

Free-surface flow simulations with floating objects using lattice Boltzmann method

Watanabe, Seiya

Research Institute for Applied Mechanics, Kyushu University

Kawahara, Jun

School of Engineering, Tokyo Institute of Technology

Aoki, Takayuki

Global Scientific Information and Computing Center, Tokyo Institute of Technology

Sugihara, Kenta

Center for Computational Science & e-Systems, Japan Atomic Energy Agency

他

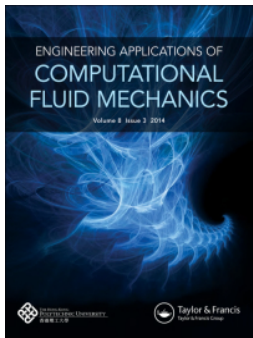
<https://hdl.handle.net/2324/7183291>

出版情報 : Engineering Applications of Computational Fluid Mechanics. 17 (1), 2023-05-12.
Taylor and Francis

バージョン :

権利関係 : © 2023 The Author(s).





Engineering Applications of Computational Fluid Mechanics

ISSN: (Print) (Online) Journal homepage: <https://www.tandfonline.com/loi/tcfm20>

Free-surface flow simulations with floating objects using lattice Boltzmann method

Seiya Watanabe, Jun Kawahara, Takayuki Aoki, Kenta Sugihara, Shinsuke Takase, Shuji Moriguchi & Hirotada Hashimoto

To cite this article: Seiya Watanabe, Jun Kawahara, Takayuki Aoki, Kenta Sugihara, Shinsuke Takase, Shuji Moriguchi & Hirotada Hashimoto (2023) Free-surface flow simulations with floating objects using lattice Boltzmann method, Engineering Applications of Computational Fluid Mechanics, 17:1, 2211143, DOI: [10.1080/19942060.2023.2211143](https://doi.org/10.1080/19942060.2023.2211143)

To link to this article: <https://doi.org/10.1080/19942060.2023.2211143>



© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 12 May 2023.



Submit your article to this journal [↗](#)



Article views: 26



View related articles [↗](#)



View Crossmark data [↗](#)

Free-surface flow simulations with floating objects using lattice Boltzmann method

Seiya Watanabe^a, Jun Kawahara^b, Takayuki Aoki^c, Kenta Sugihara^d, Shinsuke Takase^e, Shuji Moriguchi^f and Hirotsada Hashimoto^g

^aResearch Institute for Applied Mechanics, Kyushu University, Kasuga-city, Fukuoka, Japan; ^bSchool of Engineering, Tokyo Institute of Technology, Meguro-ku, Tokyo, Japan; ^cGlobal Scientific Information and Computing Center, Tokyo Institute of Technology, Meguro-ku, Tokyo, Japan; ^dCenter for Computational Science & e-Systems, Japan Atomic Energy Agency, Kashiwashi, Chiba, Japan; ^eDepartment of Civil Engineering and Architecture, Hachinohe Institute of Technology, Hachinohe, Aomori, Japan; ^fInternational Research Institute of Disaster Science, Tohoku University, Sendai, Japan; ^gGraduate School of Engineering, Osaka Metropolitan University, Osaka, Japan

ABSTRACT

In tsunami inundations or slope disasters of heavy rain, a lot of floating debris or driftwood logs are included in the flows. The damage to structures from solid body impacts is more severe than the damage from the water pressure. In order to study free-surface flows that include floating debris, developing a high-accurate simulation code of free-surface flows with high performance for large-scale computations is desired. We propose the single-phase free-surface flow model based on the cumulant lattice Boltzmann method coupled with a particle-based rigid body simulation. The discrete element method calculates the contact interaction between solids. An octree-based AMR (Adaptive Mesh Refinement) method is introduced to improve computational accuracy and time-to-solution. High-resolution grids are assigned near the free surfaces and solid boundaries. We conducted two kinds of tsunami flow experiments in the 15 and 70 m water tanks at Hachinohe Institute of Technology and Kobe University to validate the accuracy of the proposed model. The simulation results have shown good agreement with the experiments for the drifting speed, the number of trapped wood pieces, and the stacked angles.

ARTICLE HISTORY

Received 2 February 2023
Accepted 1 May 2023

KEYWORDS

Free-surface flow; floating object; tsunami; lattice Boltzmann method; adaptive mesh refinement

1. Introduction

Free-surface flows with a large amount of rubble and driftwood are usually part of tsunami and flood disasters. The damage to coastal structures is divided into hydrodynamic force and contact forces by drifting objects. According to the guidelines of the Cabinet Office in Japan, tsunami effects on coastal buildings are predicted from the hydrostatic pressure based on a tsunami inundation depth. However, since a tsunami's impact depends on the tsunami's velocity, the hydrostatic pressure model cannot be used for accurate evaluation. In addition, contact forces of drifting objects to coastal structures are large and are an important issue in designing tsunami countermeasures.

Numerical simulations are powerful tools for problems that are difficult to experiment with on a real-world scale, such as disasters. Simulating interactions between tsunami and drifting objects requires three-dimensional computations. Particle methods such as the smoothed-particle hydrodynamics (SPH) method and the moving

particle semi-implicit (MPS) method are being actively studied for a tsunami with debris (Murotani et al., 2014; Tsuzuki & Aoki, 2015) because it is easier to calculate fluid-structure interactions than using mesh-based methods. Since one particle interacts with 100 or more neighbouring particles, the computational cost of particle methods tends to be higher than that of mesh-based methods. High resolution is required in near drifting objects, making tsunami analysis with drifting objects inevitably a large-scale simulation.

A recent popular alternative for solving the Navier-Stokes equation is the lattice Boltzmann method (LBM) (Wolf-Gladrow, 2000) based on the discrete-velocity Boltzmann equation. The LBM consists of simple streaming and collision operators of the distribution functions. Due to the weakly compressible flow formulation, the time integration is fully explicit without the computation of the pressure Poisson equation. Not computing the Poisson equation by linear solvers permits an efficient parallelization for large-scale CFD simulations

CONTACT Seiya Watanabe  swatanabe@riam.kyushu-u.ac.jp  Research Institute for Applied Mechanics, Kyushu University, 6-1 Kasuga-koen, Kasuga-city, Fukuoka, Japan

(Bernaschi et al., 2011; Randles et al., 2015). The LBM achieves high performance in recent architectures such as GPU (Graphics Processing Units) because of its simplicity (Schönherr, 2015; Valero-Lara, 2016).

Improvement of the stability of the LBM is a key element when applying it to free-surface flows at high Reynolds numbers. For stability improvement, the work in Onodera and Ohashi (2016) uses a sub-grid scale large eddy simulation (LES) model with the simple single relaxation time (SRT) model. These works successfully simulate high-Reynolds free-surface flows such as a dam-breaking process. However, it is pointed out by Sato and Koshimura (2020) that SRT-LES approaches produce a problem of pressure oscillations. The authors in Janssen and Krafczyk (2010) introduce the multiple relaxation time (MRT) model (d'Humières, 2002) with a LES model for free-surface flows. The MRT model calculates the collision process in moment space and it can stabilize calculations of free-surface flows, but it is necessary to optimize the parameters of the relaxation coefficient according to the problem. In order to solve the problems of the MRT model and achieve a robust and highly accurate calculation of free surface flows, the cumulant collision model (Geier et al., 2015) is introduced to free-surface flow calculations. The combination of the cumulant LBM and the volume of fluid (VOF) method, an interface capture method, has been reported for the analysis of violent free surface flows in dam-break problems (Sato et al., 2022). We have proposed a method based on the cumulant model and the conservative Allen-Cahn equation (Watanabe & Aoki, 2021), and have succeeded in reducing pressure oscillations in free surface flows (Watanabe et al., 2021).

In this paper, we propose a lattice Boltzmann based free-surface flow model for tsunami and flood disasters with floating debris, with the following features:

- Stable simulations under very high Reynolds number flows (such as a tsunami) by coupling the cumulant LBM and the phase-field method for interface capturing (Geier et al., 2015). A single-phase free-surface model allows the calculation of violent flows that would be unstable in a gas-liquid two-phase model. The advantage of using the phase-field method, which is often used for gas-liquid two-phase flows (Wang et al., 2019), for a single-phase free surface model is that it stabilizes calculations by suppressing the generation of very small droplets in violent free-surface flows.
- It uses a particle-based method to calculate the rigid body motions (Bell et al., 2005). A floating object is assumed as a rigid body. The DEM's (Discrete Element Method) (Cundall & Strack, 1979) springs

and dashpots model the contact interactions between objects.

- Near free surfaces and objects require high-resolution computational grids, but high-resolution grids are not necessary for calculations in regions away from interfaces. We use this property to reduce computation cost, by introducing the AMR (Adaptive Mesh Refinement) method (MacNeice et al., 2000; Tu et al., 2005) which allocates high-resolution grids only near free surfaces and solid boundaries.
- To reduce execution times, computations are performed on multiple GPUs. Our code uses a domain decomposition method to support MPI parallel computing with multiple GPUs. To achieve high performance for large-scale simulations on a large number of GPUs, we use a dynamic load balancing based on the space-filling curve (Butz, 1968; Campbell et al., 2003).

To validate our model, we conduct experiments in a wave flume and compare the simulation results with the experimental results.

2. Numerical method

2.1. Lattice Boltzmann method

The lattice Boltzmann method (LBM) is an efficient approach for solving incompressible flows. The LBM usually requires an explicit time integration and a stencil accessing only immediate neighbouring lattice points. These features are very suitable for GPU computing and large-scale simulations. The LBM solves the Boltzmann equation discretized in velocity space, in contrast to classical semi-implicit solvers based on the Navier–Stokes equation. The following Boltzmann equation discretized in the time t and the space $\mathbf{x}$ for the velocity distribution functions f_{ijk} is solved on grid points

$$f_{ijk}(\mathbf{x} + \xi_{ijk}\Delta t, t + \Delta t) = f_{ijk}(\mathbf{x}, t) + \Omega_{ijk}(\mathbf{x}, t), \quad (1)$$

where $i, j, k \in \{-1, 0, 1\}$ indicate the direction of the velocity distribution function and Ω_{ijk} is a collision operator. In this study, the D3Q27 model is used, which introduces 27 discretized velocities ξ_{ijk} . The discrete velocity ξ_{ijk} is set using $c = \Delta x / \Delta t$ as

$$\xi_{ijk} = (ic, jc, kc)^T. \quad (2)$$

The density ρ and velocity $\mathbf{u}$ can be evaluated as

$$\rho(\mathbf{x}, t) = \sum_{ijk} f_{ijk}(\mathbf{x}, t), \quad (3)$$

$$\rho \mathbf{u}(\mathbf{x}, t) = \sum_{ijk} \xi_{ijk} f_{ijk}(\mathbf{x}, t). \quad (4)$$

The speed of sound is $c_s = \frac{c}{\sqrt{3}}$, and the pressure p is calculated from the isothermal equation of state

$$p = \rho c_s^2. \quad (5)$$

The BGK (Bhatnagar, Gross and Krook) collision model (Bhatnagar et al., 1954) is widely used for the collision operator in Equation (1). Although it has a simple formulation and less computational cost, the BGK model is unstable at high Reynolds numbers. Since a tsunami has a very high Reynolds number due to its scale and flow velocity, it is necessary to improve the numerical stability of the LBM. The cumulant collision model (Geier et al., 2015) is utilized in this work due to its high stability and accuracy under high-Reynolds number flows. The discrete distribution function in continuous form using the microscopic velocity $\xi = (\xi, v, \zeta)$ and the Dirac delta function is given by

$$f(\xi) = f(\xi, v, \zeta) = \sum_{ijk} f_{ijk} \delta(ic - \xi) \delta(jc - v) \delta(kc - \zeta). \quad (6)$$

The two-sided Laplace transform of this distribution function in the momentum wave number $\Xi = (\Xi, \Upsilon, Z)$ is

$$\begin{aligned} F(\Xi) &= \mathcal{L}[f(\xi)] = \int_{-\infty}^{\infty} f(\xi) e^{-\Xi \cdot \xi} d\xi \\ &= \sum_{ijk} f_{ijk} e^{-\Xi ic} e^{-\Upsilon jc} e^{-Zkc}. \end{aligned} \quad (7)$$

The coefficients of the Taylor expansion of the logarithm of the moment generating function (Equation (7)) are considered as the cumulants $C_{\alpha\beta\gamma}$:

$$\begin{aligned} C_{\alpha\beta\gamma} &= c^{-\alpha-\beta-\gamma} \\ &\times \left. \frac{\partial^\alpha \partial^\beta \partial^\gamma}{\partial \Xi^\alpha \partial \Upsilon^\beta \partial Z^\gamma} \ln(F(\Xi, \Upsilon, Z)) \right|_{\Xi=\Upsilon=Z=0}, \end{aligned} \quad (8)$$

where, $\alpha + \beta + \gamma$ indicates the order of a cumulant. The collision process of the cumulant model is

$$C_{\alpha\beta\gamma}^* = \omega_{\alpha\beta\gamma} C_{\alpha\beta\gamma}^{\text{eq}} + (1 - \omega_{\alpha\beta\gamma}) C_{\alpha\beta\gamma}, \quad (9)$$

where $C_{\alpha\beta\gamma}^*$ is the post-collision cumulant, $C_{\alpha\beta\gamma}^{\text{eq}}$ is the equilibrium value of the cumulant. The relaxation parameter $\omega_{\alpha\beta\gamma}$ is $1/\tau$ for non-diagonal second-order cumulants and one for others. The equilibrium cumulants of the second-order diagonal components are

$$C_{200}^{\text{eq}} = C_{020}^{\text{eq}} = C_{002}^{\text{eq}} = \rho c_s^2, \quad (10)$$

and other equilibrium cumulants are zero.

The gravitational force term in the cumulant LBM follows the method of the original paper on the cumulant LBM (Geier et al., 2015). Half the gravitational force is added to the pre-collision first-order moments.

