ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

LARGE-SCALE FREE-SURFACE FLOW SIMULATION USING LATTICE BOLTZMANN METHOD ON MULTI-GPU CLUSTERS

Naoyuki Onodera[1], **Kunihide Ohashi**[2]

[1] National Maritime Research Institute,
6-38-1, Shinkawa, Mitaka-shi, Tokyo, JAPAN
e-mail: onodera@nmri.go.jp

[2] National Maritime Research Institute,
6-38-1, Shinkawa, Mitaka-shi, Tokyo, JAPAN

Keywords: High-performance computing, GPU, Lattice Boltzmann method, Free-surface flow, Large-eddy simulation.

Abstract. *Turbulent free-surface flows around ship strongly affect maneuverability and safety. In order to understand the details of the turbulent flow and surface deformation, it is necessary to carry out high-order accurate and large-scale CFD simulations. We have developed a CFD code based on LBM (Lattice Boltzmann Method) with a single-phase free-surface model. Since violent flows are turbulent with high Reynolds number, a LES (Large-Eddy Simulation) model has to be introduced to solve the LBM equation. The coherent-structure Smagorinsky model is a state-of-the-art sub-grid scale model. Since this model is able to determine the model constant locally, it is suitable for a large-scale calculation containing complicated solid bodies. Our code is written in CUDA and MPI. The GPU kernel function is tuned to achieve high performance on the TSUBAME 2.5 supercomputer at Tokyo Institute of Technology. We obtained good scalability in weak scaling test. Each GPU handles a domain of 192 × 192 × 192, and 27 components are defined at a grid by the D3Q27 model. The fairly high performance of 809 MLUPS(Mega lattice update per second) is achieved by using 1000 GPUs in single precision. By executing this high-performance computation, turbulent violent flow simulation with real ship data is performed, and details of turbulent flows and free-surface deformations will be simulated with much higher accuracy than ever before.*

1 INTRODUCTION

Turbulent free-surface flows around ship strongly affect maneuverability and safety. In order to understand the details of the turbulent flow and surface deformation, it is necessary to carry out high-order accurate and large-scale CFD (Computational Fluid Dynamics) simulations. Lattice Boltzmann method (LBM) is a class of CFD methods to solve a discrete-velocity Boltzmann equation. Since LBM performs local memory access with a simple algorithm, it is suitable for a large-scale parallel computation. As an example of the large-scale calculation using LBM, the researches are nominated for Gordon bell prize of SC, and very high performance were achieved [1-3]. Since violent flows are turbulent with high Reynolds number, a sub-grid scale model has to be introduced to solve the LBM equation. Large-eddy simulation (LES) is a successful approach for unsteady turbulent flows [4]. The dynamic Smagorinsky model is a prominent subgrid-scale model based on a concept of eddy viscosity [5,6]. However, the model parameter of the DSM requires an averaging in global domain. That makes it difficult to perform a large-scale simulation. The coherent-structure model (CSM) is a remedy for these problems [7,8]. Since the model parameter can be locally determined without averaging, it is suitable for parallel computation. In this paper, we apply the CSM to a large-scale turbulent flow simulation.

For the free surface simulation, we applied interface capture method and free-surface model to the LES-LBM equation [9-11]. In this paper, we compared two types of interface capture methods based on level set and VOF. Our code is written in CUDA, and the GPU kernel function is tuned to achieve high performance on the TSUBAME 2.5 supercomputer at Tokyo Institute of Technology. By executing multi-GPU computation, a large-scale simulation with realistic ship data will be performed.

2 LATTICE BOLTZMANN METHOD

LBM solves the discrete Boltzmann equation to simulate the flow of a Newton fluid. Flow field is expressed by a limited number of particles with streaming and collision process. Physical space and time is discretized by a uniform grid. Since fluid particles move onto the neighbor lattice point after 1-time step in streaming process, discrete error of interpolation does not arise. A macroscopic diffusive and a pressure gradient is expressed by the collisional process, and we adopt BGK model. Here, in the time t and the position x, the time evolution of the discretized velocity function is denoted as follows

$$f_i(x + c_i\Delta t, t + \Delta t) = f_i(x,t) - \frac{1}{\tau}(f_i(x,t) - f_i^{eq}(x,t)). \tag{1}$$

Δt is time interval, t is the relaxation time, and f_i^{eq} *(x, t)* is the local equilibrium distribution.

