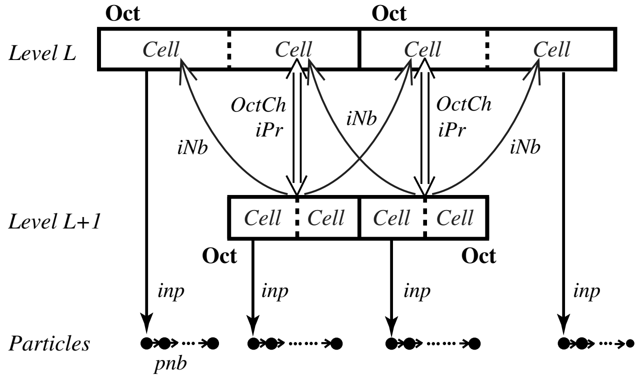

Fig. 1. Hierarchical data structure in our code. One-dimensional structure is shown for simplicity. The pointers that each cell or oct has are basically the same as those in the tree structure developed by Khokhlov [19], but we additionally give two pointers, *inp* directed to a representative superparticle from each oct, and *pnb* directed from each particle to the neighboring particle in the same oct.

physics that is incorporated in the code. An example in the case of nonlinear evolution of the plasma sheet is shown in Section 3.3. When the parent cell is refined, we check if there exist the surrounding eight cells having the same level as the parent cell. Even if one of eight cells are absent, no splitting is implemented and no child cells are created. Thus, in our tree structure, no neighboring cells have level difference greater than one, which assures that there are no sharp gradients on the boundaries of the refined regions.

The refinement procedure is similar to that in [21]. The process is started from the base level cells and recursively proceeds to reach the dynamic range level cells at each time step. First, we put flags to cells that satisfy refinement criteria. At this point, no cells are subdivided. In order to smooth the marked region, we next put flags to the eight cells surrounding each cell flagged firstly. Furthermore, we construct buffer region surrounding the refined region by giving other flags to cells that surround the originally flagged cells but have no flags for the refinement. The buffer regions are used to set boundary conditions for calculating the electromagnetic fields in the refined region as mentioned in Section 2.4. After that, we search again the whole cells with the target level. If a cell has a flag but has no child cells, the cell is subdivided and half-sized refined or buffer cells are generated. If a cell has no flag but has child cells, the child cells are removed from the hierarchical structure. If a cell has both a flag and child cells, or has neither a flag nor child cells, no operation is conducted for the cell. Thus the refinement operation is performed only to the cells that have any changes in terms of the refinement criteria at each time step, so that the refinement procedure is quickly completed.

The beaded structures of particles must be rearranged in association with the refinement procedure and the movement of particles. When a new oct is created, the particles located in the parent cell are removed from the beaded structure in the parent oct and reconnected to that in the child oct. On the other hand, when an oct is removed from the tree structure, the beaded structure is reconnected to that in the parent oct. Furthermore, particles moving away across oct boundaries are removed from the beaded structure in the oct and reconnected to those in new octs in which they are located.

### *2.2. Particle splitting and coalescence*

One of the main problems in developing the electromagnetic PIC codes using the AMR technique is decrease in the number of particles per cell in the refined regions [11]. Especially, in the vicinity of the X-line formed in association with magnetic reconnection, spatially high-resolution simulations are required because the kinetic effects of electrons are expected to be important. However, the number density of plasma in such a region is low due to inflow of the tenuous plasma in the lobe region, so that the number of particles per cell is decreased and the numerical noise is increased.

In order to solve the problem, we subdivide particles (superparticles) residing in the subdivided cells and control the number of particles per cell. We use the particle splitting-coalescence algorithm developed by Lapenta [23]. The processes should conserve the following quantities:

(1) the moments on each grid, that is, the charge and current densities,
(2) the total charge and mass of particles,
(3) the total momentum and energy of particles,
(4) the distribution function of particles.

In the splitting algorithm in the two-dimensional system, a particle with charge $q_0$, mass $m_0$, position $\boldsymbol{x}_0$, and velocity $\boldsymbol{v}_0$ can be replaced by four particles. The new particles labeled by $p$ ($p = 1, 2, 3, 4$) have charges $q_p = q_0/4$, masses $m_p = m_0/4$, and velocities $\boldsymbol{v}_p = \boldsymbol{v}_0$. The new particles are located at $x_{1,2} = x_0 \pm \Delta r$, $x_{3,4} = x_0$, $y_{1,2} = y_0$, $y_{3,4} = y_0 \pm \Delta r$ (see Fig. 2(a)). Here, $\Delta r$ should be chosen so as to suppress the local fluctuation of the spatial particle distribution, which can arise from the particle splitting, as much as possible. In this context, $\Delta r = \Delta_L/\sqrt{N_p}$ is considered to be a natural choice in the two-dimensional case, where $N_p$ is the number of particles in the cell with the size $\Delta_L$. This algorithm exactly conserve the above mentioned four quantities, if the new particles are all located in the same cell as the old particle. If one of the new particles is expected to be placed in a neighboring cell, the splitting is not implemented.

We also conduct the coalescence of particles. This algorithm is also simple. In our code, we choose two particles labeled by 1 and 2 that are located in the same cell and have same charges $q_1 = q_2 = q$ (that is, same masses $m_1 = m_2 = m$), and proximate velocities $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ which satisfy $|\boldsymbol{v}_2 - \boldsymbol{v}_1| < \alpha v_{\rm th}$, where $v_{\rm th}$ is the thermal velocity defined by $v_{\rm th} \equiv \sqrt{2T/m}$ ($T$ is temperature), and $\alpha$ is a small value which is set as $\alpha = 0.1$ in our code unless otherwise mentioned. The new particle labeled 0 has charge $q_0 = 2q$, mass $m_0 = 2m$, position $\boldsymbol{x}_0 = (\boldsymbol{x}_1 + \boldsymbol{x}_2)/2$, and velocity $\boldsymbol{v}_0 = (\boldsymbol{v}_1 + \boldsymbol{v}_2)/2$ (see Fig. 2(b)). However, the coalescence process does not exactly conserve the total energy of particles and the distribution function [23]. The difference in kinetic energy of particles between before and after the coalescence is found to be under $25\alpha^2$ percent of $mv_{\rm th}^2$, so that the error that arises during one coalescence is within 0.5% of the thermal energy of particles with mass $m$ in our code ($\alpha = 0.1$). Nonetheless, we hardly conduct the particle coalescence (usually once in a few hundred time steps), and never choose the particles in the most refined cells to avoid the numerical errors in physically important regions.

### *2.3. Particle and force weighting within the hierarchical cell*

We use the standard PIC algorithm to accumulate the charge and current densities onto each cell corner and interpolate the force on a cell corner to a particle. The charge and current that a particle located in a cell carries are assigned to the four cell corners in the following forms:

$$\rho^n_{l,m} = \sum_s \sum_j q_{sj} S_L(\boldsymbol{x}^n_{sj} - \boldsymbol{X}_{l,m}), \tag{1}$$

$$\boldsymbol{j}^{n+1/2}_{l,m} = \sum_s \sum_j q_{sj} \boldsymbol{v}^{n+1/2}_{sj} \frac{S_L(\boldsymbol{x}^n_{sj} - \boldsymbol{X}_{l,m}) + S_L(\boldsymbol{x}^{n+1}_{sj} - \boldsymbol{X}_{l,m})}{2}, \tag{2}$$

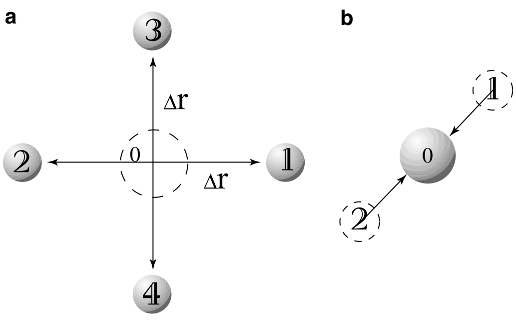

Fig. 2. Particle splitting and coalescence method in the two-dimensional system. The splitting process in (a) subdivides a particle labeled 0 into four particles numbered 1–4, while the coalescence process in (b) joins two particles numbered 1 and 2 and generates a new particle labeled 0 at the midpoint.

