Proceedings of the Twenty-second (2012) International Offshore and Polar Engineering Conference
Rhodes, Greece, June 17–22, 2012
Copyright © 2012 by the International Society of Offshore and Polar Engineers (ISOPE)
ISBN 978-1-880653-94–4 (Set); ISSN 1098-6189 (Set)

www.isope.org

Efficient simulations of long wave propagation and runup using a LBM approach on GPGPU hardware

Christian F. Janßen[1], *Stéphan T. Grilli*[1] *and Manfred Krafczyk*[2]
1. University of Rhode Island, Department of Ocean Engineering, Narragansett, RI, USA
2. Technische Universitaet Braunschweig, Institute for Computational Modeling in Civil Engineering, Germany

ABSTRACT

We present an efficient implementation of the Lattice Boltzmann method (LBM) for the numerical simulation of the propagation of long ocean waves (e.g., tsunamis), based on the Nonlinear Shallow Water (NSW) wave equation. The LBM solution of NSW equations is fully nonlinear and it is assumed that the surface elevation is single-valued (hence, waves do not break or overturn). For the treatment of wet-dry states, a simple shoreline step-slot algorithm is used.

The NVIDIA CUDA framework is used for the implementation, which gives access to the computational power of General Purpose Graphical Processing Units (GPGPUs). The initial analysis of LBM results for standard analytical benchmark problems shows a good agreement with the reference solutions. For all benchmarks, the run times of the numerical simulations are on the same order as the time scale of the real world event, or even less. The presented applications include wave runup studies on a plane beach and a more complex three-dimensional beach, as proposed in the tsunami community as part of the so-called Catalina- (Liu et al., 2008) and PMEL sets of benchmark problems. Finally, the results and the performance of the LBM solver are compared to those of the Boussinesq solver FUNWAVE.

KEY WORDS: Lattice Boltzmann method, nonlinear shallow water equations, long wave propagation, GPGPU, runup benchmark tests, tsunami forecasting

INTRODUCTION

The recent March 12, 2011 Tohuku tsunami in Japan, which killed nearly 20,000 people and caused massive destruction on the North shore of Honshu, including a major nuclear disaster at the Fukushima-Daichi power plant, reminded us how vulnerable modern society is to such natural hazard (Grilli et al., 2012). While no reliable warning exists for earthquakes, once these occur, tsunami warning centers can quickly issue a prediction of whether a tsunami was generated. With the addition of realtime observations of tsunami elevation made at some of the nearly 40 deep water (DART) buoys deployed in the world oceans, such predictions can provide a rapid assessment of tsunami source parameters. These, in turn, can be used in long wave propagation models to simulate tsunami propagation and issue realtime forecast of tsunami coastal impact. Such forecast are particularly important for far-field (trans-oceanic) areas, since it may take the tsunami several hours to reach those, which allows for proper warning and evacuation, if necessary.

Although non-hydrostatic models such as Boussinesq models (e.g., Wei et al., 1995; Shi et al., 2012; Grilli et al., 2012) are increasingly used in research and damage assessment following major tsunami events, the standard approach in realtime tsunami forecasting is still based on using models based on Nonlinear Shallow Water (NSW) equations, which are a depth-integrated (hydrostatic) version of Navier-Stokes equations (e.g., Kowalik and Murty, 1993). The NSW equations are accurate for long wave propagation and runup problems, in which the horizontal length scale is much larger than the vertical length scale, such as for most tsunamis. Besides tsunamis, these equations are widely used in the field of ocean engineering (e.g., tide and ocean modeling) and atmospheric modeling, where one length scale is dominant. If vertical variations must be taken into account, these can be separated from the horizontal ones, resulting in a set of shallow water equations for a series of horizontal fluid layers (i.e., multilayer NSW).

Although the Lattice Boltzmann method (LBM) is not widely known in the community working on free surface flow problems, several groups have already applied LBMs to standard shallow water benchmark problems and test cases. Thus, Frandsen (2006) presented a so-called D2Q9 LBM implementation for the simulation of wave runup on a sloping beach. Thömmes et al. (2007) applied a similar LBM to test cases including bed slope and friction terms. The main focus of their work was to demonstrate the ability of the method to cope with complex geometries and irregular bathymetry. This was illustrated by the simulation of the mean flow in the Strait of Gibraltar. These LBMs were based on single relaxation time (SRT) collision operators. Tubbs (2010), however, applied an extended LBM to NSW eqs. with two- and multiple-relaxation time (TRT and MRT, resp.) collision operators. Moreover, this author extended the standard LBM to a multilayer approach that considers vertical, three-dimensional effects. The model is capable of simulating wind- and density-driven circulation over irregular bathymetry.

