Large-Scale Tsunami Simulation Based on Three-Dimensional Parallel SUPG-VOF Method

***Taiki Fumuro[1], Seizo Tanaka[2], and Kazuo Kashiyama[1]**

[1]Department of Civil and Environmental Engineering, Chuo University, Japan
[2]Earthquake Research Institute, University of Tokyo, Japan

*Corresponding author: fumuro@civil.chuo-u.ac.jp

Abstract

This paper presents a large-scale tsunami simulation based on the three-dimensional parallel SUPG-VOF method. The three-dimensional Navier-Stokes equation and continuity equation are employed for solving velocity and pressure. The advection equation is employed for solving interface function between air and water. The stabilized finite element method based on unstructured grid is employed for the discretization for governing equations. In order to handle the large-scale tsunami simulations, several parallel implementations are designed by using MPI, OpenMP and hybrid method with MPI/OpenMP. The presented method is applied to several tsunami wave problems to show the validity and efficiency.

Keywords: Tsunami simulation, Stabilized finite element method, SUPG-VOF method, Parallel computing, Three-dimensional Navier-Stokes equation

Introduction

The huge tsunami wave generated by the Great East Japan Earthquake (March 11, 2011) damaged the coastal area, and it is recognized that the tsunami cause the enormous damage to the human life and economic activities. There have been presented a number of numerical methods for tsunami simulation. The shallow water equation and Boussinesq equation are normally used for the governing equations. However, in order to predict the damage of structures, the three-dimensional simulation based on Navier-Stokes equation is required. There have been presented a number of numerical methods for Navier-Stokes equation with free surface. Based on the frame of reference used, these approaches can be classified into two categories: interface-capturing method using Eulerian stationary mesh and interface-tracking method using Lagrangian moving mesh. The interface-capturing method generally utilizes the VOF method (Hirt and Nichols (1981)) and level set method (Sussman et al. (1994)). On the other hand, the interface-tracking method generally utilizes the ALE method (Hughes et al. (1981)) and space-time method (Behr and Tezduyar (1993)). In the case of the three-dimensional Navier-Stokes equation, the simulation becomes quite large-scale and it is essential to use the parallel computing techniques. Parallel computing techniques are classified into three methods (Pacheco (1997), Changra et al. (2001)): process parallelism using MPI, thread parallelism using OpenMP, and hybrid parallelism combined these two methods.

This paper investigates three types of parallel computing methods for three-dimensional tsunami simulation. Each of the method is executed on a supercomputer CRAY XE6. Three-dimensional Navier-Stokes equation with the incompressibility condition is employed as the governing equation. The interface-capturing approach based on VOF method is employed because the method is robust in the applicability: for example, the method can be usefully applied to the complicated phenomena involving breaking waves. The stabilized finite element method based on SUPG/PSPG (Tezduyar (1992)) using P1/P1 element is employed for the spatial discretization. The full implicit scheme based on Crank-Nicolson method is used for the temporal discretization.

In Section 2, we describe the governing equations. The stabilized formulations are described in Section 3. Parallel implementation is described in Section 4. The present method is applied to numerical examples in Section 5. The conclusions are stated in Section 6.

Governing Equations

To model a free surface flow, we consider two immiscible fluids, A and B, with densities ρ_A and ρ_B and viscosities μ_A and μ_B. An interface function ϕ serves a marker identifying fluids A and B with the definition ϕ= {1 for fluid A, 0 for fluid B and 0.5 for two-fluid interfaces}, as shown in Fig. 1. In this context, the density and viscosity, ρ and μ, are defined as

$$\rho = \phi\rho_A + (1-\phi)\rho_B \tag{1}$$

$$\mu = \phi\mu_A + (1-\phi)\mu_B \tag{2}$$

The evolution of the interface function is governed by a time-dependent advection equation as

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} = 0 \quad \text{on} \quad \Omega \tag{3}$$

where Ω denotes the space domain. The velocity, u_i, is obtained from the solution of the unsteady Navier-Stokes equations under the incompressibility condition as

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - f_i\right) + \frac{\partial p}{\partial x_i} - \mu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = 0 \quad \text{on} \quad \Omega \tag{4}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \quad \text{on} \quad \Omega \tag{5}$$

where p is the pressure and f_i is the external force. The following conditions are imposed at the boundary.

$$u_i = g_i \quad \text{on} \quad \Gamma_g \tag{6}$$

$$\left\{-p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right\} n_j = h_i \quad \text{on} \quad \Gamma_h \tag{7}$$

where Γ_g and Γ_h denote the Dirichlet and Neumann boundaries. δ_{ij} is the Kronecker delta.

