Computers and Mathematics with Applications 65 (2013) 211–229

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

On enhanced non-linear free surface flow simulations with a hybrid LBM–VOF model

Christian F. Janßen [a,*], Stephan T. Grilli [c], Manfred Krafczyk [b]

[a] Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Hamburg, Germany
[b] Institute for Computational Modeling in Civil Engineering, Technische Universität Braunschweig, Braunschweig, Germany
[c] Department of Ocean Engineering, University of Rhode Island, Narragansett, RI, USA

ARTICLE INFO

Keywords:
Free surface
Lattice Boltzmann method
Volume of fluid
PLIC
Potential flow
Plunging breaker

ABSTRACT

In this paper, we present extensions, extensive validations and applications of our previously published hybrid volume-of-fluid-based (VOF) model for the simulation of free-surface flow problems. For the solution of the flow field, the lattice Boltzmann method is used, where the free surface is represented by a VOF approach. The advection equation for the VOF fill level is discretized with a finite volume method, on the basis of a 3D Piecewise Linear Interface Reconstruction (PLIC) algorithm. The model is validated for several standard free surface benchmarks, such as breaking dam scenarios and a free falling jet. Finally, the hybrid algorithm is applied to the simulation of a wave breaking by overturning during shoaling, which is considered to be a demanding test case, especially for VOF solvers. In this case, the flow field is initialized early in the shoaling process with a solitary wave solution from inviscid, irrotational potential flow. The wave breaking process is then simulated with the 3D transient and turbulent LBM–VOF solver. All validation and benchmark tests confirm the accuracy of the proposed hybrid model.

1. Introduction

The lattice Boltzmann method (LBM) has become an increasingly efficient approach for solving a variety of complex fluid dynamics problems, particularly in the field of multiphysics. By contrast with classical flow solvers, which discretize and solve the macroscopic Navier–Stokes (NS) equations on a continuum mechanics basis, the LBM tackles such problems on a mesoscopic scale and represents the fluid as a field of particle distribution functions. Particle motions are found by solving discrete Boltzmann equations. Nevertheless, it can be shown that the LBM essentially solves for the same physics as classical NS models. LBM, however, offers specific advantages regarding algorithmic operations and data locality. The efficiency and accuracy of the LBM method has been demonstrated in many publications. Additionally, the method was shown to be particularly well suited for large-scale simulations on massively parallel computer clusters [1,2] and/or general purpose graphics processing units (GPGPUs; see [3–5]).

For the simulation of free surface flows, the basic LBM bulk scheme has to be extended. Technically speaking, free surface flows are multiphase flows, typically with an air and a water phase. For such multiphase flows, the ability of a numerical model to handle high viscosity ratios and high density ratios is crucial to the solver's accuracy. However, when dealing with wave flows, owing to the high density and high viscosity ratios of air and water, only the denser and more viscous phase (i.e., water) needs to be simulated. The second (less dense) phase (air) is represented by appropriate kinematic and

* Corresponding author.
E-mail addresses: mail@christian-janssen.de (C.F. Janßen), grilli@oce.uri.edu (S.T. Grilli), kraft@irmb.tu-bs.de (M. Krafczyk).

0898-1221/$ – see front matter © 2013 Published by Elsevier Ltd
doi:10.1016/j.camwa.2012.05.012

dynamic boundary conditions at the interface (i.e., here a free surface). Capillary forces can be simulated by adding additional curvature-dependent terms to the dynamic free surface boundary condition.

From a numerical point of view, a free surface represents a moving boundary, hence, leading to a transient computational domain. To calculate the movement of phase interfaces, an additional advection equation has to be solved in the LBM. To do so, the LBM provides the flow field at all times, which then is used for the advection of the phase interface. Various methods exist to deal with phase interfaces. In interface *tracking* methods, the interface location is directly accessible, e.g. by defining marker particles, that are advected with the flow in a Lagrangian manner (Marker And Cell method, MAC, [6,7]). In Smoothed Particle Hydrodynamics methods (SPH, [8–10]) or Arbitrary Lagrangian Eulerian formulations (ALE, [11,12]), the grid for the calculation of the flow field is updated at every time step and follows the free surface. In contrast with this, in interface *capturing* methods, the interface location has to be reconstructed out of one or more additional variables, such as the fill level of cells (VOF, see e.g. [13]), the position of the fluid center of mass (Moment Of Fluid Method, MOF, [14,15]), or the distance of the grid points to the phase interface (Level Set methods, LS, [16]). Also, several approaches have been developed to directly apply an LBM to free-surface problems. Thus, [17] introduced the immiscible lattice Boltzmann (ILB) model, which is technically a multiphase model with an additional anti-diffusion sweep (recoloring step) to prevent mixing between the two phases. In [18], the authors modified the ILB model and removed the second fluid phase, so that the LBM calculation steps occur only at the nodes of the liquid phase instead of solving for the flow fields of both liquid and gas phase. In [19,20], the authors combined the LBM with a VOF method. The cell fill levels are updated based on local fluxes caused by propagating LBM particle distribution functions. A straightforward geometrical representation of the interface is used to keep the interface sharp, and the advection steps apply on interface cells only. The latter work already indicates, that the computational algorithm can be split up into two independent parts: (i) the flow field is calculated with the LBM; (ii) the interface location is updated using a different and independent solver. Hence, flow field calculations and interface advection can be treated independently, making it possible using optimal methods for both purposes. In a similar way, in [21] the LBM was combined with a Lagrangian front-tracking method. The interface is represented via marker particles, which are advected on the basis of the LBM velocity field. Following the same principles, we recently presented a hybrid free surface scheme, which discretizes the advection equation in a classical macroscopic way, rather than in a mesoscopic LB framework [22]. A VOF method computes the dynamics of the interface, but the fluxes are calculated based on the LBM velocity and pressure fields. The details of this hybrid LBM–VOF algorithm already have been discussed and published. However, in the present paper, we extend this hybrid LBM–VOF advection scheme, as the simulation of demanding benchmark test cases required further improvements of the flux calculations. In contrast to our previous work, the flux terms are now evaluated for a complete set of twenty-six neighboring cells, and the time-dependency of the fill levels is considered in the advection step. Moreover, a linear extrapolation of the free surface velocity is used in the dynamic free surface boundary condition, and new boundary conditions for the advection scheme have been introduced. After the basic validation, we simulate a plunging (i.e., overturning) wave breaking during shoaling on a plane slope, and subsequent post-breaking processes, by initializing the hybrid LBM–VOF with a potential flow solution for an incident solitary wave, at an early stage of shoaling. We find that our hybrid LB-VOF-potential flow algorithm provides very good results for all the validation tests, even for the demanding wave breaking test case.

2. The lattice Boltzmann method

The LBM solves for CFD problems on a mesoscopic scale by representing the fluid as a field of particle distribution functions $f(t, \boldsymbol{x}, \boldsymbol{\xi})$. These express the probability to encounter a particle at position $\boldsymbol{x}$ and time t, with velocity $\boldsymbol{\xi}$. The Boltzmann equation (Ludwig Boltzmann, 1872) describes the time-change of such distribution functions f:

$$\frac{Df}{Dt} = \frac{\partial f(t, \boldsymbol{x}, \boldsymbol{\xi})}{\partial t} + \boldsymbol{\xi} \cdot \frac{\partial f(t, \boldsymbol{x}, \boldsymbol{\xi})}{\partial \boldsymbol{x}} = \Omega. \tag{2.1}$$

The left-hand side of this equation is an advection-type expression, while the *collision operator* Ω describes the interactions of particles on a microscopic scale. For continuum flows with low Knudsen numbers, discretized microscopic particle velocities $\boldsymbol{e}_\alpha$ can be introduced to obtain a model with reduced computational cost, leading to a set of discrete Boltzmann equations:

$$\frac{Df_\alpha}{Dt} = \frac{\partial f_\alpha(t, \boldsymbol{x})}{\partial t} + \boldsymbol{e}_\alpha \cdot \frac{\partial f_\alpha(t, \boldsymbol{x})}{\partial \boldsymbol{x}} = \Omega_\alpha. \tag{2.2}$$

In the present work, the D3Q19 model [23] is used for the discretization of the velocity space. This model introduces 19 velocities, 18 velocities pointing to neighboring nodes and the zero velocity, while the space diagonals are omitted: $\boldsymbol{e}_\alpha = \{0,0,0\}, \{\pm c,0,0\}, \{0,\pm c,0\}, \{0,0,\pm c\}, \{\pm c,\pm c,0\}, \{\pm c,0,\pm c\}, \{0,\pm c,\pm c\}, \alpha = 0,\ldots,18$. The velocity magnitude is related to a constant velocity c, which is related to the speed of sound as $c = c_s\sqrt{3}$. A finite difference discretization in space and time is introduced, on a grid such that $c = \Delta x/\Delta t = 1$ (grid spacing Δx, time stepping Δt), which finally leads to the *lattice* Boltzmann equation,

$$f_\alpha(t + \Delta t, \boldsymbol{x} + \boldsymbol{e}_\alpha \Delta t) - f_\alpha(t, \boldsymbol{x}) = \Omega_\alpha, \tag{2.3}$$

which typically is split up into a local *collision* step, which drives the particle distribution functions locally to equilibrium, and a *propagation* step, where the evolved particle distribution function is advected. It has been well-established that solutions of the lattice Boltzmann equation converge to those of the incompressible NS equations, up to errors of $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\mathrm{Ma}^2)$ [24,25], where $\mathrm{Ma} = \frac{|\boldsymbol{u}|}{c_s}$ is the Mach number. Macroscopic hydrodynamic quantities such as density fluctuation ρ and momentum $\rho_0 \boldsymbol{u}$ can be obtained from low order hydrodynamic moments of the distribution functions:

$$\rho = \sum_{\alpha=0}^{18} f_\alpha \quad \text{and} \quad \rho_0 \boldsymbol{u} = \sum_{\alpha=0}^{18} \boldsymbol{e}_\alpha f_\alpha \tag{2.4}$$

where ρ_0 is the reference density. Note that, as the flow field is described by a higher number of degrees of freedom per node than in conventional NS solvers, additional information, such as the spatial derivatives of the flow field (i.e. the stress tensor), can be obtained from the particle distribution functions by a proper design of the collision operator. In a simple approximation for the collision term Ω_i, the particle distribution functions are driven to an equilibrium state with a single relaxation rate (single relaxation time model, SRT, [26]). In the more advanced multi relaxation time (MRT) model [27], the particle distribution functions are transformed into moment space before relaxation. For the following relaxation, different relaxation rates can be used, which increases stability and allows for the development of more accurate boundary conditions [28]. The moments $\mathbf{m} = \mathsf{M} \cdot \mathbf{f}$ are labeled as

$$\mathbf{m} = \left(\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z\right)^T,$$

and denote the following: mass density $m_0 = \rho$; the part of kinetic energy independent of the density $m_1 = e$; the part of kinetic energy square independent of the density and kinetic energy $m_2 = \epsilon$; the momentum $m_{3,5,7} = j_{x,y,z}$; $m_{4,6,8} = q_{x,y,z}$ are related to heat flux; $m_{9,11,13,14,15}$ are related to the symmetric traceless viscous stress tensor; $m_{16,17,18}$ are third-order moments; and $m_{10,12}$ are fourth-order moments. The collision operator for MRT reads:

$$\Omega = \mathsf{M}^{-1} \cdot \mathsf{S} \cdot \left(\mathsf{M} \cdot \mathbf{f} - \mathbf{m}^{eq}\right) \tag{2.5}$$

M denotes the transformation matrix from distribution functions to moments ($\mathbf{m} = \mathsf{M} \cdot \mathbf{f}$ and $\mathbf{f} = \mathsf{M}^{-1} \cdot \mathbf{m}$) and the m_α^{eq} are the equilibrium moments. The diagonal collision matrix $\mathsf{S} = s_{\alpha,\alpha}$ contains the relaxation parameters, which are partly related to the kinematic viscosity ν via the relaxation time τ

$$s_9 = s_{11} = s_{13} = s_{14} = s_{15} = -\frac{\Delta t}{\tau}, \tag{2.6}$$

which is defined as

$$\tau = 3\frac{\nu}{c^2} + \frac{1}{2}\Delta t. \tag{2.7}$$

The remaining relaxation rates can be freely chosen in the range $[-2, 0]$ and tuned to improve stability [29]. The optimal values for these parameters depend on the specific geometry and the initial and boundary conditions of the system. However, reasonable values are given in [27]; here, we choose $s_\alpha = -1.0$ for the remaining parameters.