It has been shown in various publications that LBM methods are especially well suited for implementation in a General Purpose Graphical Processing Unit (GPGPU) context. The latter provides a large amount of cores (i.e. 448 on a NVIDIA Tesla C2070), resulting in update rates of more than 500 million nodal updates per second (NUPS), for a single precision implementation, on a state-of-the-art GPGPU. With such computing power, the real time simulation of long wave propagation over fine ocean-scale grids might be achievable on a desktop computer, which would represent a significant improvement, for instance, in operational tsunami forecasting.

LBM FOR SHALLOW WATER EQUATIONS

As already indicated, when the horizontal flow length scale is much larger than the vertical length scale, which is the case for numerous applications in ocean engineering, a depth-integrated version of Navier-Stokes equations can be used. According to continuity equation, the variations of the vertical velocity component of the flow field are small in such a case. Considering the momentum equation, this automatically demands the vertical pressure gradient to be nearly hydrostatic and the dynamic pressure contribution to be small. Assuming a constant horizontal velocity field over depth, a vertical integration of the momentum equation allows to remove the vertical velocity from the equations. [Although the vertical velocity is not present in the NSW equations, it is not necessarily zero and can be obtained in a post-processing step by applying continuity equation.] The resulting NSW equations read:

$$\partial_t h + \nabla(h\bar{\boldsymbol{v}}) = 0 \tag{1}$$

$$\partial_t (h\bar{\boldsymbol{v}}) + \nabla(h\bar{\boldsymbol{v}} \cdot \bar{\boldsymbol{v}}) = -\nabla\left(\frac{1}{2}gh^2\right) + \nu\nabla^2(h\bar{\boldsymbol{v}}) + \boldsymbol{F} \tag{2}$$

where ∇ denotes the horizontal gradient operator (i.e., with respect to x and y), $\bar{\boldsymbol{v}}$ is the horizontal velocity vector in water depth $h(x,y)$, and $\boldsymbol{F}$ is a forcing term defined as

$$\boldsymbol{F} = -gh\nabla z_b + \frac{\tau_w}{\rho} + \frac{\tau_b}{\rho} + \boldsymbol{E} \tag{3}$$

including the effects of bottom elevation z_b, bed shear stress τ_b, wind shear stress τ_w and a Coriolis term $\boldsymbol{E}$.

The NSW equations have been discretized and solved with various numerical methods. An LBM will be used in the current work. Such models are derived from the kinetic gas theory and describe the evolution of particle distribution functions f_α, in space and time, i.e., they consider the problem on a mesoscopic level. LBM results have been shown to converge to those of macroscopic NS equations and, hence, LBMs can be viewed as another class of NS solvers. Therefore, mesoscopic LBMs may also be used for the simulation of shallow water flows. Similar approaches to those used for the standard derivation of shallow water equations are followed. Basically, the fluid pressure is related to the water depth by using a different equation of state. A comprehensive review of LBMs for shallow water equations was given by Zhou (2003), including elaborate discussions of the treatment of bottom elevation and forcing terms. In the following, the resulting LBM for shallow water equations is briefly reviewed. Its full derivation from continuous models and the Chapman-Enskog expansions, which show that its solution converges towards depth-averaged Navier-Stokes equations, will not be given here but can be found in reference. The LBM for NSW reads:

$$f_\alpha(\boldsymbol{x} + \boldsymbol{e}_\alpha \Delta t, t + \Delta t) - f_\alpha(\boldsymbol{x}, t) = \Omega_\alpha \tag{4}$$

with particle distribution functions f_α, collision operator Ω_α, a microscopic reference velocity $c = \Delta x/\Delta t$ which is related to the grid spacing Δx and the time step Δt, and the particle direction α. The macroscopic values for water depth h and velocity $\boldsymbol{v}$ are low order moments of the distribution functions:

$$\sum_{\alpha=0}^{b-1} f_\alpha = h \qquad \sum_{\alpha=0}^{b-1} \boldsymbol{e}_\alpha f_\alpha = h\boldsymbol{v} \tag{5}$$

where b is the number of discrete particle directions. Here, a D2Q9 model is used, which introduces $b = 9$ discrete particle velocities $\boldsymbol{e}_\alpha$,

$$\boldsymbol{e}_\alpha = c \cdot \left\{ \begin{array}{ccccccccc} 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 \end{array} \right\}, \tag{6}$$

that are commonly abbreviated as geographic directions plus the rest particle velocity $\boldsymbol{e}_0$, (REST,E,N,W,S,NE,NW,SW,SE). The relation to the NS model can be found from the zeroth-order moment of the particle distribution functions, where the flow density ρ has been replaced by the water depth h in the shallow water model. A simple approximation for the collision term Ω_α is obtained from the single relaxation time (SRT) approximation, developed by Bhatnager, Gross, and Krook (BGK) (Bhatnagar et al., 1954):

$$\Omega_\alpha = -\frac{1}{\tau}(f_\alpha - f_\alpha^{eq}), \tag{7}$$

which locally drives the particle distribution functions to an equilibrium state f_α^{eq}. In the LBM-NS solution, the dimensionless relaxation time τ is linked to the kinematic viscosity. Here, we similarly find:

