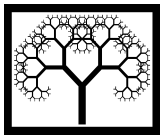
Paper 10



Proceedings of the Eighth International Conference on Engineering Computational Technology, B.H.V. Topping, (Editor), Civil-Comp Press, Stirlingshire, Scotland

The Analysis of Particle Flow using the Parallelised Smoothed Particle Hydrodynamics Method

M. Lee, M. Tak and T. Park
Department of Civil and Environmental Engineering
Hanyang University, Seoul, South Korea

Abstract

In smoothed particle hydrodynamics (SPH), an adaptive and meshfree particle method, the particle positions and velocities related to particle interactions must be updated at each time step. Thus, parallel computing is necessary to improve the numerical efficiency for the problems involving large computational domains. For the parallelization of the SPH method, the simulation domain is decomposed into subdomains with a group of particles using the new proposed parallel method. In this case, the boundary conditions are composed of the interface particles instead of the ghost zone at the other side of the subdomain. Then, the performance of the parallel computing is evaluated with the simulation of a fluid flow.

Keywords: smoothed particle hydrodynamics, particle flow, parallelization, domain decomposition, MPI, SIMD

1 Introduction

Numerical simulation using computers has been widely used to solve practical engineering problems involving fluids or solids. Meshfree methods, such as smoothed particle hydrodynamics (SPH), are very attractive numerical methods for simulating fluid flow due to their advantages over conventional grid-based methods. Unlike conventional methods in problems dealing with some complicated phenomena, meshfree methods use a set of arbitrarily distributed particles without using any grid. Thus, SPH is superior to the grid-based finite difference and finite element methods in various problems with free surface, moving interface and large deformation. SPH was originally introduced to simulate fluid dynamics in astrophysics fields by Lucy [1] and Gingold and Monaghan [2] and is extensively applied to continuum mechanics problems nowadays [3–5].

In the SPH method, the entire domain is indicated by a finite number of particles having each mass, momentum and other material properties. The particles which

don't have any connectivity are free to move and interact with each other. Then, the positions and velocities of a great number of particles in the domain vary according to time. These features of the SPH method have an effect on the computational time and expense. Therefore, parallel implementation of the SPH method is necessary to analyze the large scale problem effectively. In this case, the message passing library called Message Passing Interface (MPI) is applied between processes to deliver its own local data each other.

In the case of the parallelization of the SPH method, the decomposition of the analysis space or operation is an important step. Morris et al. [6] and Wu and Tseng [7] presented a suitable method for domain decomposition in the SPH method. There are also several decomposition approaches based on the particle, domain and operation [8–9]. In this paper, the partition of the simulation domain is proposed for the continuity of subdomain boundary under the Single Instruction Multiple Data (SIMD) structure.

2 Smoothed Particle Hydrodynamics (SPH)

2.1 Gorverning equations

The equations of motion for fluid in SPH are derived from the governing equations in Lagrangian form. The governing equations are defined using a set of partial differential equations such as Navier-Stokes equations by Navier [10] and Stokes [11]. In this case, the momentum balance equation is as follows

$$\frac{D\mathbf{v}(\mathbf{r}_a)}{Dt} = \frac{1}{\rho_a}\nabla_a \sigma(\mathbf{r}_a) \tag{1}$$

where $\mathbf{v}(\mathbf{r}_a)$ is the vector velocity which is the flow velocity and $\mathbf{r}_a$ is the position vector at the point of particle a. Also, ρ_a is the density and $\sigma(\mathbf{r}_a)$ means the total stress. In this case, the subscript a indicates the focusing on the particle a. Then, the total stress is composed of isotropic pressure p and viscous stress τ as

$$\sigma(\mathbf{r}_a) = -p(\mathbf{r}_a) + \tau(\mathbf{r}_a) \tag{2}$$

To describe Newtonian fluid behavior which means the fluid continues to flow regardless of the forces acting on it, the viscous is defined by

$$\tau(\mathbf{r}_a) = \mu(\mathbf{r}_a)\varepsilon(\mathbf{r}_a) \tag{3}$$

where μ is dynamic viscosity and ε is shear strain rate.