2.1. Smagorinsky LES

Free surface flows usually occur at high Reynolds numbers in the range 10^5–10^7, well within the turbulent regime. Although LBM can be used to perform the equivalent of Direct NS (DNS) simulations, most practical problems require using grid cells much coarser than the smallest turbulent scales. Hence, in order to capture the turbulent flow structures on the sub-grid scale, a Large Eddy Simulation model (LES; [30]) can be used. In LES, a spatial filter is applied to the velocity field, which should be fine enough that the larger turbulent structures of the flow do not get filtered out. Typically, the grid spacing Δx is used as filter length. Hence, only the dissipative effects of the sub-grid eddies on the large-scale flow structures are modeled with the LES subgrid scheme. These are included in the LBM model through an additional turbulent viscosity ν_T, which is represented by a Smagorinsky model as a function of shear rate. Details on the LES model can be found in [30].

2.2. Boundary conditions and volume forces

In the LBM, by nature, boundary conditions for the distribution functions have to be directly specified at the boundary nodes, which are quite different from standard boundary conditions in macroscopic flow solvers. For no-slip boundary conditions, we use a second-order bounce-back scheme [31], and for slip boundary conditions, the macroscopic velocity $\boldsymbol{u}$ at the boundary is calculated, then projected onto the surface normal vector $\hat{\boldsymbol{n}}$ as $\boldsymbol{u}_\parallel = \boldsymbol{u} - (\boldsymbol{u} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$ and finally applied as the boundary velocity $\boldsymbol{u}_\parallel$ using a modified version of the second-order bounce-back scheme. At the free surface boundary, the *anti-bounce back rule* enforces the equality of fluid and surrounding pressure [19]:

$$f_{\bar{\alpha}}^{t+1} = -f_\alpha^t + f_{\bar{\alpha}}^{eq}(\rho_B, \boldsymbol{u}_B) + f_\alpha^{eq}(\rho_B, \boldsymbol{u}_B) \tag{2.8}$$

where $f_{\alpha,\bar{\alpha}}^{eq}(\rho_B, \boldsymbol{u}_B)$ are Maxwellian equilibrium distribution functions, ρ_B is related to the surrounding pressure by $\rho_B = p_B c_s^{-2}$, free surface velocity $\boldsymbol{u}_B = \boldsymbol{u}(t_B, \boldsymbol{x}_B)$, time $t_B = t + \frac{1}{2}\Delta t$, and $\boldsymbol{x}_B = \boldsymbol{x} + \frac{1}{2}\hat{\boldsymbol{e}}_\alpha$.

Volume forces $\boldsymbol{F}$, such as gravity, are included by adding an additional term directly to the distribution functions f_α, at every time step [32]:

$$\Delta f_\alpha = 3\omega_\alpha \rho\, \boldsymbol{e}_\alpha \cdot \boldsymbol{F}. \tag{2.9}$$

Note, [32] compares and analyzes different forcing terms and reports a lack of accuracy of formulation Eq. (2.9), for space- and time-varying body forces $\boldsymbol{F}$. In such cases, they propose a formulation, which also depends on the relaxation time τ. Alternatively, in MRT models, volume forcing – which is, in fact, a momentum source term – can be directly added to the corresponding momenta $m_{3,5,7}$:

$$\Delta j_i = \rho \boldsymbol{F}_i \Delta t. \tag{2.10}$$

2.3. Initial conditions

Initial conditions have to be specified in all numerical simulations. Often, simulations are started from a state of rest (i.e. cold start) with zero flow velocities and dynamic pressures in the entire domain. For steady state cases, the flow field gradually converges to constant values, even an improper and inconsistent initialization of the flow field yields accurate results (at least for most of the standard steady state flow benchmarks, such as Poiseuille or Couette flows). By contrast, in transient flow simulations, the accuracy and relevance of the initial solution directly affects the quality of the overall result and the transient flow behavior. If no initial velocities are known, the flow field of a free surface flow simulation should at least be initialized with a hydrostatic pressure distribution, being consistent with the free surface pressure boundary condition and the imposed gravity field. Alternatively, for propagating surface waves, linear or higher-order wave theories may provide adequate initial flow fields in many cases. In the LBM, the particle distribution functions f are initialized, based on this velocity information, using Maxwellian equilibrium distribution functions:

$$f_\alpha = f_\alpha^{eq}(\bar{\rho}, \bar{\boldsymbol{v}}) \tag{2.11}$$

which read, in a formulation tuned for incompressible flows,

$$f_\alpha^{eq} = w_\alpha \left[\rho + \rho_0 \left(3\, \frac{\boldsymbol{e}_\alpha \cdot \boldsymbol{u}}{c^2} + \frac{9}{2} \frac{(\boldsymbol{e}_\alpha \cdot \boldsymbol{u})^2}{c^4} - \frac{3}{2} \frac{u^2}{c^2} \right) \right] \tag{2.12}$$

where w_i are lattice-dependent weighting factors. Note that the non-equilibrium parts of the moments still are zero in this initialization. In the analysis of the decaying velocity fields in a standard Taylor–Green vortex it has been observed that this lack of initialization of the non-equilibrium parts of the moments introduces systematic errors, which do not decay during the simulations. Hence, after the basic initialization detailed above, a local Poisson-type iteration should be used to estimate the non-equilibrium parts of the distribution functions [33], in which collision and propagation steps are executed, on the basis of the fixed initial velocity field.

2.4. Initial flow fields from potential flow theory

For more sophisticated free surface flow simulations, where consistent initial conditions can no longer be easily estimated and wave theories are not applicable either, models with reduced equation sets can provide the initial velocity and pressure fields. For instance, fully nonlinear wave processes can accurately be modeled, prior to breaking, using potential flow theory, which assumes inviscid and irrotational flow. As the wave approaches breaking, the potential flow simulation can be stopped and the simulation data transferred to the LBM as an initial condition, as detailed before. In this work, to study such complex nonlinear wave and breaking processes, a two-dimensional fully non-linear potential flow (2D-FNPF) solver is used for calculating the initial velocity and pressure fields. This "numerical wave tank (NWT)" is based on [34,35]'s implementation. Laplace's equation is solved using a Boundary Element Method (BEM), and 2nd-order Taylor series expansions are used, in an Eulerian–Lagrangian formulation, for the time updating of both the free surface potential and all moving boundary geometries (i.e., free surface, absorbing wavemaker). This requires solving two elliptic problems at each time step, one for the potential and one for its time derivative. Higher-order elements and very accurate numerical integration methods are used in the BEM, which make it possible to achieve extremely high accuracy of the solution and thus to perform long term simulations in the NWT without the need for smoothing or filtering of the solution. In the case of long term simulations (e.g., for periodic or irregular waves), an absorbing beach, combining an "absorbing pressure" on the free surface and a lateral absorbing piston wavemaker yields negligible reflection in NWT experiments. Various ways of generating waves are available in the NWT, including flap and piston wavemakers, exact nonlinear waves (both periodic and solitary), and internal sources. Wavemakers can also be used to generate nonlinear random waves based on standard energy spectra. A feedback control loop allows to iteratively modify the wavemaker stroke spectrum to better approach the targeted spectrum.

2.4.1. Governing equations

In accordance with 2D FNPF theory, we introduce a velocity potential $\Phi(x, z, t)$, which represents inviscid and irrotational flows in the vertical plane in such a way that the velocity is defined as the gradient of the potential $u_i^I = \nabla_i \Phi$.

Hence, continuity equation becomes Laplace's equation for the potential

$$\nabla_j \left(u_i^l\right) = \nabla_j \left(\nabla_i \Phi\right) = \nabla^2 \Phi = 0. \tag{2.13}$$

Using Green's 2nd identity, Eq. (2.13) is transformed into a boundary integral equation (BIE)

$$\alpha(\boldsymbol{x}_l)\Phi(\boldsymbol{x}_l) = \oint_{\Gamma} \left(G\left(\boldsymbol{x}, \boldsymbol{x}_l\right) \frac{\partial \Phi\left(\boldsymbol{x}\right)}{\partial n} - \Phi\left(\boldsymbol{x}\right) \frac{\partial G\left(\boldsymbol{x}, \boldsymbol{x}_l\right)}{\partial n} \right) d\Gamma \tag{2.14}$$

with the 2D free space Green's function:

$$G\left(\boldsymbol{x}, \boldsymbol{x}_l\right) = \frac{-1}{2\pi} \ln r \quad \text{and} \quad \frac{\partial G\left(\boldsymbol{x}, \boldsymbol{x}_l\right)}{\partial n} = -\frac{1}{2\pi} \frac{\boldsymbol{r} \cdot \boldsymbol{n}}{r^2} \tag{2.15}$$

with $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_l$ and $r = |\boldsymbol{r}|$ the distance from point $\boldsymbol{x} = (x, y, z)$ to a point of reference $\boldsymbol{x}_l = (x_l, y_l, z_l)$, both on the boundary, the outward normal vector $\boldsymbol{n}$, and a geometrical parameter $\alpha(\boldsymbol{x}_l) = \theta_l/(2\pi)$, function of the outer angle θ_l of the boundary at position x_l.

On the stationary parts of the boundary, a no-flow condition is prescribed by specifying zero velocity in the normal direction to the boundary,

$$\frac{\partial \Phi}{\partial n} = 0. \tag{2.16}$$

For wave generation using a wavemaker, the time-dependent position $\boldsymbol{x}_w$ and velocity $\boldsymbol{u}_w$ of the wavemaker are prescribed via

$$\frac{\partial \Phi}{\partial n} = \boldsymbol{u}_w \cdot \boldsymbol{n} \quad \text{and} \quad \boldsymbol{x} = \boldsymbol{x}_w. \tag{2.17}$$

On the free surface boundary, the non-linear kinematic and dynamic boundary conditions read, in Eulerian–Lagrangian form:

$$\frac{D\boldsymbol{R}}{Dt} = \boldsymbol{u} = \nabla \Phi \quad \text{and} \quad \frac{D\Phi}{Dt} = -gz + \frac{1}{2} \nabla \Phi \cdot \nabla \Phi - \frac{p_a}{\rho_0} \tag{2.18}$$

with free surface position $\boldsymbol{R}$, gravitational acceleration g, atmospheric pressure p_a, fluid density ρ_0 and material derivative D/Dt.