$$\nu = \frac{c^2 \Delta t}{6}(2\tau - 1). \tag{8}$$

The main differences in the LBM solution of NSW equations are the definition of the reference velocity c, which is not set to unity, and of the local equilibrium distribution functions f_α^{eq}. For LBM-NSW models derived on the basis of a D2Q9 velocity the latter read:

$$f_\alpha^{eq}(h, \boldsymbol{v}) = \begin{cases} h - \frac{5gh^2}{6c^2} - \frac{2h}{3c^2}\boldsymbol{v}^2, & \alpha = 0 \\ \frac{gh^2}{6c^2} + \frac{h}{3c^2}\boldsymbol{e}_\alpha\boldsymbol{v} + \frac{h}{2c^4}(\boldsymbol{e}_\alpha\boldsymbol{v})^2 - \frac{h}{6c^2}\boldsymbol{v}^2, & \alpha = 1\ldots4 \\ \frac{gh^2}{24c^2} + \frac{h}{12c^2}\boldsymbol{e}_\alpha\boldsymbol{v} + \frac{h}{8c^4}(\boldsymbol{e}_\alpha\boldsymbol{v})^2 - \frac{h}{24c^2}\boldsymbol{v}^2, & \alpha = 5\ldots8 \end{cases} \tag{9}$$

In the more advanced MRT model (d'Humieres et al., 2002), the particle distribution functions are transformed into moment space before relaxation. In the following, different relaxation rates are used to increase stability and allow for the development of more accurate boundary conditions (Ginzburg and D'Humieres, 2003). The collision operator for MRT thus reads:

$$\boldsymbol{\Omega} = -\mathsf{M}^{-1}\cdot\mathsf{S}\cdot\left(\mathsf{M}\cdot\mathbf{f}-\mathbf{m}^{eq}\right) \tag{10}$$

where 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^{eq}_{α} are the equilibrium moments:

$$m_0^{eq} = h,\ m_1^{eq} = \frac{h}{c^2}(3h(u_x^2+u_y^2+gh)) - 4h \tag{11}$$

$$m_2^{eq} = 4h - \frac{3h}{2c^2}(2u_x^2+2u_y^2+3gh) \tag{12}$$

$$m_3^{eq} = \frac{h}{c}u_x,\ m_4^{eq} = -\frac{h}{c}u_x,\ m_5^{eq} = \frac{h}{c}u_y,\ m_6^{eq} = -\frac{h}{c}u_y \tag{13}$$

$$m_7^{eq} = \frac{h}{c^2}(u_x^2-u_y^2),\ m_8^{eq} = \frac{h}{c^2}u_x u_y \tag{14}$$

The diagonal collision matrix $\mathsf{S} = s_{\alpha,\alpha}$ contains the relaxation parameters, which are partly related to the kinematic viscosity ν via:

$$s_7 = s_8 = \frac{1}{\tau} = \frac{2}{6\frac{\nu}{c^2\Delta t}+1}. \tag{15}$$

The remaining relaxation rates can be freely selected in the range $[0,2]$ and are tuned to improve stability (Lallemand and Luo, 2000). 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 d'Humieres et al. (2002); we will use $s_\alpha = 1.0$ for the remaining parameters.

Body forces

In shallow water models, the body force term serves to model bed stress, wind-induced shear stress or seafloor slope (see Eq. 3). A simplified force term resulting from a sloping bottom reads:

$$\boldsymbol{F} = -gh\nabla z_b \tag{16}$$

with bottom elevation $z_b(x,y)$. Zhou (2003) discusses various approaches for the detailed evaluation of the forcing term and finally proposes a centered scheme, where the term is evaluated at the mid-point between two neighboring lattice nodes:

$$\begin{aligned}\boldsymbol{F}_\alpha &= \boldsymbol{F}_\alpha\left(\boldsymbol{x}+\tfrac{1}{2}\boldsymbol{e}_\alpha\Delta t, t+\tfrac{1}{2}\Delta t\right)\\ &= -gh\left(\boldsymbol{x}+\tfrac{1}{2}\boldsymbol{e}_\alpha\Delta t, t+\tfrac{1}{2}\Delta t\right)\nabla z_b\left(\boldsymbol{x}+\tfrac{1}{2}\boldsymbol{e}_\alpha\Delta t\right).\end{aligned} \tag{17}$$

The modified Lattice Boltzmann equation reads

$$f_\alpha(t+\Delta t, \boldsymbol{x}+\boldsymbol{e}_\alpha\Delta t) - f_\alpha(t,\boldsymbol{x}) = \Omega_\alpha + F_\alpha \tag{18}$$

where F_α is a link-specific contribution of the body force vector $\boldsymbol{F}_\alpha$ per discrete particle direction α, which is evaluated as

$$F_\alpha = \frac{\Delta t}{6c^2}\boldsymbol{e}_\alpha\boldsymbol{F}_\alpha. \tag{19}$$