The local equilibrium distribution is defined as follows

$$f_i^{eq} = w_i\rho\left(1 + \frac{3c_i \cdot u}{c^2} + \frac{9(c_i \cdot u)^2}{2c^4} - \frac{3u^2}{2c^2}\right), \tag{2}$$

ρ is density, and u is velocity. The corresponding weighting factors w_i and components of velocity vector c_i of D3Q27 model are defined as follows

$$w_i = \begin{cases} \frac{8}{27} & i = 0 \\ \frac{2}{27} & i = 1 \sim 6 \\ \frac{1}{54} & i = 7 \sim 18 \\ \frac{1}{216} & i = 19 \sim 26 \end{cases} \tag{3}$$

$$c_i = \begin{cases} (0,0,0) & i=0 \\ (\pm c,0,0),(0,\pm c,0),(0,0,\pm c) & i=1\sim 6 \\ (\pm c,\pm c,0),(\pm c,0,\pm c),(0,\pm c,\pm c) & i=7\sim 18 \\ (\pm c,\pm c,\pm c) & i=19\sim 26 \end{cases} \tag{4}$$

Relaxation time in collisional process is denoted by the following formula using dynamic viscosity,

$$\tau = \frac{1}{2} + \frac{3\nu}{c^2 \Delta t}. \tag{5}$$

3 LARGE-EDDY SIMULATION

LES resolves dynamics of large-scale structures on a grid scale (GS), and the effects of smaller-scale turbulent structures are taken into account by using a subgrid-scale (SGS) model. SGS models, based on a concept of eddy viscosity, evaluate the effects of turbulence as

$$\nu_{SGS} = C\bar{\triangle}^2 \left|\bar{S}\right|, \tag{6}$$

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right). \tag{7}$$

where C is the model coefficient, $\varDelta$ is the filter width, and S_{ij} is the velocity strain tensor.

3.1 Dynamic Smagorinsky model

In the Smagorinsky model (SM), the model coefficient C is constant in an entire computational domain, and the SM does not satisfy a correct asymptotic behavior to a wall. The dynamic Smagorinsky model (DSM) improves those defects by calculating the model parameter dynamically [5,6]. The model parameter of the DSM is determined by using two types of grid filters, and the DSM is the most notable breakthrough in LES. However, the model parameter of the DSM requires averaging in global domain for numerical stability. That makes it difficult to perform a large-scale simulation with complex geometries.

3.2 Coherent-structure Smagorinsky model

The coherent structure model (CSM) is a remedy for these problems [7,8]. Since a turbulent structure determines a model parameter locally, the CSM shows good performance in complex geometries. The model coefficient C_{CSM} is calculated by the coherent-structure function F_{CS}, composed of the second invariant of the velocity gradient tensor Q and the magnitude of a velocity gradient tensor E.

$$C_{CSM} = C' \left|F_{CS}\right|^{3/2}, \tag{8}$$

$$F_{CS} = \frac{Q}{E} \quad (-1 \le F_{CS} \le 1), \tag{9}$$

$$Q = -\frac{1}{2}\frac{\partial \bar{u}_j}{\partial x_i}\frac{\partial \bar{u}_i}{\partial x_j} \tag{10}$$

$$E = \frac{1}{2}\left(\frac{\partial \bar{u}_j}{\partial x_i}\right)^2. \tag{11}$$

The coefficient $C^{'}$ is a fixed model parameter, and it is optimized for this simulation. Since the model parameter can be locally determined without averaging, it is suitable for efficient parallel computation.

Introduction of CSM to LBM enables a large-scale simulation with complicated boundary at a high Reynolds number [11]. The total relaxation time with the eddy viscosity is written as

$$\tau_* = \frac{1}{2} + \frac{3\nu_*}{c^2 \Delta t}, \tag{12}$$

$$\nu_* = \nu_0 + \nu_{SGS}. \tag{13}$$

4 FREE-SURFACE MODEL FOR LATTICE BOLTZMANN METHOD

The free surface model [9-11] is implemented with a single-phase model. In a single-phase model, three kinds of cells are defined: liquid cells, gas cells, and interface cells. Assuming the effects of gas are small and negligible, gas cells are not calculated. Liquid cells and interface cells belong to fluid region, and their velocity distribution functions are calculated. The free surface boundary conditions are implemented on the interface cells. For example, the velocity component c_i coming from the empty cell x+$c_i\Delta t$ is reconstructed as

$$f_{-i}(x, t+\Delta t) = f_i^{eq}(\rho_A, u_B) + f_{-i}^{eq}(\rho_f, u_B) - f_i(x,t). \tag{14}$$

ρ_A=*1* is atmospheric pressure and u_B is velocity of the interface cell. Since velocities in the gas phase are unknown, they are extrapolated from neighbor interface cells.

5 PERFORMANCE ON THE TSUBAME SUPERCOMPUTER

The TSUBAME 2.5 supercomputer in Tokyo Institute of Technology is equipped with more than 4,000 GPUs (NVIDIA K20X), and the theoretical peak performance is 17 PFlops in single precision. The GPU code runs on the TSUBAME by using CUDA programing and MPI library. The parallel GPU-computation is implemented by flat MPI. Since direct data transfers between GPU-to-GPU is not implemented in CUDA 7.0, MPI communications are conducted by using 3-step communications as GPU-CPU (*cudaMemcpyDeviceToHost*), CPU-CPU (*MPI communication*), and CPU-GPU (*cudaMemcpyHostToDevice*) (Fig.1).