2.2 SPH formulations

In the SPH method, the domain is distributed into a finite number of particles and these particles which have individual material properties move according to the

governing equations. In this case, the kernel approximation can be represented with smoothing kernel function which refers to a weighting function. The kernel approximation of a function $f(\mathbf{r})$ used in the SPH method is based on the integral representation over volume V

$$f(\mathbf{r}) = \int_V f(\mathbf{r}')\,\delta(\mathbf{r}-\mathbf{r}')d\mathbf{r}' \tag{4}$$

where V is the volume of computational domain, f is a function of position vector $\mathbf{r}$ and $\delta(\mathbf{r}-\mathbf{r}')$ is the Dirac delta function given by

$$\delta(\mathbf{r}-\mathbf{r}') = \begin{cases} 1 & \mathbf{r}=\mathbf{r}' \\ 0 & \mathbf{r}\neq\mathbf{r}' \end{cases} \tag{5}$$

If the Dirac delta function $\delta(\mathbf{r}-\mathbf{r}')$ is replaced by the smoothing function $W(\mathbf{r}-\mathbf{r}',h)$, the approximation function $f(\mathbf{r})$ is derived by

$$f(\mathbf{r}) = \int_V f(\mathbf{r}')\,W(\mathbf{r}-\mathbf{r}',h)d\mathbf{r}' \tag{6}$$

where W is the so-called smoothing function or kernel function and h is the smoothing length defining the influence area of the kernel function W.

In the SPH method, the selection of kernel function is one of the most important factors. This has an effect on the accuracy, efficiency and stability of the result of the modeling physical problems. Morris [5] proposed the quantic spline for kernel function based on the B-spline function as

$$W(R,h) = a \times \begin{cases} (3-R)^5 - 6(2-R)^5 + 15(1-R)^5 & 0 \le R < 1 \\ (3-R)^5 - 6(2-R)^5 & 1 \le R < 2 \\ (3-R)^5 & 2 \le R < 3 \\ 0 & R > 3 \end{cases} \tag{7}$$

where factor a is $120/h$, $7/478\pi h^2$ and $3/359\pi h^3$ in one-, two- and three-dimensional problems respectively. Also, R is the relative distance between two particles and written as $R = r/h = |\mathbf{r}-\mathbf{r}'|/h$ where r is the distance between two particles.

For computational purpose, Equation (6) is approximated by a summation over the surrounding particles in the support domain as shown in Figure 1

$$f(\mathbf{r}) = \sum_b \frac{m_b}{\rho_b} f(\mathbf{r}_b) W(\mathbf{r}-\mathbf{r}_b,h) \tag{8}$$

where m is the mass, ρ is the density of particle $b(=1,2,\ldots,N)$ and N is the number of particles within the support domain of particle b. In Figure 1, the influence area of particle a is the support domain of particle a where κ is a

constant which defines the effective area of the kernel function. This effective area means the support domain of the point $\mathbf{r}$.

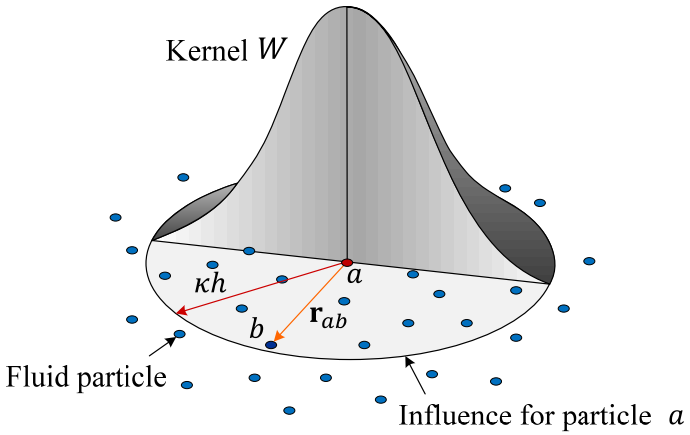


Figure 1: Particle approximation for particle a

In this case, the density is obtained by

$$\rho_a = \sum_b m_b W_{ab} \tag{9}$$

where

$$W_{ab} = W(\mathbf{r}_a - \mathbf{r}_b, h) \tag{10}$$

In the SPH method, the momentum balance equation of the Navier-Stokes equations for fluid domain, Equation (1), is expressed by applying the particle approximation concepts of Equation (8) as

$$\frac{D\mathbf{v}(\mathbf{r}_a)}{Dt} = \sum_b m_b \left[\frac{\sigma(\mathbf{r}_a)}{\rho_a^2} + \frac{\sigma(\mathbf{r}_b)}{\rho_b^2}\right] \nabla_a W_{ab} \tag{11}$$

where $\mathbf{v}(\mathbf{r}_a)$ is the vector velocity which is the flow velocity at the point of particle a and $\sigma(\mathbf{r}_a)$ is the total stress of particle a. Then, Equation (11) can be rewritten taking into account the viscous term. The approach to the dynamic viscosity is introduced by Morris et al. [4] for simulations involving low Reynolds numbers as follows