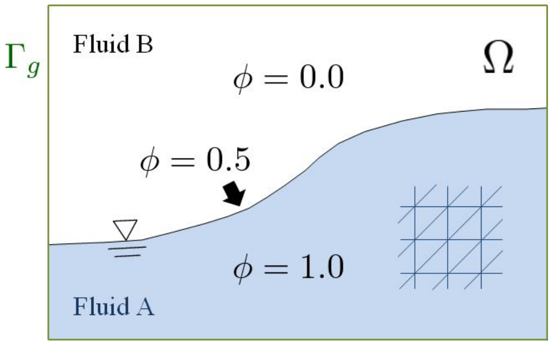


Figure. 1 Distribution of interface function

Finite Element Formulations

The stabilized finite element method based on the SUPG/PSPG method with shock capturing is employed for the governing equations. The stabilized formulation of Eqs. (4) and (5) can be written as follows.

$$\begin{aligned}
&\int_{\Omega} w_i \rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - f_i \right) d\Omega - \int_{\Omega} \frac{\partial w_i}{\partial x_i} p d\Omega + \int_{\Omega} \mu \frac{\partial w_i}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) d\Omega + \int_{\Omega} q \frac{\partial u_i}{\partial x_i} d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_e} \left(\tau_S u_k \frac{\partial w_i}{\partial x_k} + \tau_P \frac{1}{\rho} \frac{\partial q}{\partial x_k} \right) \left\{ \rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - f_i \right) + \frac{\partial p}{\partial x_i} \right\} d\Omega_e \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_C \frac{\partial w_i}{\partial x_i} \rho \frac{\partial u_j}{\partial x_j} d\Omega_e = \int_{\Gamma} w_i h_i d\Gamma
\end{aligned} \tag{8}$$

where w_i and q denote weighting functions, τ_S, τ_P and τ_C are stabilization parameters given by

$$\tau_S = \left\{ \left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\left\| u_i^e \right\|}{h_e} \right)^2 + \left(\frac{4\nu}{h_e^2} \right)^2 \right\}^{-\frac{1}{2}} \tag{9}$$

$$\tau_P = \tau_S \tag{10}$$

$$\tau_C = \frac{h_e}{2} \left\| u_i^e \right\| \xi(\mathrm{Re}_e) \tag{11}$$

where

$$\xi(\mathrm{Re}_e) = \begin{cases} (\mathrm{Re}_e/3), & \mathrm{Re}_e \le 3 \\ 1, & \mathrm{Re}_e > 3 \end{cases} \tag{12}$$

$\nu = \mu/\rho$, Δt is the time increment, h_e is the element length and Re_e is the element Reynolds number.

In Eq. (8), the first four integrals, together with the right-hand side, represent the Galerkin formulation of Eq. (4) and Eq. (5). The first series of element-level integrals in the formulation are the SUPG and PSPG stabilization terms. The second series of element-level integrals are the shock capturing terms.

The stabilized formulation of Eq. (3) can be written as follows.

$$\int_{\Omega} w \left(\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} \right) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \left(\tau_S u_j \frac{\partial w}{\partial x_j} \right) \left(\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} \right) d\Omega_e + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_\phi \frac{\partial w}{\partial x_i} \frac{\partial \phi}{\partial x_j} d\Omega_e = 0 \tag{13}$$

where τ_S and τ_ϕ are the stabilization parameters given by

$$\tau_S = \left\{ \left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\|u_i^e\|}{h_e} \right)^2 \right\}^{-\frac{1}{2}} \tag{14}$$

$$\tau_\phi = \frac{h_e}{2} \|u_i^e\| \tag{15}$$

In Eq. (13), the first integral represents the Galerkin formulation of Eq. (3). The first series of element-level integrals in the formulation are the SUPG stabilization terms. The second series of element-level integrals are the discontinuity capturing terms.