We show weak scaling results in Fig. 2. Each GPU handles a domain of 192 × 192 × 192 meshes, and 27 components are defined at a grid by the D3Q27 model. The horizontal axis indicates number of GPUs, and the vertical axis indicates the performance (MLUPS: Mega-lattice update per second) in single precision.

As a result, we achieved good scalability from 8 GPUs to 1000 GPUs; the performance on 8 GPUs is 70 MLUPS (0.809 sec/step), the performance on 125 GPUs is 109 MLUPS (0.815 sec/step), the performance on 512 GPUs is 428 MLUPS (0.846 sec/step), and the performance on 1000 GPUs is 809 MLUPS (0.875 sec/step). The algorithm/code, performed on 1000 GPUs, is 92% more efficient than when performed on 8 GPUs. Since the model parameter of our code can be locally determined, we obtained fairly good performance in the weak scalabilities.

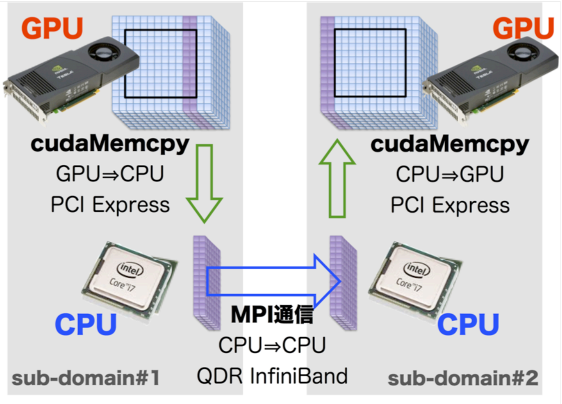

Fig. 1: MPI data transfer between GPU-GPU.

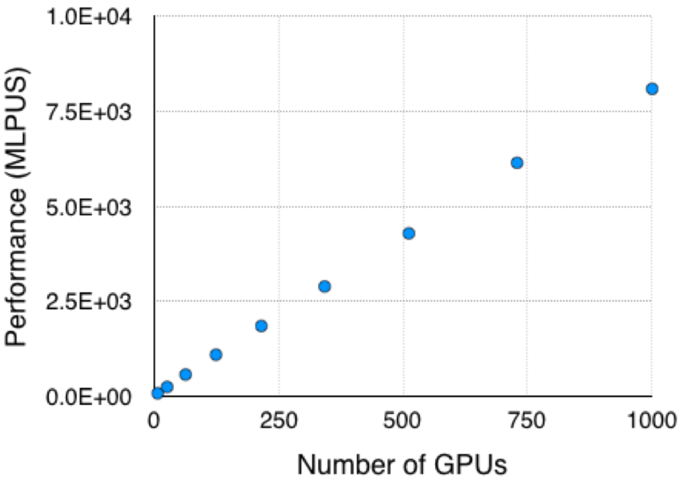

Fig. 2: Weak scalability on TSUBAME 2.5. Each GPU handles a domain of 192 × 192 × 192, and 27 components are defined at a grid by the D3Q27 model.

6 NUMERICAL EXAMPLES

6.1 Two-dimensional dam-break problem

Two-dimensional dam-break problems are considered. Two types of interface capture methods are implemented and the results are compared to experimental results of Martin and Moyce [13]. One interface capture scheme is a level set method and the other scheme is a VOF method. The level set equation is discretized by a 3rd-order WENO-scheme. The VOF equation is discretized by a THINC-WLIC scheme. Figure 3 shows the initial condition with non-dimensional parameters of *a=0.05715 m(2.25 inch)* and *n=1*. Physical parameters used here are liquid density $\rho=1000kg/m^3$, viscosity $\nu=1\times10^{-5}$, and gravity acceleration *g=9.8 m/s*. The effects of surface tension and wall wettability are not considered.

Figure 4 shows the evolutionary process of the breaking dam. Figures 5 and 6 show the time evolution of top of water column and water front. Results of three different grid resolutions (N_x*=320,512,1024)* are compared to those of experiment. Dimensionless parameters used here are top of water column $H=\eta/(an^2)$, water front *Z=z/a*, and time $\tau=t\ (g/a)^{1/2}$ and $T=nt\ (g/a)^{1/2}$.

The numerical results agree with the experimental results, except at the surge front position with the coarse grid (N_x*=320*) of the level set method. This discrepancy results from the loss of mass at the waterfront. The free surface simulated by the level set method is much smoother than that of VOF.