$$\begin{aligned}\frac{D\mathbf{v}(\mathbf{r}_a)}{Dt} = &-\sum_b m_b \left[\frac{p(\mathbf{r}_a)}{\rho_a^2} + \frac{p(\mathbf{r}_b)}{\rho_b^2}\right] \nabla_a W_{ab} \\ &+ \sum_b m_b \left[\frac{\mu(\mathbf{r}_a) + \mu(\mathbf{r}_b)}{\rho_a \rho_b} \mathbf{v}(\mathbf{r}_{ab})\right] \left(\frac{1}{|\mathbf{r}_{ab}|} \frac{\partial W_{ab}}{\partial \mathbf{r}_{ab}}\right)\end{aligned} \tag{12}$$

where $|\mathbf{r}_{ab}|$ is the distance between the two particles a and b, $p(\mathbf{r}_a)$ is the pressure of particle a. The particle pressure in the SPH method is derived by the artificial equation of state using the local particle density as

$$p(\mathbf{r}_a) = c^2 \rho_a \tag{13}$$

where c is the speed of sound.

3 Parallelization of the SPH method

3.1 Traditional domain decomposition

Domain decomposition is the most widely applied by distribution method in parallel computing that divides the problem domain into subdomains. In the SPH method, each subdomain includes a group of particles which can interact with each other. The particles located near the boundary of segmented domain have an effect on the particles in the opposite subdomain within the specific distance connected with smoothing length. Thus, the communication between the subdomains is significantly necessary for domain decomposition.

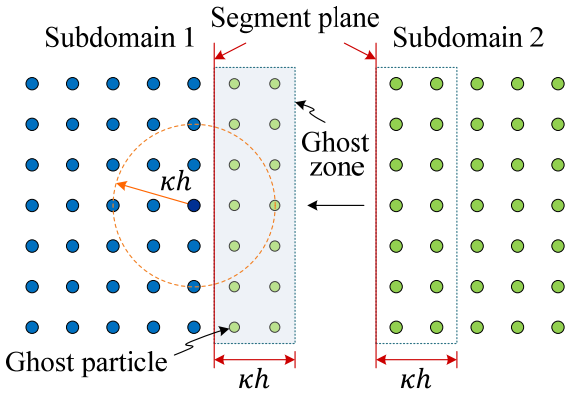


Figure 2: Domain decomposition with the ghost zone

The boundary conditions are composed by the ghost particles from the opposite side of the computational domain shown in Figure 2. The ghost zone duplicated by the other side of subdomain is developed along the separated plane to ensure accuracy of domain decomposition. Then, the ghost particles have the data about position, velocity and internal energy respectively. These data are used to calculate the forces acting on the particles near the boundary between subdomains. In this case, a great number of ghost particles are applied in the ghost zone.

3.2 Neighboring particle search

The detection of the neighbor particles is required in the SPH method that contributes to a significant proportion of the computational time. To accelerating the search of neighbors and saving in computational time, spatial hashing is one of a general method proposed by Monaghan and Gingold [12]. In the spatial hashing, entire domain is divided into hashed regions and all the particles are assigned to regions or cells which have regular intervals of κh as shown in Figure 3.

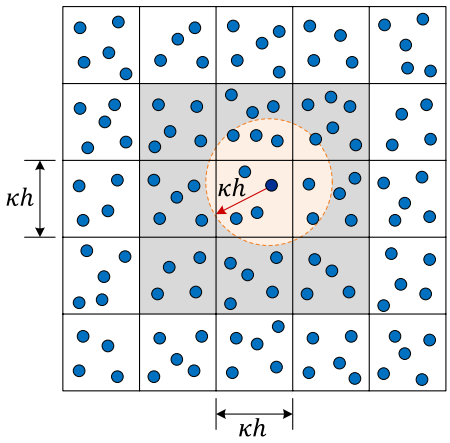


Figure 3: Spatial hashing for searching the neighboring particles

The particles only interact with particles existing in the same cell. When the neighbors are placed on other cells, the information about particles needs to be exchanged each other. In this case, it is only necessary to search for neighboring particles over the eight adjacent cells in two dimensional space instead of entire domain. The linked list by Monaghan [13] is generally used to managing spatial hashing cells.

4 Proposed parallel method for SPH

4.1 Improvement in domain decomposition

The proposed parallel method for domain decomposition is based on grouping of fluid particles and assigning to a particular processor. In this case, the data transfer is significant step for interactions between particles in each processor. Therefore, the subdomain boundary is modeled by the ghost particles called interface particles which do not have a mass. The interface particles are placed right on the segment plane of the simulation domain and play an important part in a neighboring particle search. In the conventional method for searching the neighboring particles such as the spatial hashing, the dimension of subdomain is κh for smoothing kernel function. In this case, there are can be included many unnecessary particles in the

adjoining cells. Therefore, the proposed method filters the particles near the boundary through an interface particle for reducing the computational expense shown in Figure 4.