Introducing the arithmetic mean for h and a central finite difference scheme for the bottom slope gradient $\nabla\boldsymbol{z}_b$ yields the following expression to evaluate the force term:

$$F_\alpha = \frac{\Delta t}{6c^2}\boldsymbol{e}_\alpha(-g)\frac{h(\boldsymbol{x})+h(\boldsymbol{x}+\boldsymbol{e}_\alpha\Delta t)}{2}\,\frac{z_b(\boldsymbol{x})-z_b(\boldsymbol{x}+\boldsymbol{e}_\alpha\Delta t)}{\|\boldsymbol{e}_\alpha\Delta t\|}\,\frac{\boldsymbol{e}_\alpha}{\|\boldsymbol{e}_\alpha\|} \tag{20}$$

where h and z are water depth and bottom boundary elevation, respectively, of the corresponding lattice nodes. After simplifications, we have:

$$F_\alpha = -\frac{g}{12c^2}(h(\boldsymbol{x})+h(\boldsymbol{x}+\boldsymbol{e}_\alpha\Delta t))(z_b(\boldsymbol{x})-z_b(\boldsymbol{x}+\boldsymbol{e}_\alpha\Delta t)). \tag{21}$$

Boundary conditions

Boundary conditions in the LB context have to be directly specified for the particle distribution functions and can not be specified by solely prescribing macroscopic quantities for the primary variables, here, water height and depth-averaged velocity. For no flow boundary conditions, a simple bounce back scheme may be used. In analogy to the fully three dimensional flow kernel, particles bounce off the boundaries and are reflected back into the domain,

$$f_\alpha(\boldsymbol{x}, t+\Delta t) = f_{\bar{\alpha}}(\boldsymbol{x}, t), \tag{22}$$

where $\bar{\alpha}$ is the inverse direction to α. No-slip and slip boundary conditions can be modeled this way. In both cases, the velocity perpendicular to the wall and hence the flow through the wall equals zero. For moving boundaries, a modified bounce-back scheme (Ladd, 1994) serves to adjust the resulting fluid velocity; a velocity-dependent term is added to the particle distribution functions at each time step. Note that this boundary conditions also leads to a zero-lateral velocity, as it is a modified no-slip boundary condition. For non-trivial Dirichlet and Neumann boundary conditions, the modified bounce-back scheme for moving boundaries may be used. This scheme also is proposed by Zhou (2003), although his notation differs. The simplified expressions for a west boundary, for instance, read

$$f_E = f_W + \frac{2q_x}{3c}, \tag{23}$$

$$f_{NE} = f_{SW} + \frac{q_x}{6c} + \frac{f_S - f_N}{2}, \tag{24}$$

$$f_{SE} = f_{NW} + \frac{q_x}{6c} + \frac{f_N - f_S}{2}, \tag{25}$$

where the eastward-pointing distribution functions (E,NE,SE) are missing. We see that only one boundary flux term q_x is included in these equations. Hence, if a water depth $\bar{h}$ must be specified instead of a flux $\bar{q}_x$, it can be converted to an equivalent flux as: $\tilde{q}_x = \bar{h}v_x$. In the literature, this formulation is used for various boundary conditions, whether velocity, flux, or water depth is specified (see, Thömmes et al. (2007)). In our test cases, however, this has been shown to lead to numerical oscillations, likely because the product of two factors is used for the flux calculation and at least one factor (v_x) has to be extrapolated in either space or time. Hence, we proposed an alternative formulation, similar to the anti-bounce back pressure boundary condition in standard LBMs (Körner et al., 2005), to set the boundary water depth h_B at the boundary $\boldsymbol{x}_B$:

$$f_\alpha(\boldsymbol{x}_B, t+\Delta t) = -f_{\bar{\alpha}}(\boldsymbol{x}_B, t) + f_{\bar{\alpha}}^{eq}(h_B, \boldsymbol{v}_B) + f_\alpha^{eq}(h_B, \boldsymbol{v}_B), \tag{26}$$

with equilibrium distribution functions f_α^{eq} and $f_{\bar{\alpha}}^{eq}$ according to Eq. 9. The boundary value for the velocity $\boldsymbol{v}_B$ is obtained by extrapolation.

Stability requirements

Nearly all explicit numerical methods must satisfy more or less severe stability requirements. For the NSW-LBM, these are perhaps less straightforward to establish than for standard LBMs. Zhou (2003) identified the most stringent requirements as:

$$\mathrm{CFL} = \frac{|\boldsymbol{v}|}{|\boldsymbol{c}|} \leq 1.0 \quad \text{and} \quad \mathrm{Fr} = \frac{|\boldsymbol{v}|}{\sqrt{gh}} \leq 1.0 \tag{27}$$

$$\nu = \frac{c^2 \Delta t}{6}(2\tau - 1) > 0 \quad \text{and} \quad \frac{\mathrm{CFL}}{\mathrm{Fr}} = \frac{\sqrt{gh}}{|\boldsymbol{c}|} \leq 1.0 \tag{28}$$

where CFL denotes the Courant number and Fr the Froude number. We see that, in addition to their upper bound of unity, their ratio must be smaller than one as well, and a negative viscosity coefficient is not permitted. Typically, the space discretization is set first, by specifying Δx. Then, a proper time step Δt is selected, and the macroscopic reference velocity $c = \Delta x / \Delta t$ is defined.