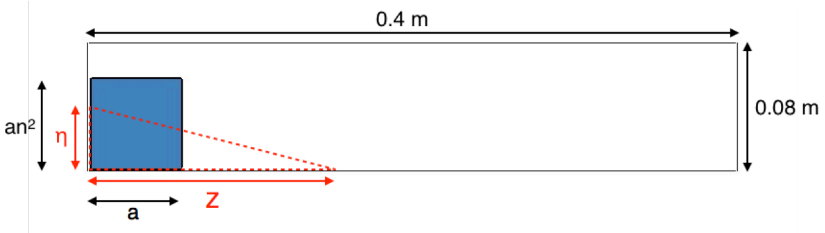

Fig. 3 : Computational condition of breaking dam

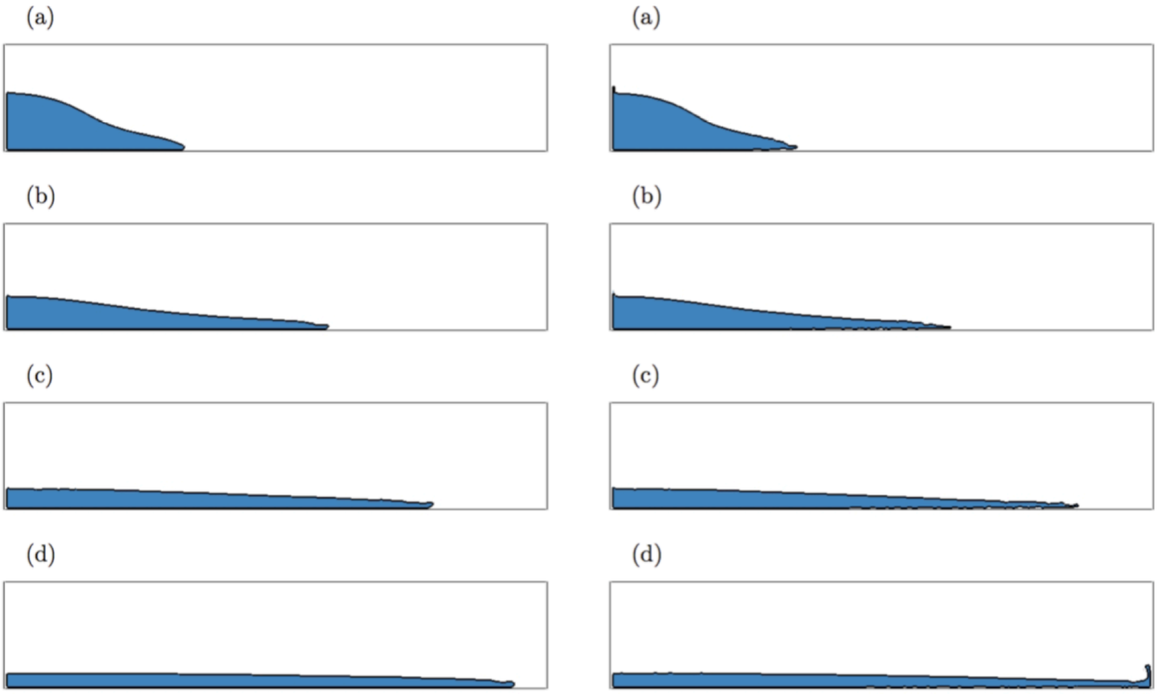

Fig. 4: The process of the 2d-breaking dam at time (a) 0.1, (b) 0.2, (c) 0.3, and (d) 0.4 sec. Left: Level set method; right: THINC-WLIC scheme (512 × 102 cells).

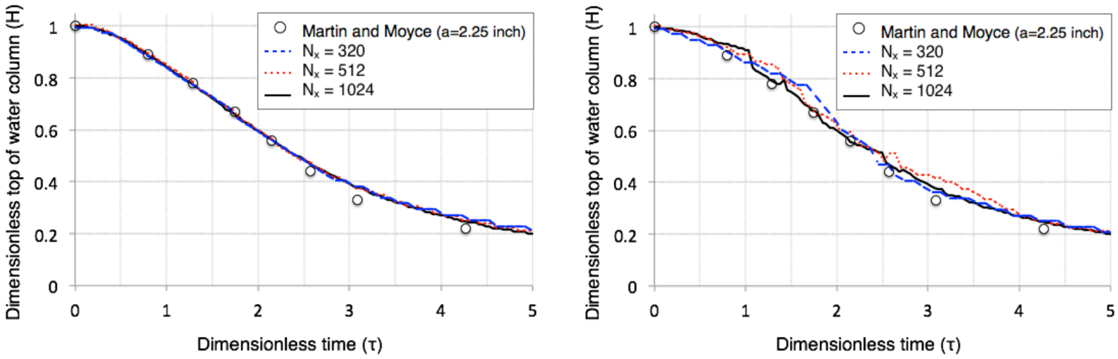

Fig. 5: Comparison of numerical and experimental results for the position of the water column top. Left: level set method; right: THINC-WLIC scheme.