The solution for the velocity potential and its derivatives along the boundary is obtained in the BEM. Velocities at any arbitrary point inside the domain u_i^l can be explicitly obtained, in a postprocessing step, as a function of the boundary solution for Φ as

$$u_i^l(\boldsymbol{x}) = \oint_{\Gamma} \left(\nabla G\left(\boldsymbol{x}, \boldsymbol{x}_l\right) \frac{\partial \Phi\left(\boldsymbol{x}\right)}{\partial n} - \Phi\left(\boldsymbol{x}\right) \nabla \frac{\partial G\left(\boldsymbol{x}, \boldsymbol{x}_l\right)}{\partial n} \right) d\Gamma. \tag{2.19}$$

For the calculation of the velocity gradient $\nabla_i u_i^l(\boldsymbol{x})$, the ∇ operator is applied once more to Eq. (2.19).

2.4.2. Mapping to the LBM

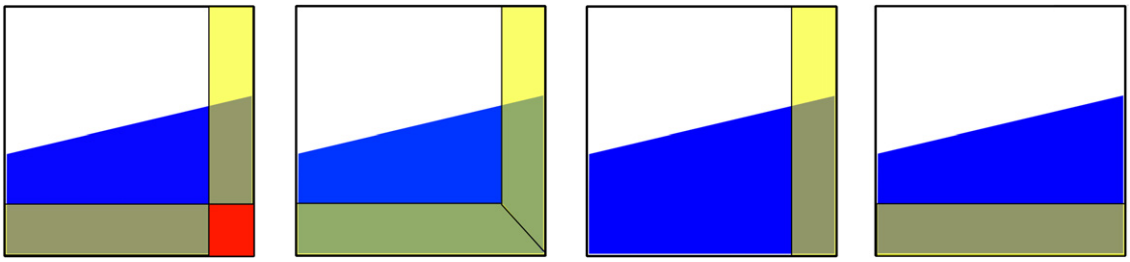
In order to transfer simulation results from the NWT to the LBM, a parametrization has to be established, to select the LBM parameters for grid spacing Δx, Mach number Ma, forcing g_{LB}, and viscosity ν. This involves three steps. First, as LBM solutions satisfy NS equations, up to an order $\mathcal{O}(\mathrm{Ma}^2)$, it is necessary prescribe the maximum Mach number Ma and hence the maximum velocity $v_{\max} = \mathrm{Ma} \cdot c_s = \mathrm{Ma} \cdot \frac{c}{\sqrt{3}}$ in the LBM simulation. Second, free surface flows are typically qualified by their Froude number, i.e., a dimensionless number that compares inertia and gravitational forces, $\mathrm{Fr} = v_{\max}\left(gh\right)^{-0.5}$ using the reference water depth h. The Froude numbers in the NWT and the LBM must be identical, so that based on a given LB discretization we can calculate the LB gravitational term $g_{\mathrm{LB}} = v_{\max}^2 \mathrm{Fr}^{-2} h_{\mathrm{LB}}^{-1}$. Third, in experimental benchmarks, the Reynolds number of experiments and numerical simulations should be the same. The NWT, however, is based on inviscid potential flow theory, so that a Reynolds number cannot be assigned. Nonetheless we can calculate a corresponding Reynolds number via $\mathrm{Re} = \frac{vD}{\nu_{water}}$ and, consequently, find the resulting LBM viscosity as $\nu_{\mathrm{LB}} = v_{\mathrm{LB}} h_{\mathrm{LB}} \mathrm{Re}^{-1}$.

Finally, we find that NWT results for velocity and pressure should be transferred to the LB simulations by applying the following scaling factors

$$\Delta v = \frac{v_{\max,\mathrm{LB}}}{v_{\max,\mathrm{NWT}}} \quad \text{and} \quad \Delta x = \frac{h_{\mathrm{LB}}}{h_{\mathrm{NWT}}} \tag{2.20}$$

$$\Delta t = \frac{\Delta x}{\Delta v} \quad \text{and} \quad \Delta p = \frac{\rho_{\mathrm{LB}} g_{\mathrm{LB}} h_{\mathrm{LB}}}{\rho_{\mathrm{NWT}} g_{\mathrm{NWT}} h_{\mathrm{NWT}}}. \tag{2.21}$$

It turns out that the only free parameters in these equations are the grid resolution (which directly governs Δx) and the Mach number limit ($\mathrm{Ma}_{\max} = 0.1$ is considered a reasonable maximum value for an incompressible limit).



(a) Unsplit method. (b) Split method.

Fig. 3.1. Split and unsplit flux calculation.

3. Free surface advection scheme

In this work, we use an interface capturing method on the basis of a VOF approach to follow the interface position. We describe the *fill level* of a control volume V_{cv} using a fluid fraction variable ε, i.e. the volume fraction being filled with fluid:

$$\varepsilon = \frac{V_{\text{fluid}}}{V_{cv}}. \tag{3.1}$$

A fill level of $\varepsilon = 0$ indicates a completely empty cell, while a fill level of $\varepsilon = 1$ indicates a completely filled cell. For the weakly compressible LBM approach, the sum of fill levels is *not* conserved due to compressibility effects, so that the interface advection is governed by the full continuity equation:

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \nabla \cdot (\boldsymbol{v}\rho) = 0. \tag{3.2}$$

This equation can be discretized with various methods. In algebraic models, the gradient operator is discretized with higher-order numerical methods, which leads to a diffusive phase interface with an interface thickness of typically two to five cells. By contrast, in a geometric discretization, the interface is kept sharp. The equation is discretized with a classical finite volume method and the interface position is reconstructed at every time step to perform the interface advection. These two essential VOF steps of the latter kind of method will be described in the following.

3.1. Geometric discretization of the advection equation

Integrating mass conservation equation over a control volume Ω and applying the divergence theorem, we obtain a surface integral for the convective terms:

$$\int_\Omega \frac{\partial \rho}{\partial t} d\Omega + \int_\Omega \nabla \cdot (\boldsymbol{v}\rho) d\Omega = \frac{\partial}{\partial t} \int_\Omega \rho \, d\Omega + \int_\Gamma (\boldsymbol{v}\rho) \cdot \hat{\boldsymbol{n}} d\Gamma \tag{3.3}$$

where $\hat{\boldsymbol{n}}$ is the unit outward normal vector on the corresponding face of the control volume. A further discretization in time with an explicit Euler finite difference scheme leads to

$$m^{t+1} = m^t - \sum_{i=0}^{d} \Phi_i \tag{3.4}$$

where Φ_i denotes the flux through the ith face of the control volume

$$\Phi_i = \left(\boldsymbol{v} \cdot \hat{\boldsymbol{n}}\rho\right)_i^t \varepsilon_i^t dt \tag{3.5}$$

where ε_i^t denotes the wet surface area at the corresponding face of the control volume. The new VOF fill level of an interface cell is calculated by summing up all flux terms:

$$\varepsilon^{t+1} = \frac{m^{t+1}}{\rho^{t+1}} = \frac{m^t - \sum_{i=0}^{d} \Phi_i}{\rho^{t+1}} = \frac{\varepsilon^t \rho^t - \sum_{i=0}^{d} \Phi_i}{\rho^{t+1}}, \tag{3.6}$$

for a model with d exchange directions. In this unsplit method, all d flux terms are calculated at the same time (Fig. 3.1a), while in split methods the advection directions are alternated (Fig. 3.1b). For the calculation of Φ_i, one needs to account for changes in cell fill during the advection. If this is ignored, the fluxes may be higher than the actual cell retention volume, leading to the need to redistribute excessive mass (see e.g. [22]). In contrast to that, a time-linear evaluation of the volume

fluxes does not introduce excessive mass, since by construction the maximum volume fluxes can never be exceeded:

$$\Phi_i = \left(\boldsymbol{v}\cdot\hat{\boldsymbol{n}}\rho\right)_i^t \cdot \int_t \varepsilon_i(t)dt. \tag{3.7}$$

Here, the temporal change of wet surface area ε_i is considered. In a standard LB free-surface approach, the VOF control volume Ω then is assigned to the LB nodes [19], and the flux terms Φ_i are expressed in terms of LB particle distribution functions. In contrast to these approaches, we propose a staggered grid layout, where every cell (i.e., control volume) is spanned by eight LB nodes. The mass exchange between neighboring cells is no longer calculated in terms of LB distribution functions f, but must be evaluated from macroscopic velocity and density information on the surface Γ of the control volume Ω.

The resulting advection algorithm has recently been published in [22]. Initially, the mass exchange was evaluated for six neighboring cells only ($d = 6$). Moreover, the red marked volume in Fig. 3.1a (left) was fluxed multiple times, both to the adjacent right and bottom cells, instead of using a more sophisticated algorithm with tracing the characteristics (Fig. 3.1a, right). Hence, with a maximum LB velocity of $v_{\max} \approx 0.05$, on a Cartesian grid with $\Delta x = \Delta t = 1.0$, the maximum volume error yields $0.05^3 \approx 1.2510^{-4}$ and we considered this to be negligible [22]. However, although the model was successfully validated for various benchmark problems, such as breaking dams, it turned out that more demanding flows could only be addressed by modeling a complete mass exchange to all the neighboring cells with $d = 26$. More particularly, in the simulations of a plunging breaker, where high velocities and a high interface curvature occur, the reduced six-direction advection model failed. Hence, in this work, all 26 geometrical neighbor cells are used for computing free surface advection. This way, the diagonal fluxes are much better represented, in particular leading to an improved overall breaker shape.

3.2. Piecewise Linear Interface Reconstruction

The interface reconstruction itself can be executed independently of the order of flux calculations and split/unsplit type of methods. In this work, this is done using the PLIC method [36,37]. This type of methods determine the interface location and hence the flux terms more accurately than simpler models, and avoid oscillations on the phase interface. In PLIC, the free surface is represented as a series of linear segments (in 2D) or planes (in 3D), which are uniquely defined by their unit normal vector $\boldsymbol{n}$ and the distance α to a point of origin:

$$\boldsymbol{x}\cdot\boldsymbol{n} = \alpha. \tag{3.8}$$

The normal vector is approximated by the gradient of the fluid volume fraction ε:

$$\boldsymbol{n} = -\frac{\nabla\boldsymbol{\varepsilon}}{\|\nabla\boldsymbol{\varepsilon}\|}. \tag{3.9}$$

The gradient $\nabla\boldsymbol{\varepsilon}$ can be calculated from the surrounding cell fill levels, using the finite-difference ansatz discussed in [38,39] and tested in [22]. As the fill level in a cell is limited to the range $\varepsilon \in [0, 1]$, it can be shown that the resulting surface normal is between first- and second-order accurate, depending on the interface inclination.