2.2. Smagorinsky LES

Tsunamis usually appear under very high Reynolds numbers in the turbulence region. Although the cumulant collision model works as an implicit LES (Large Eddy Simulation), in order to stabilize more by suppressing the dispersion of very small droplets in violent free-surface flows, we introduce a coherent structure Smagorinsky LES model (Kobayashi, 2005) to the cumulant LBM. The effect of small eddies in the sub-grid scale is modelled as an eddy viscosity ν_{SGS} , which is given as

$$\nu_{\text{SGS}} = C \bar{\Delta}^2 |\bar{S}|, \quad (11)$$

$$|\bar{S}| = \sqrt{2 \bar{S}_{ij} \bar{S}_{ji}}, \quad (12)$$

where, C is the Smagorinsky constant, $\bar{\Delta}$ is the filter width, with $\bar{\Delta} = \Delta x$, and $\bar{S}_{ij}$ is the strain rate tensor which is calculated by

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} \right). \quad (13)$$

The barred variables indicate the grid-scale components, and the macro variables computed in the LBM are used as the grid-scale values. The eddy viscosity is assumed to affect the flow as well as the molecular viscosity ν_0 , and the relaxation parameter τ is set to

$$\tau = \frac{1}{2} + \frac{(\nu_0 + \nu_{\text{SGS}})}{c_s^2 \Delta t}. \quad (14)$$

The Smagorinsky constant C is set using the coherent structure function as

$$C = C_{\text{CSM}} |F_{\text{CS}}|^{3/2}, \quad (15)$$

$$F_{\text{CS}} = \frac{Q}{E}, \quad (16)$$

$$Q = -\frac{1}{2} \frac{\partial \bar{u}_j}{\partial x_i} \frac{\partial \bar{u}_i}{\partial x_j}, \quad (17)$$

$$E = \frac{1}{2} \frac{\partial \bar{u}_j}{\partial x_i} \frac{\partial \bar{u}_j}{\partial x_i}, \quad (18)$$

where, we set $C_{\text{CSM}} = 0.05$. The coherent structure Smagorinsky model has variable model coefficients C at each grid point. Therefore, it can be introduced into turbulent flows with object boundaries without using wall functions.

2.3. Free-surface model

We use a phase-field variable ϕ as a smooth marker function distinguishing the liquid phase ($\phi = 1$) and the gas phase ($\phi = 0$). The interface represents the smooth profile of ϕ in several grid points. The time evolution of ϕ follows the conservative Allen-Cahn equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \nabla \cdot \left[M \left\{ \nabla \phi - \frac{1 - 4(\phi - \phi^{\text{ave}})^2}{W} \mathbf{n} \right\} \right], \quad (19)$$

where, W , M , and $\mathbf{n}$ are the interface thickness, the mobility parameter, and the unit normal vector of an interface, respectively. In this study, we use $W = 3\Delta x$ and $M = 0.05$. ϕ^{ave} is the average value of ϕ for the liquid and gas phases, in this study $\phi^{\text{ave}} = 1/2$.

The conservative phase-field LBM (Geier et al., 2015) solves the conservative Allen-Cahn equation. Although the phase-field LBM uses more memory than solving the conservative Allen-Cahn equation directly, it has the advantage of simplifying the handling of moving boundary conditions. We define a distribution function h_{ijk} for the phase field variable ϕ , and in this study, the D3Q27 velocity model is also applied for h_{ijk} . The lattice Boltzmann equation for the phase field model is

$$h_{ijk}(\mathbf{x} + \xi_{ijk}\Delta t, t + \Delta t) = \frac{1}{\tau_\phi} h_{ijk}^{\text{eq}}(\mathbf{x}, t) + \left(1 - \frac{1}{\tau_\phi}\right) h_{ijk}(\mathbf{x}, t), \quad (20)$$

where, τ_ϕ is a relaxation parameter defined as

$$\tau_\phi = \frac{1}{2} + \frac{M}{c_s^2 \Delta t}. \quad (21)$$

The distribution function at equilibrium state h_{ijk}^{eq} is defined as

$$h_{ijk}^{\text{eq}} = \phi \Gamma_{ijk} + \theta w_{ijk} \xi_{ijk} \cdot \mathbf{n}, \quad (22)$$

$$\Gamma_{ijk} = w_{ijk} \left\{ 1 + \frac{\xi_{ijk} \cdot \mathbf{u}}{c_s^2} + \frac{(\xi_{ijk} \cdot \mathbf{u})^2 - c_s^2 |\mathbf{u}|^2}{2c_s^4} \right\}, \quad (23)$$

$$\theta = \frac{M}{c_s^2} \left\{ \frac{1 - 4(\phi - \phi^{\text{ave}})^2}{W} \right\}, \quad (24)$$

where, w_{ijk} is the weighting parameter of the D3Q27 model. The unit normal vector $\mathbf{n}$ of the free surface is computed from the level set function ψ of the interface by the second-order central difference. The level set function ψ is created every time step from the phase field ϕ by the re-initialization process (Peng et al., 1999; Watanabe

et al., 2021). The phase field variable ϕ is calculated from the sum of h_{ijk} as

$$\phi(\mathbf{x}, t) = \sum_{ijk} h_{ijk}(\mathbf{x}, t). \quad (25)$$

To improve the stability, the lattice Boltzmann equation is solved only in the liquid phase with free-surface boundary conditions (Körner et al., 2005) in our free-surface model. Lattice points with $\phi \geq \phi^{\text{ave}}$ are treated as the liquid phase and those with $\phi < \phi^{\text{ave}}$ as the gas phase. The distribution function f_{ijk} for the gas phase is not defined. Therefore, at lattice points near an interface, the missing distribution functions streaming from gas cells are reconstructed as

$$f_{ijk}(\mathbf{x}, t + \Delta t) = -f_{\overline{ijk}}(\mathbf{x}, t) + f_{ijk}^{\text{eq}}(\rho_0, \mathbf{u}) + f_{\overline{ijk}}^{\text{eq}}(\rho_0, \mathbf{u}), \quad (26)$$

where, the subscript $\overline{ijk}$ indicates the opposite direction of ijk , and ρ_0 is the reference density. The equilibrium distribution function $f_{ijk}^{\text{eq}}(\rho, \mathbf{u})$ is

$$f_{ijk}^{\text{eq}}(\rho, \mathbf{u}) = w_{ijk} \rho \left\{ 1 + \frac{\xi_{ijk} \cdot \mathbf{u}}{c_s^2} + \frac{(\xi_{ijk} \cdot \mathbf{u})^2 - c_s^2 |\mathbf{u}|^2}{2c_s^4} \right\} \quad (27)$$

where w_{ijk} is the weighting parameter of the D3Q27 model.

Although the gas velocity is required to solve the conservative Allen-Cahn equation, our free-surface approach only calculates the liquid phase by using the LBM, which leaves the gas velocity undefined. These undefined velocities are found by extrapolating the interface velocities to satisfy $\nabla \mathbf{u} \cdot \mathbf{n} = 0$ in the gas phase by using the velocity extension method (Peng et al., 1999). At each time step, the following equation is iteratively solved using the first-order Euler method and the first-order upwind difference:

$$\frac{\partial \mathbf{u}}{\partial t_\psi} = -S(\psi) \frac{\nabla \psi}{|\nabla \psi|} \cdot \nabla \mathbf{u}, \quad (28)$$

$$S(\psi) = \begin{cases} 0, & \psi < 0 \\ 1, & \psi \geq 0 \end{cases}. \quad (29)$$

where, t^ψ is the fake time set as $\Delta t^\psi = \frac{\Delta x}{2}$, and ψ is the level-set function of the interface. The extrapolated velocity in the gas phase region is used in Equation (23) to update the phase field variable in the gas region.

2.4. Object motion

A particle-based approach for rigid body dynamics (Bell et al., 2005) calculates the motion of objects such as driftwood or debris. The shape of an object is represented by

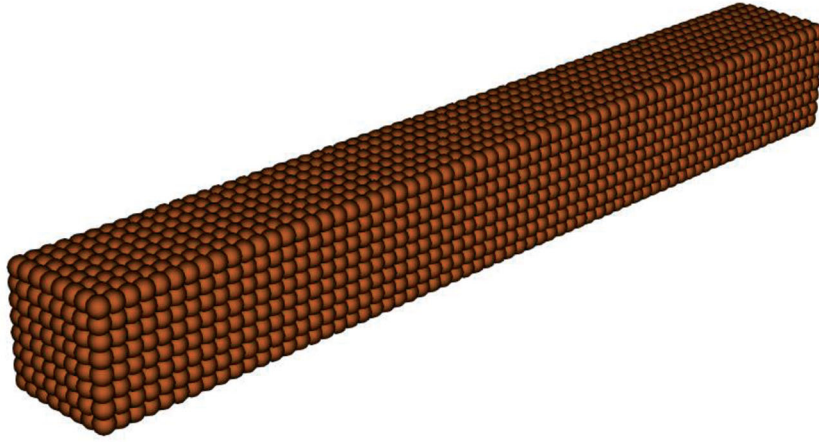


Figure 1. Object represented by a set of spherical particles.

a signed distance function and a set of spherical particles of finite size as shown in Figure 1. The signed-distance function is used to calculate the object's overall mass and moment of inertia and LBM boundary conditions. To reduce the number of particles, spherical particles are placed only on the surface of the object. The set of spherical particles calculates collisions between rigid objects. The sphere size is determined empirically by trial and error, and in this paper, the sphere size is set to 1.5 times bigger than the grid spacing of the fluid calculation.

The motion of a rigid body is computed by dividing motion into translation and rotation. The translation of the centre of mass is governed by

$$m_s \frac{d\mathbf{v}_s}{dt} = \sum_i \mathbf{F}_i^{\text{DEM}} + \mathbf{F}_s^{\text{fluid}} + m_s \mathbf{g}, \quad (30)$$

where, m_s and $\mathbf{v}_s$ are the mass and the velocity of the rigid body, respectively, $\mathbf{F}_s^{\text{fluid}}$ is a hydrodynamic force, and $\mathbf{g}$ is the gravity acceleration. $\mathbf{F}_i^{\text{DEM}}$ indicates the contact force acting on the i th particle constructing the object. The contact force applied to the object is the sum of the forces exerted on all particles of the object.

The rotation is computed based on the equation of the time derivative of the rigid body's angular momentum $\mathbf{L}_s$:

$$\frac{d\mathbf{L}_s}{dt} = \sum_i \mathbf{r}_i \times \mathbf{F}_i^{\text{DEM}} + \mathbf{M}_s^{\text{fluid}}, \quad (31)$$

where, $\mathbf{M}_s^{\text{fluid}}$ is a fluid torque and $\mathbf{r}_i$ is a relative position of the acting point from the centre of mass. The angular velocity ω_s is calculated as

$$\omega_s(t) = \mathbf{I}_s(t)^{-1} \mathbf{L}_s(t), \quad (32)$$

where, $\mathbf{I}_s(t)$ is the inertial tensor at time t obtained from the rotation matrix $\mathbf{R}(t)$

$$\mathbf{I}_s(t)^{-1} = \mathbf{R}(t) \mathbf{I}_s(0)^{-1} \mathbf{R}(t)^T. \quad (33)$$

The rotation matrix is represented by a quaternion. Equations (30) and (31) are both solved by the first-order Euler's method.

The contact force acting on a solid particle that represents an object is estimated by using the DEM model (Cundall & Strack, 1979). The force working between i th and j th particles is decomposed into a normal component and a tangential component,

$$\mathbf{F}_{ij}^{\text{DEM}} = \mathbf{F}_{n,ij}^{\text{DEM}} + \mathbf{F}_{t,ij}^{\text{DEM}}, \quad (34)$$

where, subscripts n and t indicate the normal components and the tangential components, respectively. In the DEM, the contacting forces are modelled by springs and dashpots as

$$\mathbf{F}_{n,ij}^{\text{DEM}} = k_n \delta_{n,ij} + \eta_n \mathbf{v}_{n,ij}, \quad (35)$$

$$\mathbf{F}_{t,ij}^{\text{DEM}} = \min\{k_t \delta_{t,ij} + \eta_t \mathbf{v}_{t,ij}, \mu |\mathbf{F}_{n,ij}^{\text{DEM}}| \mathbf{t}_{ij}\}, \quad (36)$$

where, k , δ , η and $\mathbf{v}$ are the spring coefficient, the displacement, the damping coefficient and the relative velocity, respectively. μ and $\mathbf{t}$ are the friction coefficient and the tangential vector, respectively. The function min returns the smaller absolute value between the two values. We set the spring constant and the damping coefficient based on Young's modulus E , Poisson's ratio ν and coefficient of restitution e (Luding, 2008),

$$k_n = \frac{2E}{3(1-\nu^2)} \sqrt{\frac{r}{2}}, \quad (37)$$

$$\eta_n = 2 \sqrt{\frac{mk_n}{1 + (\pi/\ln e)}}, \quad (38)$$

$$k_t = \frac{k_n}{2(1+\nu)}, \quad (39)$$

$$\eta_t = 2 \sqrt{\frac{mk_t}{1 + (\pi/\ln e)}}. \quad (40)$$

The mass m and the radius r of the small sphere constituting the rigid body are used to determine the coefficients. However, because using the actual Young's modulus E requires very small discrete time Δt^{DEM} , E is generally set small enough not to affect the object's motion.

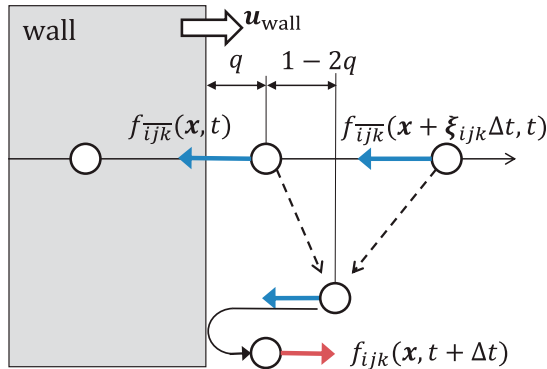
The total force $\mathbf{F}_i^{\text{DEM}}$ acting on the i th particle is the sum of the force $\mathbf{F}_{ij}^{\text{DEM}}$ acting from j th contacting particles,

$$\mathbf{F}_i^{\text{DEM}} = \sum_j \mathbf{F}_{ij}^{\text{DEM}}. \quad (41)$$

2.5. Fluid-solid coupling

The computation of the fluid-solid coupling is based on the momentum exchange at the solid surface. The motion of an object is introduced into the LBM as a moving boundary condition for a curved shape. The non-slip boundary condition based on an interpolated bounce-back scheme (Bouzidi et al., 2001) is utilized at the solid surface. As shown in Figure 2, if a distribution function moves into a solid node, the reflected distribution function is calculated using the second-order interpolation scheme

$$f_{ijk}(\mathbf{x}, t + \Delta t) = \begin{cases} 2qf_{ijk}(\mathbf{x}, t) + (1 - 2q)f_{ijk}(\mathbf{x} + \xi_{ijk}\Delta t, t) + \frac{6\omega_{ijk}\rho(\xi_{ijk} \cdot \mathbf{u}_{\text{wall}})}{c^2}, & 0 < q \leq 0.5 \\ \frac{1}{2q} \left\{ f_{ijk}(\mathbf{x}, t) + \frac{6\omega_{ijk}\rho(\xi_{ijk} \cdot \mathbf{u}_{\text{wall}})}{c^2} \right\} + \frac{(2q - 1)}{2q}f_{ijk}(\mathbf{x}, t), & 0.5 < q \leq 1.0 \end{cases}, \quad (42)$$



where, f_{ijk} indicate the distribution function moving to the opposite direction of the f_{ijk} and $\mathbf{u}_{\text{wall}}$ is the wall velocity. The normalized distance q from the node to the solid surface is calculated using the signed distance function s as:

$$q = \frac{|s(\mathbf{x}, t)|}{|s(\mathbf{x}, t)| + |s(\mathbf{x} - \xi_{ijk}\Delta t, t)|}. \quad (43)$$

If the liquid phase is as thin as one mesh or a triple point, the neighbouring points referenced by the interpolated bounce-back method may not be in the liquid phase. The interpolated bounce-back rule cannot be applied in such situations because the distribution function f is not defined for the gas and solid phases. As an exception, the boundary condition is switched to the halfway bounce-back model setting $q = 1/2$ in Equation (42), which does not require an adjacent point reference.