The linear tetrahedral element is employed for the discretization in space and the Crank-Nicolson method is employed for the discretization in time. The advection speed is approximated on the basis of the second-order Adams-Bashforth method. The GPBi-CG method is employed for solving the simultaneous linear equations. Also, the interface-sharpening/mass-conservation algorithm (Aliabadi and Tezduyar (2000)) is employed in order to express the interface clearly and conserve the mass for each fluid. In this approach, the interface function ϕ which is calculated from the simultaneous liner equations is replaced by $\hat{\phi}$ as follows.

$$\hat{\phi} = c^{1-a}\phi^a, \qquad 0 \le \phi \le c \tag{16}$$

$$\hat{\phi} = 1-(1-c)^{1-a}(1-\phi)^a, \qquad c \le \phi \le 1 \tag{17}$$

$$\phi \leftarrow \hat{\phi} \tag{18}$$

where a is a sharpning parameter, and $0 \le c \le 1$ is a mass conservation level.

Parallel Implementation

Parallel implementation is a technique for fast computation and realization of large-scale computing. Parallel computing techniques are classified into three methods of program parallelism: process parallelism, thread parallelism, and hybrid parallelism combined these two methods. MPI is used for the process parallelism and OpenMP is used for the thread parallelism. To minimize the amount of interprocessor communication, the automatic mesh decomposer, METIS, is employed.

Table 1 shows the specifications of the parallel supercomputer. Fig. 2 shows the architecture of the CPUs and memories in a node for CRAY XE6. In this paper, four type of parallel computing methods are investigated, as shown in Fig. 3.

Table 1. Specifications of the parallel supercomputer used to calculate

	CRAY XE6
CPU	AMD Opteron 6238 (2.9GHz)
Memory size (1 Node)	64GB
Number of cores	16cores × 2CPUs × 940nodes
O.S.	SUSE Linux Enterprise Server 11
Compiler	Intel Composer XE2011