Subsequently, the only remaining unknown for the linear surface reconstruction is the distance α between the surface segment and a coordinate origin, which can be calculated from the surface normal vector $\boldsymbol{n} = n_i$ and the fill level ε. In 3D, the expression for the cell fill level, which in fact corresponds to the cut volume of the reconstructed interface plane segment and a unit cell with $\Delta x_i = 1.0$, is given by

$$\varepsilon(\boldsymbol{n},\alpha) = \frac{1}{2n_1n_2n_3}\left[\alpha^3 - \sum_{j=1}^{3} H(\alpha - n_j)\left(\alpha - n_j\right)^3 + \sum_{j=1}^{3} H(\alpha - \alpha_{\max} + n_j)\left(\alpha - \alpha_{\max} + n_j\right)^3\right] \tag{3.10}$$

with H the Heaviside step function and $\alpha_{\max} = \sum_{j=1}^{3} n_j$. Solving analytically for α is not possible since an inverse of this expression is not generally possible to derive. Therefore, a case distinction is introduced, depending on the Heaviside terms H in Eq. (3.10). The values of ε, for which these Heaviside functions switch from zero to one are typically referred to as *critical volumes* ε_i^c. These can be calculated easily using Eq. (3.10) as

$$\varepsilon_1^c = \frac{1}{6}\frac{n_1^2}{n_2n_3} \tag{3.11}$$

$$\varepsilon_2^c = \frac{1}{6}\frac{n_1^2 + 3n_2^2 - 3n_1n_2}{n_2n_3} \tag{3.12}$$

$$\varepsilon_3^c = \begin{cases} \dfrac{1}{2}\dfrac{n_1+n_2}{n_3} & n_1+n_2 < n_3 \\ \dfrac{1}{6}\dfrac{n_2^3+n_1^3-n_3^3+3n_2n_3^2+3n_1n_3^2-3n_2^2n_3-3n_1^2n_3}{n_1n_2n_3} & n_1+n_2 > n_3 \end{cases} \tag{3.13}$$

$$\varepsilon_4^c = 1-\varepsilon_3^c, \qquad \varepsilon_5^c = 1-\varepsilon_2^c, \qquad \varepsilon_6^c = 1-\varepsilon_1^c. \tag{3.14}$$

Table 3.1
Analytical solutions for the plane coefficient α.

Region	Result
$\varepsilon < \varepsilon_1^c$	$\alpha = \sqrt[3]{6n_1n_2n_3\varepsilon}$
$\varepsilon_1^c < \varepsilon < \varepsilon_2^c$	$\alpha = \frac{n_1}{2} - \frac{1}{6}\sqrt{-3n_1n_2 + 72n_2n_3\varepsilon}$
$\varepsilon_2^c < \varepsilon < \varepsilon_3^c$	$\emptyset$
$\varepsilon_3^c < \varepsilon < \varepsilon_4^c$	$\alpha = \frac{1}{2}n_1 + \frac{1}{2}n_2n_3\varepsilon$, if $n_1 + n_2 < n_3$
	$\emptyset$, if $n_1 + n_2 > n_3$
$\varepsilon_4^c < \varepsilon < \varepsilon_5^c$	$\emptyset$
$\varepsilon_5^c < \varepsilon < \varepsilon_6^c$	$\alpha = \alpha_{\max} - (\frac{n_1}{2} - \frac{1}{6}\sqrt{-3n_1n_2 + 72n_2n_3(1-\varepsilon)})$
$\varepsilon > \varepsilon_6^c$	$\alpha = \alpha_{\max} - \sqrt[3]{6n_1n_2n_3(1-\varepsilon)}$

Note that, in this derivation, the normal vector components are first permuted and sorted, so that $n_1 \le n_2 \le n_3$ holds and further case distinctions can be avoided. After the calculation of ε_i^c is done, Eq. (3.10) can be piecewise inverted, leading to the values of the plane coefficient α that are given in Table 3.1. Note that for three configurations, an algebraic inversion is not possible: $\varepsilon_2^c < \varepsilon < \varepsilon_3^c$, $\varepsilon_3^c < \varepsilon < \varepsilon_4^c$ (if $n_1 + n_2 > n_3$), and $\varepsilon_4^c < \varepsilon < \varepsilon_5^c$ ($\emptyset$ in Table 3.1). In these three cases, the value for α is determined iteratively, e.g. by a secant method, or the faster-converging *Brent's method* [40].

The position of the interface plane now is uniquely determined, since both $\boldsymbol{n}$ and α are known, so that fluxes Φ_i can be calculated. For the latter, we use a geometric flux calculation, in which fluxes are calculated as the intersection of the fluid body and the fraction of the cell volume that is advected, for example, $V_{0,cell} = v_0 \Delta t \Delta x_1 \Delta x_2$ in the x_0-direction. The cut volume of a plane and an arbitrary cuboidal control volume, with extent Δx_i is given by

$$V(\boldsymbol{n}, \alpha, \Delta x_i) = \frac{1}{2n_1n_2n_3}\left[\alpha^3 - \sum_{j=1}^{3} H(\alpha - \Delta x_j n_j)\left(\alpha - \Delta x_j n_j\right)^3 + \sum_{j=1}^{3} H(\alpha - \tilde{\alpha}_{\max} + \Delta x_j n_j)\left(\alpha - \tilde{\alpha}_{\max} + \Delta x_j n_j\right)^3\right] \tag{3.15}$$

with $\tilde{\alpha}_{\max} = \Delta \boldsymbol{x} \cdot \boldsymbol{n} = \sum_{j=1}^{3} \Delta x_j n_j$. This volume V is evaluated for the reconstructed linear surface (normal vector $\boldsymbol{n}$, plane parameter α) and the portion of the cell volume which is advected to the neighboring cell, $\boldsymbol{v}_i \Delta t$:

$$\Phi_i = V(\boldsymbol{n}, \alpha, \boldsymbol{v}_i \Delta t). \tag{3.16}$$

If one or two components of the surface normal vector are zero, reduced one- or two-dimensional PLIC schemes are used for the flux calculation.

4. Details on the overall hybrid algorithm

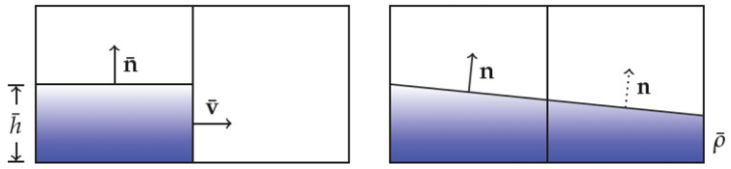
The previously described methods for calculating the flow field and the interface advection are now coupled, resulting in a hybrid LBM–VOF free surface flow model. The LBM serves to calculate the flow field, which then advects the free surface using the VOF–PLIC method detailed above. Topological changes resulting from interface advection are then transferred back to the LBM domain.

4.1. Extensions to the LBM

The LB solver has to be modified to incorporate the free surface capturing capabilities. As the computational domain changes in time, there will be inactive lattice nodes outside the fluid domain. Hence, an additional flag field is used to track whether a node is inside or outside the fluid domain and, hence, has to execute collision or propagation steps, respectively. A pressure boundary condition is applied at the free surface. Apart from the prescribed boundary value for the atmospheric pressure, the free surface velocity $\boldsymbol{u}(t_B, \boldsymbol{x}_B)$ for the anti-bounce back scheme (Eq. (2.8)) along the lattice link i is obtained by linear extrapolation from the fluid velocities inside the domain:

$$\boldsymbol{u}_i(t_B, \boldsymbol{x}_B) = \boldsymbol{u}(t_B, \boldsymbol{x}) + 0.5\left(\boldsymbol{u}(t_B, \boldsymbol{x}) - \boldsymbol{u}(t_B, \boldsymbol{x} - \boldsymbol{e}_i \Delta t)\right). \tag{4.1}$$

Accurate velocity extrapolation is crucial to ensuring accurate free surface results, especially in regions with high gradients and high interface curvature as, e.g., in breaking wave jets. A higher-order extrapolation might further improve the results. However, according to our experience, the linear order is sufficient, as it corresponds to the order of the pressure boundary condition (2.8) and to the order of the surface reconstruction scheme, which is of first to second order accuracy due to the normal vector calculation in (3.9). In future work, as soon as proper second-order boundary conditions for the flow field are available, higher-order interpolations will be addressed. In any case, the previously proposed constant extrapolation definitely leads to inaccurate results.



(a) Inflow. (b) Outflow.

Fig. 4.1. Inflow and outflow boundary conditions.

4.2. Extensions for the advection scheme

Similar to the flow solver, the advection scheme must be extended as well, especially regarding the treatment of interface cells next to solid boundaries, boundary conditions and the time stepping.

4.2.1. Boundary conditions

On the free surface boundary, the kinematic boundary condition is inherently fulfilled by the advection scheme. After reconstruction, the interface is advected in a Lagrangian way, so that the total flux through the interface surface is zero. On solid boundaries, a no-flow boundary condition guarantees that no fluid leaves the boundary cells: if a face i of the control volume contains solid nodes only, the resulting flux is set to zero, $\Phi_i = 0$. Along boundaries where a flux q is prescribed, the information is split up. The velocity v is used for the boundary condition of the flow solver, and the corresponding water height h is used as boundary condition for the advection scheme: the water level and the interface normal at boundary cells are kept identical in the flow solver and the advection scheme. These are fixed to the desired boundary value (Fig. 4.1a). At pressure boundaries of the fluid solver, which are mainly used as outflow boundary conditions, zero-gradient conditions for the surface normal are assumed (Fig. 4.1b). Alternatively, if the flow direction is clear, a backward estimation of the surface normal can be used.

4.2.2. Interface cells at the boundary

For interface cells in the vicinity of solid walls, several model extensions have to be introduced in addition to the no-flow boundary condition in the wall normal direction. First, the size of the control volume may change, since a cell can contain parts of the solid phase, in addition to gas and fluid. For this purpose, a *solid fill level* function $\zeta(\boldsymbol{x})$ is introduced to quantify the relative amount of the cell volume that is occupied by solid obstacles. Eq. (3.6) is modified accordingly to

$$\varepsilon^{t+1} = \frac{\varepsilon^t \rho^t (1-\zeta) - \sum_i \Phi_i}{\rho^{t+1}(1-\zeta)}. \tag{4.2}$$

Additionally, if an edge of the control volume is cut by a solid obstacle, the mean solid volume in the corresponding cell is evaluated and serves as a weighting factor for the flux term Φ_i:

$$\Phi_i^* = \Phi_i \cdot \left[1 - \frac{\zeta(\boldsymbol{x}) + \zeta(\boldsymbol{x} + \boldsymbol{e}_i \Delta t)}{2}\right]. \tag{4.3}$$

Finally, the finite difference evaluation of the surface normal vector fails for boundary cells, where the application of centered finite differences is not possible, since the solid cell does not contain valid information on the fill level. Instead, an asymmetric finite difference scheme is used near solid boundaries.