The halfway bounce-back method is used for the distribution function h of the phase field variable at the solid boundary. The reason is that the mass conservation of the phase field variable is not satisfied if the interpolated bounce-back model is utilized.

The fluid force acting on the object is determined by the momentum exchange method (Wen et al., 2014). In the liquid phase, the fluid force acted from the reflected distribution function on the solid surface is calculated as,

$$\mathbf{F}_{ijk}(\mathbf{x}, t) = (\xi_{ijk} - \mathbf{u}_{\text{wall}})f_{ijk}(\mathbf{x}, t) - (\xi_{ijk} - \mathbf{u}_{\text{wall}})f_{ijk}(\mathbf{x}, t + \Delta t). \quad (44)$$

In the gas phase, we assumed that the force based on the atmospheric pressure works on the object, and the fluid force on a gas node is obtained by

$$\mathbf{F}_{ijk}(\mathbf{x}, t) = (\xi_{ijk} - \mathbf{u}_{\text{wall}})f_{ijk}^{\text{eq}}(\rho_0, \mathbf{u} = 0) - (\xi_{ijk} - \mathbf{u}_{\text{wall}})f_{ijk}^{\text{eq}}(\rho_0, \mathbf{u} = 0), \quad (45)$$

where, ρ_0 is a reference density that represents the atmospheric pressure $p_0 = \rho_0 c_s^2$. The fluid force $\mathbf{F}_s$ and fluid

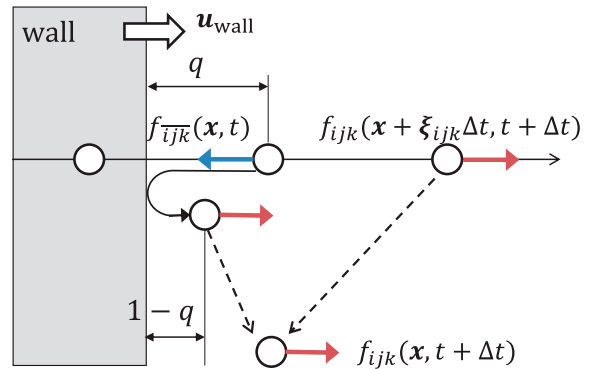


Figure 2. Interpolated bounce-back scheme (left: $0 < q \leq 0.5$, right: $0.5 < q \leq 1$).

torque $\mathbf{M}_s$ acting on the object are calculated by the summation of $\mathbf{F}_{ijk}$ on the solid surface region Ω_s as,

$$\mathbf{F}_s(t) = \sum_{\mathbf{x} \in \Omega_s} \sum_{ijk \in q^*} \mathbf{F}_{ijk}(\mathbf{x}, t), \quad (46)$$

$$\mathbf{M}_s(t) = \sum_{\mathbf{x} \in \Omega_s} \left\{ (\mathbf{x} - \mathbf{x}_s) \times \sum_{ijk \in q^*} \mathbf{F}_{ijk}(\mathbf{x}, t) \right\}, \quad (47)$$

where, $\mathbf{x}_s$ is the centre of gravity of the object. q^* determines the set of the directions of the distribution functions moving from the liquid or gas node to the solid node. As suggested in Aidun et al. (1998), to improve the stability of the fluid-solid coupling with floating objects, the fluid force $\mathbf{F}_s^{\text{fluid}}$ in Equation (30) and the fluid torque $\mathbf{M}_s^{\text{fluid}}$ in Equation (31) are estimated as the averaged value of the force and the torque on present and previous time step

$$\mathbf{F}_s^{\text{fluid}} = \frac{1}{2} \{ \mathbf{F}_s(t) + \mathbf{F}_s(t - \Delta t) \}, \quad (48)$$

$$\mathbf{M}_s^{\text{fluid}} = \frac{1}{2} \{ \mathbf{M}_s(t) + \mathbf{M}_s(t - \Delta t) \}. \quad (49)$$

A difference between discrete times of the LBM and the DEM is a problem in fluid-solid coupling simulations. To satisfy the weak compressibility assumption of the LBM, the Mach number Ma should be small enough. The authors in Li et al. ((2022) assume that density variations can be neglected when $Ma \leq 0.1$, in which case the discrete time of the LBM should be satisfied,

$$Ma = \frac{U_0}{C_s} = \sqrt{3} \frac{U_0}{\Delta x} \Delta t^{\text{LBM}} \leq 0.1, \quad (50)$$

which gives

$$\Delta t^{\text{LBM}} \leq \frac{0.1 \Delta x}{\sqrt{3} U_0}, \quad (51)$$

where, U_0 is the typical velocity. At some grid points, such as droplet splashes or gaps between objects, the velocity is greater than the typical velocity. To satisfy the small Mach number at any lattice point, the discrete time of the LBM is set by trial and error. Since the discrete time of the DEM must be smaller than the natural period of the spring-mass system, Δt^{DEM} should satisfy,

$$\Delta t^{\text{DEM}} < 2\pi \sqrt{\frac{m}{k_n}}, \quad (52)$$

where, m is the mass of the particle and k_n is the spring coefficient of the normal component. In many cases, the discrete time of the DEM is much smaller than that of the LBM. Although it is possible to calculate the LBM using the same discrete time of the DEM, the computational cost increases significantly. To solve this, we use a sub-time stepping that calculates the DEM time integration

with N^{sub} steps for each step of the LBM time integration. To satisfied Equation (52), Δt^{DEM} is set as

$$\Delta t^{\text{DEM}} = \frac{1}{N^{\text{sub}}} \Delta t^{\text{LBM}}. \quad (53)$$

During the calculation of the sub-step of the DEM, the hydrodynamic force acting on the object is assumed to be constant.

A simplified flowchart is shown in Figure 3 to summarize the coupling method of the free-surface LBM and particle-based rigid body dynamics. The flowchart shows one time step of the LBM and does not include the initialization, mesh generation, and output. Inter-node data transfers frequently occur in parallel computation on multiple nodes, but data communication is omitted to simplify the flowchart. The effect of the flow field on the objects is given by the forces and torques calculated by the momentum exchange method. The motions of the objects are transmitted to the LBM fluid through an update of the signed distance function.

2.6. Block-based octree AMR

In order to reduce memory usage and speed up the computation, we introduce an adaptive mesh refinement method (MacNeice et al., 2000; Tu et al., 2005) to free-surface flow simulations by the lattice Boltzmann method. The adaptive mesh refinement is performed by using a series of blocks. All the blocks have a uniform mesh with the same number of lattice points. The LBM calculation point is placed at the centre of the cell. The octree data structure handles the block refinement and coarsening. In the numerical simulation for tsunamis and flood disasters, the computational domain is often wide in the horizontal directions. Since one octree corresponds to a cubic domain, multiple octrees are used to manage the blocks in the rectangular computational domain. A data structure that uses multiple octree data structures is called a forest of octrees. As shown in Figure 4, high-resolution blocks are dynamically assigned near the free surface and the solid boundaries. At the resolution boundary, the trilinear interpolation method is used to connect the velocity distribution functions on coarse and fine meshes. As the refinement/coarsening criteria, we use the value $\phi(1 - \phi)$ calculated from the phase-field variable and the absolute value of the signed distance function of solid $|s|$. The block checks the maximum value of $\phi(1 - \phi)$ and the minimum value of $|s|$ in itself and the neighbouring blocks. When the maximum value of $\phi(1 - \phi)$ is larger than 0.09 or the minimum value of $|s|$ is smaller than $2\Delta x$, the block creates eight new child blocks. If the eight child blocks do not satisfy the criteria, their parent becomes the new block. For both

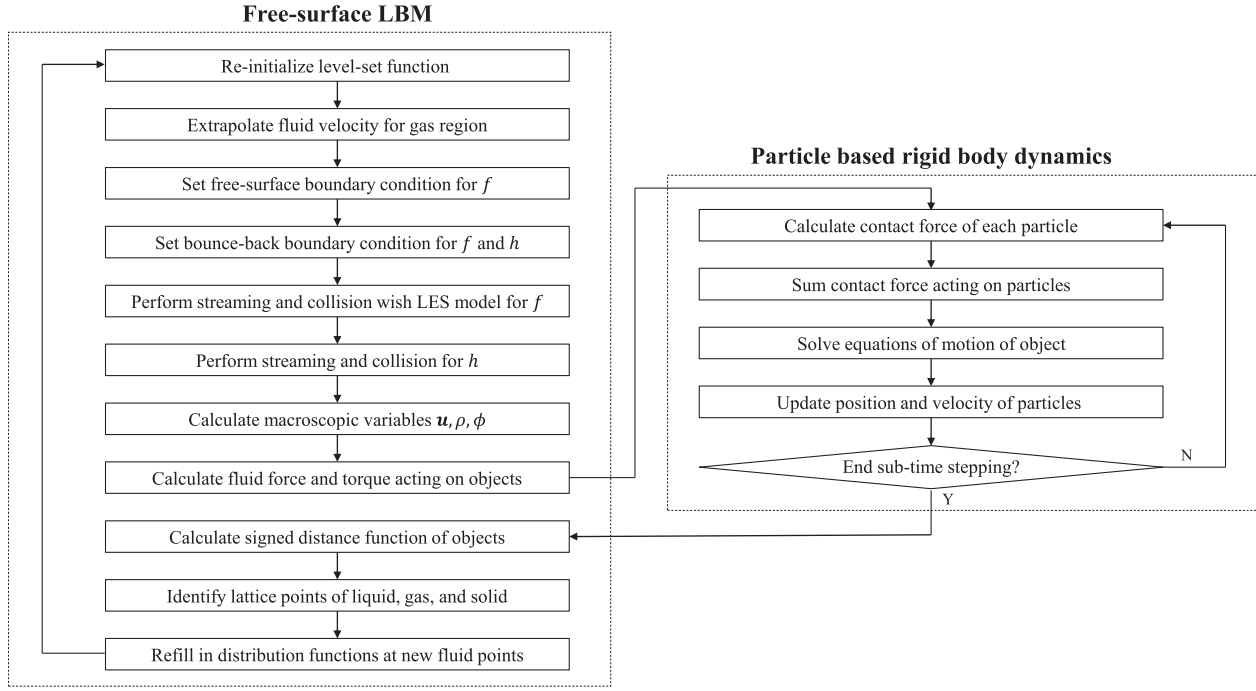


Figure 3. Flowchart of coupling simulation of free-surface LBM and particle-based rigid body dynamics.

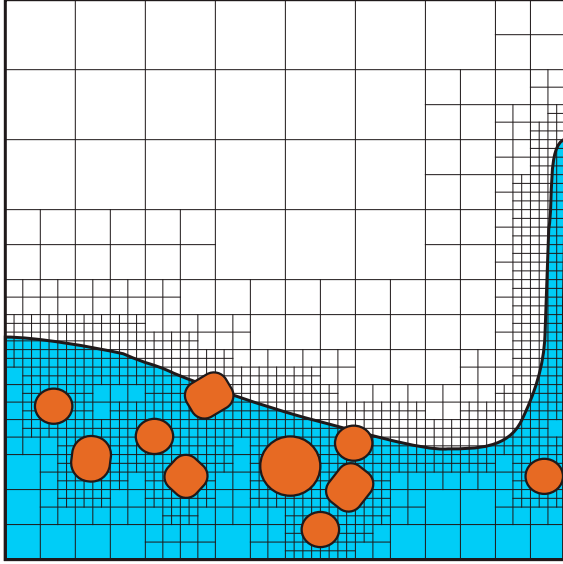


Figure 4. Schematic drawing of AMR method assigning high-resolution blocks to free surfaces and objects.

refinement and coarsening the distribution functions f and h of the new block are calculated by the trilinear interpolation. A rescaling of the non-equilibrium component of the velocity distribution function is not applied in the grid refinement/coarsening process. The mesh refinement/coarsening process is performed once every 256 steps. This roughly corresponds to the time required for free surfaces or solid objects to move a distance of one finest block at a reference speed.

We use multiple GPU computing with the dynamic domain decomposition method to run free surface flow simulations with our solver. To reduce the system dependency of the code, MPI parallelization is implemented by the Flat-MPI that allocates one GPU to one MPI process. The computational domain is decomposed into sub-domains, each assigned to a different MPI rank with a corresponding GPU. In our implementation, the host CPU manages the tree data structure and the partitioning while the GPU handles stencil and particle computations. Dynamic load balancing is easily and quickly performed by mapping three-dimensional blocks to one dimension using the Morton curve (Campbell et al., 2003). Each subdomain has the same number of blocks. Since the computational cost of each block is different, the computational load on each GPU is not uniform, but the memory usage of each GPU can be made nearly equal in this way. Since the memory capacity of NVIDIA GPU Tesla P100 used in this study is limited to 16 GB, it is important to avoid allocating more blocks than the memory capacity. Details of multiple GPU implementations of the lattice Boltzmann method using the octree AMR are described in (Watanabe & Aoki, 2021). Since the number of DEM particles representing objects is much smaller than the number of LBM grid points, MPI parallelization for the particle computations is not implemented and each GPU redundantly calculates all particles.