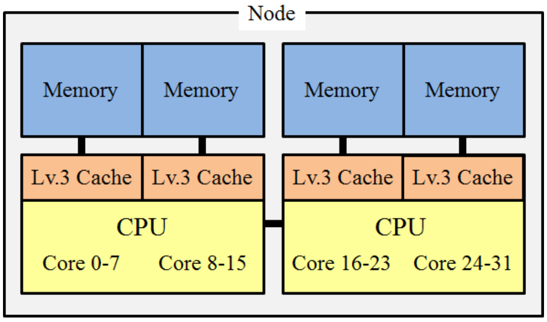


Figure 2. Architecture of the CPUs and memories in a node

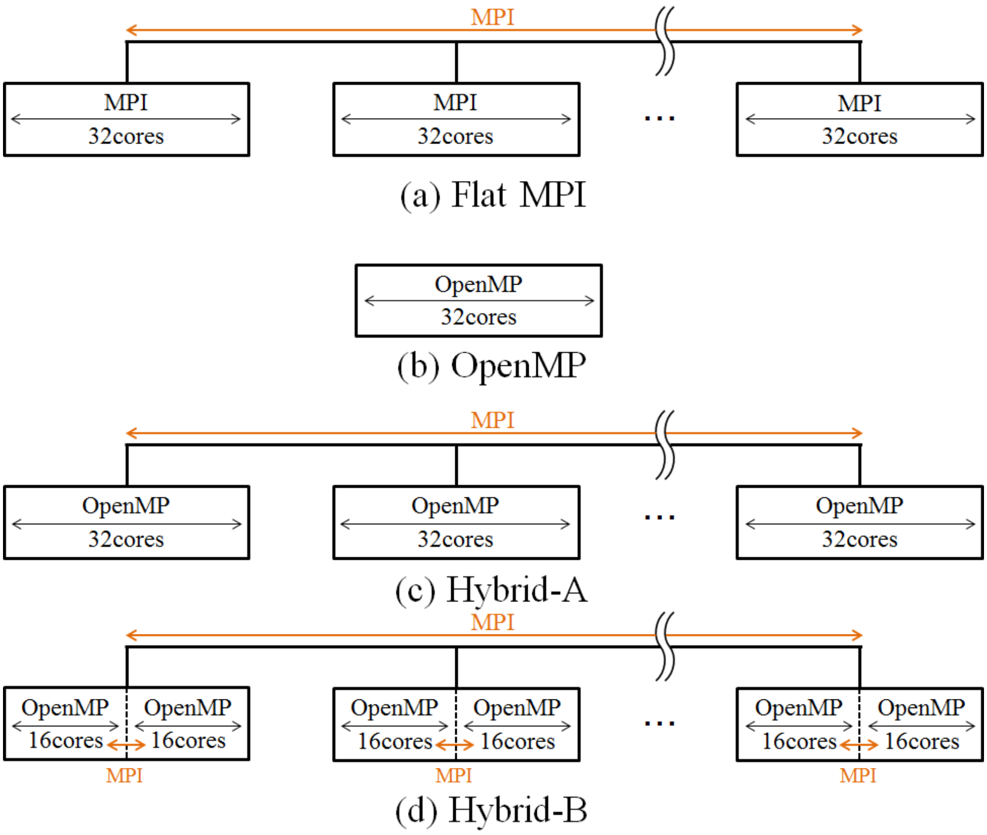


Figure 3. The four type of parallel computing methods

Numerical Examples

The four type of parallel computing methods are applied to two numerical examples: the dam-break problem and large-scale tsunami simulation.

Dam-Break Problem

The parallel computing methods are applied to the dam-break problem, as shown in Fig. 4. The computational domain was discretized by a uniform finite element mesh with 233 × 40 × 190 elements ($x \times y \times z$ direction). The total number of nodes and elements are 1,832,454 and 10,624,800, respectively. The density/viscosity of water and air are assumed as 1000.0kg/m^3 / 1.0×10^{-3}Pa$\cdot$s and 1.293kg/m^3 / 1.8×10^{-5}Pa$\cdot$s. The slip condition is employed at the wall boundary condition. The time increment Δt is assumed to be 0.0001 s.

Fig. 5 shows the time history of the waterfront line. The computed result obtained by the present method is good agreement with the experimental results (Koshizuka et al. (1995) and Martin, Moyce (1952)). Fig. 6(a) and 6(b) show the speed-up ratio and parallel efficiency versus the total number of cores/nodes. In this figures the normalization are performed using the flat MPI using 1 node. From the results of parallel performance, it can be seen that the good parallel efficiency are obtained in all approaches and the significant difference is not appeared.