4.2.3. Adaptive time stepping

In hybrid approaches, the idea of adaptive time stepping is tempting, especially when both solvers act on different time scales. The resulting maximum velocity magnitude in an LBM simulation is typically limited by the Mach number, which controls the convergence of the LBM scheme to the incompressible NS equations. Reasonable Mach number values should be selected in the range of 0.005–0.05, corresponding to maximum LBM velocity magnitudes of ≈0.0025–0.025. Opposite to that, the advection scheme has shown to produce good results even for CFL numbers around 0.1 or higher [22]. Hence, an adaptive time stepping for the advection scheme has been introduced in order to save computational cost. During runtime, the maximum global velocity magnitude is evaluated in certain intervals. Then, the time interval for the interface updating is changed and the velocities are rescaled before the evaluation of the flux terms (Eq. (3.16)) according to

$$t_{UI} = \frac{\mathrm{CFL}_{\max}}{\mathrm{CFL}_{actual}} \quad \text{and} \quad \boldsymbol{v}_{UI} = \frac{t_{UI}}{t_{\mathrm{LB}}} \boldsymbol{v}_{\mathrm{LB}}. \tag{4.4}$$

For typical LBM simulations, the interface is updated each 35–350 time steps without substantial negative impact on the overall result quality. Especially for gravity-driven simulations, as e.g. falling drops or dambreak scenarios, where the fluid is accelerated from a state of rest, the performance gain of proceeding in this manner during the initial stages of the simulation is high.

4.2.4. Calculation of cell densities and face velocities

The advection scheme relies on the cell-centered values of fill level ε and mass m of each cell. An average cell density ρ is needed to convert between these intensive and extensive quantities. It is obtained by averaging the fluid densities of all cell fluid nodes according to

$$\rho_{cell}^{t} = \frac{1}{n_f} \sum_{n_f} \rho_i^t \tag{4.5}$$

for a cell with n_f fluid nodes. A more sophisticated interpolation using information on the exact fluid distribution in a cell (by means of a *node fill level*) did not further improve the results.

The treatment and calculation of free surface velocities has to be done with great care. In standard VOF methods, the advection velocities are defined on each face centroids of the control volume. In most cases, the same staggered grid layout is also used for the solution of NS solutions, so that inter- or extrapolation is not needed. By contrast, in the present hybrid LBM–VOF method, the fluid velocities are known on the eight lattice nodes in the corners of each cell. In a low-order ansatz, the face values of density ρ_i and velocity v_i are determined from the corresponding nodal values:

$$\boldsymbol{v}_i = \sum_i \boldsymbol{v}_i \cdot \hat{\boldsymbol{n}} \quad \forall \text{ fluid nodes } i. \tag{4.6}$$

If the cell is not entirely filled with fluid, gas nodes appear, which do not have valid particle distribution functions and hence no valid velocity information. In the worst case of an under-resolved wave front, this interpolation is constant in space, and grid refinement is necessary to improve the accuracy. For all other cases, the interpolation is linear, which is sufficient since this corresponds to the order of the free surface reconstruction scheme and the time discretization. For testing purposes, an extrapolation approach, which is consistent with the calculation of the free surface velocity for the pressure boundary condition, was implemented: if a face is a fluid face and enough fluid nodes are available in the neighboring cells, linear interpolation is applied to obtain the corresponding velocity values. Again, the results were not significantly improved.

4.3. Algorithm

The resulting advection algorithm is included in the LBM framework referred to as VirtualFluids [41,42] and is given in Algorithm 1. After the LBM collision and propagation steps, the position of the interface is updated on the basis of the new valid velocity and density fields. Note that the advection algorithm loops over interface cells only; for further details the interested reader is referred to [22].

5. Validation

Several state-of-the-art benchmark test cases were used to validate the new hybrid model. First, two- and three-dimensional breaking dam benchmarks were used to verify that the hybrid model is capable of simulating real world hydraulic engineering fluid applications. The presence of solid objects as obstacles to the flow will validate the basic approach of accounting for solid bodies. Wave impact on solid boundaries is also checked and validated as part of these test cases, including the treatment of corner points and geometric discontinuities. The breaking dam test cases do not require the use of elaborate boundary conditions, so that after these initial breaking dam validations, the behavior of a free falling water jet is analyzed to validate the use of proper inflow and outflow boundary conditions. After that, the validation of the LBM coupling to a potential flow solver is performed for the case of a breaking of a wave during shoaling on a 1:10 slope. The simulations were run on a single core of a 2.6 GHz Intel Quad X5550 CPU. The performance of the code is measured in *million node updates per second* (MNUPS). The typical performance of LBM bulk flow solvers without free surface capturing on such hardware, for the purpose of comparison, is between 4 and 5 MNUPS.

5.1. Breaking dam with obstacle

The classical dam breaking benchmark is one of the most widely used free surface flow validations. Neither advanced boundary conditions nor complex initial conditions are needed. In [22], the standard dam breaking benchmark of [43] was successfully simulated. Here, this benchmark case is extended with the addition of a solid obstacle in the center of the domain, according to the experiments of [44] (Fig. 5.1).

Upon removal of the dam, the collapsing of the water column is initiated and the front evolves and finally impacts the solid obstacle. At this moment, the flow pattern changes drastically, similar to what occurs during wave impact on structures. As the water level decreases, a small, highly curved jet evolves, and the water is deflected to the top of the obstacle. A set of snapshots is given in Fig. 5.2 for four selected points in time, which illustrate this flow evolutions in model results and experiment.

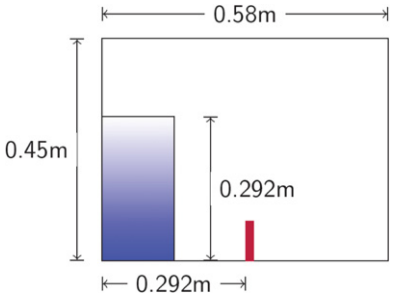
In Fig. 5.3, the numerical results are compared to experimental observations by Kölke [44] for two different grid resolutions. Very good agreement can be seen for the fine grid, at least for this qualitative comparison. The jet evolves slightly faster in computations than in the experiment, but the time-lag decreases with grid refinement. Hence, we conclude that a proper grid resolution is essential, to capture at least the large-scale sprays and the resulting energy dissipation, and to accurately represent the solid body. We verified that, in the LBM, a careful treatment of the slip–slip corner boundaries is

Algorithm 1 Update interface algorithm (PLIC surface reconstruction)

```
if node type == fluid then
    collision
    forcing
    propagation
    apply boundary conditions
end if
if cell type == interface then
    determine face values for density ρ_i and velocity u_i
    determine surface normal vector n
    reconstruct surface (i.e. determine plane parameter α)
    determine face fill levels ε_i
    calculate mass fluxes Φ
    evaluate new fill level ε_i^{t+1}
    if new fill level ε_i^{t+1} < 0.0 then
        add cell to container with new gas cells
        set new fill level to 0.0 and store excessive mass portion
    end if
    if new fill level ε_i^{t+1} > 1.0 then
        add cell to container with new fluid cell
        set new fill level to 1.0 and store excessive mass portion
    end if
    set new cell states
    check consistency of node and cell states
    set new node states
    initialize new fluid nodes
    check and adjust interface cell states
    if applicable, locally distribute the excessive mass
end if
```

Param.	Value
Domain	0.58m x 0.45m
Water	0.146m x 0.292m
Obstacle	0.015m x 0.08m
Lattice	58x1x45
	116x1x90
	232x1x180
	464x1x360
Δx	0.01m, ..., 0.00125m

(a) Parameters.



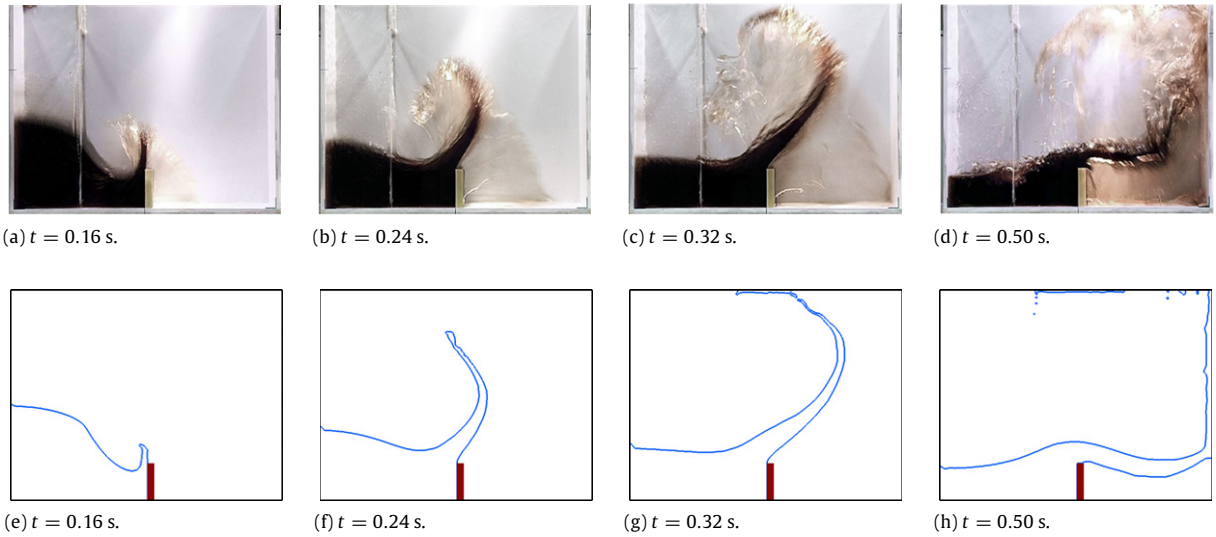
(b) Geometry.

Fig. 5.1. Breaking dam with solid obstacle.

crucial to achieve good results. For slip BCs an accurate normal vector is needed in the LBM nodes, which is discontinuous in corner points. If a smoothed geometry is used, where the corner is "rounded out" and the corner node uses a linear interpolation of the neighboring surface normal vectors, the jet shape is affected. For this reason, the normal vector at the obstacle corners is set to the normal vector of each respectively participating face. With this definition, the overtopping jet can be accurately simulated. The node update rate for this test case was about 2.55 MNUPS, leading to a computational time of 6 h, for the $464 \times 1 \times 360$ nodes' grid configuration.

5.2. Breaking dam with obstacle and pressure probes

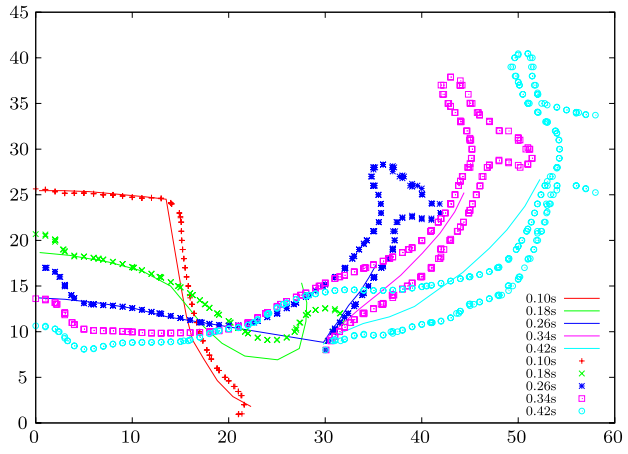
After qualitatively validating the jet geometry, this next test case serves to validate results for the free surface–structure interaction in terms of water elevation and impact pressure. The corresponding experiments have been carried out at the Maritime Research Institute Netherlands (MARIN) [45,46]. A laboratory tank with an open roof was used (Fig. 5.4).