In the proposed parallel method, the fictitious Cartesian grid made up of cells is used which is fixed in time during the entire simulation and a group of fluid particles is contained on each cell. The distance and angle between fluid particle and interface particle are calculated on each subdomain above all as

$$\theta_a = \tan^{-1}|\mathbf{r}_{\text{int}} - \mathbf{r}_a| \tag{14}$$

where θ_a is the angle from an interface particle to fluid particle a using the distance written as $R_a = |\mathbf{r}_{\text{int}} - \mathbf{r}_a|$. Then, the filtering of the neighboring particles can be implemented with the data having the interface particle. To cut the unnecessary particles on other cells, the first process of checking is progressed by

$$R_a \cos\theta_a + R_b \cos\theta_b \leq \kappa h \tag{15}$$

where κ is a constant which defines the effective area of the kernel function and h is the smoothing length. Figure 4 illustrates this step where α means the considered range of adjacent cells based on particle a.

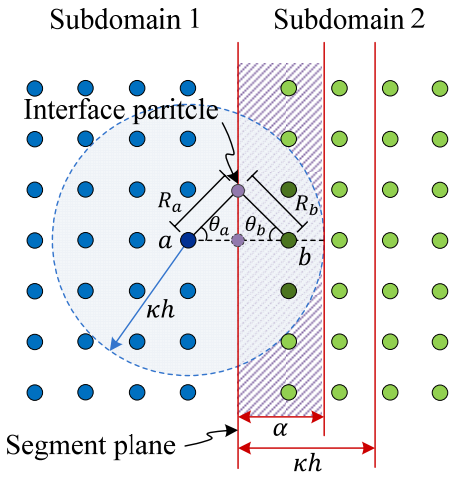


Figure 4: Filtering the neighboring particles with the interface particle

4.2 Parallel algorithm

The proposed parallel algorithm is applied for the improved domain decomposition in the SPH method. The non-blocking communication of the MPI library is used that does not block and runs the next step whether the request has completed successfully or not. Then, by the SIMD structure, the same instruction is executed in the multiple processors having different data, respectively.

In two dimensional spaces, if the simulation domain is decomposed into three processes having a group of particles and the neighboring particles of the particles in

CPU 1 and 3 are placed on CPU 2, the parallel algorithm can be expressed as Figure 5. In this case, step II. is the procedure of filtering the neighboring particles and calculating the density and pressure with only particles existing in the same process. Finally, the acceleration, velocity and displacement of all particles are calculated in step V.

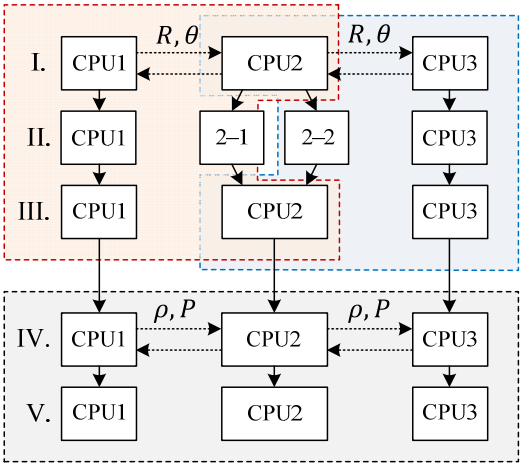


Figure 5: Procedure of parallel algorithm for the proposed method

4.3 Verification of numerical efficiency

For the verification of the proposed parallel method, the Linux cluster of 2.4GHz Neahlem CPUs is used. The system consists of 5 nodes with 8 cores and the MPICH2 [14] for MPI library is applied to developed the C++ program. In this verification, two dimensional fluid models are used and input properties are described as Table 1.

Smoothing length h	0.05 m
Time increment	0.001 $sec.$
Fluid density ρ	1,000 kg/m^3
Speed of sound c	5 m/s

Table 1: Input material properties of SPH simulation

In this model, the total number of particles is increased to four steps as 768, 1488, 2928 and 3648 at a regular rate as shown in Figure 6. The model has 48 side particles fixedly and 16, 31, 61 and 76 particles on the horizontal axis. The number of time step is 100 and the initial mesh spacing sets the minimum to $3h$. Then, one interface particle is placed on one domain boundary which is assigned to two CPUs respectively. In the numerical test for the proposed method, the domains is decomposed from two to ten elements and the parallel analysis is progressed by the two cases such as the traditional domain decomposition and the proposed method as expressed in Figure 4. In order to express numerical efficiency, CPU running time of

the SPH method is measured by the number of subdomains. Also, the running time are compared with each other.