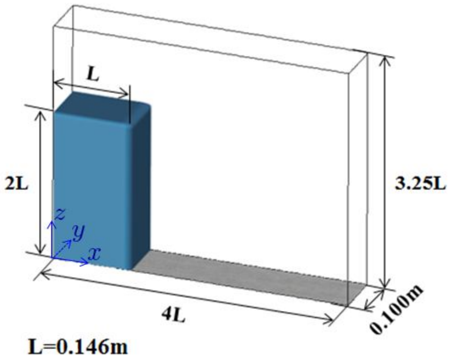


Figure 4. Computational domain and initial condition

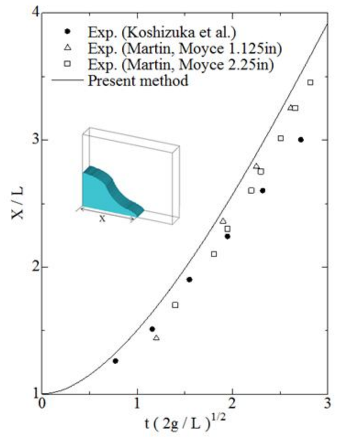
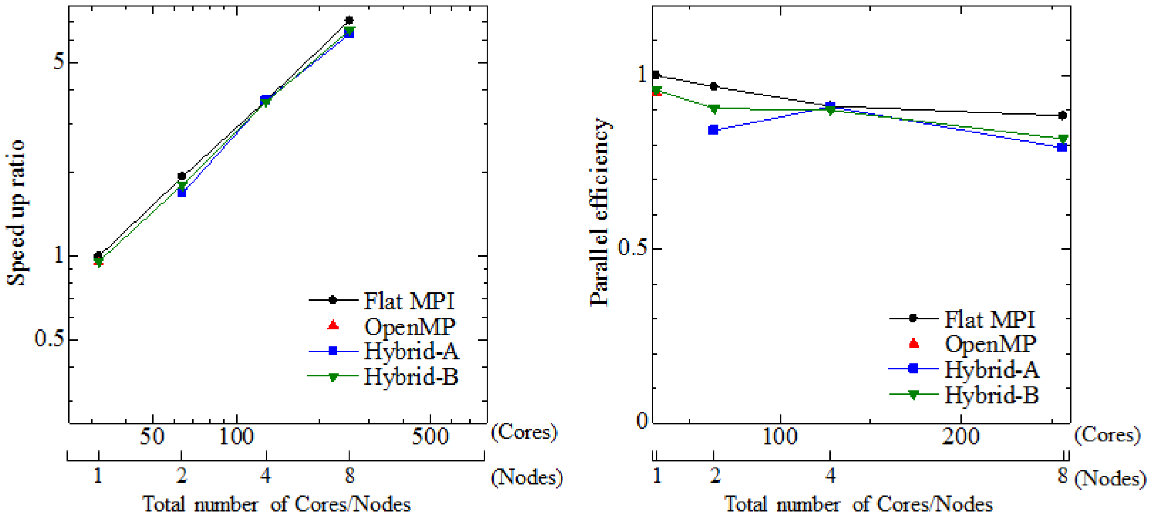


Figure 5. The time history of the waterfront line



(a)

(b)

Figure 6. Comparison of speed-up and efficiency

Large-scale Tsunami Simulation

The parallel computing methods are applied to the large-scale tsunami simulation. Fig. 7 shows the initial condition and the finite element mesh. The total number of nodes and elements are 6,738,732 and 37,560,556, respectively. The minimum mesh size is assumed to be 0.50m around the water surface and structures. The density and viscosity of water and air are same as the dam-break problem. The slip boundary condition is employed at solid boundary. The time increment Δt is set to be 0.050 s.

Fig. 8 shows the computed results at $t = 30.0$s and 45.0s. Fig. 9(a) and 9(b) show the speed-up ratio and parallel efficiency versus the total number of cores/nodes. In this figures, the normalization are performed using the flat MPI using 2 nodes. From the result of parallel performance, it can be seen that the result by flat MPI shows a better parallel efficiency compared with those using Hybrid A and Hybrid B, and the result by Hybrid-B is better than that by Hybrid-A by the effect of the conflict of memory access. Fig. 10 shows the mass conservation ratio versus the time. From this figure, it can be seen that the mass of each fluid is conserved.