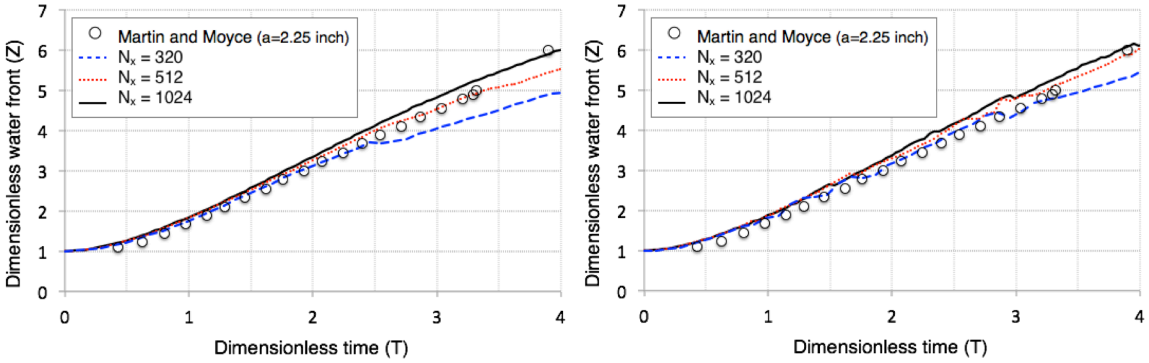

Fig. 6: Comparison of numerical and experimental results for the position of the surge front. Left: level set method; right: THINC-WLIC scheme.

6.2 Three-dimensional dam-break problem

Three-dimensional dam-break problems are considered. A domain size is *0.72 m × 0.36 m × 0.12 m*. The width of water column is set to *15 cm*, and the height of thin water surface is set to *1.8 cm*. The number of grid points is $(N_1, N_2, N_3) = (1024, 512, 176)$, and the grid resolution is *0.703 mm*. Figure 7 shows water surface profiles at time step: (a) 0.2, (b) 0.3, (c) 0.4, and (d) 0.5 sec. Although free surface profiles of the level set method and the VOF method are roughly the same, there are large differences at the bubbles in the fluid. Consequently, the level set method does not guarantee the conservation of mass locally. The volume of water by using the level set method increases by *27 %,* and volume of the water by using the VOF method decreased by *0.4 %* at *time = 1 sec*.

6.3 Three-dimensional dam-break problem with a real ship data

A large-scale free-surface problem is considered. We have generated the solid object based on real ship data (Fig. 8 : bulk carrier) and carried out the domain decomposition for multiple-GPU computing. The VOF method (THINC-WLIC) is applied for interface capturing method. We used 80 GPUs for the computation with a *2560 × 296 × 736* mesh. The solid object does not move in this calculation. Figures 9 and 10 show profiles of free-surface at time step (a)6, (b)9, (c)12, (d)15, (e)18, and (f) 21 sec. As shown above, our code can calculate turbulent violent flows with high Reynolds number, and it is concluded that the present scheme is a promising candidate to simulate free-surface flows.

7 CONCLUSION

This paper has presented a large-scale free-surface simulation based on real ship data. The LES-based lattice Boltzmann method is applied to turbulent free-surface flow with a high Reynolds number. Our code is written in CUDA, and the GPU kernel function is tuned to achieve high performance on the TSUBAME 2.5 supercomputer at Tokyo Institute of Technology. We obtained good scalabilities from 8 GPUs to 1000 GPUs. The fairly high performance of 809 MLUPS (Mega lattice update per second) using 1000 GPUs is achieved in single precision. By executing this high-performance computation, turbulent violent flow simulation with a real ship data is considered, and details of the free surface will be simulated with much higher accuracy than ever before.

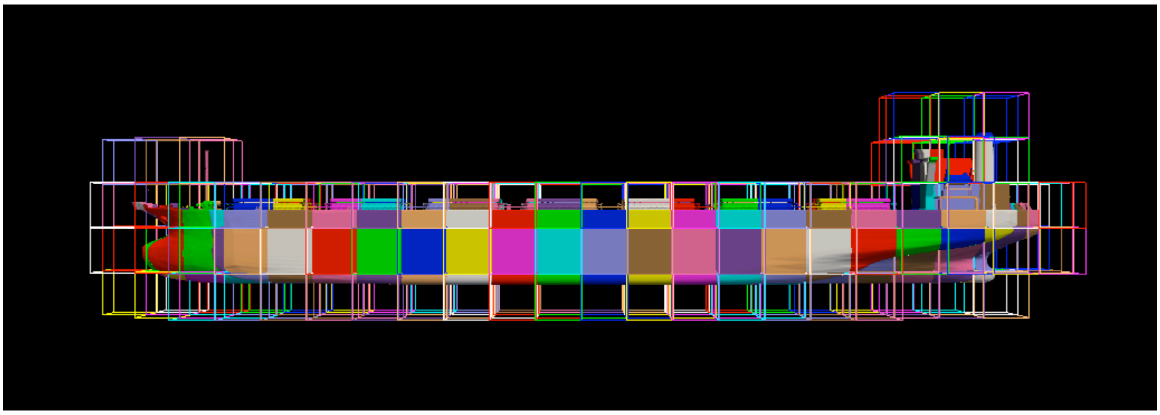

Fig. 8: Level set data of a bulk carrier.

8 ACKNOWLEDGEMENT

This research was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI, a Grant-in-Aid for Young Scientists (B) No. 15K20996.

REFERENCES

[1] Rahimian, I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, et al. Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures. In Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, pages 1–11. IEEE Computer Society, 2010.