To evaluate the parallel performance of the proposed solver, we measured the weak scalability from 4 to 64 GPUs on the TSUBAME3.0 supercomputer at Tokyo

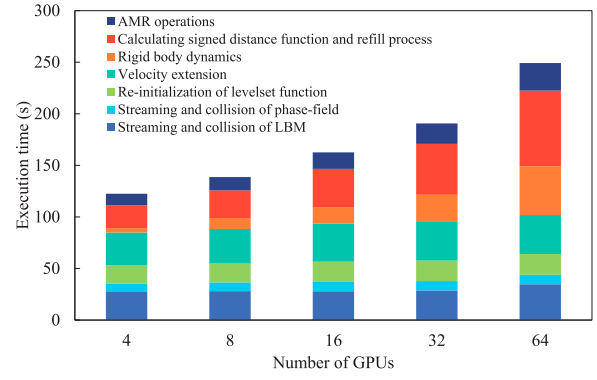
Table 1. Conditions and results of weak scaling study.

# GPUs	# lattice points	# objects	# particles	Performance (MLUPS)	Efficiency
4	62,291,968	256	98,816	229	1
8	124,583,936	512	197,632	405	0.883
16	249,167,872	1,024	395,264	691	0.754
32	498,335,744	2,048	790,528	1,179	0.643
64	996,671,488	4,096	1,581,056	1,804	0.492

Institute of Technology. Compute nodes of TSUBAME3.0 are equipped with two sockets with a 2.4-GHz 14-core Intel Xeon E5-2680 V4 and four GPUs of NVIDIA Tesla P100 with NVLink. The calculation conditions are shown in Table 1. Each GPU is allocated a cube computation domain of 1 m length represented by one tree structure. A water surface was set at a height of 0.8 m, and 64 cube objects of 0.05 m per side were floated. Each object was modelled with 386 spherical particles. The depth of the tree structure was set as 2 to 6, and fine meshes were assigned to the area near the water surface and the objects. There were $8 \times 8 \times 8$ grid points per block. The finest mesh size was 1.953 mm, which corresponds to 512 lattice points per the edge of the cube domain. The number of lattice points per GPU was 15,572,992. As the number of GPUs increased, the computational domain was extended in two dimensions (x and y axes) other than the direction of gravity. We recorded the time taken to run 512 iterations on the finest grid, including two remeshes. The number of iterations of level set reinitialization and velocity extension was both 15. The number of sub-iterations of rigid body calculations was set to 10.

Table 1 shows the weak scaling results in terms of performance and efficiency. The performance was evaluated by MLUPS (Mega Lattice Update Per Second), which is commonly used in LBM studies and indicates the number of grid points updated in one second. Performance increased as the number of GPUs increased, while parallelization efficiency decreased, which was not perfect for weak scaling. Although the weak scaling performance of the free-surface LBM alone in the previous study (Watanabe & Aoki, 2021) was also not good, the performance was even worse in the present coupled free-surface flows and rigid-body calculations.

Figure 5 shows the breakdown of the execution time of the weak scaling study. The execution time of subroutines related to the LBM and the phase-field calculations had not increased significantly with the increase in the number of GPUs. The execution time for the rigid-body calculations in orange also increased as the number of GPUs increased. This is because the rigid-body calculations are not parallelized for multiple GPUs. For 32 GPUs with 2048 objects, the rigid-body analysis time was 9.8%

**Figure 5.** Breakdown of computational time of weak scaling results.

of the total time. Since the number of objects in applications shown in the next sections is 112 at most, the overhead is considered small even without parallelization of the subroutine for the rigid-body computation. The execution time for the calculation of the solid signed distance function and the lattice refilling process increased with the number of GPUs, i.e. with the number of objects. The reason is that the search time for objects in the neighbourhood of each lattice point increases with the number of objects. The time required to generate the AMR mesh also increased with the number of GPUs. This is due to the fact that non-parallel mesh generation increases the execution time as the tree structure grows.

3. Tsunami flow with driftwood

For validation, we conducted an experiment on how driftwood were carried away by a tsunami flow using the ship model basin at Kobe university and then compared these results with the simulation results. The setup of the experiment and the simulation is shown in Figure 6. A stage with a length of 10 m and a height of 1 m was placed at the starting end of the water tank, and water with a depth of 0.4 m was initially set on the stage. The stage was connected to the floor of the water tank by a slope with a length of 6 m. When the experiment or simulation was started, the gate was opened at 0.119 m/s, and the water began to flow due to gravity. The water density was $\rho = 1000 \text{ kg/m}^3$, the kinematic viscosity was $\nu = 1.0 \times 10^{-6} \text{ m}^2/\text{s}$, and the gravity was $g = 9.8 \text{ m/s}^2$. Ten wood logs were placed at 50 cm intervals, 15 m from the gate. The size of the wood logs was $10 \text{ cm} \times 150 \text{ cm} \times 10 \text{ cm}$, and the density was 540 kg/m^3 . Non-slip boundary conditions were set on the walls, the gate, and the wood logs. We used 1 cm grid spacing and a uniform mesh of $4000 \times 600 \times 200$ lattice points. We calculated up to $t = 15 \text{ s}$, with the discrete time $5.0 \times 10^{-5} \text{ s}$.

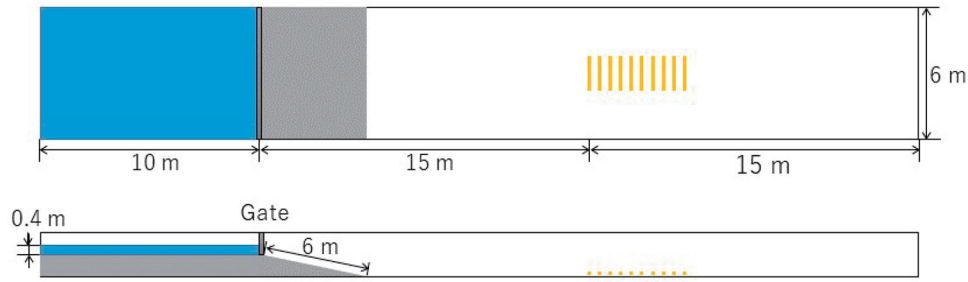


Figure 6. Setup of simulation and experiment for tsunami flow with driftwood.

For each parameter used in the DEM calculation, the discrete times was 5.0×10^{-6} s, the friction coefficient was 0.3, the Poisson's ratio of 0.33 and the coefficient of restitution was 0.25. Young's modulus was set such that Δt^{DEM} is 1/20 of the natural period of the spring-mass system in Equation (52). The simulation was conducted on 32 GPUs of NVIDIA Tesla P100 on the TSUBAME3.0 supercomputer.

Snapshots of the experiment and the simulation at four physical time instances are shown in Figure 7. It was confirmed that the front position of the tsunami flow and the motion of wood logs were in good agreement between the experiment and the simulation. Figure 8 shows the time history of the water height near the wall at 15 m from the gate. The arrival time of the tsunami flow and the trend of temporal changes in water height was consistent between the experiment and the simulation. The water height changes abruptly around $t = 5$ s, when the wave-front passes the measurement point, due to the torn water surface at the wave-front in the simulation. In Figure 9, the positions of object No. 1 (nearest from the gate) and object No. 10 (farthest from the gate) are shown. For both objects No. 1 and No. 10, the times when they started to be carried away by the tsunami flow were in good agreement between the experiment and the simulation. From these results, we consider that our simulation method can calculate the tsunami flow with drifting objects with sufficient accuracy. Figure 10 shows the time history of the error rate from the initial value of the total liquid mass. The mass was not exactly conserved, with errors of about 0.5% from the initial state. The proposed method solves the conservative Allen-Cahn equation, but mass conservation is not satisfied due to the moving boundary. Before the water reached the wood logs, mass was not conserved due to the gate motion. Once the wood logs began to be drifted by the water, the mass error increased more due to the movement of the wood logs. Although the proposed scheme has a liquid mass error due to the motion of the objects, the error rate is sufficiently small that it does not significantly affect the calculation results.

4. Tsunami flow passing through a tide protection forest

As an application, we use the proposed free surface model to compute a tsunami flow against a tide protection forest. A tide protection forest generally refers to a forest constructed on the coast for a purpose of reducing damage from disasters such as tsunamis and storm surges. Old tsunami reports and recent studies (Harada & Imaura, 2005; Shuto, 1987; Tanaka et al., 2007) show the following effects of tide protection forests for tsunami disasters:

- A tide protection forest acts as a hydraulic resistance on tsunami. Forest resistance can reduce tsunami energy, inundation depth, and inundation area. A tsunami passing through a tide protection forest weakens, and the damage of the tsunami behind the forest decreases.
- Trees of a tide protection forest stop floating objects in a tsunami such as boats, ships, driftwood, and debris. As a result, tide protection forests can reduce the secondary disaster caused by the impacts of floating objects on houses and buildings behind the forest.

The tide protection forests have known issues as well (Shuto, 1987):

- A tide protection forest cannot completely prevent tsunami disasters, and tsunami floods and still can damage an area behind the forest.
- A more serious problem of a tide protection forest is that if a huge tsunami hits the forest, the trees could be destroyed and carried with the flood. The drifted trees could cause secondary damage by hitting structures and buildings. This suggests that tide protection forests may increase the tsunami damage to urban areas behind the forest.

Thus, tide protection forests have advantages and disadvantages in case of a tsunami.

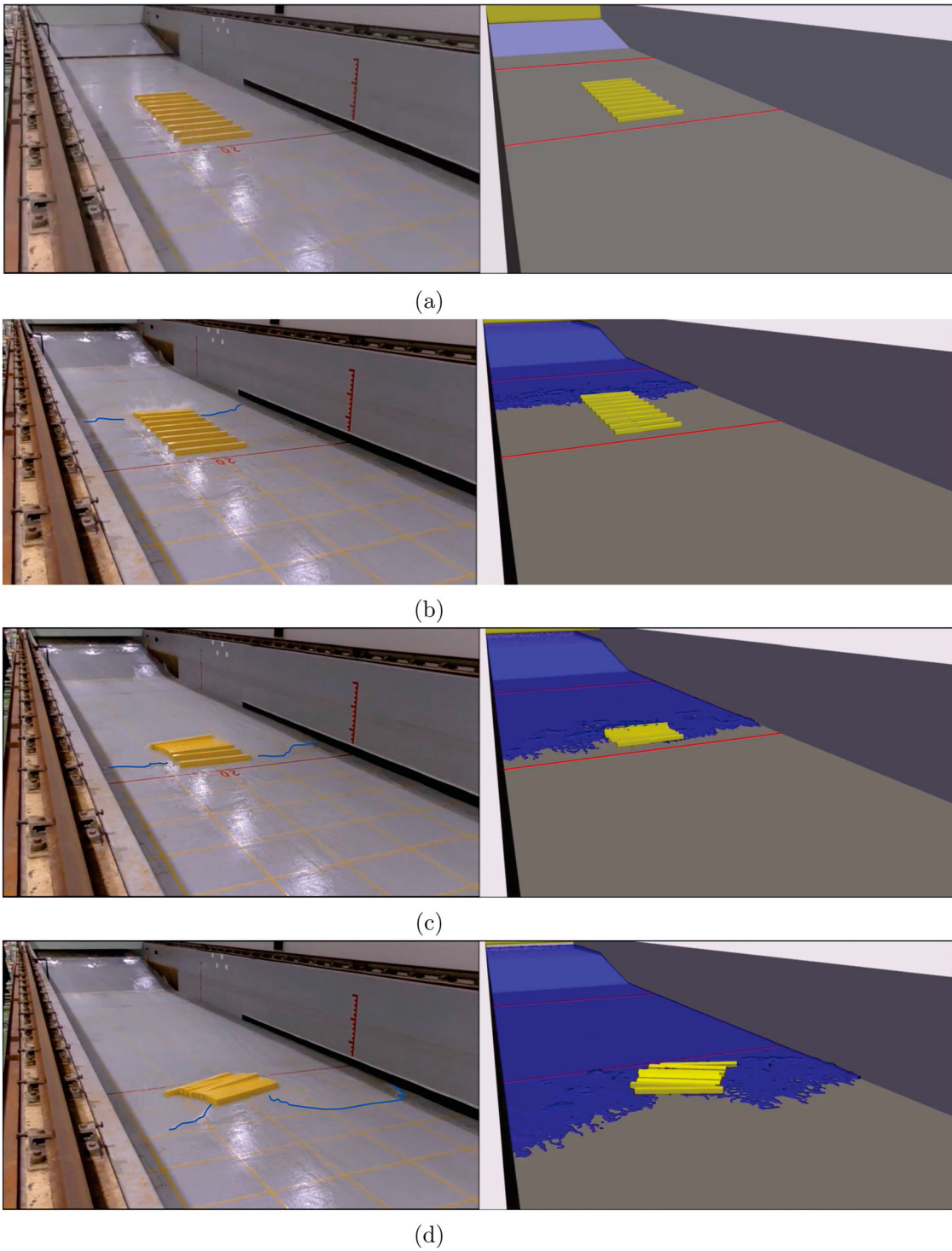


Figure 7. Snapshots of experiment (left) and simulation (right) of tsunami flow with driftwood. Wet/dry interfaces of experiment are highlighted with blue lines. Note that blue lines are not drawn in areas where wet/dry interfaces are not clearly visible in camera images. (a) $t = 0.0$ s, (b) $t = 5.0$ s, (c) $t = 6.0$ s and (d) $t = 7.0$ s.

To evaluate the prevention effects of a forest against a tsunami and the damage caused by the tsunami to the forests, we conducted two experiments and simulations. In Section 4.1, we show that a tsunami flow without

debris passes through a tide protection forest. We measure the water level around the forest in the experiment and the simulation to evaluate the energy reduction of the tsunami due to the forest. We also evaluate the fluid force

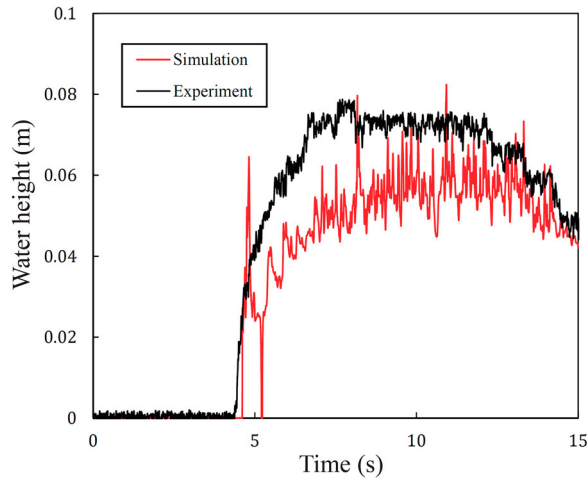


Figure 8. Time history of water height.