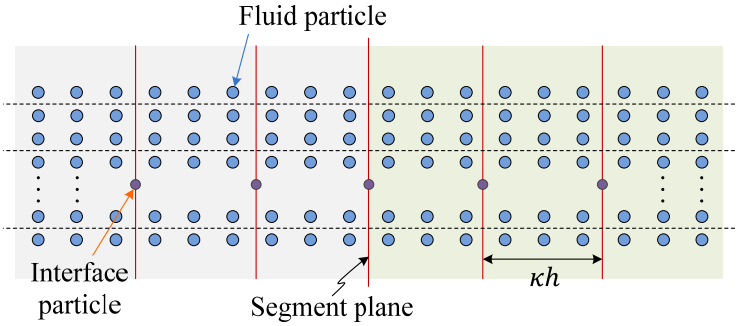
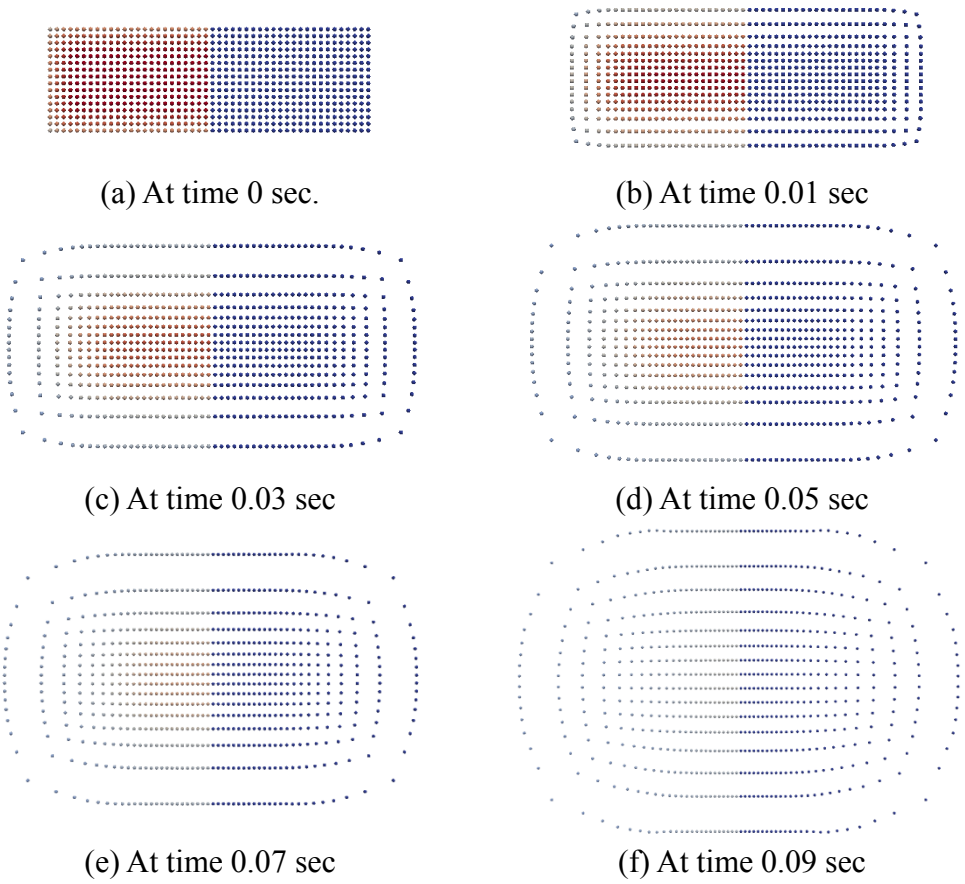


Figure 6: Decomposed simulation domain in two dimensions

First, the output files of the post processor program Paraview by the proposed C++ parallel algorithm can be confirmed as shown in Figure 7. In this case, the number of total particles is 768 and the domain is decomposed into two subdomains. The movement of particles in each subdomain has the symmetric form. This is because the particles are influenced by the neighboring particles placed on the other subdomains.



(a) At time 0 sec. (b) At time 0.01 sec

(c) At time 0.03 sec (d) At time 0.05 sec

(e) At time 0.07 sec (f) At time 0.09 sec

Figure 7: Domain decomposition using two CPUs on 768 particles

To confirm how many unnecessary particles in the adjoining cells are filtered by the proposed parallel method, the number of particle iteration is compared with the traditional domain decomposition following the total particle number. As shown in Figure 8, the number of particle iteration reduces rapidly according to the filtering of the neighboring particles in the proposed method.