Moving shoreline algorithm

Solutions of the NSW equations are only valid for a water depth h greater than zero. For a moving shoreline, however, there will be areas of wave runup-rundown where parts of the computational domain will successively be dry or wet. The numerical treatment of such drying or wetting boundaries requires additional considerations. In the NSW-LBM, a moving shoreline algorithm is used to artificially adjust the domain size in the wave runup-rundown region, depending on the rising or falling water level at the boundary nodes. In this simple step-slot algorithm, the water level in the vicinity of solid boundaries is checked via evaluating:

$$z_b(\boldsymbol{x}) + h(\boldsymbol{x}) > z_b(\boldsymbol{x} + \boldsymbol{e}_\alpha \Delta t) + \delta \quad \forall \alpha \tag{29}$$

where z_b denotes the elevation of bottom LBM grid cells, h is the water depth, and δ is a proper threshold value. If the water is overflowing a given cell, the neighboring inactive solid cell is activated and converted to a fluid cell. The equilibrium distribution functions for an extrapolated water elevation and velocity are defined as:

$$f_\alpha = f_\alpha^{eq}(h_{init}, \boldsymbol{v}_{init}) \quad \forall \alpha. \tag{30}$$

This approach leads to accurate results for test cases where the characteristic direction of wave propagation is aligned with the coordinate axes. For more complex bottom topographies, however, more advanced runup models must be used. Balzano (1998) analyzed and discussed ten different moving shoreline algorithms, mainly for implicit methods, and compared these for demanding runup test cases. Lynett et al. (2002) proposed an extrapolation technique to determine artificial fluid properties in the dry region, near the wet-dry interface. It appears from these references that the calculation of h_{init} and $\boldsymbol{v}_{init}$ is a challenging problem. In a very simple model, h_{init} is set to a constant value (with $h_{init} > \delta$) and the initial velocities are set to zero ($\boldsymbol{v}_{init} = \mathbf{0}$). This constant extrapolation of water levels works well for most test cases, but might lead to instabilities in some cases, due to inconsistent initialization of velocities. In such cases, a linear extrapolation might improve the runup, but based on Eq. 27, Froude number stability limitations must also be considered. Specifically, the NSW-LBM is only applicable to subcritical flows, so that an upper bound or similar technique has to be satisfied during runup, to limit the Froude number value (i.e., such that $|\boldsymbol{v}_{init}| \leq \sqrt{gh_{init}}$) and avoid inconsistent node initializations. It was observed that the present runup model is very sensitive to such limitations, which may lead to instabilities, especially for low fluid viscosities ν. Specifically, an improper choice of h_{init} induces shock waves that can lead to instabilities. Similar effects have been reported in NS-LB simulations of fluid-structure interaction problems, where nodes have to be initialized with proper values for fluid pressure and velocity (Geller, 2006). So far, to tackle this problem, we used increased values for the fluid viscosity. In future work, both a proper extrapolation and an enhanced initialization of particle distribution functions will be tested and implemented in the runup model.

VALIDATION AND RESULTS

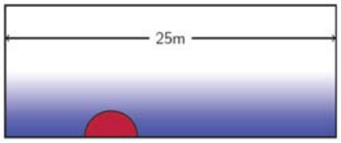
In the following, the NSW-LBM code, implemented on a GPGPU, is applied to three benchmark problems widely used in the tsunami community: (i) 2D flow over a semi-circular obstacle; (ii) 2D tsunami runup over a plane beach; and (iii) tsunami runup over a complex 3D natural beach (Monai valley; Liu et al. (2008)). Note, in those, benchmark problem (iii) is a demanding case, with very complex bottom topography. In all cases, the LBM solution is compared to the available analytical/numerical or experimental reference solutions, and to other results from the literature. In addition, the performance of the GPGPU-LBM solver is compared to results of FUNWAVE, a state-of-the-art Boussinesq model used for tsunami propagation, in terms of accuracy and run-times.

2D Flow over a bump

A very simple test case for the shallow water solver, which includes a complex bottom elevation, is that of the flow over a bump (Zhou, 2003; Thömmes et al., 2007) (Fig. 1).