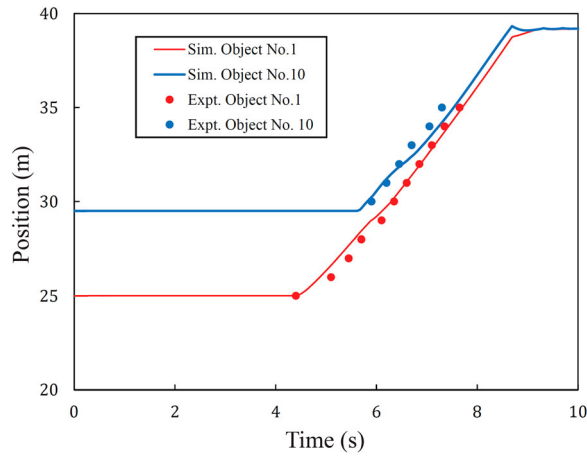


Figure 9. Comparison of experimental and simulated driftwood motion. Object No. 1 and No. 10 mean nearest and farthest object from gate, respectively.

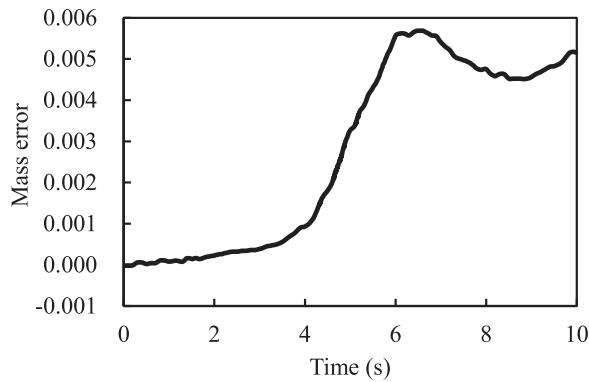


Figure 10. Liquid mass error against time.

acting on each tree in the forest. In Section 4.2, we calculate tsunamis with driftwood, and confirm the function for stopping debris based on the density of trees and the size of driftwood.

The simulations were compared with experiments conducted in the water tank of $15.6 \text{ m} \times 0.6 \text{ m} \times 0.6 \text{ m}$ at the Hachinohe Institute of Technology. Cylindrical poles with a diameter of 50 mm and a height of 400 mm model the tide protection forests. In the simulations, the finest grids were assigned near the free-surface, poles, and wood logs. The depth of the AMR tree was set from 1 to 5. Due to the long computational domain in the stream-wise direction, 26 tree structures were used. Each AMR block had a uniform grid of $8 \times 8 \times 8$. The sizes of the coarsest and finest grid were 37.5 and 2.343 mm, corresponding to a uniform grid of $416 \times 16 \times 16$ and $6656 \times 256 \times 256$, respectively. The time step of the finest grid was $2.3 \times 10^{-5} \text{ s}$. The solid surface uses non-slip boundary conditions. The water density was $\rho = 1000 \text{ kg/m}^3$, the kinematic viscosity was $\nu = 1.0 \times 10^{-6} \text{ m}^2/\text{s}$, and the gravity was $g = 9.8 \text{ m/s}^2$. These simulations used 32 GPUs of Tesla P100 on the TSUBAME 3.0 at the Tokyo Institute of Technology.

4.1. Function to reduce tsunami inundation height

In this section, we calculated a tsunami flow passing through a tide protection forest and then evaluated the effects of the forest on the water height and the fluid force acting on each pole. As shown in Figure 11, a water column of $5.5 \text{ m} \times 0.6 \text{ m} \times 0.3 \text{ m}$ was set initially in a region separated by a gate. The water began to flow by gravity, and the wave passed through the forest modelled by 45 poles placed as shown in Figure 11. The diameter of the cylindrical poles was 0.05 m. The first poles were located 10.93 m from one end of the computational domain, i.e. 5.43 m from the gate.

Snapshots of the simulation zooming around near the tide protection forest are shown in Figure 12. The water level gradually decreases from upstream to downstream around the tide protection forest. The experimental result under the same conditions as the simulation is shown in Figure 13. Similar to the simulation results, the water level in the upstream region of the poles tended to be higher than that in the downstream area. Figure 14 shows a top view of the free surface near the poles. The bow wave in front of the cylinders and the free surface disturbance behind the cylinders can be observed, and they interfere with each other generated by the 45 cylinders. Figure 15 shows the time history of the water height at three measurement points: the upstream, the inside, and the downstream of the poles as shown in Figure 11. We confirmed that the temporal change in the water level trend was a good agreement between the simulation and the experiment. However, the simulation results were a few centimetres different from the experimental results at the measurement points in the upstream and

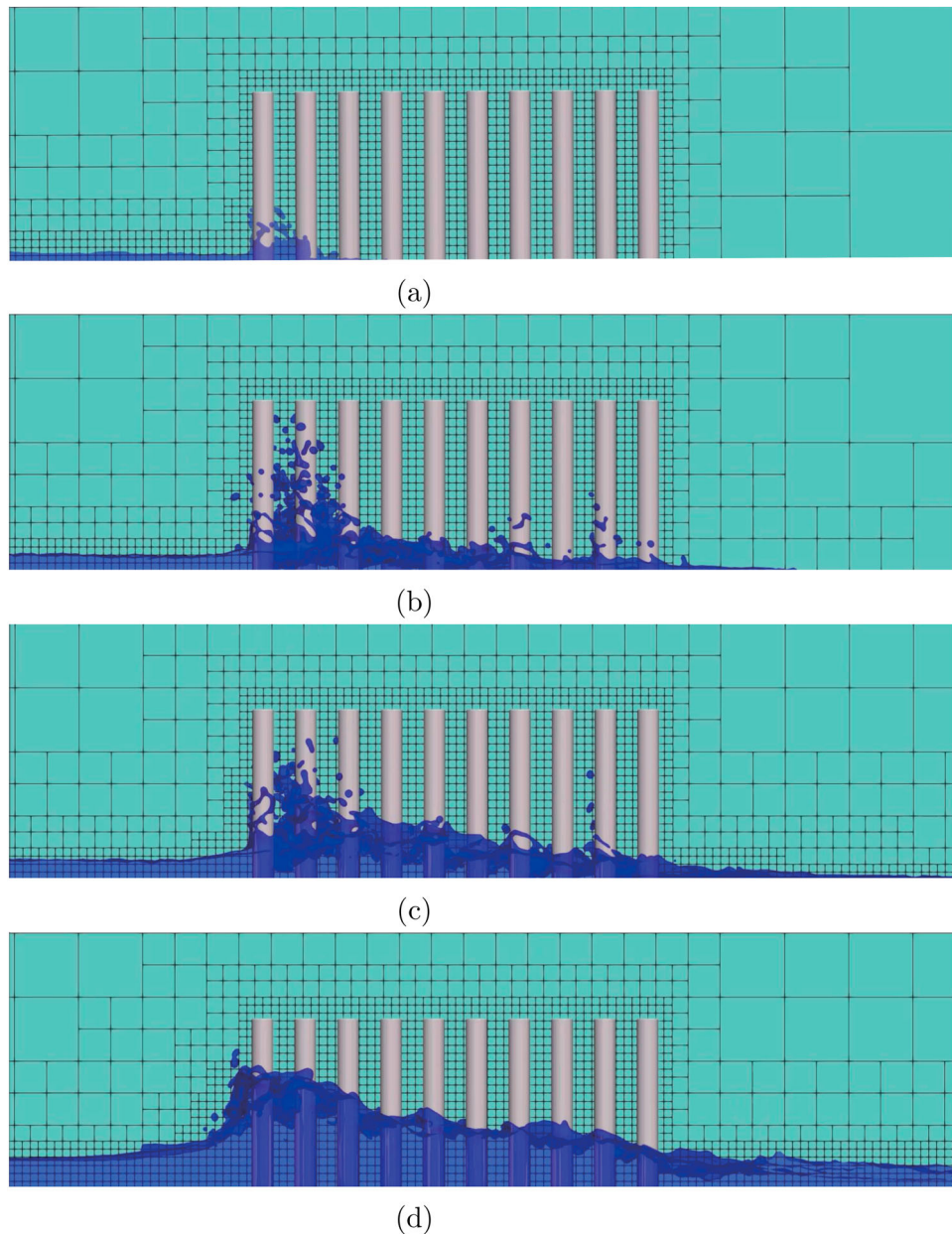


Figure 12. Simulation results of tsunami against tide protection forest. The mesh indicates blocks of a uniform grid of $8 \times 8 \times 8$ points. (a) $t = 2.40$ s, (b) $t = 2.88$ s, (c) $t = 3.36$ s and (d) $t = 4.80$ s.

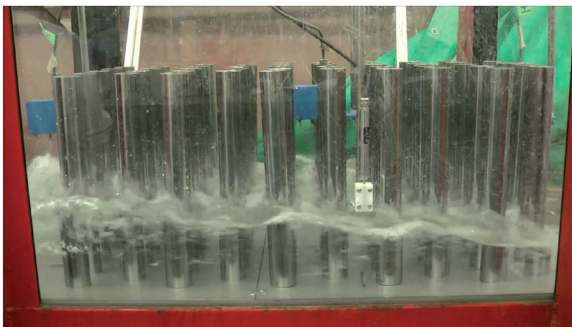


Figure 13. Experimental result of tsunami against tide protection forest at $t = 4.80$ s.

and the poles as shown in Figure 18 and performed simulations like those in Section 4.1. We performed experiments and simulations under the four conditions shown in Table 2. The number of poles was set to 8 or 18, and the length of driftwood was 15 or 20 cm. Figure 19 shows the details of the pole arrangement. The gap between the poles when 18 poles were arranged was narrower than when eight poles were arranged. The number of wood logs was 47, and the initial placements are shown in Figure 20. The density of the wood logs was 580 kg/m^3 which makes them float in water. In the simulations, the woods were assumed as rigid bodies. The parameters of the discrete element method were the Poisson's ratio of



Figure 14. Top view of simulation result of tsunami against tide protection forest.

0.33, the coefficient of restitution of 0.25 and the coefficient of friction of 0.3. The discrete times of the LBM and the DEM were 2.3×10^{-5} s and 2.3×10^{-6} s, respectively. The surface of the wall, poles and driftwood use non-slip conditions.

The simulation results of each case are shown in Figures 21–24. In case 1 and case 2, since the length of the driftwood was as short as 15 cm, almost all the driftwood passed through the cylindrical poles. We observed that the driftwood logs were decelerated by the tide protection forest (the poles) and then passed through the gap

between the poles. In case 1, only one driftwood was trapped between the pole and the wall. In case 2, one driftwood was stopped by the two poles in the second row. The length of the driftwood in case 3 was 20 cm, which is longer than that used in case 1, however, the simulation result was similar, and almost all the driftwood flowed through the gap between poles. In case 4, the poles trapped 25 driftwood which represented about half of the total driftwood. Most of the trapped wood logs were stopped by the four poles at the head of the tide protection forest. In these simulations, we confirmed that

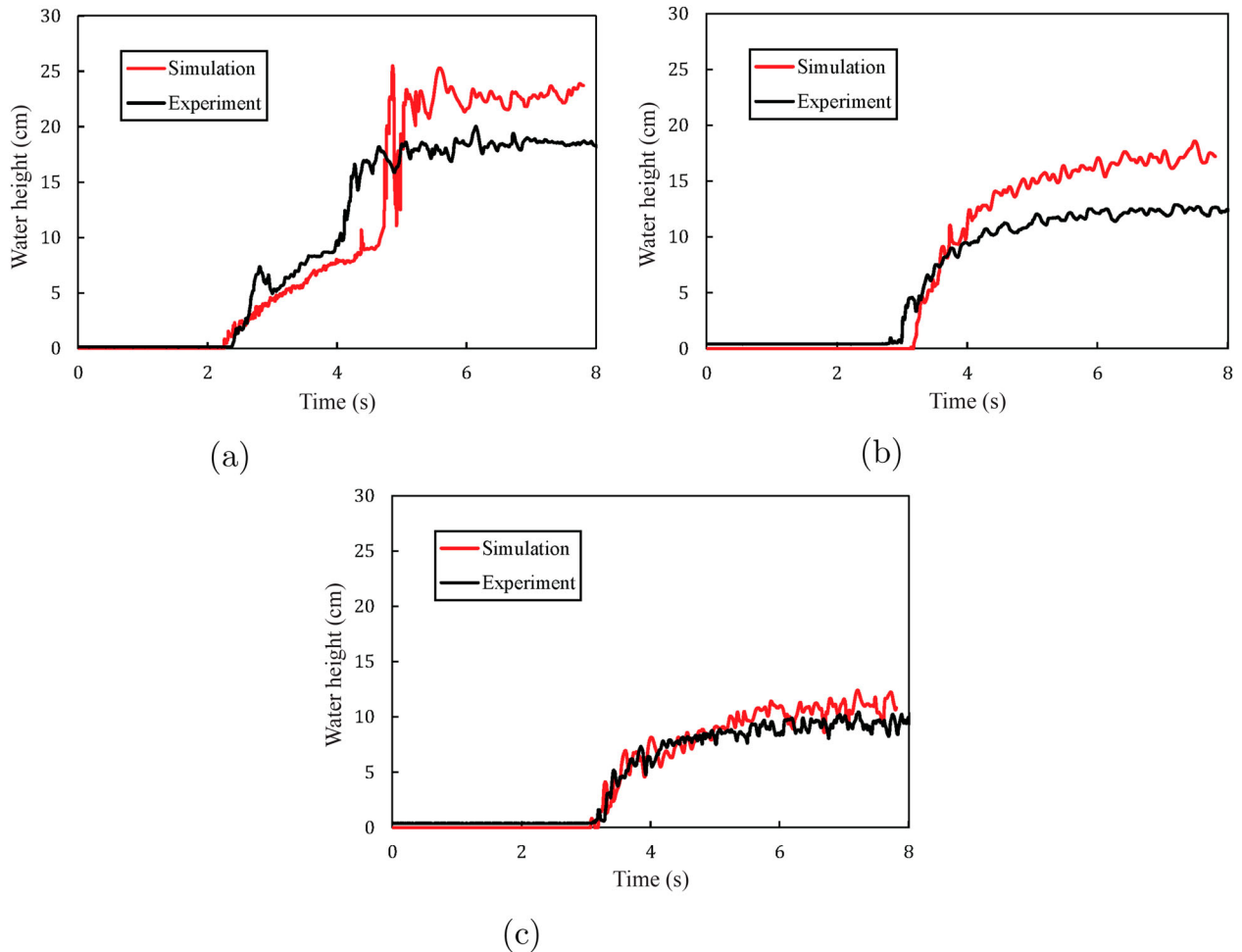


Figure 15. Comparison of simulated and experimental water height. (a) Upstream, (b) inside and (c) downstream.