[2] X. Wang and T. Aoki. Multi-gpu performance of incompressible flow computation by lattice boltzmann method on gpu cluster. Parallel Computing, 2011.

[3] M. Bernaschi, M. Fatica, S. Melchionna, S. Succi, and E. Kaxiras. A flexible high-performance lattice boltzmann gpu code for the simulations of fluid flows in complex geometries. Concurrency and Computation: Practice and Experience, 22(1):1–14, 2009.

[4] H. Yu, S.S. Girimaji, and L.S. Luo. Dns and les of decaying isotropic turbulence with and without frame rotation using lattice boltzmann method. Journal of Computational Physics, 209(2):599–616, 2005.

[5] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. Physics of Fluids A: Fluid Dynamics, 3:1760, 1991.

[6] DK Lilly. A proposed modification of the germano subgrid-scale closure method. Physics of Fluids A: Fluid Dynamics, 4:633, 1992.

[7] H.Kobayashi, F. Ham, and X. Wu. Application of a local sgs model based on coherent structures to complex geometries. International Journal of Heat and Fluid Flow, 29(3), 2008.

[8] H. Kobayashi. Large eddy simulation of magneto hydrodynamic turbulent channel flows with local subgrid-scale model based on coherent structures. Physics of Fluids, 18:045107, 2006.

[9] U Rude and N Thurey. Free surface lattice- boltzmann fluid simulations with and without level sets. In Vision, Modeling, and Visualization 2004: Proceedings, November 16-18, 2004, Standford, USA, page 199. IOS Press, 2004.

[10] Nils Thurey, C Korner, and U Rude. Inter- active free surface fluids with the lattice boltz- mann method. Technical Report05-4. University of Erlangen-Nuremberg, Germany, 2005.

[11] H. Yu, S.S. Girimaji, and L.S. Luo. Dns and les of decaying isotropic turbulence with and without frame rotation using lattice boltzmann method. Journal of Computational Physics, 209(2):599–616, 2005.

[12] T. Shimokawabe, T. Aoki, T. Takaki, T. Endo, A. Yamanaka, N. Maruyama, A. Nukada, and S. Matsuoka. Peta-scale phase-field simulation for dendritic solidification on the tsubame 2.0 supercomputer. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, page 3. ACM, 2011.

[13] J.C. Martin and W.J. Moyce, “An experimental study of the collapse of liquid columns on a rigid horizontal plane” Phil. Trans. Royal Soc. of London, Vol. 244, pp.312-324, 1952