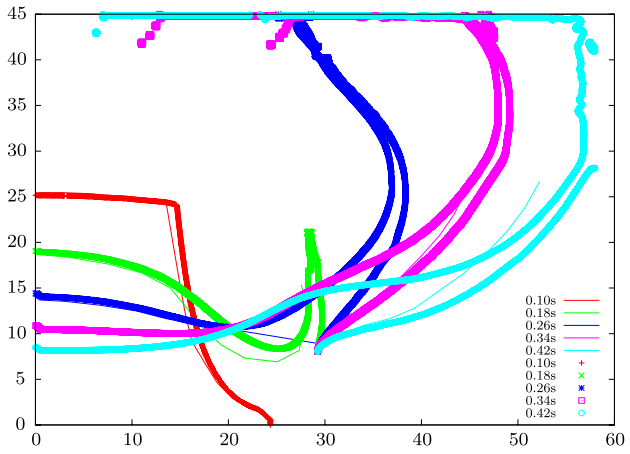
(a) $t = 0.16$ s. (b) $t = 0.24$ s. (c) $t = 0.32$ s. (d) $t = 0.50$ s.

(e) $t = 0.16$ s. (f) $t = 0.24$ s. (g) $t = 0.32$ s. (h) $t = 0.50$ s.

Fig. 5.2. Breaking dam with an obstacle (Fig. 5.1): experimental results of [44] compared to present numerical results (fine grid).



(a) Coarse grid ($58 \times 1 \times 45$).

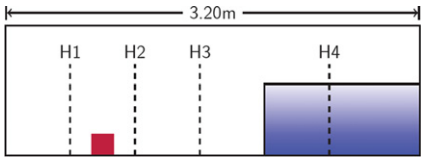


(b) Fine grid ($464 \times 1 \times 360$).

Fig. 5.3. Results for dambreak test case with obstacle (Fig. 5.1) for two different grid resolutions, in comparison to experimental data (solid lines).

Param.	Value
Domain	3.20m x 1.00m
Water	1.20m x 0.55m
Obstacle	0.16m x 0.16m
Lattice	512x1x160
	$(\Delta x = 0.00625m)$

(a) Parameters.



(b) Geometry.

Fig. 5.4. Breaking dam (3D).

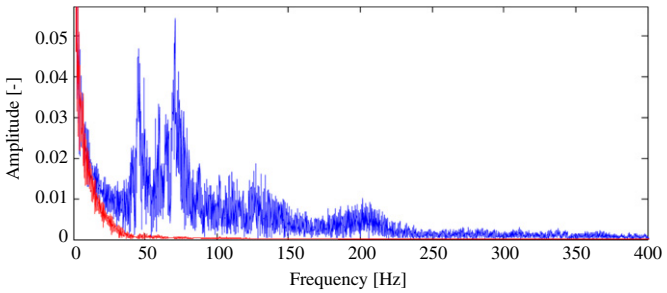


Fig. 5.5. Fourier spectrum of the P1 pressure signal, before (blue) and after (red) filtering. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

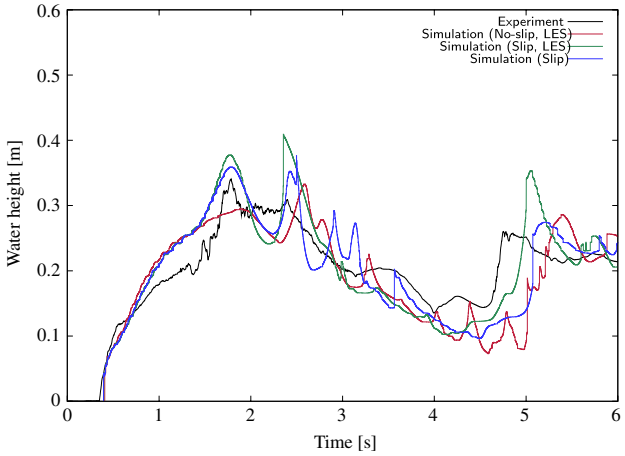
Behind $x = 2$ m in the back part of the tank, a water column is made to collapse. In these experiments, unlike the thin diaphragm used in [43], the water column is constrained by a door which is pulled up by releasing a weight at initial time. The water elevation then was measured in the tank at four locations (H1 in the reservoir and H2, H3, H4 in the tank at $x_i = 0.5, 1.0, 1.5, 2.66$ m).

The same setup is used to perform two sets of experiments. In the first one, performed by Kleefsman [45], a box was placed in the tank and was covered by eight pressure sensors, four on the front of the box at height $z = 0.025, 0.063, 0.099$ and 0.136, and four on the top of the box at $x = 0.806, 0.769, 0.733$ and 0.696. The experimental results for the height probes H2 and H4 and the pressure probes P1 and P7 are public domain and can be downloaded [47]. The same experimental setup, without the box, is used by Wemmenhove [46] to examine tank sloshing phenomena with flexible walls.

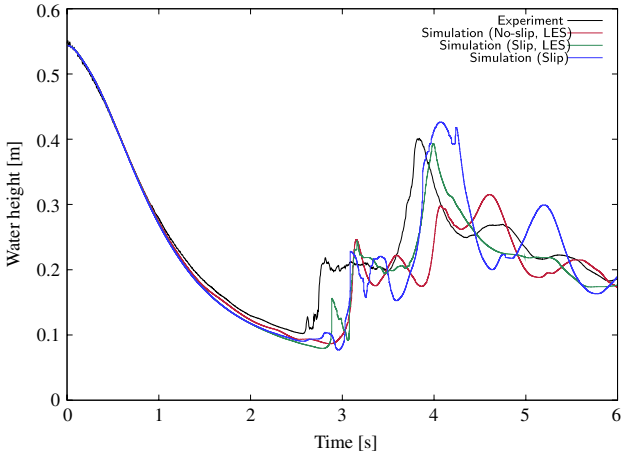
In the LBM, a 2D equivalent test case is used for this numerical simulation, assuming periodicity in the third spatial direction. In Fig. 5.6, the results for the surface elevation gages H2 and H4 on a grid with $256 \times 1 \times 80$ nodes are compared to the experimental data for three different configurations of boundary conditions. In general, good agreement can be observed. More specifically, the initial peak at gage H2 cannot be reproduced if no-slip bottom boundary conditions are assumed, whereas the simulation with frictionless bottom overpredicts the surface elevation near the obstacle. The measured arrival time of the reflected wave at $t \approx 5$ s is shifted by about 0.4 s in all three simulations, which is most likely due to the simplified 2D setup. It is notable that the simulation without bottom friction, but using an LES model, predicts a higher value for the reflected wave amplitude than the slip-simulation without LES. Results for gage H4 show similar characteristics, including a comparable time-lag. However, the slip simulation with an LES model produces the best results. Results at pressure probes P1 and P7 are shown in Fig. 5.7 for the no-slip LES simulation in the finest grid ($512 \times 1 \times 160$ lattice nodes). The pressure at probes P1 and P4 qualitatively corresponds to the experimental data and we see that the time lag between incident and reflected waves is small. The amplitudes match reasonably well, although the pressure signals of the simulation were processed with a low-pass filter to remove the high-frequency pressure noise.

The observed pressure noise is mainly caused by inconsistent node initializations during the free surface advection, that potentially leads to shock waves. Due to the weak compressibility of the numerical method, these shock waves propagate in the computational domain with a finite speed, i.e. the speed of sound. Once released, they are only damped by viscous dissipation. However, in high-Reynolds number simulations at low viscosities, these shock waves are not remarkably damped, but are reflected back and forth in between solid domain boundaries. A Fourier analysis of the original LB pressure signal reveals two dominant noise frequencies, $f_1 = 45$ Hz and $f_2 = 70$ Hz, that strongly dilute the pressure signal. Hence, two moving average filters with corresponding filter widths $T_i = f_i^{-1}$, i.e. $T_1 = 0.022$ s and $T_2 = 0.014$ s, have been used to smooth the pressure signal and to reduce the noise. The pressure spectrum before and after the smoothing process is shown in Fig. 5.5.

The simulations for this test case took approximately 2 h, on the fine grid with $512 \times 1 \times 160$ nodes, which corresponds to a node update rate of 2.65 MNUPS.



(a) Water level probe H2.



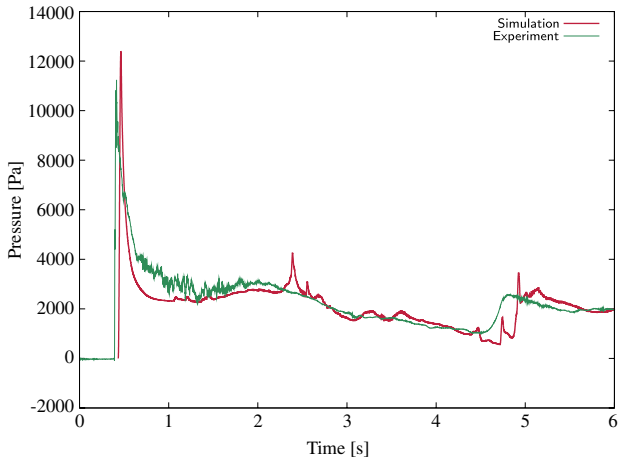
(b) Water level probe H4.

Fig. 5.6. 3D dam breaking (Fig. 5.4), results for height probes H2 and H4 for three simulation setups ($\Delta x = 0.0125$ m).

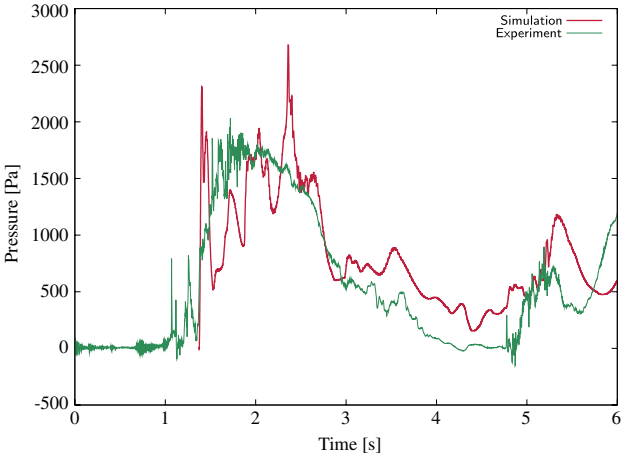
5.3. *Free falling jet*

The earlier cases of breaking dam scenarios were demanding in terms of free surface advection, pressure evaluation, and representation of solid obstacles in the flow. However, in those cases, complex boundary conditions were not needed as the entire domain boundary was made of impermeable no-slip or slip-walls. Hence, for validating the inlet and outlet boundary conditions in the model, we investigate a free falling jet over a horizontal slip-surface (Fig. 5.8). Initially, the LBM domain is empty and an inflow on the left side is specified, with a flow rate $q = 0.0643\ \mathrm{m}^2/\mathrm{s}$ per unit width. The inflow water depth $h = \kappa y_c$ depends on the critical water depth, here $y_c = 0.075$ m, similar to the experimental setups of [48]. We examine the cases $\kappa = 0.49, 0.68, 1.00$, corresponding to Froude numbers $\mathrm{Fr} = 1.0, 1.78$ and 2.91. Gravity is given as $g = 9.81\ \mathrm{m/s}^2$.