where $q_{sj}$, $\boldsymbol{x}_{sj}^{n}$, and $\boldsymbol{v}_{sj}^{n}$ are the charge, position, and velocity of $j$th particle of species $s$ (ion and electron) at time $n\Delta t$ ($\Delta t$ is a time interval and equivalent for all particles), respectively, and $\boldsymbol{X}_{l,m}$ represents the spatial position $(l\Delta_L, m\Delta_L)$ of a cell corner. $S_L(\boldsymbol{x})$ is the shape function and defined by using the cell size $\Delta_L$ on each refinement level $L$ in the form,

$$S_L(x,y) = \begin{cases} \frac{(\Delta_L-|x|)(\Delta_L-|y|)}{\Delta_L^2} & \text{if } |x| \leqslant \Delta_L \text{ and } |y| \leqslant \Delta_L, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Each particle is belonging to the finest oct including the particle, which means that it is only the oct that has a direct connection to the particle without passing through the other octs. However, we solve Maxwell's equations and the Poisson equation on every refinement level independently as described in Section 2.4, so that the charge and current densities must be assigned onto every cell regardless of whether or not the oct including the cell has a pointer to a particle. This assignment process using Eqs. (1) and (2) is very heavy and time consuming. So we do not adopt this manner for all the cells. Alternatively this process is carried out only for the cells belonging to the octs that have direct pointers to the particles locating in the cells, and the following operation is conducted for the other cells,

$$\begin{aligned} A_{l,m}^{L} = \frac{1}{16} \{ & 4A_{2l,2m}^{L+1} + 2(A_{2l-1,2m}^{L+1} + A_{2l,2m-1}^{L+1} + A_{2l+1,2m}^{L+1} + A_{2l,2m+1}^{L+1}) \\ & + (A_{2l-1,2m-1}^{L+1} + A_{2l-1,2m+1}^{L+1} + A_{2l+1,2m-1}^{L+1} + A_{2l+1,2m+1}^{L+1})\}, \end{aligned} \tag{4}$$

where $A_{l,m}^{L}$ is a physical quantity (charge density and current density in the current case) at the spatial position of $(l\Delta_L, m\Delta_L)$ on the level $L$ cells. The charge and current densities calculated by the use of Eq. (4) is found to be identical to those derived by Eqs. (1) and (2), if they are known on the finer cells. This procedure is very fast compared with Eqs. (1) and (2).

The electromagnetic fields at the positions of the level $L$ particles are interpolated from the cell corners on the refinement level $L$ in order to calculate the forces that works on the particles. The interpolation is also implemented by using the PIC algorithm:

$$\boldsymbol{E}^{L}(\boldsymbol{x}_{sj}) = \sum_{l}\sum_{m} \boldsymbol{E}_{l,m}^{L} S_L(\boldsymbol{x}_{sj} - \boldsymbol{X}_{l,m}), \tag{5}$$

$$\boldsymbol{B}^{L}(\boldsymbol{x}_{sj}) = \sum_{l}\sum_{m} \boldsymbol{B}_{l,m}^{L} S_L(\boldsymbol{x}_{sj} - \boldsymbol{X}_{l,m}), \tag{6}$$

where $\boldsymbol{E}_{l,m}^{L}$ and $\boldsymbol{B}_{l,m}^{L}$ are the electric and magnetic fields at the position of $(l\Delta_L, m\Delta_L)$ on the refinement level $L$, respectively.

### 2.4. Integration of the field equations

We use the complete set of Maxwell's equations to derive the electromagnetic fields. The time advancements of the electric field $\boldsymbol{E}$ and magnetic field $\boldsymbol{B}$ are described by Ampère's law and Faraday's law, respectively, that is,

$$\frac{\partial \boldsymbol{E}}{\partial t} = c^2 \nabla \times \boldsymbol{B} - \frac{1}{\varepsilon_0}\boldsymbol{j}, \tag{7}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E}, \tag{8}$$

where $c$ is the velocity of light, and $\varepsilon_0$ is the dielectric constant in vacuum. The time differences of Eqs. (7) and (8) are

$$\frac{\boldsymbol{E}^{n+1} - \boldsymbol{E}^{n}}{\Delta t} = c^2 \nabla \times \tilde{\boldsymbol{B}}^{n+1/2} - \frac{1}{\varepsilon_0}\tilde{\boldsymbol{j}}^{n+1/2}, \tag{9}$$

$$\frac{\boldsymbol{B}^{n+1} - \boldsymbol{B}^{n}}{\Delta t} = -\nabla \times \tilde{\boldsymbol{E}}^{n+1/2}, \tag{10}$$

where

$$\tilde{\boldsymbol{E}}^{n+1/2} = \frac{\boldsymbol{E}^n + \boldsymbol{E}^{n+1}}{2}, \tag{11}$$

$$\tilde{\boldsymbol{B}}^{n+1/2} = \frac{\boldsymbol{B}^n + \boldsymbol{B}^{n+1}}{2}. \tag{12}$$

Here, the superscripts denote the time levels.

The initial fields $\boldsymbol{E}^0$ and $\boldsymbol{B}^0$ are determined self-consistently with the spatial distributions of the initial charge and current densities. To advance the field data, one eliminates $\boldsymbol{B}^{n+1}$ in Eq. (9) by the use of Eqs. (10)–(12), and obtains,

$$\left(1 + \frac{c^2 \Delta t^2}{4} \nabla \times \nabla \times\right) \boldsymbol{E}^{n+1} = \left(1 - \frac{c^2 \Delta t^2}{4} \nabla \times \nabla \times\right) \boldsymbol{E}^n + c^2 \Delta t \nabla \times \boldsymbol{B}^n - \frac{\Delta t}{\varepsilon_0} \boldsymbol{j}^{n+1/2}. \tag{13}$$

The right-hand side consists of the known fields and currents, while the left-hand side has the elliptic operator. Since the boundaries of the refined regions can have an arbitrary shape in our algorithm, methods to solve the equation are restricted and we choose the conjugate gradient method which is one of the relaxation methods [26]. If $\boldsymbol{E}^{n+1}$ is obtained from Eq. (13), then $\boldsymbol{B}^{n+1}$ is calculated from Eqs. (10) and (11).

However, the assignments of the charge and current densities by using Eqs. (1) and (2), or equivalently Eq. (4), may violate the continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \boldsymbol{j} = 0. \tag{14}$$

If this equation is not satisfied, the straightforward integration of Eqs. (7) and (8) will lead to the evolution of nonphysical fields that do not satisfy $\nabla \cdot \boldsymbol{E} = \rho/\varepsilon_0$. Thus we need to make a correction to ensure that $\nabla \cdot \boldsymbol{E} = \rho/\varepsilon_0$ is maintained [22]. If $\boldsymbol{E}$ is a corrected electric field and $\overline{\boldsymbol{E}}$ is that computed from Eq. (7), we assume $\boldsymbol{E} = \overline{\boldsymbol{E}} - \nabla \delta\phi$. Then $\delta\phi$ is calculated by solving the Poisson equation,

$$-\nabla^2 \delta\phi = \left(1 + \frac{\Delta_L^2}{6} \nabla^2\right) \frac{\rho}{\varepsilon_0} - \nabla \cdot \overline{\boldsymbol{E}}. \tag{15}$$

The first term of the right-hand side is worth noticing. The second term in the parentheses arises from the truncation error of the second-order central differences of $\nabla \cdot \boldsymbol{E}$. $\nabla \cdot \overline{\boldsymbol{E}}$ in the difference equation has the truncation error within the second order in space ($\sim o(\Delta_L^2)$), while the replacement of $\nabla \cdot E = \rho/\varepsilon_0$ has no truncation so that no errors are included. This difference in truncation error can yield nonphysical electrostatic field because the error can become the source in the Poisson equation. If the difference form of $\nabla \cdot \boldsymbol{E}$ in one-dimensional system is given by $(E_{x,i+1} - E_{x,i-1})/2\Delta_L$ (the subscripts $i+1$ and $i-1$ denote the positions of cell corners), this term is approximated in the following form in the spatial fourth-order accuracy,

$$\frac{E_{x,i+1} - E_{x,i-1}}{2\Delta_L} \simeq \left.\frac{\partial E_x}{\partial x}\right|_i + \left.\frac{\Delta_L^2}{6} \frac{\partial^3 E_x}{\partial x^3}\right|_i. \tag{16}$$

The second term of the right-hand side of the equation expresses the truncation error within the spatial second order. We add this term explicitly. Since $\partial E_x/\partial x = \rho/\varepsilon_0$ is exactly satisfied, the right-hand side of this equation becomes the first term of the right-hand side of Eq. (15). In two- or three-dimensional system, Eq. (15) is also satisfied within the second-order accuracy. The Poisson equation is solved by using the conjugate gradient method again.