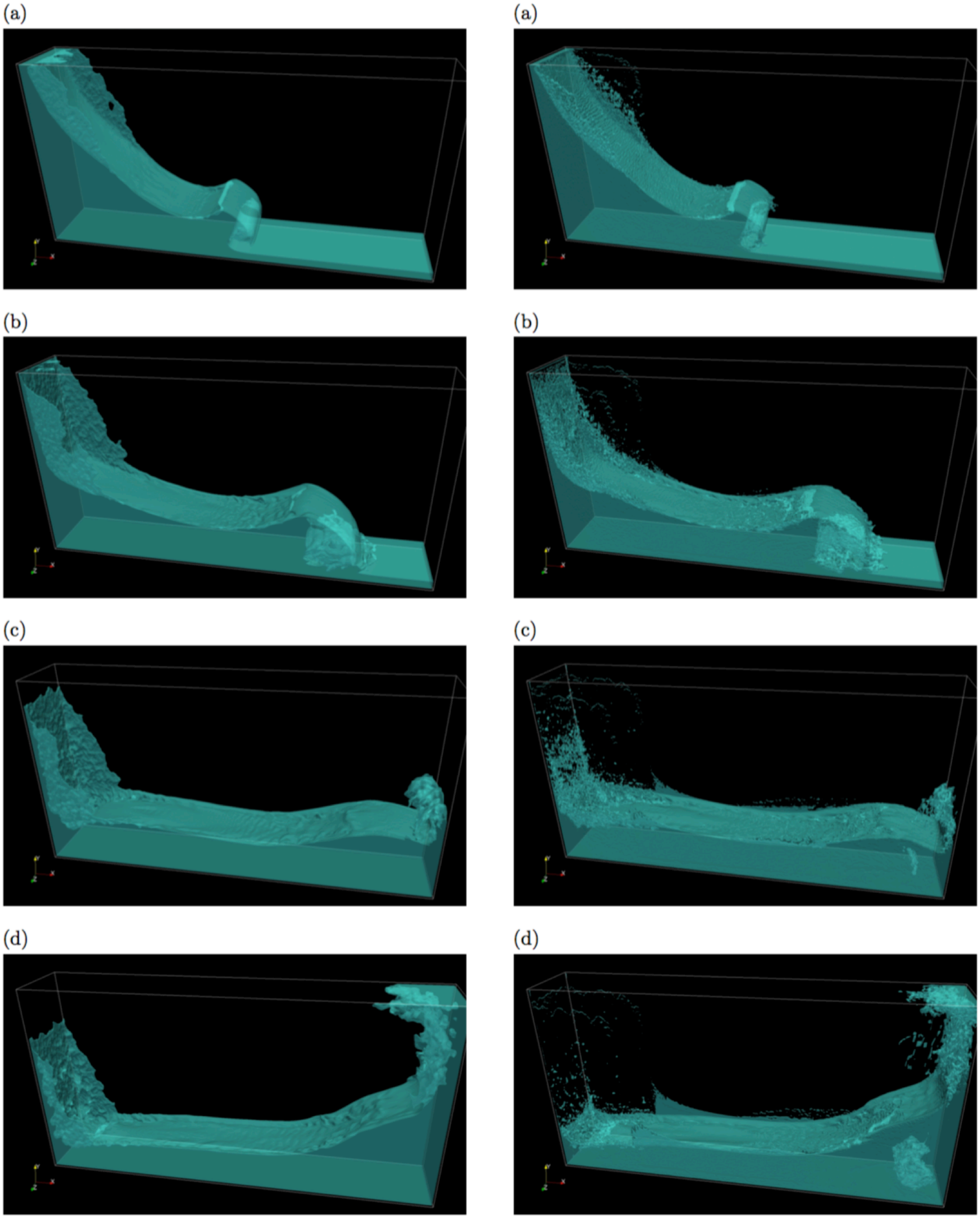

Fig. 7: The water surface profiles of the 3d-breaking dam problem at time step (a) 0.2, (b) 0.3, (c) 0.4, and (d) 0.5 sec. Left: Level set method; right: THINC-WLIC scheme (1024 × 512 × 176 cells).