After the flow develops over the domain solid part at $x = 0.3$ m, it falls freely as a jet and adjusts its shape until a quasi-steady geometry is reached. During free fall, the fluid is accelerated in the vertical direction and the jet narrows. In Fig. 5.9, the computed free surface geometry is compared to the experimental data [48] for two different grid resolutions, comparing the geometry of the free surface over the obstacle and that of the lower part of the free falling jet. Good overall agreement is observed for all three cases in both grids. Moreover, comparing results for two different grid resolutions, we see a clear convergence to the experimental data. The free jet location in the coarse grid ($\Delta x = 0.25$ cm, Fig. 5.9a) is slightly lower than the experimental values, indicating that the horizontal velocity component is too low. Grid refinement yields a higher jet position ($\Delta x = 0.125$ cm, Fig. 5.9b), which agrees better with experiments. Again, the proper handling of the surface normal vector discontinuity at the corner, $\boldsymbol{x}_s = (0.3, 0.3)$ is crucial in this test case. If the corner was rounded out, the jet would stick to the wall and its geometry would be distorted. For this reason, the normal vector at $\boldsymbol{x}_s$ is assumed to be $\boldsymbol{n}_s = (0, 1)$. The computational time for this test case was between 13.5 h ($\kappa = 1.0$) and 24 h ($\kappa = 0.49$), for the $480 \times 1 \times 320$ grid configuration.



(a) Pressure probe P1 (low-pass filtered).

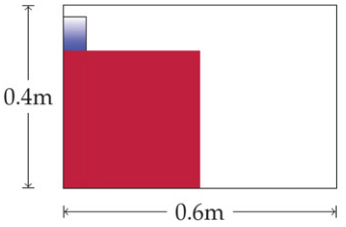


(b) Pressure probe P7 (low-pass filtered).

Fig. 5.7. 3D dam breaking (Fig. 5.4), results for pressure probes, high-resolution simulation with LES and no-slip boundary conditions ($\Delta x = 0.00625$ m).

Param.	Value
Domain	0.6m x 0.4m
κ	0.49,0.68,1.00
Fr	1, 1.78, 2.91
Lattice	240x1x160
	480x1x320
Δx	0.25cm, 0.125cm

(a) Parameters.

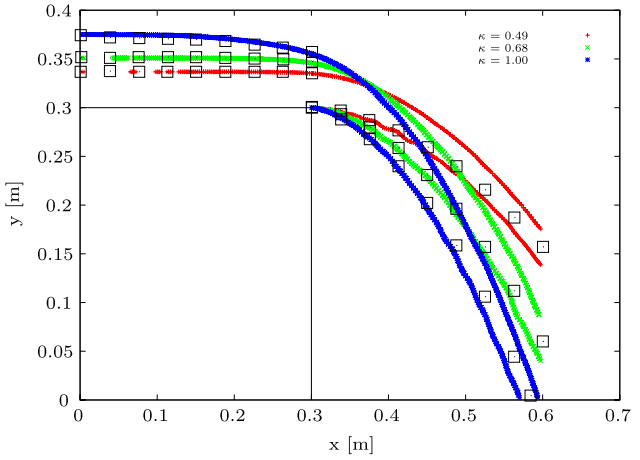


(b) Geometry.

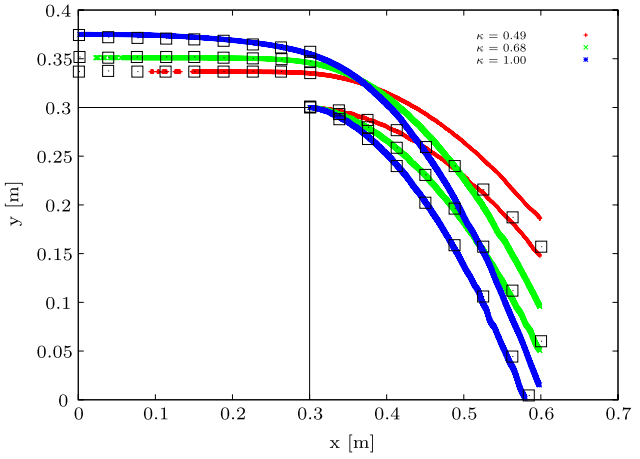
Fig. 5.8. Free falling jet.

5.4. Breaking wave during shoaling

As a final validation benchmark, we analyzed the case of a solitary wave that breaks during shoaling over a plane slope in a series of increasingly refined LBM grids. As discussed before, these computations were initialized using the geometry,



(a) Coarse grid.



(b) Fine grid.

Fig. 5.9. Free falling jet (Fig. 5.8): position of the free jet for two different grid resolutions in comparison to reference data (black squares, [48]).

dynamic pressure, and velocity fields computed in a very accurate Numerical Wave Tank (NWT), based on fully nonlinear potential flow theory. Results of the NWT simulations for the initialization data are shown in Fig. 5.10, for an incident solitary wave of height $H = 0.5$ m over a 1:15 slope (note, in the NWT we used the fully nonlinear solitary wave solution of Tanaka to initialize the computations, [49]). Once the wave starts breaking (by overturning on the slope), this test case is extremely demanding in terms of the free surface capturing scheme, as was already noted by other investigators using other VOF methods [50–52].

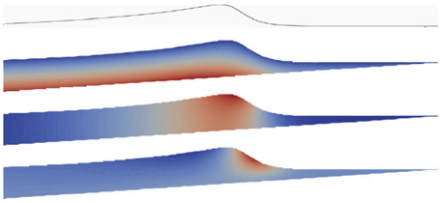
The LBM simulations are run for four different grid resolutions and two Mach numbers, as summarized in Fig. 5.10. A slip boundary condition is used along the bottom boundary, whereas at the offshore boundary a constant water depth $\bar{h}$ is prescribed. As illustrated in Fig. 5.11 for time $t = 3.0$ s, in the finer grid with Ma $= 0.001$ the overall flow behavior in the LBM is as expected, and the breaker jet geometry agrees very well with results of purely potential flow simulations (e.g., [53]). While the overall wave flow is not too influenced, a high dependency of results in the breaker jet on grid resolution and Mach number can be seen in Fig. 5.12.

The test case was the most challenging validation example during the development of the new VOF advection scheme. Even though more simple validations, such as the breaking dam setup, worked fine even with less sophisticated free surface capturing methods, for the breaking wave, more accurate concepts were required to achieve sufficient accuracy. Thus, here, at the free surface boundary, a linear extrapolation of surface velocity is used for the pressure boundary condition, and the flux calculation is extended to twenty-six directions, as already mentioned in Section 3.1.

Note that – even though this test case is 2D – the simulation was carried out with the 3D solver using periodic boundary conditions in the transverse y direction. Hence, the simulation of 3D plunging breakers or deep water freak waves is possible without further modifications of the code. The performance of the code for this test case yields between 2.5 and 3.0 MNUPS. Due to the very low Mach number and the resulting small time step, this leads to a computational time of 18 h, on the $1200 \times 1 \times 140$ grid.

Param.	Value
Domain	12m x 1.40m
Lattice	$300 \times 1 \times 45$
	$600 \times 1 \times 90$
	$1200 \times 1 \times 180$
	$1800 \times 1 \times 270$
Re	$\approx 630,000$
Fr	≈ 0.84
Ma	0.01,0.001
g	$9.81 \frac{m}{s^2}$
ν	$1 \cdot 10^{-6} \frac{m^2}{s}$

(a) Parameters.



(b) Initialization (interface position, ρ, vx_1, vx_3 from top to bottom).

Fig. 5.10. Breaking wave during shoaling.

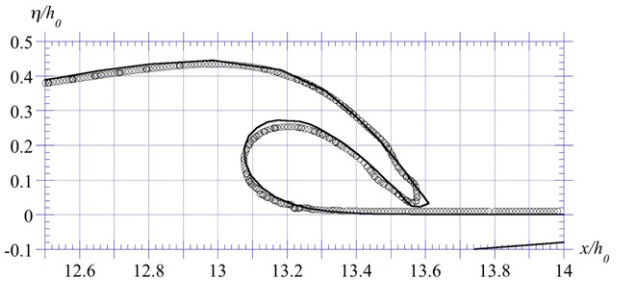


Fig. 5.11. Breaking solitary wave jet (Fig. 5.10): comparison of LBM–VOF model to the FNPF results of [53], finest grid, Ma = 0.001.

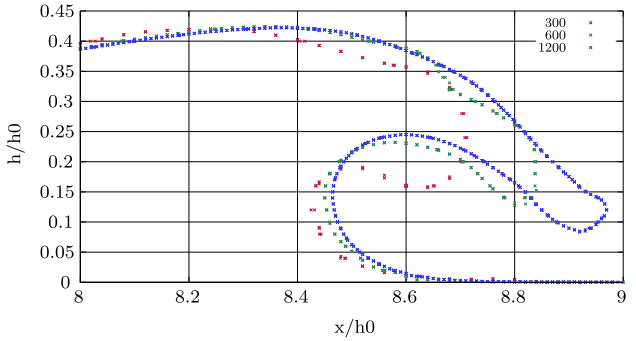
6. Conclusions and outlook

In this paper, we presented the extension and detailed validation of a hybrid scheme for free-surface flow simulations. For the simulation of flow fields, an LBM model is used, on the basis of MRT collision operators with a Smagorinsky LES subgrid scheme. The free surface is represented with a VOF method and a PLIC-based interface reconstruction. The surface normals for the PLIC are obtained by a finite-difference approximation.

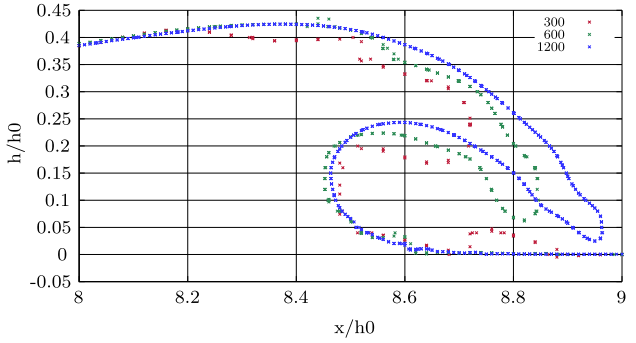
The extended hybrid algorithm was applied to several state-of-the-art validation problems, such a breaking dam or free falling jet problems, for which experimental results are available. The numerical results were found to agree with experiments quite well in all cases, and convergence to the experimental data was observed. Finally the even more demanding case of an overturning breaking wave during shoaling was successfully simulated.

For the breaking dam experiments with elevation and pressure gages, the difference between experimental and numerical results was still significant in some instances. However, it can be shown that this is due to an improper or underresolved modeling of the flow field rather than to errors in the phase advection scheme itself. This emphasizes that a physically sound modeling, including a proper choice of boundary conditions, initial conditions and turbulence models, is key to meaningful simulation results.