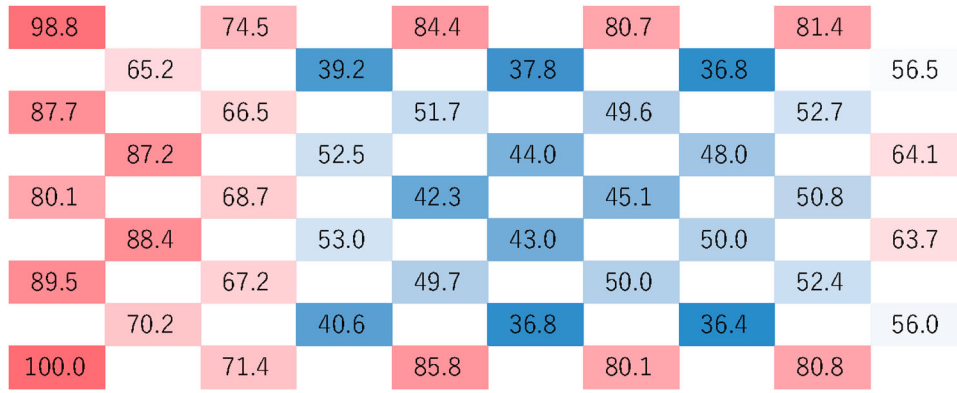


Figure 16. Time-averaged fluid force acting on each pole as percentage of maximum force.

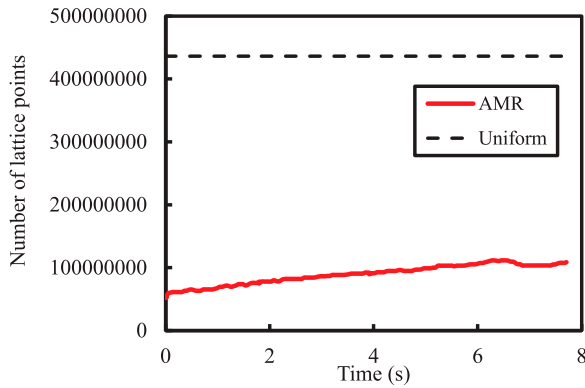


Figure 17. Number of lattice points in computation for tsunami against tide protection forest.

only case 4 could stop the driftwood. However, in case 1, case 2, and case 3, the velocities of the driftwood were at least reduced by the collision of the driftwood with the poles. From these cases, we see that even if a tide protection forest cannot trap all the driftwood, it can slow down the speed of floating objects and reduce the damage to the urban area.

Figure 25 shows a comparison between the simulations and the experiments at $t = 4.8$ s for case 3 and case 4. We observed in the simulations and the experiments, that the number of driftwood logs trapped by the poles strongly depends on the number of poles. In the simulations of case 4, the driftwood logs' arrangement and

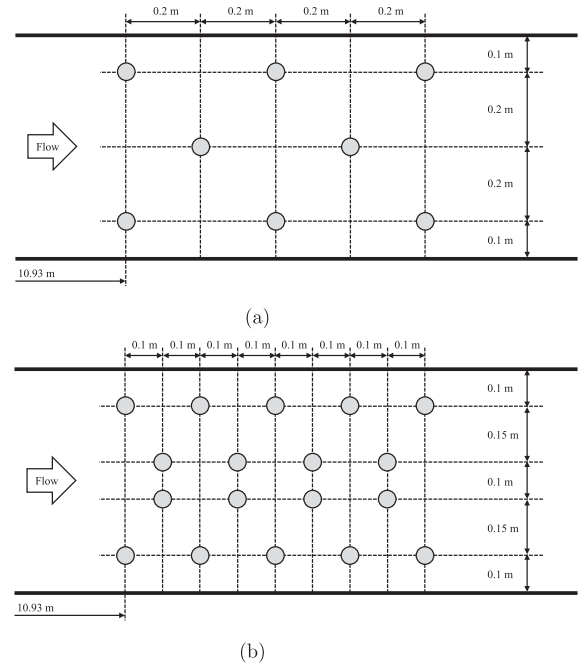


Figure 19. Placement of poles for tsunami with driftwood against tide protection forest. (a) 8 poles and (b) 18 poles.

angle are in very good agreement with the experiment. Table 3 shows the comparison of the number of driftwood logs trapped by the poles between the simulations and the experiments. The experiments were performed

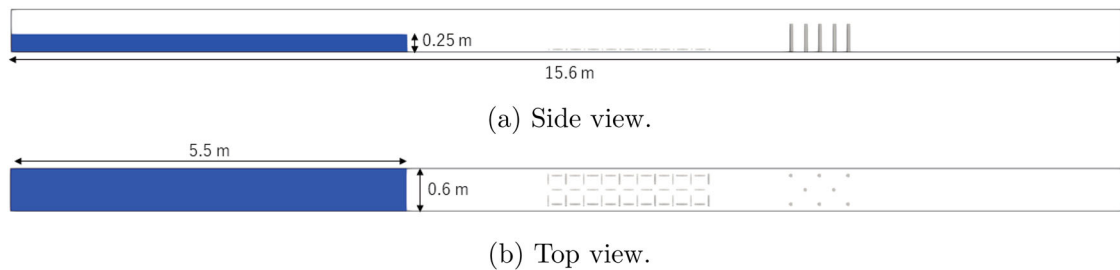


Figure 18. Simulation setup for tsunami with driftwood against tide protection forest. (a) Side view and (b) top view.

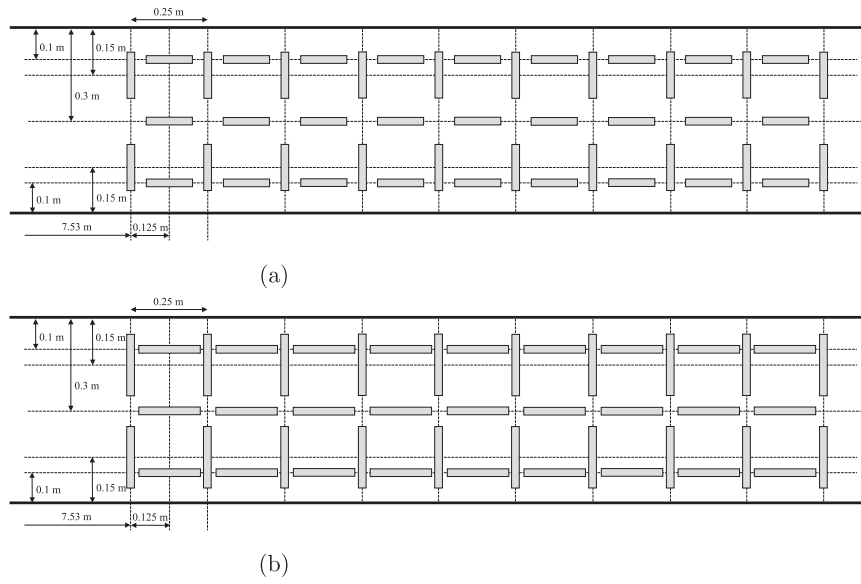


Figure 20. Initial placement of wood logs. (a) $2\text{ cm} \times 2\text{ cm} \times 15\text{ cm}$ wood log. (b) $2\text{ cm} \times 2\text{ cm} \times 20\text{ cm}$ wood log.

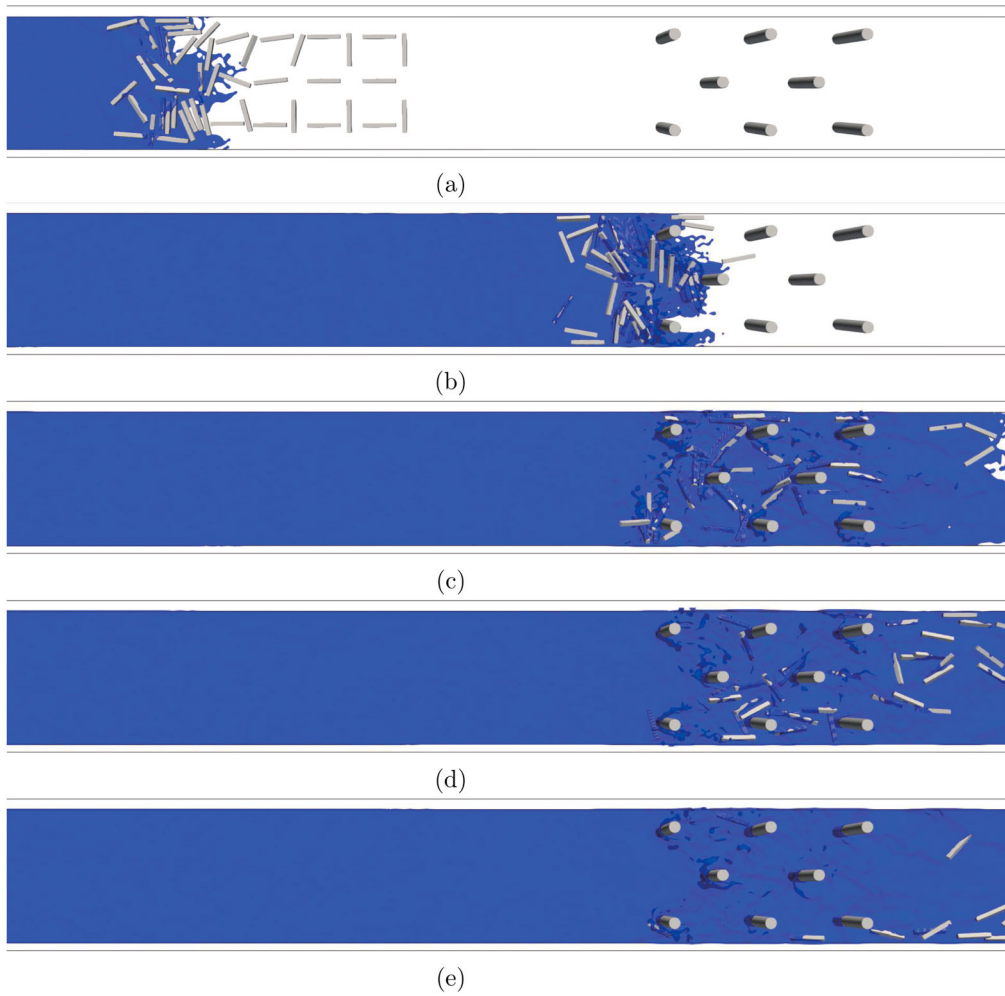


Figure 21. Simulation results of tsunami with driftwood against tide protection forest (case 1). The size of driftwood is $2\text{ cm} \times 2\text{ cm} \times 15\text{ cm}$, and the number of poles is 8. (a) $t = 1.68\text{ s}$, (b) $t = 2.88\text{ s}$, (c) $t = 3.84\text{ s}$, (d) $t = 4.80\text{ s}$ and (e) $t = 6.24\text{ s}$.

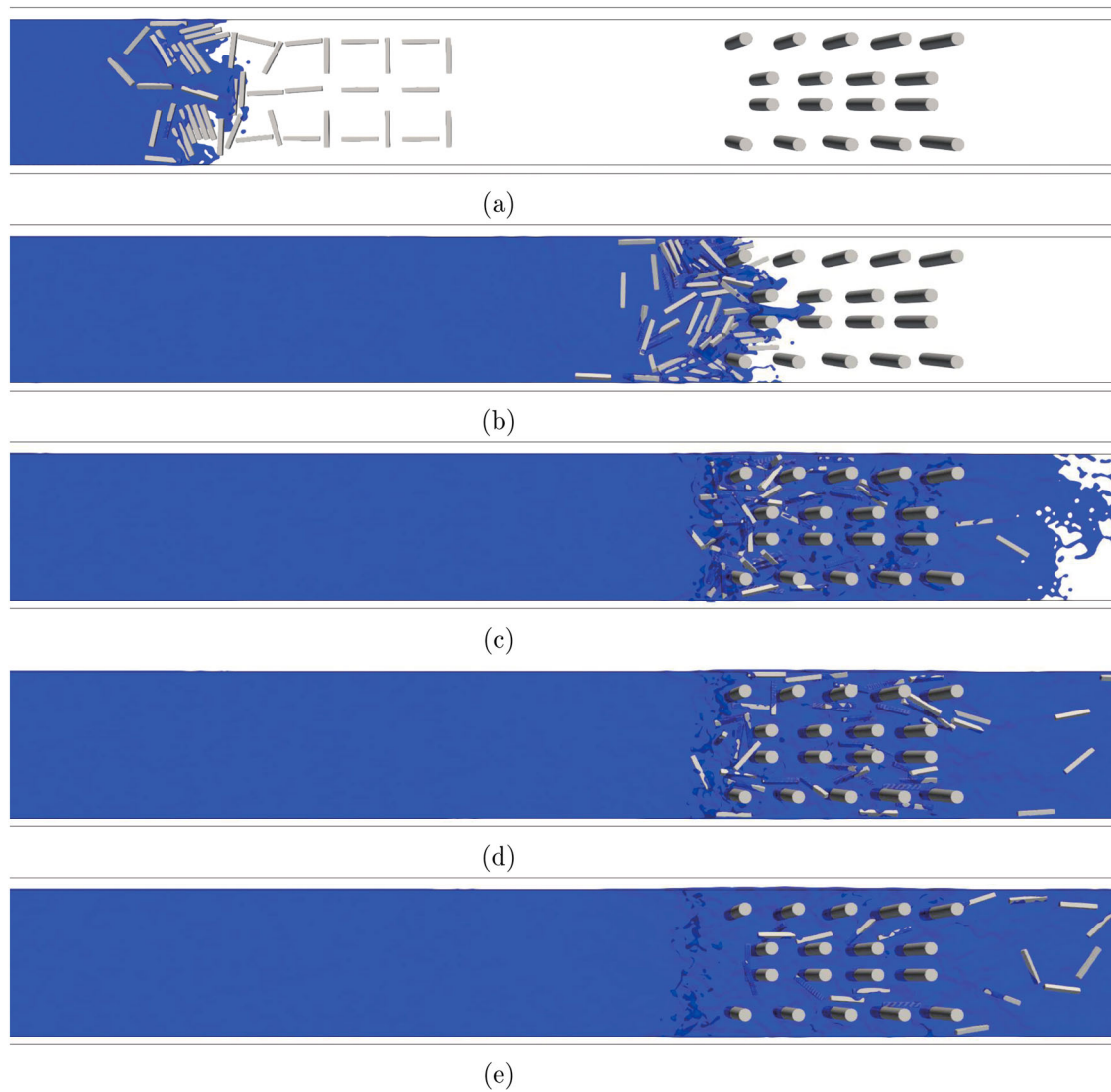


Figure 22. Simulation results of tsunami with driftwood against tide protection forest (case 2). The size of driftwood is $2\text{ cm} \times 2\text{ cm} \times 15\text{ cm}$, and the number of poles is 18. (a) $t = 1.68\text{ s}$, (b) $t = 2.88\text{ s}$, (c) $t = 3.84\text{ s}$, (d) $t = 4.80\text{ s}$ and (e) $t = 6.24\text{ s}$.