Grid	$nx \times ny$ nodes Fig. 2, Tab. 1
c	15 m/s
$1/\tau$	1/1.5
BC x	$q = 4.42m^2/s$ $h = 2m$
BC y	periodic

(a) Parameters



(b) Geometry

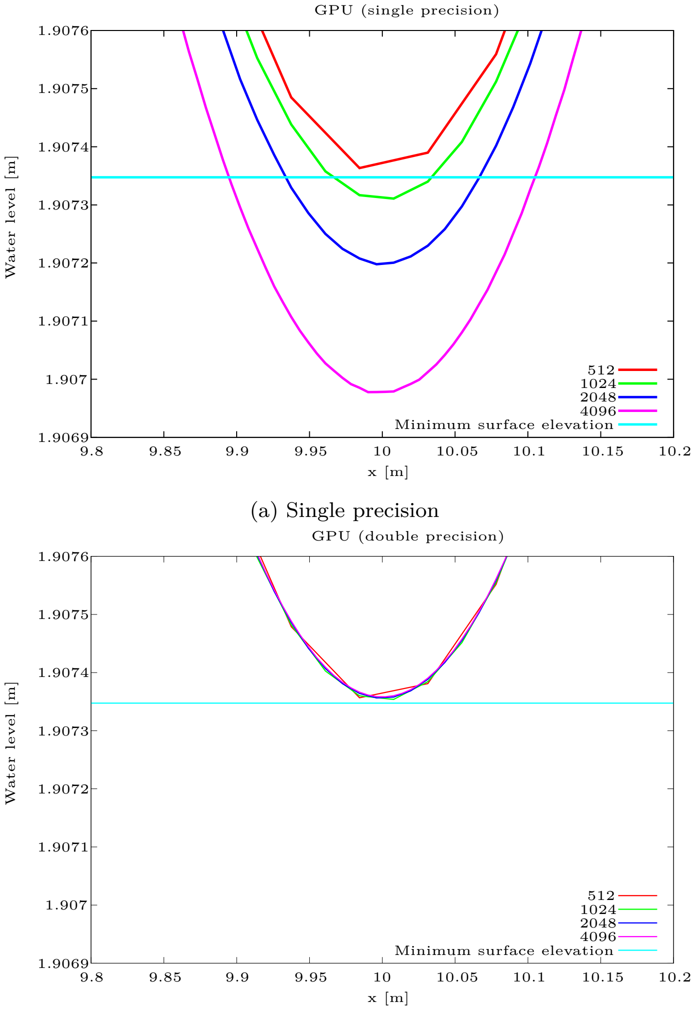
Figure 1: Flow over a bump

In simulations, the channel is 25 m long, and the bottom elevation is defined as:

$$z_b(x) = \begin{cases} 0.2m - 0.05\frac{1}{m}(x - 10m)^2 & 8m < x < 12m \\ 0.0m & x \leq 8m, x \geq 12m \end{cases} \tag{31}$$

The function $z_b(x)$ is continuous, but not continuously differentiable. The discontinuities of its slope at $x = 8$ and 12 m lead to a demanding test case in terms of bottom elevation forcing terms. At the leftward inflow boundary, a constant mass flux of $q = 4.42$ m^2/s is specified, while the outflow boundary specifies the water level at $h = 2$ m. Hence, the resulting flow velocity far away from the obstacle is $v = 2.21$ m/s, which is also used in the grid initialization process. Values for macroscopic reference velocity and relaxation time are set in accordance to Zhou (2003) to $c = 15$ m/s and $\tau = 1.5$.

Results show that, above the obstacle, the flow velocity increases while the water level decreases below the surrounding mean water level. This could be expected from a simple application of mass conservation and Bernoulli equations: the surface elevation contributes linearly to the overall energy while the velocity has a quadratic contribution; hence when velocity increases due to flow constriction over the obstacle, the water level decreases. Specifically, in the cross section directly above the obstacle, at $x = 10$ m, Bernoulli equation predicts a minimum water level of 1.90735 m and a maximum flow velocity of 2.5888 m/s.



(a) Single precision

(b) Double precision

Figure 2: Water surface elevation above the obstacle (at $x = 10$ m; Fig. 1) for single and double precision GPU simulations, as a function of grid resolution, in comparison to the expected minimum water level.

Fig. 2 shows results for the water depth over the obstacle, computed in several gradually refined grids and for simulations performed with either single or double precision floating point variables. 512 to 8192 lattice nodes have been used in flow direction (25m domain length), leading to a grid spacing from $\Delta x_{512} \approx$ 0.049m to $\Delta x_{8192} \approx$ 0.0031m on the finest grid. As the macroscopic reference velocity is kept constant on all grid configurations, the time step is simultaneously refined, from $\Delta t_{512} = \Delta x_{512}/c \approx$ 0.0033s down to $\Delta t_{8192} \approx$ 0.0002s. Convergence can be clearly observed, although the single precision kernel (Fig. 2a) shows inconsistent convergence behavior. If double precision variables are used (Fig. 2b), the results converge to the correct analytical solution. Similar effects were reported by Schönherr et al. (2011) in the GPU simulation of bulk flows. These authors compared compressible and incompressible LBM formulations and found a higher sensitivity of the compressible LBM approach, due to the treatment of single precision float variables. As the present NSW-LBM is analogous to the compressible formulation for bulk flows, it apparently suffers from similar numerical effects. More elaborate discussions of the effects of single and double precision variables on the accuracy of LBM simulations can be found in Harvey et al. (2008). Nonetheless the relative error in water depth h above the obstacle is below 10^{-4} and hence does not have a serious impact on the quality of the overall results.