In future work, surface tension will be included in the hybrid free surface model. Several algorithms for doing so were published and successfully applied to VOF free surface representations, e.g. [46,54]. Moreover, a more sophisticated solid boundary representation should be introduced with solid boundaries and moving solid objects. In analogy to the PLIC representation of the fluid surface, the solid surface itself could be represented by a solid fill level, and additional surface normal information. Such a more complete solid boundary representation is likely of great importance for the accurate simulation of dynamic fluid–structure interaction problems, in which the solid boundary moves, deforms as a result of hydrodynamic forces, and inclined and curved solid boundaries appear. Moreover, the solid fill levels are time-dependent and have to be carefully updated at each time step. Initial validation of such wave–structure interaction simulations already has been done [55,56] and will be part of an upcoming publication.



(a) Ma 0.01.



(b) Ma 0.001.

Fig. 5.12. Breaking solitary wave jet (Fig. 5.10): results for two different Mach numbers and three grid configurations (300, 600 and 1200 lattice nodes in *x* direction).

With regard to further applications of the hybrid solver, the wave breaking study will be extended to full 3D cases, and more breaker types and wave heights will be examined [52,57]. This way, more insight on wave breaking processes and turbulent energy dissipation can be gained.

Acknowledgments

CJ and SG acknowledge support from grants OCE-09-27014 and OISE-PIRE-05-30203, from the US National Science Foundation.

References

[1] S. Freudiger, J. Hegewald, M. Krafczyk, A parallelization concept for a multi-physics lattice Boltzmann solver based on hierarchical grids, Progress in Computational Fluid Dynamics 8 (1–4) (2008) 168–178.
[2] S. Geller, M. Krafczyk, J. Tölke, S. Turek, J. Hron, Benchmark computations based on Lattice–Boltzmann, finite element and finite volume methods for laminar flows, Computer and Fluids 35 (2006) 888–897.
[3] J. Tölke, M. Krafczyk, Implementation of a lattice Boltzmann kernel using the compute unified device architecture developed by nVIDIA, Computing and Visualization in Science 1 (2008) 29–39.
[4] J. Tölke, M. Krafczyk, TeraFLOP computing on a desktop PC with GPUs for 3D CFD, International Journal of Computational Fluid Dynamics 22 (2008) 443–456.
[5] C. Janßen, M. Krafczyk, Free surface flow simulations on GPGPUs using LBM, Computers and Mathematics with Applications 61 (12) (2011) 3549–3563.
[6] F.H. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics of Fluids 8 (1965) 2182–2189.
[7] J. Welch, F. Harlow, J. SHannon, B. Daly, The MAC method: a computing technique for solving viscous, incompressible, transient fluid flow problems involving free surfaces, Los Alamos Scientific Laboratory Report LA-3425.
[8] J.J. Monaghan, Simulating free surface flows with SPH, Journal of Computational Physics 110 (2) (1994) 399–406.
[9] R.A. Dalrymple, A. Herault, Levee breaching with GPU-SPHysics code, in: Proceedings of the 4th International SPHERIC Workshop Nantes, France, May 27–29 2009.
[10] A. Herault, G. Bilotta, R.A. Dalrymple, SPH on GPU with CUDA, Journal of Hydraulic Research 48 (Extra Issue) (2010) 74–79.
[11] E. Walhorn, Ein simultanes Berechnungsverfahren für Fluid-Struktur-Wechselwirkungen mit finiten Raum-Zeit-Elementen, Ph.D. Thesis, Institut für Statik, TU Braunschweig 2002.
[12] S. Tanaka, K. Kashiyama, ALE finite element method for FSI problems with free surface using mesh re-generation method based on background mesh, International Journal of Computational Fluid Dynamics 20 (2006) 229–236.
[13] C. Hirt, B. Nichols, Volume of fluid method for dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–221.
[14] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Tech. Rep., Los Alamos National Laboratory, Oct 2005.
[15] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2) (2007) 2096–2132.

[16] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, Journal of Computational Physics 79 (1998) 12–49.
[17] A.K. Gunstensen, D.H. Rothman, S. Zaleski, G. Zanetti, Lattice Boltzmann model of immiscible fluids, Physical Review A 43 (8) (1991) 4320–4327.
[18] I. Ginzburg, K. Steiner, Lattice Boltzmann model for free-surface flow and its application to filling process in casting, Journal of Computational Physics 185 (1) (2003) 61–99.
[19] C. Körner, M. Thies, T. Hofmann, N. Thürey, U. Rüde, Lattice Boltzmann model for free surface flow for modeling foaming, Journal of Statistical Physics 121 (18) (2005) 179–196.
[20] N. Thürey, U. Rüde, Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids, Computing and Visualization in Science (2008).
[21] P. Lallemand, L.-S. Luo, Y. Peng, A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two dimensions, Journal of Computational Physics 226 (2) (2007) 1367–1384.
[22] C. Janßen, M. Krafczyk, A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids, Computers and Mathematics with Applications 59 (7) (2010) 2215–2235.
[23] Y.H. Quian, D. d'Humieres, P. Lallemand, Lattice BGK models for Navier Stokes equations, Europhysics Letters 17 (1992) 479–484.
[24] U. Frisch, D. d'Humieres, B. Hasslacher, P. Lallemand, Y. Pomeau, J.-P. Rivet, Lattice gas hydrodynamics in two and three dimensions, Journal of Complex Systems 1 (1987) 75–136.
[25] M. Junk, A. Klar, L.-S. Luo, Asymptotic analysis of the lattice Boltzmann equations, Journal of Computational Physics 210 (2) (2005) 676–704.
[26] P.L. Bhatnagar, E.P. Gross, M. Krook, A Model for collision processes in gases, I, small amplitude processes in charged and neutral one-component systems, Physical Review 94 (3) (1954) 511–525.
[27] D. d'Humieres, I. Ginzburg, M. Krafczyk, P. Lallemand, L.-S. Luo, Multiple relaxation-time lattice Boltzmann models in three-dimensions, Royal Society of London Philosophical Transactions Series A 360 (2002) 437–451.
[28] I. Ginzburg, D. d'Humieres, Multireflection boundary conditions for lattice Boltzmann models, Physical Review E 68 (6) (2003) 066614.1–066614.30.
[29] P. Lallemand, L.-S. Luo, Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability, Physical Review E 61 (2000) 6546–6562.
[30] M. Krafczyk, J. Tölke, L.-S. Luo, Large-eddy simulations with a multiple-relaxation-time LBE model, International Journal of Modern Physics B 17 (2003) 33–39.
[31] M. Bouzidi, M. Firdaouss, P. Lallemand, Momentum transfer of a lattice-Boltzmann fluid with boundaries, Physics of Fluids 13 (2001) 3452–3459.
[32] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Physical Review E 65 (4) (2002) 046308.1–046308.6.
[33] R. Mei, L.-S. Luo, P. Lallemand, D. d'Humieres, Consistent initial conditions for lattice Boltzmann simulations, computers & Fluids 35 (8–9) (2006) 855–862.
[34] S.T. Grilli, R. Subramanya, Numerical modeling of wave breaking induced by fixed or moving boundaries, Computational Mechanics 17 (6) (1996) 374–391.
[35] S. Grilli, J. Horrillo, Numerical generation and absorption of fully nonlinear periodic waves, Journal of Engineering Mechanics 123 (10) (1997) 1060–1069.
[36] D. Youngs, Time-dependent multimaterial flow with large fluid distortion, in: K. Morton, M. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, 1982, pp. 273–285.
[37] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, Journal of Computational Physics 152 (2) (1999) 423–456.
[38] E.G. Puckett, J.S. Saltzman, A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics, Physica D 60 (1–4) (1992) 84–93.
[39] B. Parker, D. Youngs, Two and three dimensional Eulerian simulation of fluid flow with material interfaces, Tech. Rep., UK Atomic Weapons Establishment, February 1992.
[40] R.P. Brent, Algorithms for Minimization Without Derivatives, in: American Scientist, vol. 61, Prentice Hall, 1973, p. 374. (reviewed); Mathematical Programming 4 (1973) 349; Computer Journal 16 (1973) 314; Mathematics of Computation 28 (1974) 865–866.
[41] S. Freudiger, Entwicklung eines parallelen, adaptiven, komponentenbasierten Strömungskerns für hierarchische Gitter auf Basis des Lattice Boltzmann Verfahrens, Ph.D. Thesis, TU Braunschweig, 2009.
[42] S. Geller, Ein explizites Modell für die Fluid-Struktur-Interaktion basierend auf LBM und *p*-fem, Ph.D. Thesis, Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften der Technischen Universität Carolo-Wilhelmina zu Braunschweig, 2010.
[43] J.C. Martin, W.J. Moyce, Part IV, an experimental study of the collapse of liquid columns on a rigid horizontal plane, Royal Society of London Philosophical Transactions Series A 244 (1952) 312–324.
[44] A. Kölke, Modellierung und Diskretisierung bewegter Diskontinuitäten in randgekoppelten Mehrfeldsystemen, Ph.D. Thesis, Technische Universität Braunschweig, 2005.
[45] K. Kleefsman, Water impact loading on offshore structures—a numerical study, Ph.D. Thesis, University of Groningen, 2005.
[46] R. Wemmenhove, Numerical simulation of two-phase flow in offshore environments, Ph.D. Thesis, Rijksuniversiteit Groningen, 2008.
[47] A. Veldman, ComFLOW—dambreak experiment. URL http://www.math.rug.nl/~veldman/comflow/dambreak.html, 2005.
[48] E. Marchi, On the free overfall, Journal of Hydraulic Research 31 (1993) 777–790.
[49] M. Tanaka, The stability of solitary waves, Physics of Fluids 29 (3) (1986) 650–655.
[50] S. Guignard, S.T. Grilli, R. Marcer, V. Rey, Computation of shoaling and breaking waves in nearshore areas by the coupling of BEM and VOF methods, in: Proceedings of the 9th Offshore and Polar Engineering Conference III, 1999, pp. 304–309.
[51] C. Lachaume, B. Biausser, S.T. Grilli, S. Guignard, Modeling of breaking and post-breaking waves on slopes by coupling of BEM and VOF methods, in: Proceedings of the 13th International Offshore and Polar Engineering Conference, 2003, pp. 353–359.
[52] B. Biausser, S. Grilli, P. Fraunie, R. Marcer, Numerical analysis of the internal kinematics and dynamics of three-dimensional breaking waves on slopes, International Journal of Offshore and Polar Engineering 14 (4) (2004) 247–256.
[53] S. Grilli, I. Svendsen, R. Subramanya, Breaking criterion and characteristics for solitary waves on slopes, Journal of Waterway Port Coastal and Ocean Engineering 123 (3) (1997) 102–112.
[54] M. Meier, G. Yadigaroglu, B.L. Smith, A novel technique for including surface tension in PLIC-VOF methods, European Journal of Mechanics—B/Fluids 21 (1) (2002) 61–73.
[55] C.F. Janßen, Kinetic approaches for the simulation of non-linear free surface flow problems in civil and environmental engineering, Ph.D. Thesis, TU Braunschweig, 2011.
[56] S. Geller, C. Janssen, S. Kollmannsberger, M. Krafczyk, E. Rank, The lattice Boltzmann method for fluid–structure interaction phenomena, in: Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, 2011.
[57] P. Guyenne, S. Grilli, Numerical study of three-dimensional overturning waves in shallow water, Journal of Fluid Mechanics 547 (2006) 361–388.