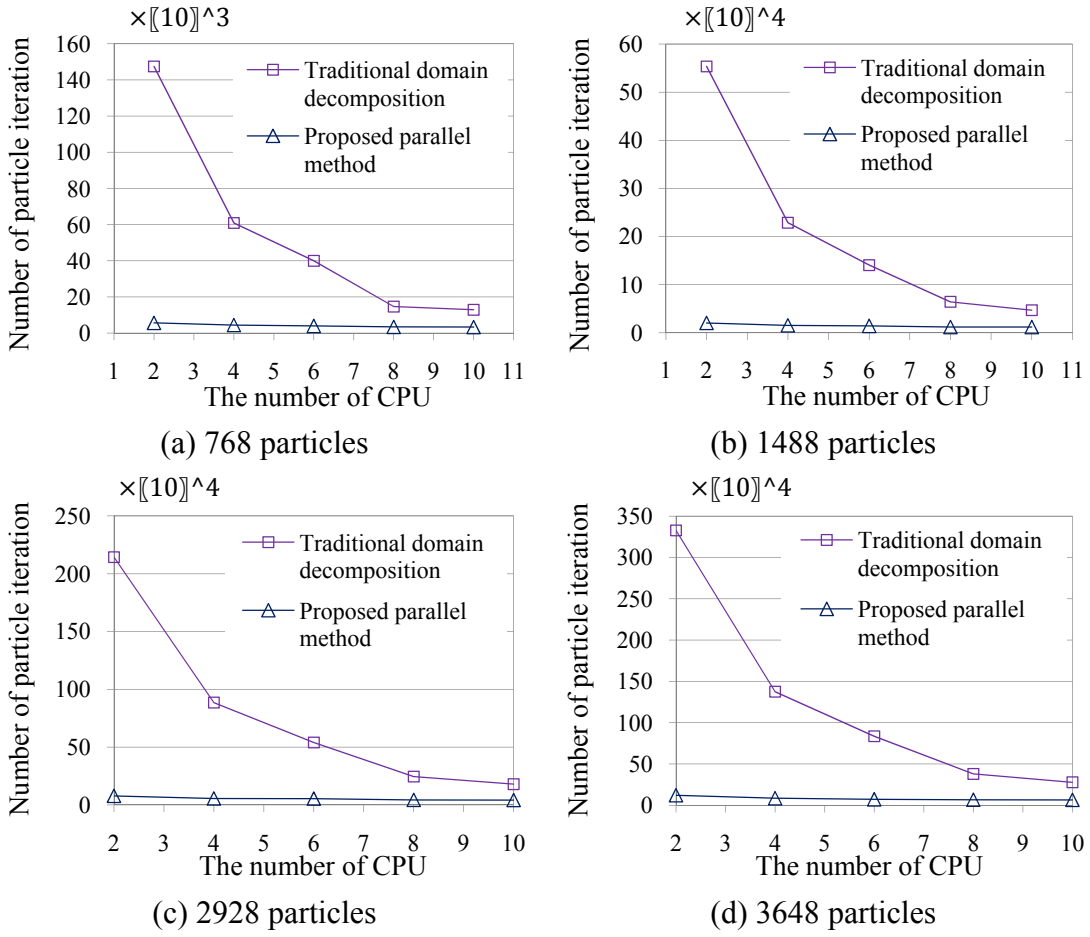


Figure 8: The number of particle iteration about total particle number

Then, two parameters such as speedup and efficiency are measured for evaluating the performance of the proposed parallel method. Speedup and efficiency are commonly used to confirm the actual scalability of the parallel implementation. Speedup according to domain decomposition for an iteration including multi time step and sub iteration is considered that is value of comparison between parallel algorithm and serial algorithm. The speedup is defined as the ratio of the running time by

$$S_p = \frac{T_1}{T_p} \tag{16}$$

where p is the number of process, T_1 and T_p are the running time at one CPU and pth CPUs respectively. In general, the numerical efficiency is the best when the speedup is identified with p that is defined linear speedup.