ny	Collision model	512	1024	2048	4096	8192
16	MRT	9	21	36	65	101
	SRT	11	21	36	57	109
64	MRT	34	59	93	163	233
	SRT	34	56	98	157	230
128	MRT	61	99	153	227	297
	SRT	58	92	156	237	296

Table 1: Performance (Million node updates per second, MNUPS) of the simulation of the cases of Figs. 1 and 2, for different grid configurations ($nx \times ny$ lattice nodes in x and y direction, resp.) and LBM collision models

The performance of the solver for both SRT and MRT collision operators on different grid configurations is given in Tab. 1. First, the performance of the SRT and MRT models is almost the same, as expected, because the transformation into moment space and back (Eq. 10) can explicitly be implemented in the code, without time-consuming matrix-vector operations or matrix inversions. Second, the performance of the solver is highly dependent on grid size. The bigger the computational domain, the higher the resulting performance. On the smaller grids, the maximum performance can not be achieved, as the amount of computations is not large enough to schedule the threads (and the memory accesses) efficiently, and the computational overhead of boundary conditions is large. Third, the maximum performance for this test case yields about 300 MNUPS, which is on the same order of magnitude as for LBM-GPGPU solvers for the full NS equations (500-600 MNUPS). The performance loss is mainly due to the additional operators needed for the calculation of the bottom elevation forcing term (Eq. 21), that requires additional non-local memory accesses and computations.

2D tsunami runup over a plane beach

After the previous basic validation of the NSW-LBM, we solve the standard benchmark problem of 2D tsunami runup onto a plane beach. This problem was initially proposed as part of the “Third international workshop on long-wave runup models”, which took place in 2004 on Catalina Island, California (Liu et al., 2008). The benchmark reference data was obtained from the analytical solution of NSW equations of Carrier et al. (2003), for the initial tsunami wave profile shown in Fig. 3b (without initial velocity). This and other benchmark test cases can be accessed online[1].

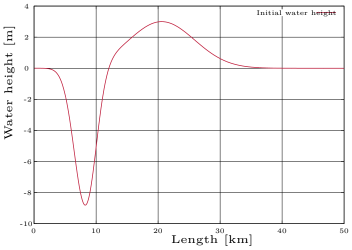
For this 2D runup problem, both the beach slope (1:10) and the initial wave profile (Fig. 3b) were given. The benchmark task was

[1] http://isec.nacse.org/workshop/2004_cornell/benchmark.html

to compute snapshots of the free surface elevation in the runup region, at three times ($t = 160, 175, 220$ s), and to compare those to the reference solution. Hence, the accuracy of the algorithm used for the treatment of the air-water-beach interface at the shoreline is the key point of this benchmark. The LBM simulation parameters are given in Fig. 3a.

Domain	51.2km x 6.25 m
Lattice	8192 x 4
Δx	6.25 m
c	1000 m/s
Δt	6.25 ms
$1/\tau$	1.95
δ	0.0001
BC x	runup
	No-slip
BC y	Periodic

(a) Parameters



(b) Initial wave profile

Figure 3: Tsunami runup onto a plane beach

Fig. 4 compares the resulting free surface profile in the runup region to the benchmark reference data. A very good agreement can be seen between these, even at a relatively low resolution in the LBM. The larger differences for the last few nodes in the runup region are likely due to the fairly low grid resolutions used here, with $\Delta x = 6.25$ m, resulting in a vertical bottom elevation difference of $\Delta z = 0.625$ m between two neighboring lattice nodes. This is a limiting factor, for this simple runup model, which does not perform an extrapolation or uses a complex reconstruction of the bottom elevation. Grid refinement would further improve the runup behavior, while the overall flow is already well-represented, even at such low grid resolutions. Note that the Carrier solution is valid for the inviscid NSW equations, while our model includes viscous effects. Hence, for this benchmark, the fluid viscosity was iteratively minimized, and convergence to the presented results was observed. The viscosity for the given LB parameters in Fig. 3a still is several orders of magnitude higher than the viscosity of water ($\nu \approx 25$ m^2/s), but a further decrease does not significantly change the results.