In our code, the electromagnetic fields are first calculated from Eqs. (13) and (10) on the base level ($L_{\mathrm{B}}$) cells, because the boundary conditions at the edge of the simulation area are assumed to be given. The solutions are interpolated onto the buffer cell corners on the next level cells as the boundary conditions as shown in Fig. 3 [40]. The interpolations are simply performed in the linear form,

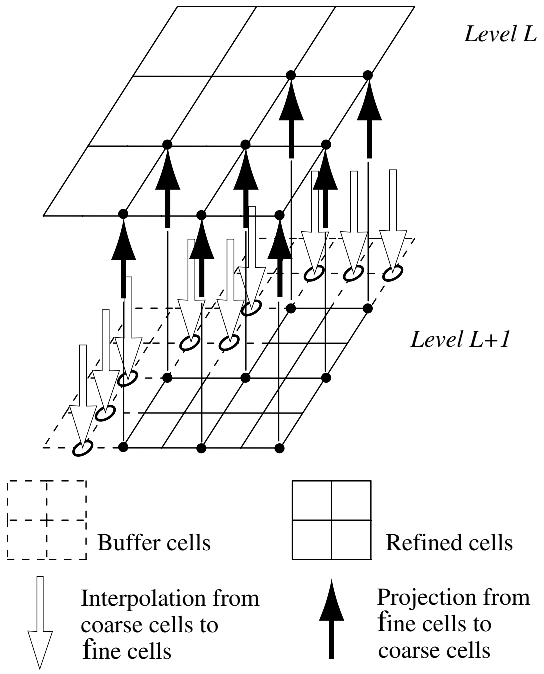

Fig. 3. Interactions of the field data between the levels $L$ and $L+1$. The solutions on the level $L$ cells are interpolated onto the buffer cell corners on the level $L+1$ cells as the boundary conditions. The electromagnetic fields solved on the level $L+1$ cells are in turn projected to the level $L$ cells.

$$
\begin{aligned}
&A^{L+1}_{2l,2m} = A^{L}_{l,m},\\
&A^{L+1}_{2l+1,2m} = \frac{1}{2}(A^{L}_{l,m} + A^{L}_{l+1,m}),\\
&A^{L+1}_{2l,2m+1} = \frac{1}{2}(A^{L}_{l,m} + A^{L}_{l,m+1}),\\
&A^{L+1}_{2l+1,2m+1} = \frac{1}{4}(A^{L}_{l,m} + A^{L}_{l+1,m} + A^{L}_{l,m+1} + A^{L}_{l+1,m+1}).
\end{aligned}
\tag{17}
$$

Then the electromagnetic fields are solved on this level ($L_B + 1$) cells by the use of Eqs. (13) and (10). These solutions are also interpolated to the buffer cell corners on the next level cells, and the procedure is recursively carried out until the solutions on the dynamic range level ($L_D$) cells are obtained. After then, the field data on the $L_D$ cells are projected onto the $L_D - 1$ cells, and the process are also recursively proceeded until the data on the $L_B$ cells are replaced (see Fig. 3). The projections are performed by using Eq. (4) which is found to eliminate the aliasing. Finally, the Poisson solver described in Eq. (15) must be executed to maintain $\nabla \cdot \boldsymbol{E} = \rho/\varepsilon_0$ on each level cells. The electromagnetic fields obtained by this procedure without recalculating under the corrected boundary conditions, however, do not give proper solutions on each refinement level, because they are not continuously differentiable at the boundaries of the refined regions [17,37]. Nevertheless, the test simulations described in Section 3 indicate that numerical errors in association with the cell refinement are almost negligible, so that we consider that the influence of the inconsistency in the field solutions between refined regions is not significant. More accurate treatment of the elliptic equations will be examined in the future work.

### *2.5. Integration of the equations of motion*

The initial particle distribution in velocity space is chosen to be Maxwellian or shifted-Maxwellian. The Maxwell's distribution is obtained using the Box–Muller method [26], in which particle velocities with a thermal velocity $v_{\text{th}}$ are described in the form,

$$\begin{aligned} v_1 &= v_{\mathrm{th}}\sqrt{-2\ln x_1}\cos 2\pi x_2, \\ v_2 &= v_{\mathrm{th}}\sqrt{-2\ln x_1}\sin 2\pi x_2, \end{aligned} \tag{18}$$

where $x_1$ and $x_2$ are the uniform random numbers. The velocities $v_1$ and $v_2$ are independent of each other.

The particle equations of motion to be integrated are

$$\frac{\mathrm{d}\boldsymbol{v}_{sj}}{\mathrm{d}t} = \frac{q_{sj}}{m_{sj}}(\boldsymbol{E}(\boldsymbol{x}_{sj}) + \boldsymbol{v}_{sj} \times \boldsymbol{B}(\boldsymbol{x}_{sj})), \tag{19}$$

$$\frac{\mathrm{d}\boldsymbol{x}_{sj}}{\mathrm{d}t} = \boldsymbol{v}_{sj}, \tag{20}$$

where $m_{sj}$ is the mass of *j*th particle of species *s*. $\boldsymbol{E}$ and $\boldsymbol{B}$ must be those on the cells whose level is the same as the particle level. The time differences of Eqs. (19) and (20) are

$$\frac{\boldsymbol{v}_{sj}^{n+1/2} - \boldsymbol{v}_{sj}^{n-1/2}}{\Delta t} = \frac{q_{sj}}{m_{sj}}\left(\boldsymbol{E}^n(\boldsymbol{x}_{sj}^n) + \frac{\boldsymbol{v}_{sj}^{n+1/2} + \boldsymbol{v}_{sj}^{n-1/2}}{2} \times \boldsymbol{B}^n(\boldsymbol{x}_{sj}^n)\right),$$

$$\frac{\boldsymbol{x}_{sj}^{n+1} - \boldsymbol{x}_{sj}^n}{\Delta t} = \boldsymbol{v}_{sj}^{n+1/2}.$$

Above equations represent a standard second-order leapfrog integration scheme. We use the Buneman–Boris method to calculate $\boldsymbol{v}_{sj}^{n+1/2}$, in which the electric and magnetic forces are separately treated [6].