Table 2. Conditions for size of driftwood and number of poles.

	Size of driftwood	Number of poles
Case 1	$2\text{ cm} \times 2\text{ cm} \times 15\text{ cm}$	8
Case 2	$2\text{ cm} \times 2\text{ cm} \times 15\text{ cm}$	18
Case 3	$2\text{ cm} \times 2\text{ cm} \times 20\text{ cm}$	8
Case 4	$2\text{ cm} \times 2\text{ cm} \times 20\text{ cm}$	18

five times under each condition. The simulation reproduced the trend that the number of trapped driftwood was small in case 1, case 2 and case 3. More than half of the driftwood logs were stopped by the poles in both the simulation and the experiments of case 4.

In the simulations of each case, we measured the water height at three points shown in Figure 11. The time histories of the water height are shown in Figure 26. In case 1

Table 3. Comparison of number of driftwood stopped by modelled tidal forest in simulations and experiments.

Run	Case 1	Case 2	Case 3	Case 4
Simulation	1	1	2	25
Experiment 1	0	6	1	36
Experiment 2	2	1	8	18
Experiment 3	1	1	7	38
Experiment 4	1	0	4	38
Experiment 5	1	0	6	22

and case 3 with 8 poles, the water height at Point 1 located upstream of the poles was lower than that at Point 3 located downstream. It was not confirmed that the water height tended to decrease from upstream to downstream as in the simulation that did not include driftwood presented in Section 4.1. On the other hand, in case 2 and

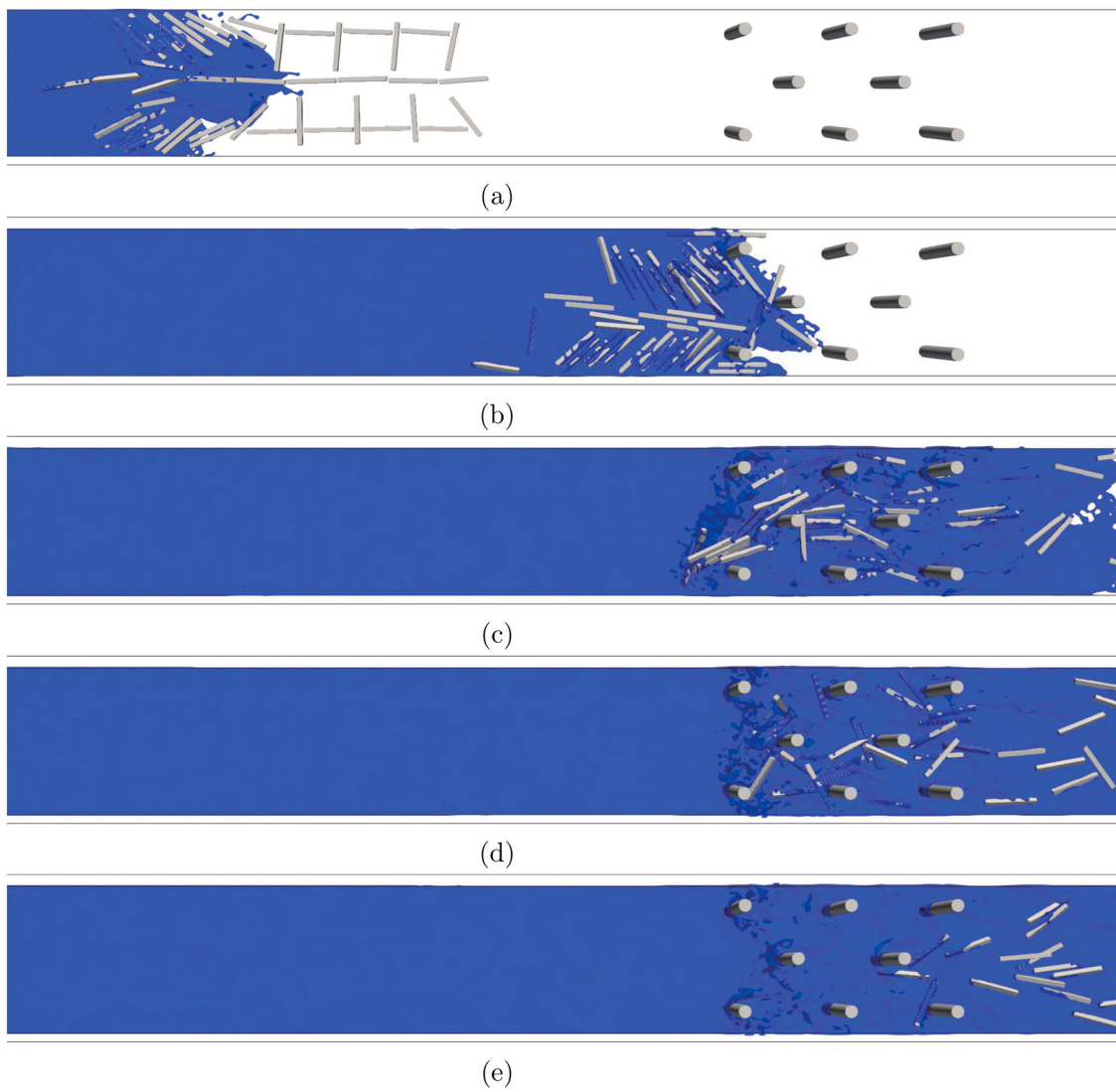


Figure 23. Simulation results of tsunami with driftwood against tide protection forest (case 3). The size of driftwood is $2\text{ cm} \times 2\text{ cm} \times 20\text{ cm}$, and the number of poles is 8. (a) $t = 1.68\text{ s}$, (b) $t = 2.88\text{ s}$, (c) $t = 3.84\text{ s}$, (d) $t = 4.80\text{ s}$ and (e) $t = 6.24\text{ s}$.

case 4 with 18 poles, we can observe that the water height in upstream was higher than that in downstream. From these results, we consider that the density of the poles has an influence on the reduction of the water height downstream of the poles. Comparing case 2 and case 4, the water height at Point 1 (upstream) of case 4 was higher than that of case 2, and at Point 2 (inside), the height of case 4 was lower than that of case 2. In other words, case 4 showed a stronger trend of water level reduction than case 2. We consider that driftwood logs trapped in front of the poles have an effect on reducing the water height downstream.

5. Conclusion

We developed a free-surface flow model for tsunamis with floating debris, based on the cumulant LBM and the particle approach to the rigid body dynamics. The model uses an AMR method and runs parallel on multiple GPUs to reduce the number of the grid points and computational time. These features allow us to compute tsunami simulations with high-resolution grids near the free-surface and floating objects.

We carried out simulations that reproduce tsunami experiments in large water tanks to examine the accuracy

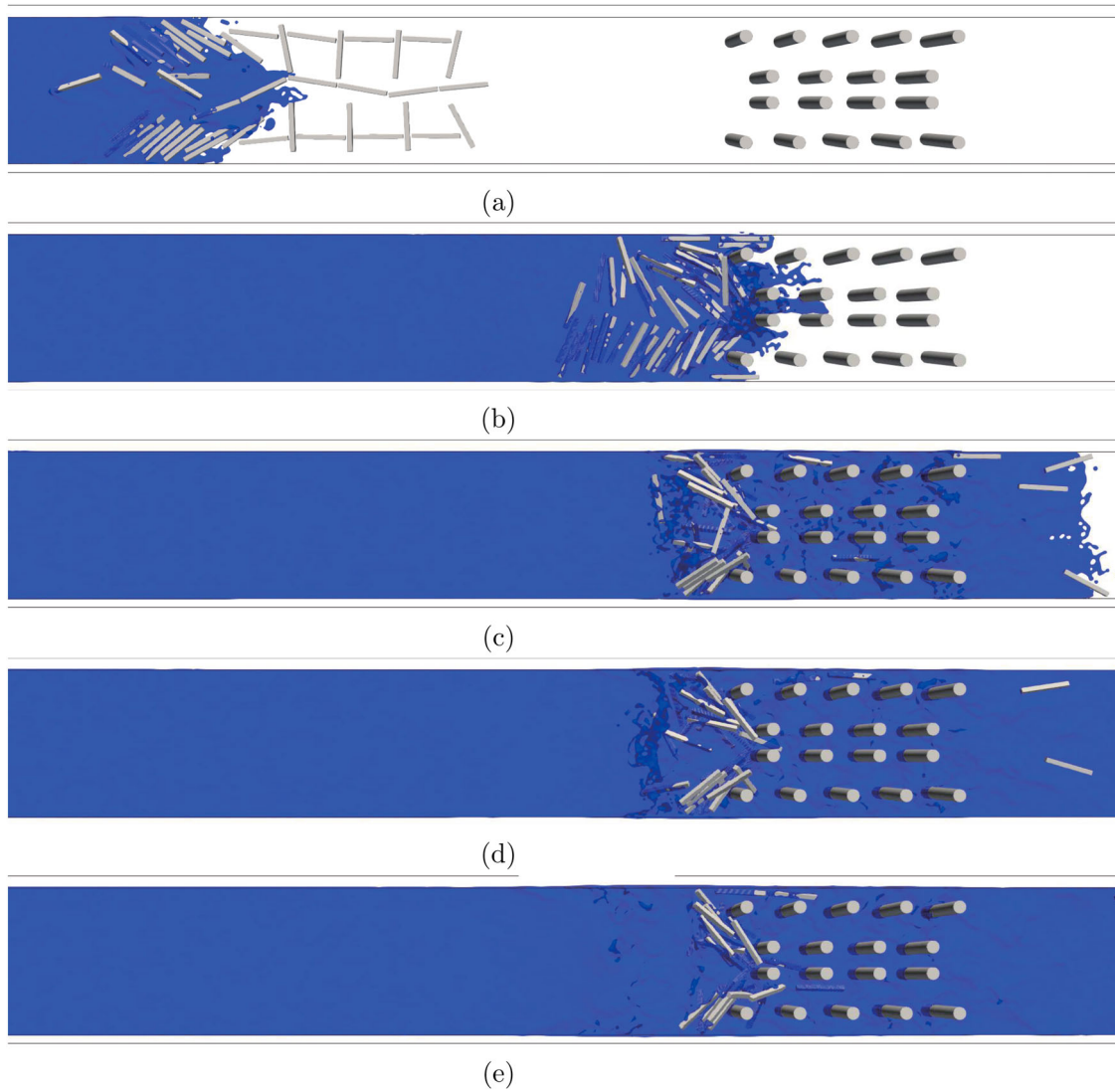


Figure 24. Simulation results of tsunami with driftwood against tide protection forest (case 4). The size of driftwood is $2\text{ cm} \times 2\text{ cm} \times 20\text{ cm}$, and the number of poles is 18. (a) $t = 1.68\text{ s}$, (b) $t = 2.88\text{ s}$, (c) $t = 3.84\text{ s}$, (d) $t = 4.80\text{ s}$ and (e) $t = 6.24\text{ s}$.

and reliability of the proposed model. In Section 3, we calculated a tsunami flow with drifting objects and compared it with the experiment on the ship-model basin at Kobe University. The water height and the motions of the driftwood were in good agreement with the experiment. In Section 4, as an application of the solver, several simulations and experiments were conducted to check the effects of tide protection forests on tsunamis. In both simulations and experiments, it was shown that the tide forest has the effect of reducing the inundation depth of the tsunami behind the forest. The fluid force on each pole was evaluated by the simulation, and it was shown that the front poles receive a larger hydrodynamic

force. In order to evaluate the effect of the tidal forest stopping drifting objects, tsunami flows including 47 driftwood were calculated under four conditions with different driftwood sizes and densities of the forest. The simulation results showed good agreement with the experiments for the number of trapped woods, and the stacked angles. The above results showed that the proposed method could be used to simulate tsunamis with debris.

Although this paper showed simulations of tsunami experiments on the water tanks, we plan to carry out simulations reproducing past tsunami disasters using topographical data in a future study.