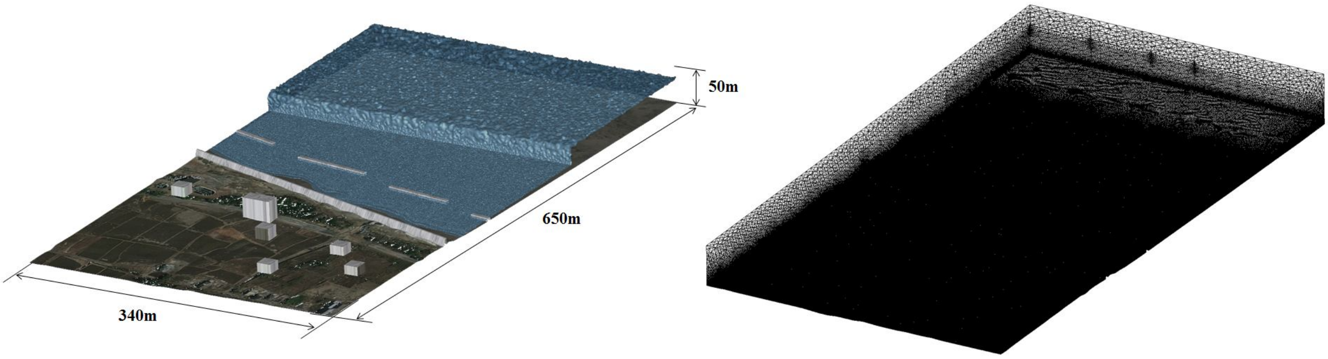


Figure 7. Initial condition and the finite element mesh

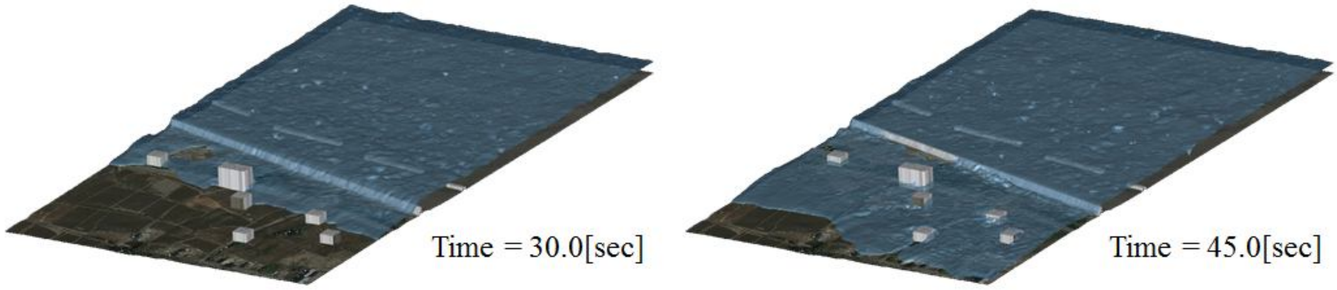


Figure 8. Computed results

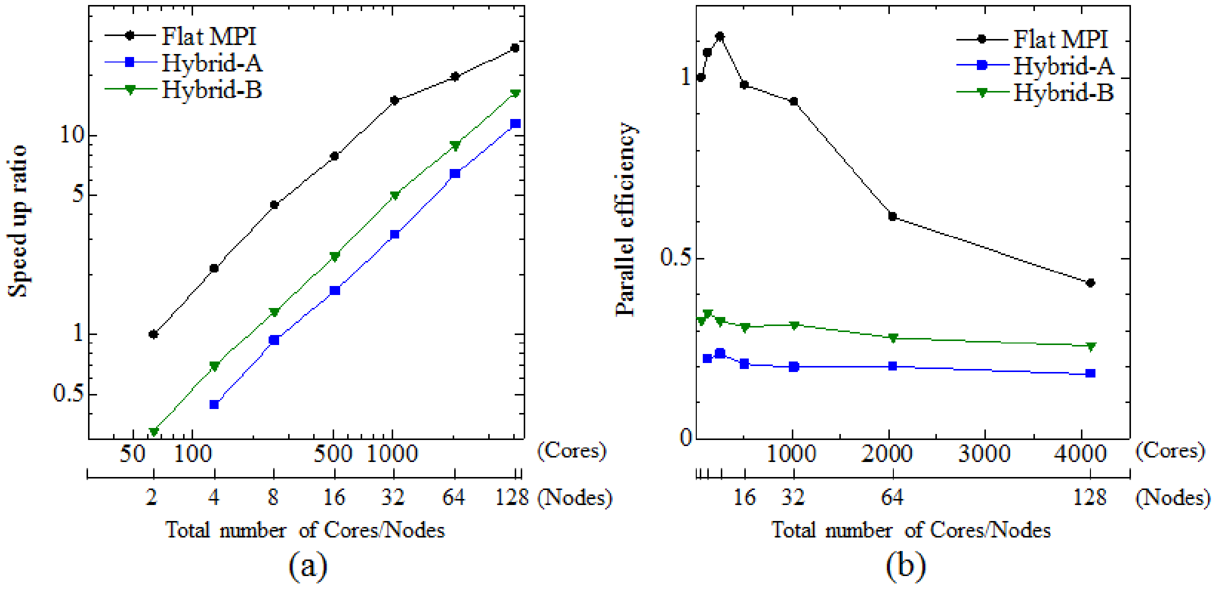


Figure 9. Comparison of speed-up and efficiency

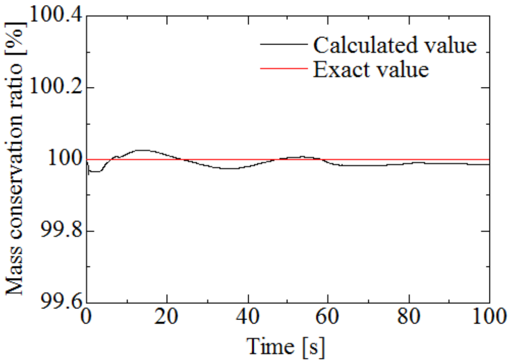


Figure 10. Mass conservation ratio

Conclusions

A parallel computational methods using MPI, OpenMP and hybrid method with MPI/OpenMP are investigated for three dimensional tsunami simulation based on SUPG/VOF method. The following conclusions can be made:

- The good parallel efficiency is obtained in all approaches and the significant difference is not appeared in the small-scale simulation example.
- The good parallel efficiency is obtained in flat MPI compared with those using Hybrid-A and Hybrid-B in the large-scale simulation example, and the result by Hybrid-B is better than that by Hybrid-A by the effect of the conflict of memory access.

From the results obtained in this paper, it can be concluded that the proposed computational method with MPI parallelization is a useful and powerful tool for the large-scale tsunami simulation.

References

Hirt, C. W. and Nichols, B. D. (1981), Volume of fluid(VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39, pp. 201-225.

Sussman, M., Smereca, P. and Osher, S. (1994), A level set approach for computing solutions for incompressible two-phase flow. *Journal of Computational Physics*, 144, pp. 146-159.