## 3. Test simulations

### 3.1. Landau damping of the Langmuir waves

Our code is firstly checked against the Landau damping of the Langmuir waves that propagate across the refined regions. The initial setting of the Langmuir waves in the two-dimensional system is given as one-dimensional plane waves with an appropriate dispersion relation, which are assumed to propagate in the *x* direction (the horizontal direction in the simulation area) and to be uniform in the *y* direction. The boundaries of the system are periodic in both *x* and *y* directions. The number of the base level cells is $N_x \times N_y = 64 \times 64$, unless otherwise mentioned, which means $L_\mathrm{B} = 6$. In this and next subsections, we assume the number of refined layers is only one so that $L_\mathrm{D} = 7$ in order to avoid considerable numerical noise due to small number of particles per cell in the case that we do not conduct the particle splitting for comparison. In the run including the AMR, we initially subdivide the base level cells that satisfy the condition, $|X_j - l_x/2| < 16\Delta_{L_\mathrm{B}}$, where $X_j$ is the central *x*-coordinate of the *j*th cell, and $l_x$ represents the horizontal size of the simulation area. The refined region is assumed to be fixed to the base level region and not changed, even though a wave ridge moves from one place to another as time goes on. Thus we have an interest only in whether the waves propagate properly across the boundaries from the base region to the refined region and vice versa, and the hierarchical cells with the particle splitting-coalescence algorithm are adequately applied to the conventional PIC codes. In the runs including the AMR and particle splitting algorithms, the particles that are located in or enter the refined region can be split into four particles, and the child particles can be carefully coalesced if they move outside the region. Other initial parameters are set as $m_\mathrm{i}/m_\mathrm{e} = 10^4$, $T_\mathrm{i}/T_\mathrm{e} = 1.0$, $\omega_\mathrm{ce}/\omega_\mathrm{pe} = 1.0$, and $c/v_{\mathrm{th,e}} = 10^2$, where $m_s$ and $T_s$ represent the mass and temperature of species *s*, $\omega_\mathrm{ce}$ and $\omega_\mathrm{pe}$ are the cyclotron frequency and plasma frequency of electrons, respectively, and $v_{\mathrm{th,e}}$ is the electron thermal velocity defined as $v_{\mathrm{th,e}} \equiv \sqrt{2T_\mathrm{e}/m_\mathrm{e}}$. Adoption of an unrealistic high mass ratio indicates that we have an interest only in the electron scale processes, but not those of the ion scale. The size of the base level cells is set as $\Delta_{L_\mathrm{B}} = 1.0 \times 10^{-2}\lambda_\mathrm{e}$, where $\lambda_\mathrm{e}$ is the electron inertial length. The time step is chosen $\Delta t\omega_\mathrm{pe} = 5.0 \times 10^{-3}$ in the whole region. The initial number of particles is 64 per base level cell, so that the total number is approximately $2.6 \times 10^5$.

Fig. 4 shows the time evolutions of the field energy of the wave component $E_x$ normalized by the initial values. We have examined three cases of $\lambda = 8\Delta_{L_\mathrm{B}}$, $12\Delta_{L_\mathrm{B}}$, and $16\Delta_{L_\mathrm{B}}$, where $\lambda$ is the wavelength. In the runs of $\lambda = 12\Delta_{L_\mathrm{B}}$, the number of cells in the base level is set as $N_x \times N_y = 96 \times 64$ that is compatible with the cyclic boundary conditions in this case. Energy is averaged over the base level cells. The solid, dashed, and dotted

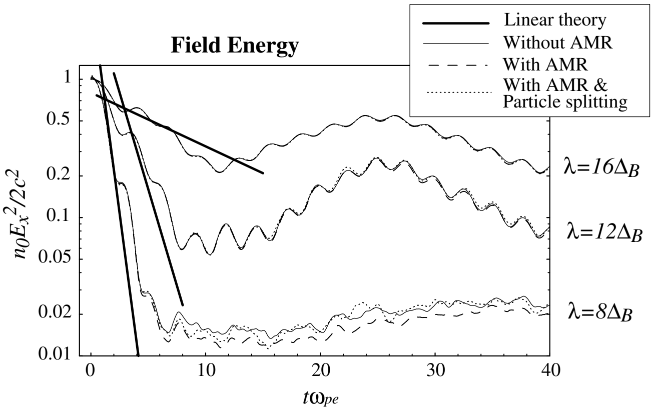

Fig. 4. Time evolution of the field energy of the Langmuir wave $E_x$, normalized by the initial values. We examined three cases of wavelength $\lambda = 8\Delta_{L_{\mathrm{B}}}$, $12\Delta_{L_{\mathrm{B}}}$, and $16\Delta_{L_{\mathrm{B}}}$. These energies are averaged over the base level cells. The thick lines denote the theoretical profiles deduced from the linear analysis of the Landau resonance. The solid, dashed, and dotted lines represent the runs without the AMR, with the AMR but not including the particle splitting, and including both the AMR and particle splitting algorithms, respectively.

lines denote the runs without the AMR, with the AMR but not including the particle splitting, and including both the AMR and particle splitting, respectively. The thick lines represent the theoretical profiles of the growth rate described as $\gamma = \pi\omega_{\mathrm{pe}}^2/2n_0kv(\partial f_0/\partial v)_{v=\omega/k}$, which is deduced from the linear analysis of the Landau resonance. Here, $n_0$ is the plasma number density averaged over the entire simulation area, $k$ is the wavenumber, $\omega$ is the angular frequency, and $f_0$ is the background part of the electron distribution function assumed to be Maxwellian. We find that each simulation result provides a good fit to the theoretical profile at the stage of the linear evolution. We also find that the results of 'With AMR' and 'With AMR and Particle splitting' runs are in good agreement with 'Without AMR' run for each wavelength. We can see a little discrepancy at the nonlinear stage of the case $\lambda = 8\Delta_{L_{\mathrm{B}}}$. This is thought to arise from the difference in noise distribution due to thermal plasma and can be affected by the short wave reflections at the boundaries of the refined region. However, the discrepancy is not essential because physical phenomena we try to understand using PIC codes should have much larger amplitudes compared with the noise level. The effects of the wave reflections at the boundaries are described in detail in the next subsection.

### 3.2. Wave reflections at the boundaries of the refined regions

One of the issues when we combine the PIC and AMR techniques is wave reflections at the boundaries surrounding the refined regions. Wave dispersion relations in the framework using finite-sized particles are different from those in the system using the actual point particles, and dependent on the cloud sizes, that is, the grid spacings in our code [6]. Thus if a wave in a coarse cell region propagates into a refined region or vice versa, the inconsistency in the wave dispersions between the hierarchical cell layers can cause wave reflections at the boundary. In order to check whether the reflections are significant, we test again the Langmuir waves that propagate across the boundaries. The initial settings of the hierarchical cells and plasma condition are the same as those in the previous subsection. In order to avoid wave damping due to the Landau resonance, we examine the relatively long wavelength, $\lambda = 32\Delta_{L_{\mathrm{B}}}$, of which the phase velocity is much larger than the electron thermal velocity, so that the resonance between the Langmuir waves and background electrons is very weak.

In Fig. 5(a), the wave spectral density of $E_x$ in the $\omega$–$k$ space is shown for the run including the AMR and particle splitting processes. Strong peak that arises at $(k,\omega) \simeq (20\lambda_{\mathrm{e}}^{-1}, \omega_{\mathrm{pe}})$ represents the normal Langmuir wave initially loaded. If the wave reflects at the boundaries of the refined region, the reflected wave should have strong peak at $(k,\omega) \simeq (-20\lambda_{\mathrm{e}}^{-1}, \omega_{\mathrm{pe}})$. However, such a peak is not shown in Fig. 5(a). The difference in wave spectral density between the 'With AMR and Particle splitting' and 'Without AMR' runs is described in Fig. 5(b). This difference is thought to represent any wave modulation caused by the presence of the refined cells. However, the scale of color contour in Fig. 5(b) is two order smaller than that in Fig. 5(a), which

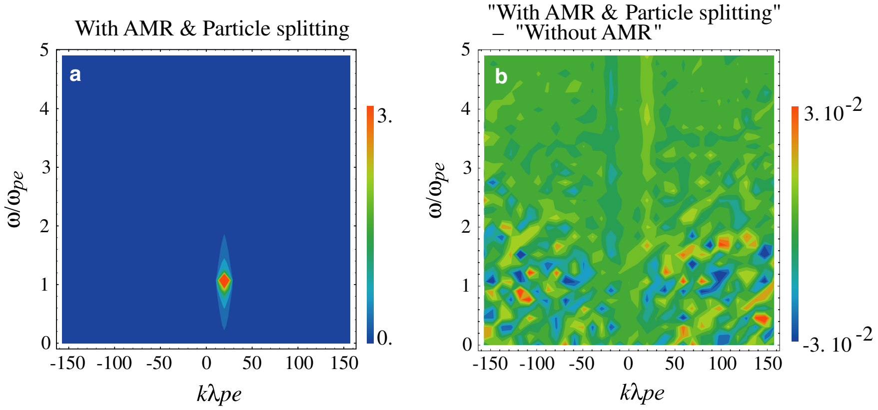

Fig. 5. (a) Wave spectral density of $E_x$ in the $\omega$–$k$ space for the 'With AMR and Particle splitting' run in the case of $\lambda = 32\Delta_{L_B}$, in which the Landau resonance between the Langmuir wave and the background electrons is negligible. (b) Difference in wave spectral density between the runs of 'With AMR and Particle splitting' and 'Without AMR'. The wave lengths are again $\lambda = 32\Delta_{L_B}$.

indicates that the modulations including the wave reflections at the boundaries are very small, if any, compared with the source wave amplitude and can be masked by the thermal fluctuations.