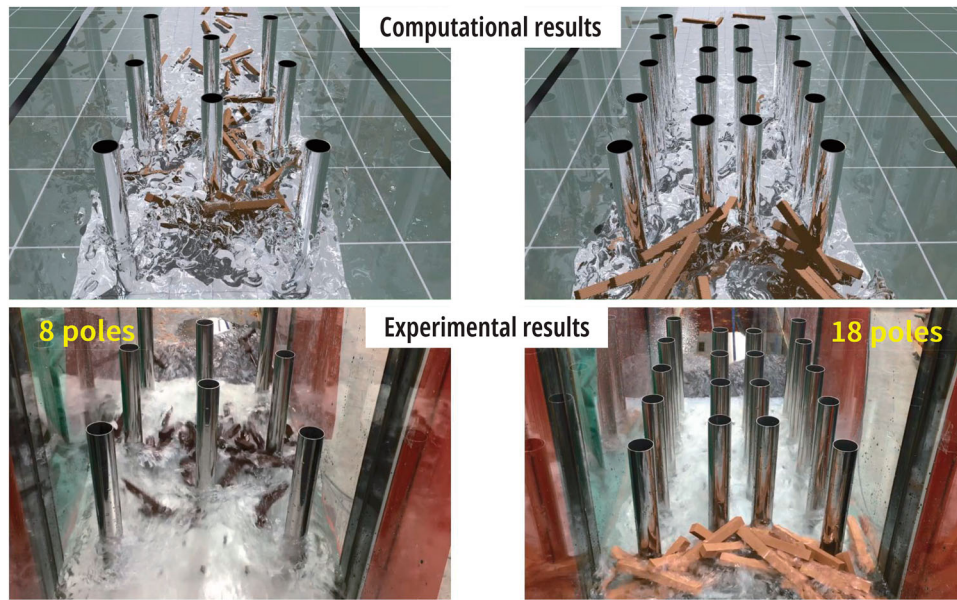


Figure 25. Comparison of computational results and experimental results for tsunami with driftwood against tide protection forest at $t = 4.80$ s (left: case 3, right: case 4).

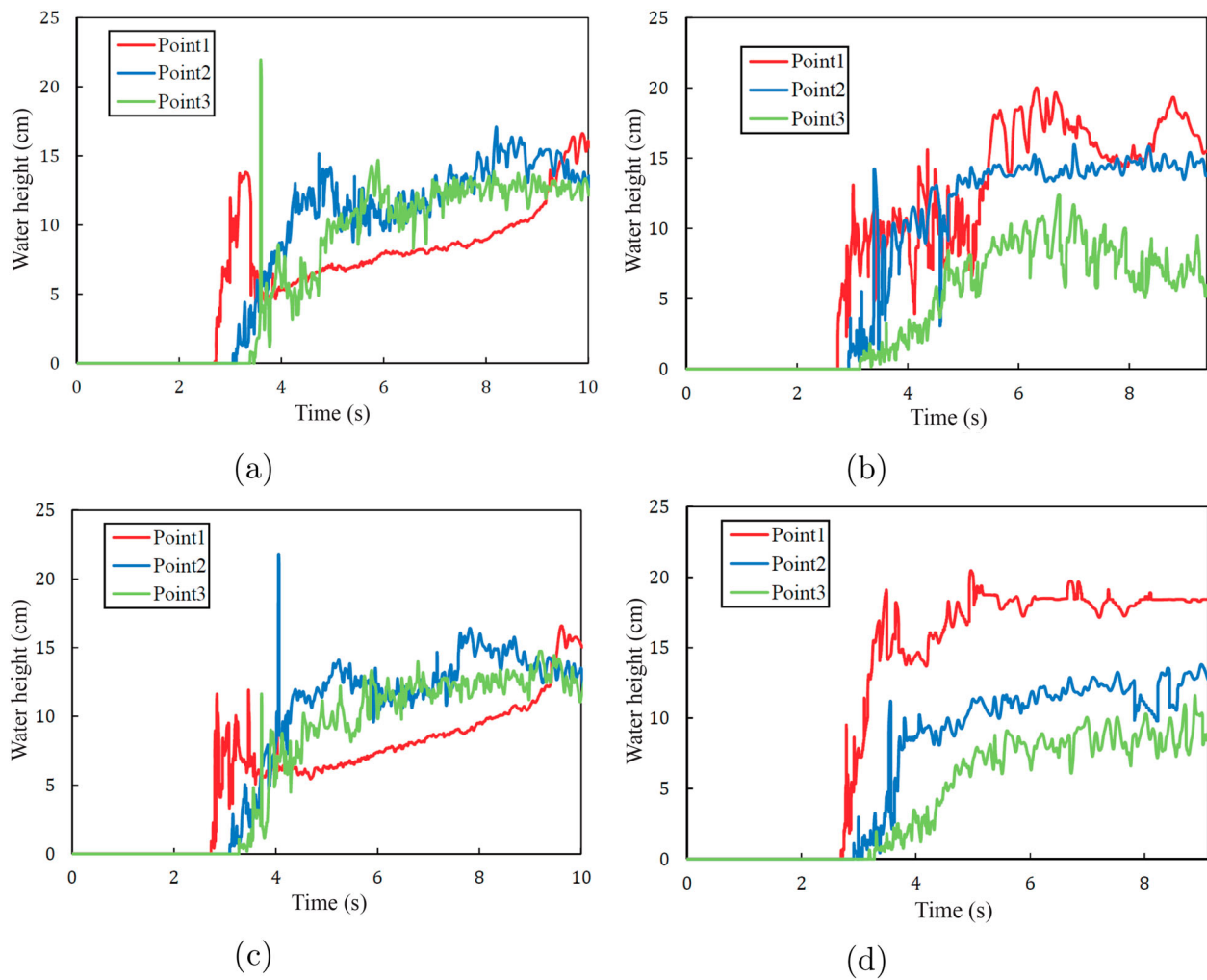


Figure 26. Water height of simulations for tsunami with driftwood against tide protection forest. (a) Case 1, (b) case 2, (c) case 3 and (d) case 4.

Acknowledgments

The numerical calculations were carried out using the computing resources of the TSUBAME3.0 supercomputer at Tokyo Institute of Technology.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was partly supported by KAKENHI, Grant-in-Aid for Scientific Research (S) JP19H05613, (A) JP20H00284 and JP22K14181 from Japan Society for the Promotion of Science (JSPS) and 'Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures (JHPCN)' jh210031 and 'High Performance Computing Infrastructure (HPCI)' hp190130.

ORCID

Seiya Watanabe  <http://orcid.org/0000-0002-9290-3914>

References

- Aidun, C. K., Lu, Y., & Ding, E. J. (1998). Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. *Journal of Fluid Mechanics*, 373, 287–311. <https://doi.org/10.1017/S0022112098002493>
- Bell, N., Yu, Y., & Mucha, P. J. (2005). Particle-based simulation of granular materials. In *Proceedings of the 2005 ACM Siggraph/Eurographics Symposium on Computer Animation* (pp. 77–86). Association for Computing Machinery. <https://doi.org/10.1145/1073368.1073379>
- Bernaschi, M., Bisson, M., Endo, T., Matsuoka, S., Fatica, M., & Melchionna, S. (2011). Petaflop biofluidics simulations on a two million-core system. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. Association for Computing Machinery. <https://doi.org/10.1145/2063384.2063389>
- Bhatnagar, P. L., Gross, E. P., & Krook, M. (1954). A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Reviews*, 94(3), 511–525. <https://doi.org/10.1103/PhysRev.94.511>
- Bouzidi, M., Firdaouss, M., & Lallemand, P. (2001). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11), 3452–3459. <https://doi.org/10.1063/1.1399290>
- Butz, A. R. (1968). Space filling curves and mathematical programming. *Information and Control*, 12(4), 314–330. [https://doi.org/10.1016/S0019-9958\(68\)90367-7](https://doi.org/10.1016/S0019-9958(68)90367-7)
- Campbell, P. M., Devine, K. D., Flaherty, J. E., Gervasio, L. G., & Teresco, J. D. (2003). *Dynamic octree load balancing using space-filling curves* (Tech. Rep. No. CS-03-01). Williams College Department of Computer Science.
- Cundall, P. A., & Strack, O. D. (1979). A discrete numerical model for granular assemblies. *Geotechnique*, 29(1), 47–65. <https://doi.org/10.1680/geot.1979.29.1.47>
- d'Humières, D. (2002). Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 360(1792), 437–451. <https://doi.org/10.1098/rsta.2001.0955>
- Geier, M., Fakhari, A., & Lee, T. (2015). Conservative phase-field lattice Boltzmann model for interface tracking equation. *Physical Review E*, 91(6), 063309. <https://doi.org/10.1103/PhysRevE.91.063309>
- Geier, M., Schönherr, M., Pasquali, A., & Krafczyk, M. (2015). The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Computers & Mathematics with Applications*, 70(4), 507–547. <https://doi.org/10.1016/j.camwa.2015.05.001>
- Harada, K., & Imamura, F. (2005). Effects of coastal forest on tsunami hazard mitigation: A preliminary investigation. *Advances in Natural and Technological Hazards Research*, 23, 279–292. https://doi.org/10.1007/1-4020-3331-1_17
- Janssen, C., & Krafczyk, M. (2010). A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids. *Computers & Mathematics with Applications*, 59(7), 2215–2235. <https://doi.org/10.1016/j.camwa.2009.08.064>
- Kobayashi, H. (2005). The subgrid-scale models based on coherent structures for rotating homogeneous turbulence and turbulent channel flow. *Physics of Fluids*, 17(4), 045104. <https://doi.org/10.1063/1.1874212>
- Körner, C., Thies, M., Hofmann, T., Thürey, N., & Rüde, U. (2005). Lattice Boltzmann model for free surface flow for modeling foaming. *Journal of Statistical Physics*, 121(1–2), 179–196. <https://doi.org/10.1007/s10955-005-8879-8>
- Li, Z., Oger, G., & Le Touzé, D. (2022). A partitioned framework for coupling LBM and FEM through an implicit IBM allowing non-conforming time-steps: Application to fluid-structure interaction in biomechanics. *Journal of Computational Physics* 449 110786. <https://doi.org/10.1016/j.jcp.2021.110786>
- Luding, S. (2008). Introduction to discrete element methods: Basic of contact force models and how to perform the micro-macro transition to continuum theory. *European Journal of Environmental and Civil Engineering*, 12(7–8), 785–826. <https://doi.org/10.1080/19648189.2008.9693050>
- MacNeice, P., Olson, K. M., Mobarrey, C., de Fainchtein, R., & Packer, C. (2000). PARAMESH: A parallel adaptive mesh refinement community toolkit. *Computer Physics Communications*, 126(3), 330–354. [https://doi.org/10.1016/S0010-4655\(99\)00501-9](https://doi.org/10.1016/S0010-4655(99)00501-9)
- Murotani, K., Koshizuka, S., Tamai, T., Shibata, K., Mitsume, N., Yoshimura, S., Tanaka, S., Hasegawa, K., Nagai, E., & Fujisawa, T. (2014). Development of hierarchical domain decomposition explicit mps method and application to large-scale tsunami analysis with floating objects. *Journal of Advanced Simulation in Science and Engineering*, 1(1), 16–35. <https://doi.org/10.15748/jasse.1.16>
- Onodera, N., & Ohashi, K. (2016, January). Large-scale free-surface flow simulation using lattice Boltzmann method on multi-GPU clusters. In *VII European Congress on Computational Methods in Applied Sciences and Engineering* (pp. 1294–1304). ECCOMAS Congress 2016. <https://doi.org/10.7712/100016.1886.8019>
- Peng, D., Merriman, B., Osher, S., Zhao, H., & Kang, M. (1999). A PDE-based fast local level set method. *Journal of Computational Physics*, 155(2), 410–438. <https://doi.org/10.1006/jcph.1999.6345>

- Randles, A., Draeger, E. W., Oppelstrup, T., Krauss, L., & Gunnels, J. A. (2015). Massively parallel models of the human circulatory system. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Association for Computing Machinery. <https://doi.org/10.1145/2807591.2807676>
- Sato, K., Kawasaki, K., & Koshimura, S. (2022). A comparative study of the cumulant lattice Boltzmann method in a single-phase free-surface model of violent flows. *Computers & Fluids*, 236, 105303. <https://doi.org/10.1016/j.compfluid.2021.105303>
- Sato, K., & Koshimura, S. (2020). Validation of the MRT-LBM for three-dimensional free-surface flows: An investigation of the weak compressibility in dam-break benchmarks. *Coastal Engineering Journal*, 62(1), 53–68. <https://doi.org/10.1080/21664250.2019.1672124>
- Schönherr, M. (2015). *Towards reliable LES-CFD computations based on advanced LBM models utilizing (Multi-) GPGPU hardware* [Unpublished doctoral dissertation]. Technische Universität Braunschweig.
- Shuto, N. (1987). The effectiveness and limit of tsunami control forests. *Coastal Engineering in Japan*, 30(1), 143–153. <https://doi.org/10.1080/05785634.1987.11924470>
- Tanaka, N., Sasaki, Y., Mowjood, M. I., Jinadasa, K. B., & Homchuen, S. (2007). Coastal vegetation structures and their functions in tsunami protection: Experience of the recent Indian Ocean tsunami. *Landscape and Ecological Engineering*, 3(1), 33–45. <https://doi.org/10.1007/s11355-006-0013-9>
- Tsuzuki, S., & Aoki, T. (2015). A large-scale suspension-flow simulation directly solving fluid-structure interactions with dynamic load balance on a GPU supercomputer. In *High performance computing symposium 2015* (pp. 111–119). Information Processing Society of Japan.
- Tu, T., O'Hallaron, D., & Ghattas, O. (2005). Scalable parallel octree meshing for terascale applications. In *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing* (pp. 4–). IEEE Computer Society. <https://doi.org/10.1109/SC.2005.61>
- Valero-Lara, P. (2016, December). Leveraging the performance of LBM-HPC for large sizes on GPUs using ghost cells. In *Algorithms and architectures for parallel processing* (Vol. 10048, pp. 417–430). Springer International Publishing. https://doi.org/10.1007/978-3-319-49583-5_31
- Wang, H., Yuan, X., Liang, H., Chai, Z., & Shi, B. (2019). A brief review of the phase-field-based lattice Boltzmann method for multiphase flows. *Capillarity*, 2(3), 33–52. <https://doi.org/10.26804/capi>
- Watanabe, S., & Aoki, T. (2021). Large-scale flow simulations using lattice Boltzmann method with AMR following free-surface on multiple GPUs. *Computer Physics Communications*, 264, 107871. <https://doi.org/10.1016/j.cpc.2021.107871>
- Watanabe, S., Fujisaki, S., & Hu, C. (2021). Numerical simulation of dam break flow impact on vertical cylinder by cumulant lattice Boltzmann method. *Journal of Hydrodynamics*, 33(2), 185–194. <https://doi.org/10.1007/s42241-021-0028-6>
- Wen, B., Zhang, C., Tu, Y., Wang, C., & Fang, H. (2014). Galilean invariant fluid–solid interfacial dynamics in lattice Boltzmann simulations. *Journal of Computational Physics*, 266, 161–170. <https://doi.org/10.1016/j.jcp.2014.02.018>
- Wolf-Gladrow, D. A. (2000). *Lattice-gas cellular automata and lattice Boltzmann models: An introduction*. Springer Science & Business Media.