Hughes, T. J. R., Liu, W. K., Zimmermann, T. K. (1981), Lagrangian-Eulerian finite-element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29, pp. 329-349.

Behr, M. and Tezduyar, T. E. (1993), Finite element solution strategies for large-scale flow simulations, *Computer Methods in Applied Mechanics and Engineering*, 112, pp. 3-24.

Pacheco, P. S. (1997), *Parallel Programming with MPI*, Morgan Kaufmann

Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., Menon, R. (2001), *Parallel Programming in OpenMP*, Morgan Kaufmann.

Tezduyar, T. E. (1992), Stabilized finite element formulations for incompressible flow computations. *Advance in Applied Mechanics*, 28, pp. 1-44.

Aliabadi, S. and Tezduyar, T. E. (2000), Stabilized-finite-element/interface-capturing technique for parallel computation of unsteady flows with interface. *Computer Methods in Applied Mechanics and Engineering*, 190, pp. 243-261.

Koshizuka, S., Tamako, H. and Oka, Y. (1995), A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal*, 4, No. 1, pp. 29-46.

Martin, J. C. and Moyce, W. J. (1952), An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London*, Series A, vol. 244, p. 312.

Hord, R. M. (1999), *Understanding Parallel Supercomputing*, p. 356, IEEE Press.

Kumar, V., Grama, A., Gupta, A. and Karypis, G. (1994), Introduction to Parallel Computing, *Design and Analysis of Algorithms*, p. 597, The Benjamin/Cummings Publishing Company, Inc.

Karypis, G. and Kumar, V. (1998), Multilevel k-way partitioning scheme for irregular graphs, *Journal of Parallel and Distributed Computing*, 48, No. 1, pp. 96-129.