### 3.3. Nonlinear evolution of the plasma sheet

In this subsection, we show the electron dynamics that evolves nonlinearly from the initial plasma sheet. We use the Harris-type current sheet [14] as an initial condition, which is given as $B_x(z) = -B_0 \tanh(z/\lambda)$, where $\lambda$ is the half width of the initial plasma sheet. We set as $\lambda = 0.5\lambda_{i0}$, where $\lambda_{i0}$ is the ion inertial length defined by the initial plasma sheet density, $n_{ps}$. In addition to the equilibrium, we put a small perturbation of the form,

$$\begin{aligned} B_{xp}(x,z) &= -2\delta/\lambda \cos(2\pi x/l_x)\,\mathrm{sech}^2(z/\lambda)\,\tanh(z/\lambda), \\ B_{zp}(x,z) &= 2\pi\delta/l_x\,\sin(2\pi x/l_x)\,\mathrm{sech}^2(z/\lambda), \end{aligned} \tag{21}$$

where $\delta$ is given as $\delta = 0.03B_0\lambda_{i0}$ in the current case, so that the initial value of the reconnected magnetic field defined by $\psi \equiv \int_0^{l_x} |B_z(x, z = 0)|\,\mathrm{d}x$ is only $0.12B_0\lambda_{i0}$, where $l_x$ is the system size in the $x$ direction. Furthermore, we add the background plasma such as $n_b = n_{b0}\tanh^2(z/\lambda)$ in order to describe the lobe plasma in the Earth magnetotail. Here, $n_{b0}$ is the asymptotic lobe density and $n_{b0} = 0.044n_{ps}$ is loaded in the current runs. We do not give any perturbation in the initial current density consistent with the initial field perturbation. However, once the simulation starts, the initial current profile is quickly modified to adjust the field perturbation. As a result, a thinner current sheet is formed around the center of the simulation area, so that the tearing instability selectively develops therein. The system size is $l_x \times l_z = 15.4\lambda_{i0} \times 15.4\lambda_{i0}$, we assume the periodic boundary in the $x$ direction and the conducting wall in the $z$ direction, that is, the normal component of the electric field ($E_z$) and the tangential components of the magnetic field ($B_x, B_y$) has no gradients and the charge and current densities are vanished at the $z$ boundaries. The initial plasma condition is $m_i/m_e = 100$, $T_{i,ps}/T_{e,ps} = 8.0$, $T_{i,lobe}/T_{e,lobe} = 1.0$, $T_{e,lobe}/T_{e,\,ps} = 1.0$, and $c/v_{th,e} = 5.0$, where $T_{s,ps}$ and $T_{s,lobe}$ are the temperatures of the species $s$ at the central plasma sheet and the lobe region, respectively. Time step is $\Delta t\omega_{ci} = 8.0 \times 10^{-4}$ for all particles and refined regions in order to satisfy the Courant condition on the level $L_D$ cells, where $\omega_{ci} = eB_0/m_i$ is the ion cyclotron frequency in the asymptotic lobe region. The cell shape is assumed to be square in the current runs.

We examine three runs for comparison: 'Without AMR', 'With AMR', and 'With AMR and Particle splitting' runs. Table 1 shows information on these three runs, showing whether the AMR or particle splitting algorithms are used, the base level ($L_B$), the dynamic range level ($L_D$), the number of cells, and the number of particles. In the 'Without AMR' run, the whole simulation area is initially covered only by the fine cells with level $L = 10$ and the number of particles is large enough to suppress the numerical noise even in the fine cells. The numbers of cells and particles are not changed during the run, so that we assume results of the 'Without AMR' run are identical with those in conventional PIC simulations. In the 'With AMR' run, the AMR processes are included but the particle splitting is not implemented. Thus the number of cells is greatly reduced according to the refinement condition described later, but the number of particles is the same as the 'Without AMR' run. Finally, the 'With AMR and Particle splitting' run includes both the AMR and particle splitting processes, so that the numbers of cells and particles are reduced and changed dynamically in association with the evolution of the plasma sheet. The refinement condition is defined by three physical values. The first is the local electron Debye length, $\lambda_{\mathrm{De}} = v_{\mathrm{th,e}}/\sqrt{2}\omega_{\mathrm{pe}}$, which is required to avoid a numerical heating of local plasma, where $\omega_{\mathrm{pe}}$ is the electron plasma frequency. We use the initial value of $v_{\mathrm{th,e}}$ for calculation of $\lambda_{\mathrm{De}}$, because electrons are expected to be heated in the region where the electron dynamics is important, that is, we use the minimum value of $\lambda_{\mathrm{De}}$. The second is the out-of-plane electron flow velocity ($V_{\mathrm{e}y}$), because the inertial term in the generalized Ohm's law gets dominant in the region where the electron flow is strong, so that the dissipation of the magnetic field becomes strong and higher resolution is required. The third is the in-plane electron current density ($j_{\mathrm{e}xz} = \sqrt{j_{\mathrm{ex}}^2 + j_{\mathrm{ez}}^2}$), which is required because strong currents can excite microinstabilities so that high resolution is needed (e.g. [10]). In each time step, if $\lambda_{\mathrm{De}}$, $V_{\mathrm{e}y}$, and $j_{\mathrm{e}xz}$ calculated at the center of a cell satisfy the condition, $\Delta_L \geqslant 2.0\lambda_{\mathrm{De}}$ or $V_{\mathrm{e}y} > 2.0V_{\mathrm{A}}$ or $j_{\mathrm{e}xz} > 0.5en_{\mathrm{ps}}V_{\mathrm{A}}$, the cell is subdivided and four child cells are produced, otherwise, the child cells are removed if any. Here, $V_{\mathrm{A}} = B_0/\sqrt{\mu_0 n_{\mathrm{ps}} m_i}$ is the Alfvén velocity, and $\mu_0$ is the magnetic permeability in vacuum. The process begins from the base level cells and proceed up to the level $L_{\mathrm{D}} - 1$ cells. However, for the level $L_{\mathrm{D}} - 1$ cells we do not impose the criterion on $\lambda_{\mathrm{De}}$ in order to avoid making the level $L_{\mathrm{D}}$ cell region patchy. The particle splitting is performed only for the background particles with level except for $L_{\mathrm{D}}$ in the present runs. The particle coalescence algorithm is also implemented for the background particles once in a few hundred time steps, but we never choose particles in the level $L_{\mathrm{D}}$ cells to avoid the numerical errors in physically important regions.

