Astrophysical Supercomputing Using Particles
IAU Symposium, Vol. 208, 2003
J. Makino and P. Hut, eds.

N-body Code with Adaptive Mesh Refinement

Hideki Yahagi & Yuzuru Yoshii

Institute of Astronomy, University of Tokyo
2-21-1 Osawa Mitaka Tokyo 181-0015

Abstract. We have developed a simulation code with the techniques which enhance both spatial and time resolution of the PM method for which the spatial resolution is restricted by the spacing of the structured mesh. The adaptive mesh refinement (AMR) technique subdivides the cells which satisfy the refinement criterion recursively. On the other hand, the technique of hierarchical time steps (HTS) changes the time step, from particle to particle, depending on the size of the cell in which particles reside. Our N-body code with these AMR and HTS is fully vectorized including the operations of the mass assignment.

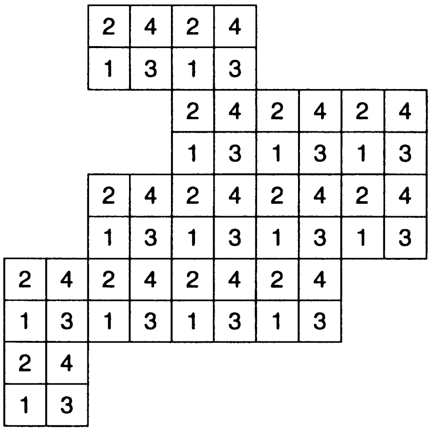
1. Vectorization of the code

The details of our N-body code with AMR and HTS are described in Yahagi & Yoshii (2001). Thus, we only describe the vectorization scheme of our code in this paper. However, our vectorization scheme is applicable to all PM-based methods, such as the P^3M method.

Operations treating cells and particles separately are vectorized trivially. However, it is hard to vectorize operations which need cell data and particle data simultaneously. In our code, particles in a cell are chained, and the cell keeps the pointer to the first particle in it. Thus, operations except for the mass assignment is vectorized simply by processing particles in the width-first order in stead of the depth-first order, as Hernquist (1990) vectorized the TREECODE. Furthermore, the mass assignment is also vectorized by dividing a loop into 2^D loops, where D is the dimension of the system. An example of the two dimensional case is shown in Figure 1. However, constructing the hashing data structure described in the above contains a recursive loop. If this particle hashing is executed only once in a simulation, it is impossible to reduce it. To the contrary, the particle hashing is called in each step. Thus, we treat particles which cross the cell borders in a step and those which do not separately. This makes the particle hashing for the latter vectorizable, and computational time of the hashing for the former negligible, because the time steps are controlled so that particles do not move over the mesh spacing in a step in general.

2. LCDM simulation

We performed an LCDM simulation with $\Omega_0 = 0.3, \lambda_0 = 0.7, h = 0.7$ and $\sigma_8 = 1.0$. The size of the simulation box is $70h^{-1}$ Mpc, and the number of



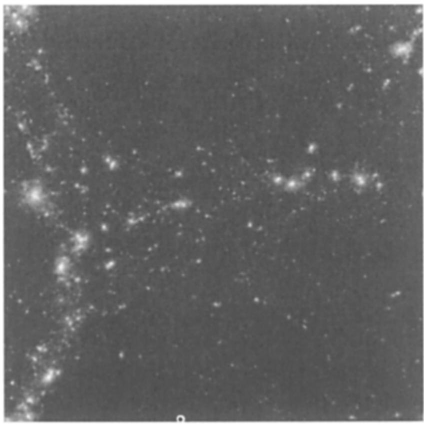


Figure 1. (Left panel) An example of the order of the vectorized mass assignment. First, particles in cells numbered one assign their mass to the corners of the cells. Then, particles in cells numbered two and so on. This avoids the recursive accesses to the density on the grids and vectorizes the mass assignment loop.

Figure 2. (Right panel) Map of projected density in the logarithmic unit at $z = 0$ for the LCDM universe. Parameters used are described in the text.

particles is 256^3. The levels of the base mesh and the finest mesh are 8 and 15, respectively. It takes about 4 days to complete this simulation from $z \sim 44.5$ to $z = 0$ on the one node of VPP5000 installed at the Astronomical Data Analysis Center of the National Astronomical Observatory, Japan.

3. Summary

Our N-body code with AMR is almost fully vectorized including the operations of the mass assignment. Cosmological N-body simulation using 256^3 particles is peformed in a short time using only a single vector processor. Now we are parallelizing the code to carry out still larger simulations in a shorter time.

References

Hernquist, L. 1990, J. Comp. Phys., 87, 137

Yahagi, H. & Yoshii, Y. 2001, ApJ, 558, 463