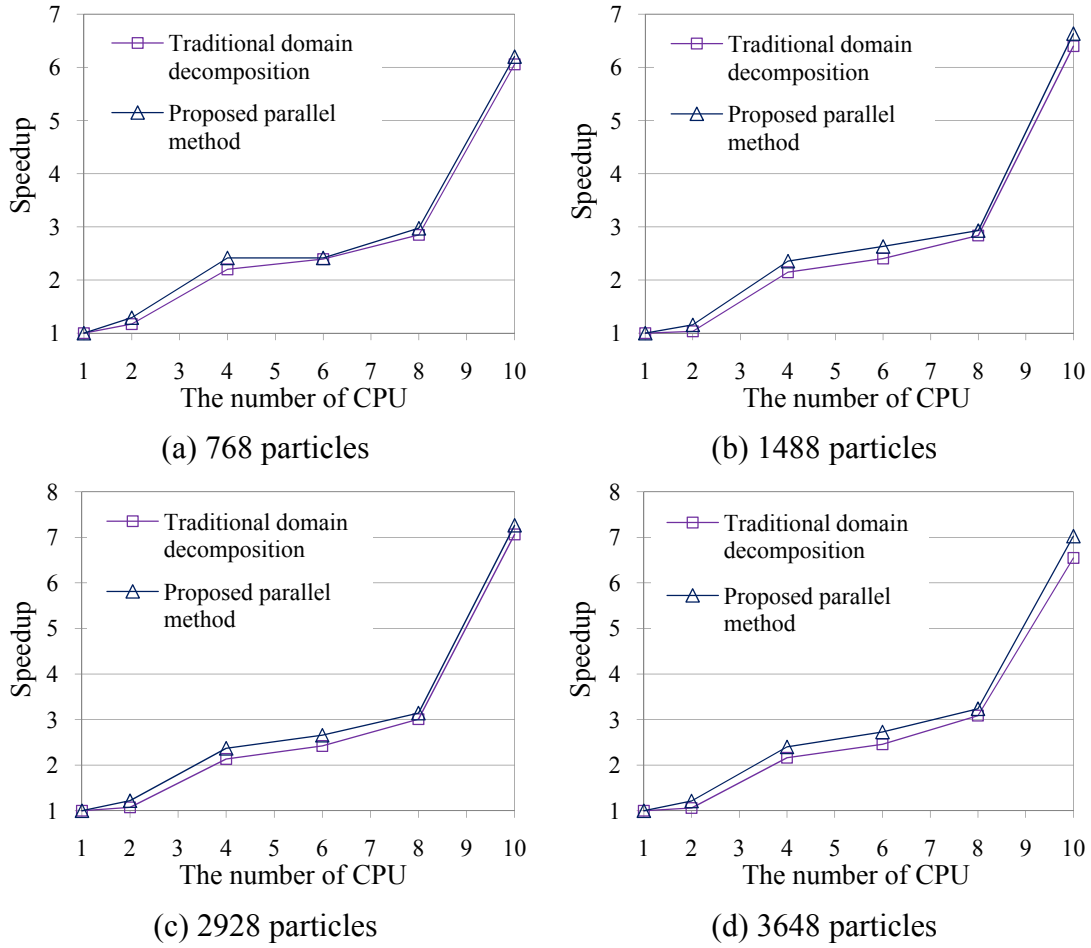


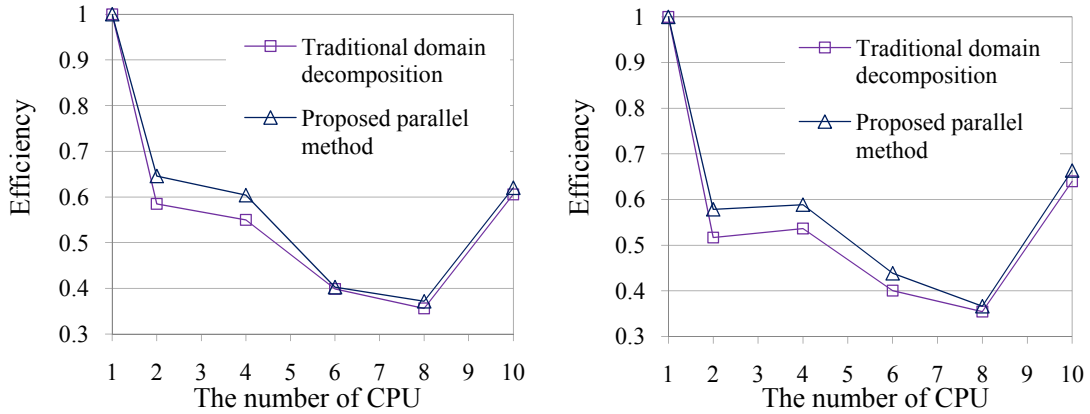
Figure 9: Speedup according to the number of CPU

According to increasing the number of decomposition and progress the filtering step, the CPU time is more saved as shown in Figure 9. The numerical efficiency is defined the waste time of the communication and synchronization as