Time evolutions of the reconnected magnetic flux $\psi$ are shown in Fig. 6 for the three runs: the 'Without AMR' (black solid line), 'With AMR' (black dashed line), and 'With AMR and Particle splitting' (black dotted line) runs. We found that, although the runs with the AMR require slightly more time before the onsets of fast reconnection occur, each run has the almost identical gradient so the same reconnection rate after the onset. Furthermore, the saturation levels for the runs with the AMR are clearly in agreement with that for the 'Without AMR' run. Thus we can say that the fast reconnection processes described in the three runs are physically identical, so that the adaptation of the AMR and particle splitting algorithms to the conventional PIC codes is successful. We think that the discrepancy of the onset time is caused by the difference in the noise level around the central current sheet at the initial stage in each run. Actually the initial cell size around the central current sheet in the runs with the AMR is twice of that in the 'Without AMR' run, because we do not impose the criterion on $\lambda_{\mathrm{De}}$ for the level $L_{\mathrm{D}} - 1$ cells. Thus the number of particles per cell is increased four times in the former, which indicates that the initial noise level around the central current sheet is lower in the runs with the AMR. In order to make sure whether this reasoning is correct or not, we also examined a simulation which imposes the refinement criterion on $\lambda_{\mathrm{De}}$ for the level $L_{\mathrm{D}} - 1$ cells. In this case, the criterion on $\lambda_{\mathrm{De}}$ is given as $\Delta_{\mathrm{L}} \geqslant 1.4\lambda_{\mathrm{De}}$, instead of $\Delta_{\mathrm{L}} \geqslant 2.0\lambda_{\mathrm{De}}$, only for the level $L = L_{\mathrm{D}} - 1$ cells to avoid the patchy refined regions with level $L_{\mathrm{D}}$. The results for two cases are shown in Fig. 6. One describes the run using only the AMR technique (red dashed line) and the other shows that using the AMR and particle splitting techniques (red dotted line). It is found that these two runs are in good agreement with the run without the AMR, which indicates that the initial noise level around the central plasma sheet is very crucial to determine the time when the onsets of magnetic reconnection occur. Indeed, it is very difficult to discuss the time scale in which magnetic reconnection grows from noise to a significant size, because it is strongly dependent not only on the initial noise level but also on many other conditions specific to the system (e.g. [30]). Thus we do not treat such a problem in this

Table 1
Simulation information

| Run | AMR | Particle splitting | $L_{\mathrm{B}}$ | $L_{\mathrm{D}}$ | Total number of cells | Total number of particles |
|---|---|---|---|---|---|---|
| Without AMR | No | No | 10 | 10 | $1.0 \times 10^6$ | $2.2 \times 10^7$ |
| With AMR | Yes | No | 7 | 10 | $8.4 \times 10^{4\mathrm{a}}$ | $2.2 \times 10^7$ |
| With AMR and Particle splitting | Yes | Yes | 7 | 10 | $9.1 \times 10^{4\mathrm{a}}$ | $4.8 \times 10^{6\mathrm{a}}$ |

[a] Averaged value over the run.

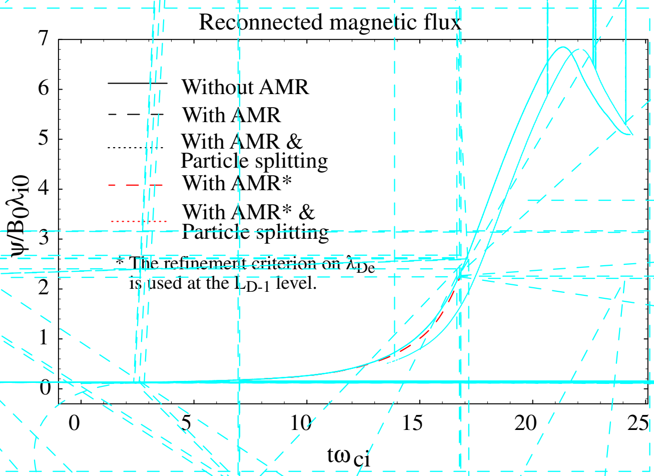

Fig. 6. Time evolutions of $\psi \equiv \int_0^{l_x} |B\,(x, z = 0)|\, \mathrm{d}x$, the where $l$ is the system $x$ direction.
The black solid line shows the result of the non-
background particles, while the black
as in the non-AMR run. The black dotted
background particles is greatly
criterion on $\lambda_{\mathrm{De}}$ is used also at the $L_{\mathrm{D}} - 1$ level are plotted in red lines.

paper. We only compare the description of fast reconnection, which allows us to start simulations with coarser cells around the central plasma sheet.

Efficiency of our code is measured by the elapsed time to complete each run, which is presented in Fig. 7. Our code including both the AMR and particle splitting algorithms is shown clearly to be very efficient compared with the conventional PIC code without the AMR. This efficiency is achieved by reducing the numbers of cells and particles greatly (see Table 1).

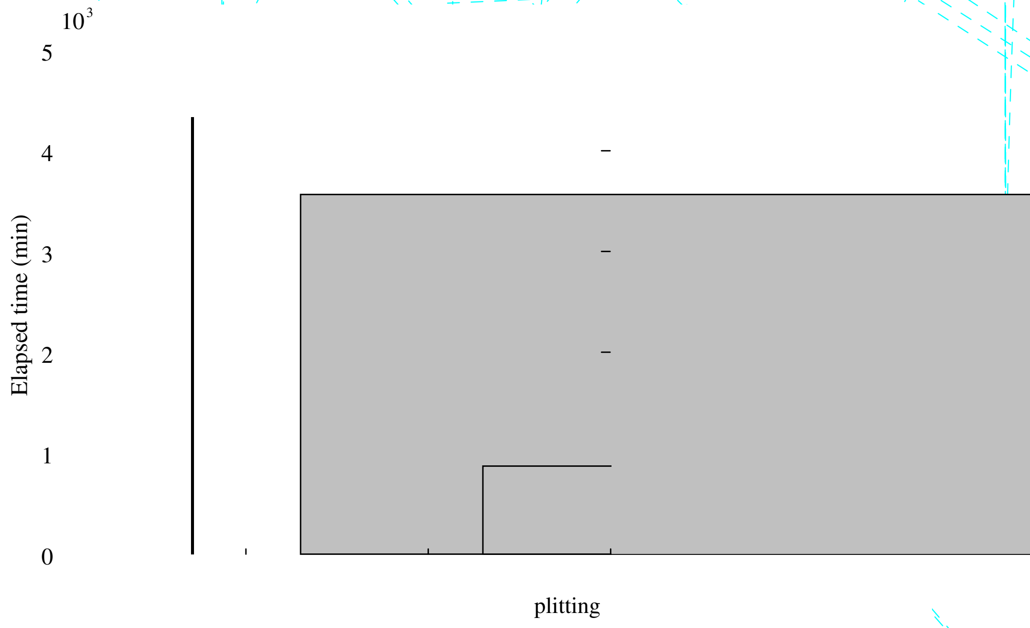

Fig. ‘Without AMR’, ‘With AMR’, and ‘With AMR and Particle splitting’ runs, Elapsed time to complete simulations for the cases of
which correspond to the solid, dashed, and dotted lines in Fig. 6, respectively. Each run has the same ‘With
splitting’ run is faster than ‘Without AMR’ and ‘With AMR’ runs by a factor of 560 and 4.1, respectively.

Fig. 8 shows the evolution of the out-of-plane current density ($J_y$) for the run including the AMR and particle splitting algorithms. A strong thin current sheet is formed in the vicinity of the X-line ($t\omega_{\mathrm{ci}} = 14.72$) and then fast reconnection occurs (see Fig. 6). After the initially loaded plasma in the current sheet is carried away from the central region, the background lobe plasma enters the diffusion region and quasi-steady electron diffusion region is produced ($t\omega_{\mathrm{ci}} = 17.92, 20.48$). In the electron diffusion region, electrons are unmagnetized from the ambient magnetic field, and conduct the meandering motions with acceleration due to the out-of-plane electric field, which generate a double-peaked current sheet therein as clearly shown in the figure. Similar properties are described also in other full particle simulations on magnetic reconnection (e.g. [16,28,29]). The hierarchical cell distributions in association with the current sheet evolution are shown in the right-hand column of Fig. 8. We find that the finest cells are only distributed around the diffusion region, separatrices extending from the X-line, and plasmoid, where the electron kinetic effects are expected to be important.