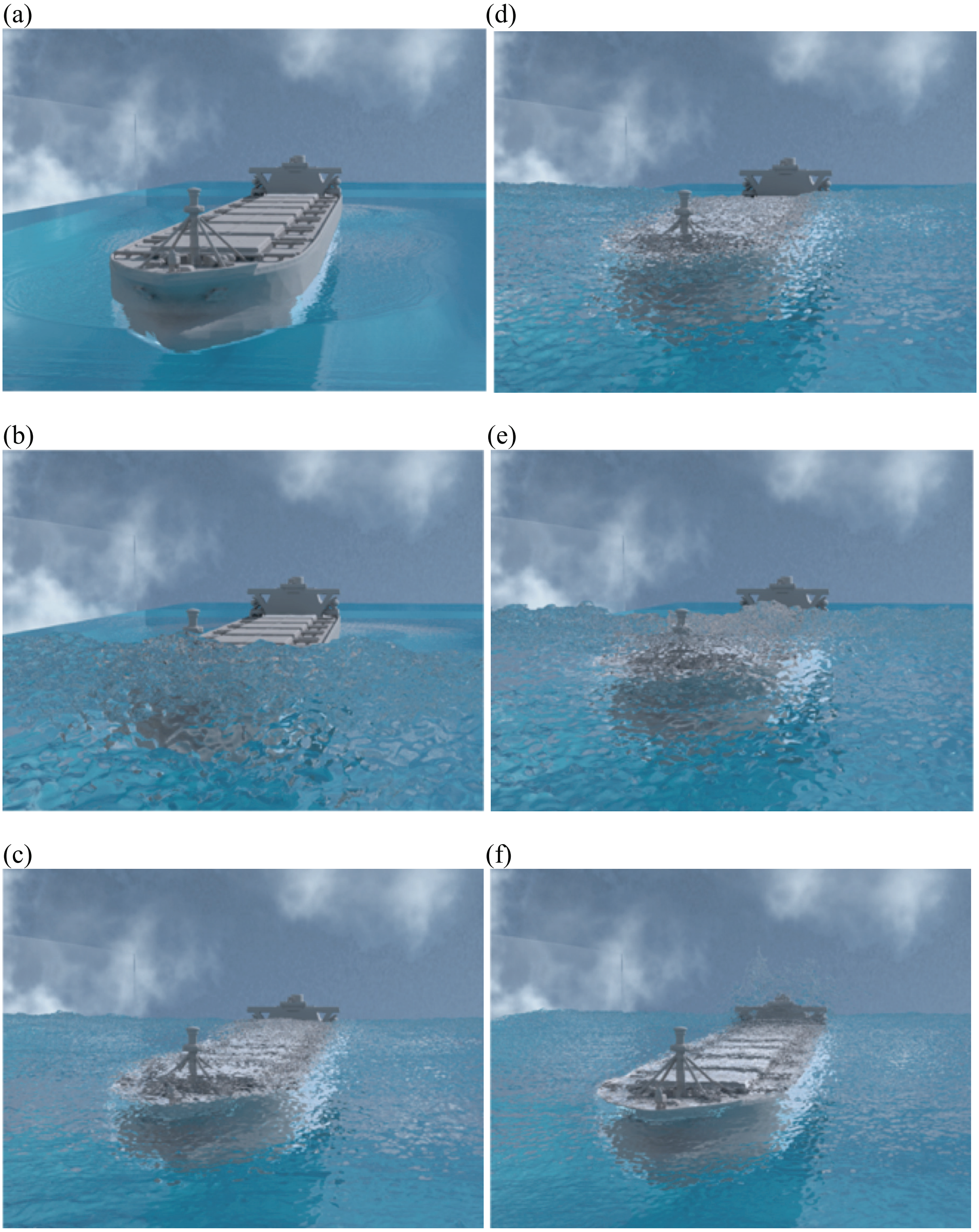

Fig. 9: The evolutionary process of the 3d-breaking dam with a bulk carrier data at time step (a) 6, (b) 9, (c) 12, (d) 15, (e) 18, and (f) 21 sec. (THINC-WLIC scheme : 2560 × 296 × 736 cells).

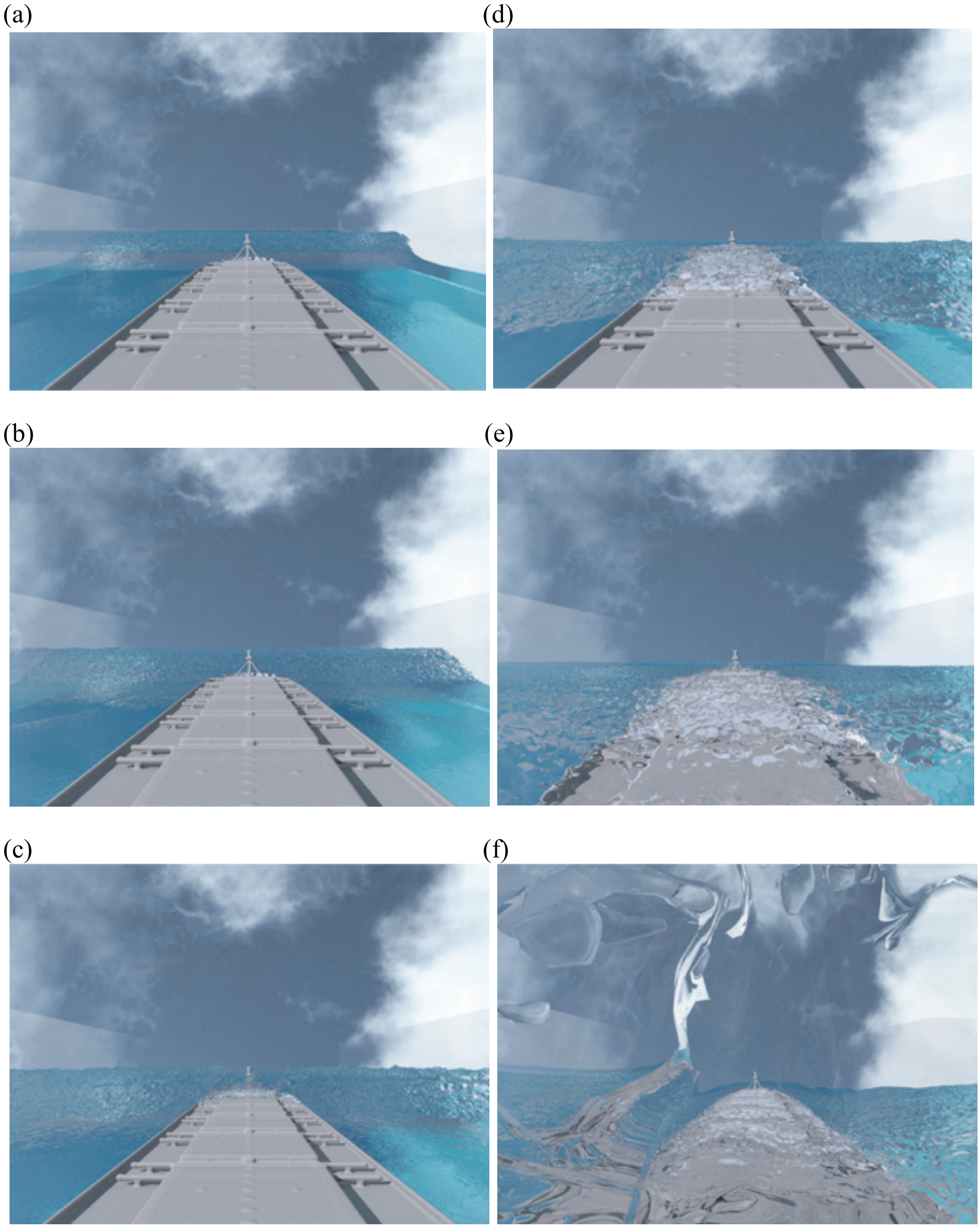

Fig. 10: The evolutionary process of the 3d-breaking dam with a bulk carrier data at time step (a) 6, (b) 9, (c) 12, (d) 15, (e) 18, and (f) 21 sec. (THINC-WLIC scheme : 2560 × 296 × 736 cells).