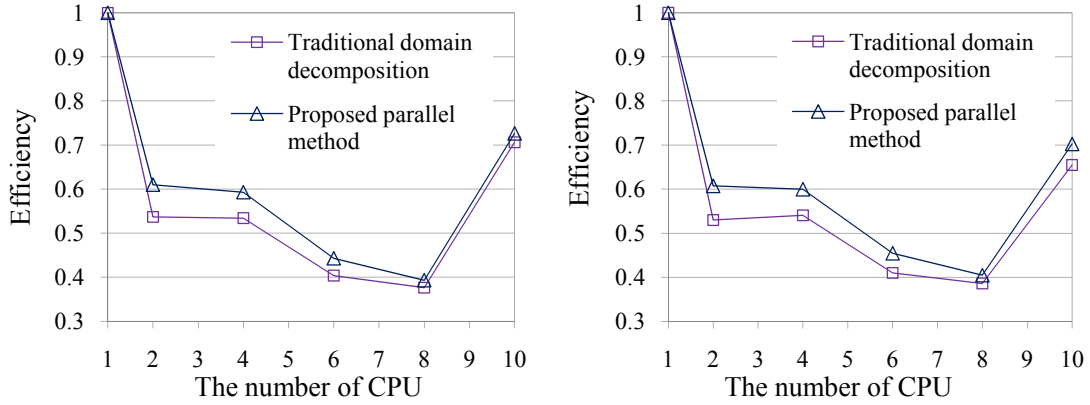
$$E_p = \frac{T_1}{pT_p} \tag{17}$$

In the case of linear speedup and algorithm using one CPU, the value of efficiency is obtained 1. Then, the value of efficiency about the total particle numbers is represented in Figure 10 that means the efficiency value is more effective as increasing the CPU numbers and filtering the neighboring particles. In the case of using the ten CPUs, the efficiency becomes less effective and this is because of sharing the information between the nodes. Thus, using the eight CPUs is the most

ideal condition for this input data. Overall, the analysis using the proposed parallel method is more effective than using the traditional domain decomposition.



(a) 768 particles (b) 1488 particles



(c) 2928 particles (d) 3648 particles

Figure 10: Efficiency according to the number of CPU

5 Conclusion

For domain decomposition in the SPH method, the proposed parallel method composes the boundary conditions to the interface particles instead of the ghost zone. In this case, the interface particle performs a role of filtering the particles near the boundary. The existing method for detection of neighbor particles can be improved using the proposed method. By the developed C++ codes, running time of subdomains and the number of particle iteration are compared according to the number of decomposed domain. Then, the proposed method is verified to the speedup and the numerical efficiency using the simple simulation. By this process, it is confirmed that the proposed parallel method is more effective than the existing method and is expected to be an efficient analysis for the problems involving large and complex computational domains. For optimization technique of this method, the

load balancing should be considered and then the proposed method will be applied in three-dimensional space.

Acknowledgement

This research was supported by WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R32–2008–000–20042–0).

References

[1] L.B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis", The Astronomical Journal, 82(12), 1013–1024, 1977.
[2] R.A. Gingold, J.J. Monaghan, "Smoothed Particle Hydrodynamics: Theory and Application to Non-spherical Stars", Monthly Notices of the Royal Astronomical Society, 181, 375–389, 1977.
[3] J.J. Monaghan, "Simulating Free Surface Flows with SPH", Journal of Computational Physics, 110, 399–406, 1994.
[4] J.P. Morris, P.J. Fox, Y. Zhu, "Modeling Low Reynolds Number Incompressible Flows using SPH", Journal of Computational Physics, 136(1), 214–226, 1997.
[5] J.P. Morris, "Analysis of Smoothed Particle Hydrodynamics with Applications", PhD Thesis, Monash University, 1996.
[6] J.P. Morris, Y. Zhu, P.J. Fox, "Parallel Simulation of Pore-scale Flow through Porous Media", Computers and Geotechnics, 25, 227–246, 1999.
[7] J.S. Wu, K.C. Tseng, "Parallel DSMC Method using Dynamic Domain Decomposition", International Journal for Numerical Methods in Engineering, 63(1), 37–76, 2005.
[8] D.W. Holmes, J.R. Williams, P. Tilke, "A Framework for Parallel Computational Physics Algorithms on Multi-core: SPH in Parallel", Advances in Engineering Software, 42(11), 999–1008, 2011.
[9] G.R. Liu, M.B. Liu, "Smoothed Particle Hydrodynamics: A Meshfree Particle Methods", World Scientific, 2003.
[10] C.L. Navier, "Memoire sur les Lois du Movement des Fluides", Mem. Acad. Sci. Inst. France, 6(2), 389–440, 1822.
[11] G.G. Stokes, "On the Theories of the Internal Friction of Fluids in Motion and of the Equilibrium and Motion of Elastic Solids", Transactions of the Cambridge Philosophical Society, 8(22), 287–319, 1845.
[12] J.J. Monaghan, R.A. Gingold, "Shock Simulation by the Particle Method of SPH", Journal of Computational Physics, 52, 374–381, 1983.
[13] J.J. Monaghan, "Particle Methods for Hydrodynamics", Computer Physics Report, 3, 71–124, 1985.
[14] W. Gropp, E. Lusk, R. Thakur, "Using MPI-2: Advanced Features of the Message-Passing Interface", MIT Press, 382, 1999.