Fig. 9 displays the profiles of (a) the number density of electrons ($n_{\mathrm{e}}$), (b) the out-of-plane current density ($J_y$), (c) the inflow electron velocity ($V_{\mathrm{e}z}$) and $E \times B$ drift velocity (dashed line), and (d) the out-of-plane electric field ($E_y$), along the $x/\lambda_{\mathrm{i0}} = 7.9$ axis at the time $t\omega_{\mathrm{ci}} = 17.92$ for the run including the AMR and particle splitting algorithms. The electron diffusion region can be defined as an area where the electron bulk velocity is inconsistent with the local $E \times B$ drift velocity. Thus the area from $z/\lambda_{\mathrm{i0}} = -0.34$ to 0.34 is the electron

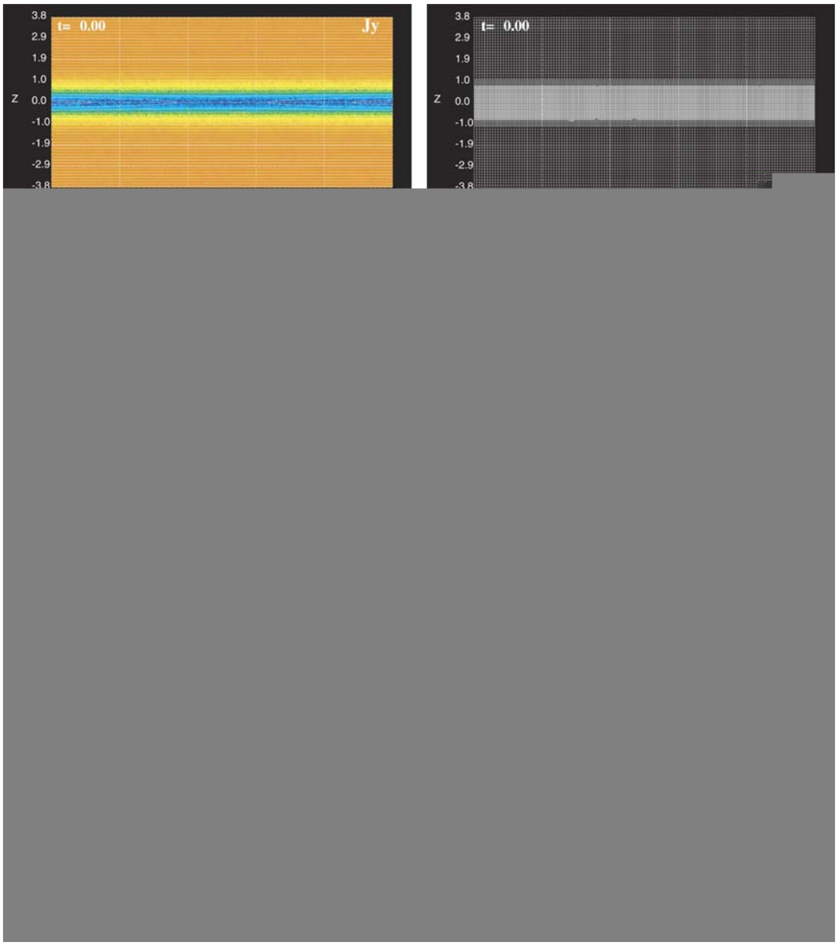

Fig. 8. Example of the evolution of the hierarchical cell distribution associated with the current sheet evolution for the run with the AMR and particle splitting processes. Magnetic field and current density (color-coded) evolutions are shown in the left-hand column and the hierarchical cells at the times corresponding to them are described in the right-hand column. Finest cells are only distributed around the diffusion region, the separatrices extending from the X-line, and the plasmoid, where the electron kinetic effects are expected to be important.

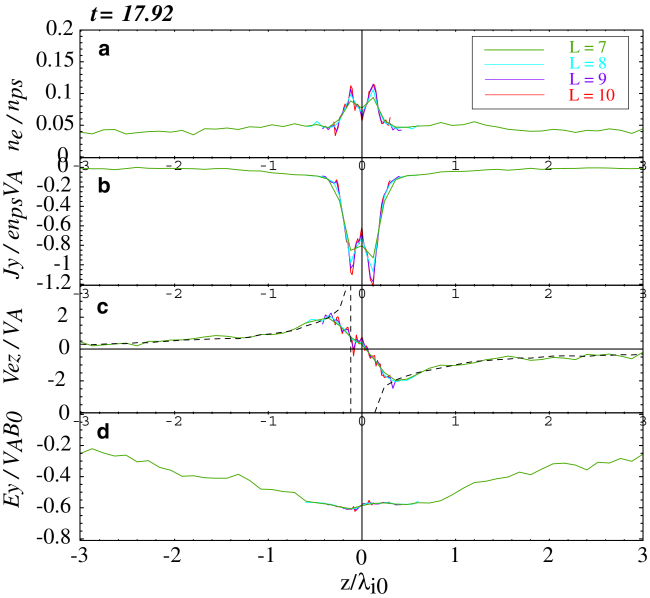

Fig. 9. Profiles of (a) the number density of electrons, (b) the out-of-plane current density, (c) the inflow electron velocity and $E \times B$ drift velocity (dashed line), and (d) the out-of-plane electric field, along the $x/\lambda_{i0} = 7.9$ axis at the time $t\omega_{ci} = 17.92$ for the same run as in Fig. 8. The green, blue, purple, and red lines represent the profiles on the level $L = 7$, 8, 9, and 10 cells, respectively.

diffusion region in this case. The double-peaked profiles of $n_e$ and $J_y$ inside the electron diffusion region are also shown in Figs. 9(a) and (b). The peak separation is estimated at $0.3\lambda_{i0}$, which is approximately same as the electron inertial length defined by the electron number density averaged within the electron diffusion region, $n_e/n_{ps} \simeq 0.071$. Shay et al. [29] have constructed a model around the electron diffusion region, in which $v_{ez}$ at the edge of the electron diffusion region and $E_y$ within the region are given by $v_{ez} \sim B_d/\mu_0 e n_e l$ and $E_y \sim -B_d^2/\mu_0 e n_e l$, respectively, where $B_d$ is the $x$ component of the magnetic field at the edge of the electron diffusion region and $l$ is the length of the region along the $x$ axis. When we estimate $l$, we assume that electrons continue to be accelerated along $x$ in the electron diffusion region, so that the electron bulk velocity $V_{ex}$ should be maximized at the edge of the region. In the current case, $B_d/B_0 \simeq 0.29$ and $l/\lambda_{i0} \simeq 2.0$ are estimated from our simulation results, so that $v_{ez}/V_A \simeq 2.1$ and $E_y/V_A B_0 \simeq -0.61$ are derived, which are approximately consistent with our results described in Figs. 9(c) and (d). Thus the electron dynamics described in the run including the AMR and particle splitting algorithms is in good agreement with the conventional model.

## 4. Summary and conclusions

We have described a new electromagnetic full particle code with the AMR and particle splitting algorithms. The AMR technique subdivides and removes cells dynamically in accordance with the refinement criteria and enhances the spatial resolution. On the other hand, the particle splitting algorithm divides particles that locate in finer cells, conserving the ratio of mass to charge, the total momentum, energy, and distribution function of particles, and the moments on grids. The particle splitting is indispensable for adapting the AMR to the electromagnetic particle codes, because the numerical noise increases on the subdivided cells due to the decrease in the number of particles per cell, especially around the X-line formed in association with magnetic reconnection [11].

We conduct several test simulations and compare three runs without the AMR, with the AMR, and with the AMR and particle splitting. Test simulations on the propagation of the Langmuir waves indicate that the AMR and particle splitting algorithms are successfully applied to the conventional PIC codes, and we have found that the wave reflections at the boundaries of the refined regions are not essential. The nonlinear evolution of the Harris-type current sheet and the electron dynamics around the X-line using our AMR code are basically the same as those in other reconnection simulations. However, the numbers of cells and particles are greatly reduced in the run including the AMR and particle splitting algorithms, so that the time to complete the simulation is considerably shortened. Thus we conclude that we have realized effectively high-resolution simulations on the evolution of the current sheet by the use of the AMR technique and particle splitting algorithm. Our code enables us to conduct large-scale particle simulations on magnetic reconnection in a wide range of the order $10^4$–$10^5$.

Our next challenge is a massively parallel computation for the AMR code using several computational nodes. Our code at the current version is parallelized only within a node of shared memory processors using OpenMP. It is worth mentioning that the accumulation of the charge and current densities onto each cell corner is parallelized by preparing temporal lists of charge and current densities for each processor. In other words, each processor has private lists of charge and current densities, and assigns particle to the lists independently of the other processors. After a sweep of the particle loop, the charge and current densities assigned to each processor are summed up, respectively, and substituted into the shared lists. This procedure is effectively parallelized at the expense of the computational memory. Difficulties in parallelizing our code arise mainly from the Poisson solver to calculate $\delta\phi$ by Eq. (15). The relaxation methods as used in the code are not adequate to a parallel computing. This is conceptually because a global distribution of the source field contributes to each local potential, and technically because the iteration loop repeated until the residual error becomes small enough must be carried out sequentially. In order to avoid solving the Poisson equation, we can use the charge conservation method for example, in which the current density is determined directly from knowledge of charge motion so as to satisfy the charge continuity equation (e.g. [38]). The improved version of our code will be discussed in a separate article.

## Acknowledgments

We are very grateful for discussions with M. Tanaka of the National Institute for Fusion Science. K.F. thanks the staffs of the Academic Center for Computing and Media Studies of Kyoto University, especially A. Hirano and K. Asaoka, for their technical advices to improve the performance of our code. Some calculations were performed by FUJITSU HPC2500 installed at the Academic Center for Computing and Media Studies of Kyoto University, the Information Technology Center of Nagoya University, and the Research Institute for Sustainable Humanosphere (RISH) of Kyoto University. K.F. acknowledges the Research Fellowships of the Japan Society for the Promotion of Science (JSPS) for Young Scientists.

This work has been partly supported by the Grant-in-Aid for JSPS Fellows.

## References

[1] J. Barnes, P. Hut, A hierarchical O($N\log N$) force-calculation algorithm, Nature 324 (1986) 446–449.
[2] W. Baumjohann, G. Paschmann, C.A. Cattell, Average plasma properties in the central plasma sheet, J. Geophys. Res. 94 (1989) 6597–6606.
[3] W. Baumjohann, R.A. Treumann, Basic Space Plasma Physics, Imperial College Press, London, 1997.
[4] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.
[5] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.
[6] C.K. Birdsall, A.B. Langdon, Plasma Physics via Computer Simulation, IOP, London, 1991.
[7] J. Birn et al., Geospace environmental modeling (GEM) magnetic reconnection challenge, J. Geophys. Res. 106 (2001) 3715–3719.
[8] W. Daughton, The unstable eigenmodes of a neutral sheet, Phys. Plasmas 6 (1999) 1329–1343.
[9] J. Dreher, R. Grauer, Racoon: a parallel mesh-adaptive framework for hyperbolic conservation laws, Parallel Computing (2005), doi:10.1016/j.parco.2005.04.011.
[10] K. Fujimoto, S. Machida, An electron heating mechanism in the outflow region from the X-type neutral line, J. Geophys. Res. 108 (2003), doi:10.1029/2002JA009810.
[11] K. Fujimoto, S. Machida, Full particle simulation of the plasma sheet using adaptive mesh refinement (AMR) technique, Adv. Space Res. (2005), doi:10.1016/j.asr.2005.03.096.
[12] S. Gelato, D.F. Chernoff, I. Wasserman, An adaptive hierarchical particle-mesh code with isolated boundary conditions, Astrophys. J. 480 (1997) 115–131.
[13] M. Griebel, G. Zumbusch, Parallel multigrid in an adaptive PDE solver based on hashing and space-filling curves, Parallel Computing 25 (1999) 827–843.
[14] E.G. Harris, On a plasma sheath separating regions of oppositely directed magnetic field, Nuovo Cimento 23 (1962) 115–121.
[15] M. Hesse, D. Winske, Electron dissipation in collisionless magnetic reconnection, J. Geophys. Res. 103 (1998) 26,479–26,486.

[16] M. Hoshino, T. Mukai, T. Terasawa, I. Shinohara, Suprathermal electron acceleration in magnetic reconnection, J. Geophys. Res. 106 (2001) 25,979–25,997.
[17] J. Huang, L. Greengard, A fast direct solver for elliptic partial differential equations on adaptively refined meshes, SIAM J. Sci. Comput. 21 (2000) 1551–1566.
[18] C. Jessop, M. Duncan, W.Y. Chau, Multigrid methods for N-body gravitational systems, Comput. Phys. 115 (1994) 339–351.
[19] A.M. Khokhlov, Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations, J. Comput. Phys. 143 (1998) 519–543.
[20] A. Knebe, A. Green, J. Binney, Multi-level adaptive particle mesh (MLAPM): a c code for cosmological simulations, Mon. Not. R. Astron. Soc. 325 (2001) 845–864.
[21] A.V. Kravtsov, A.A. Klypin, A.M. Khokhlov, Adaptive refinement tree: a new high-resolution N-body code for cosmological simulations, Astrophys. J. Suppl. Ser. 111 (1997) 73–94.
[22] A.B. Langdon, B.F. Lasinski, Electromagnetic and relativistic plasma simulation models, Methods Comput. Phys. 16 (1976) 327–366.
[23] G. Lapenta, Particle rezoning for multidimensional kinetic particle-in-cell simulations, J. Comput. Phys. 181 (2002) 317–337.
[24] E.N. Parker, The solar-flare phenomenon and the theory of reconnection and annihilation of magnetic fields, Astrophys. J. Suppl. Ser. 8 (1963) 177–212.
[25] H.E. Petschek, Magnetic field annihilation, in: W.N. Ness (Ed.), AAS-NASA Symposium on the Physics of Solar Flares, SP-50, NASA Space Publication, Washington, DC, 1964, p. 425.
[26] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in FORTRAN, second ed., Cambridge University Press, Cambridge, 1992.
[27] P.L. Pritchett, Effects of electron dynamics on collisionless reconnection in two-dimensional magnetotail equilibria, J. Geophys. Res. 99 (1994) 5935–5941.
[28] P.L. Pritchett, Geospace environment modeling magnetic reconnection challenge: simulations with a full particle electromagnetic code, J. Geophys. Res. 106 (2001) 3783–3798.
[29] M.A. Shay, J.F. Drake, B.N. Roger, R.E. Denton, Alfvénic collisionless magnetic reconnection and the Hall term, J. Geophys. Res. 106 (2001) 3759–3772.
[30] M.A. Shay, J.F. Drake, M. Swisdak, B.N. Rogers, The scaling of embedded collisionless reconnection, Phys. Plasmas 11 (2004) 2199–2213.
[31] B.U.Ö. Sonnerup, Magnetic field reconnection, in: L.J. Lanzerotti, C.F. Kennel, E.N. Parker (Eds.), Solar System Plasma Physics, vol. III, North-Holland, New York, 1979, p. 45.
[32] I. Suisalu, E. Saar, An adaptive multigrid solver for high-resolution cosmological simulations, Mon. Not. R. Astron. Soc. 274 (1995) 287–299.
[33] P.A. Sweet, The neutral point theory of solar flares, in: B. Lehnert (Ed.), Electromagnetic Phenomena in Cosmical Physics, Cambridge University Press, London, 1958, p. 123.
[34] M. Tanaka, Macro-particle simulations of collisionless magnetic reconnection, Phys. Plasmas 2 (1995) 2920–2930.
[35] R. Teyssier, Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES, Astron. Astrophys. 385 (2002) 337–364.
[36] V.M. Vasyliunas, Theoretical models of magnetic field line merging, 1, Rev. Geophys. 13 (1975) 303–336.
[37] J.L. Vay, P. Colella, A. Friedman, D.P. Grote, P. McCorquodale, D.B. Serafini, Implementations of mesh refinement schemes for particle-in-cell plasma simulations, Comput. Phys. Commun. 164 (2004) 297–305.
[38] J. Villasenor, O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, Comput. Phys. Commun. 69 (1992) 306–316.
[39] J.V. Villumsen, A new hierarchical particle-mesh code for very large scale cosmological N-body simulations, Astrophys. J. Suppl. Ser. 71 (1989) 407–431.
[40] H. Yahagi, Y. Yoshii, N-body code with adaptive mesh refinement, Astrophys. J. 558 (2001) 463–475.