Our simulation was run on a NVIDIA Tesla C2070 GPGPU, with an average performance of 115 MNUPS, leading to an overall simulation time of approximately 30 s, nearly ten times faster than real time. Similar to earlier LBM bulk flow simulations, however, although impressive at 115 MNUPS, the full performance of the GPGPU cannot really be assessed in this benchmark because of the small amount of lattice nodes used in the y-direction for this 2D flows (only four). Frandsen (2008) computed the same test case with a D1Q3 LB model on a scalar processor, using a lattice of 100,000 nodes (resulting in a grid spacing $\Delta x = 0.5$ m, and choosing a time step $\Delta t = 0.002$ s). The CPU time for the whole simulation was between 6,700 and 11,000 s for a similar numerical accuracy as our results, at least for the free surface elevation in the runup region (Fig. 5). Frandsen (2008) also presented results for the flow velocities and the detailed shoreline movement, which still have to be compared to our solution, but presumably will be more accurate owing to the much higher grid resolution.

It should be noted that a more physically complete solution of this initial value problem, than the NSW solution of Carrier et al. (2003), can be obtained by directly solving Euler equations (i.e., the inviscid Navier-Stokes equations). This can be done in a fully nonlinear potential flow (FNPF) formalism, using a higher-order Boundary Element model (Grilli and Subramanya, 1996), which essentially yields a numerically exact solution. This solution was presented by Enet and Grilli during the 2004 workshop and is also plotted in Fig. 4 for comparison. We see that there are significant differences in both the free surface location and the runup; e.g., for $t = 220$ s, the NSW model underpredicts the maximum runup by almost 30% (Fig. 4).

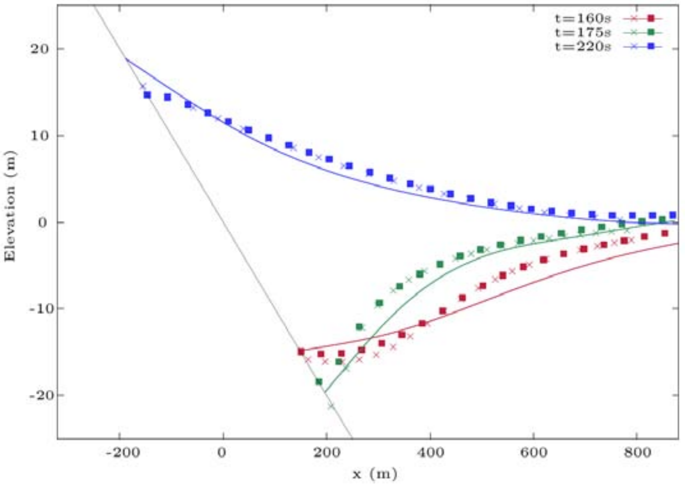


Figure 4: Free surface in the runup region. Square symbols denote the numerical solution with the NSW-LBM; cross symbols denote the reference NSW solution (Carrier et al., 2003); and solid lines denote the FNPF solution (Enet and Grilli, 2004).

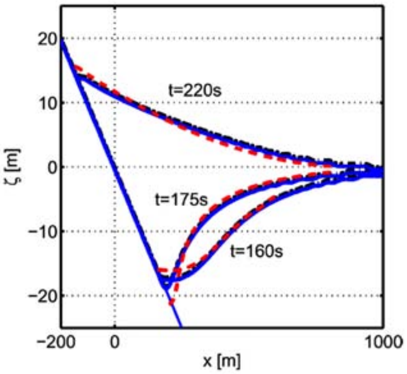


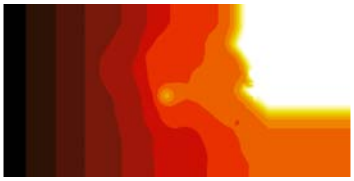
Figure 5: Free surface in the runup region. Dash lines denote a D1Q3 LBM solution on a 100,000 grid (Frandsen, 2008), solid lines denote the reference NSW solution (Carrier et al., 2003)

Tsunami runup over a complex 3D natural beach

Finally, we tested the application of our new NSW-LBM code to a real-world event, the Okushiri 1993 tsunami, for which laboratory experiments were also performed as part of the set of benchmark problems used in the 2004 Catalina Island workshop (Liu et al., 2008). The Okushiri tsunami caused unexpectedly disastrous damage on the coast, whose survey provided numerous field data that can be used to validate and benchmark tsunami runup models. Additionally, laboratory experiments at a 1/400 scale of runup in the Monai Valley were conducted, aimed at reproducing and better understanding the coastal impact of the Okushiri tsunami. The latter are being modeled here as part of our third validation benchmark.

Param.	Value
Domain	5.488 m × 3.416 m
Grid	392 × 244
Δx	0.014 m
c	56 m/s
Δt	0.25 ms
$1/\tau$	1.8
ν	$\approx$ 0.01 m^2/s
δ	0.00001

(a) Parameters



(b) Coastal topography

Figure 6: Tsunami runup onto a complex three-dimensional beach

The scale model dimensions and parameters are given in Fig. 6. The 5.4 by 3.4 m horizontal extent of the experimental setup was discretized in the LBM with a grid spacing $\Delta x = 1.4$ cm (as recommended in the benchmark setup information). The time step is set to $\Delta t = 0.25$ ms, which also sets the macroscopic reference velocity to $c = 56$ m/s. A collision rate $1/\tau$ of 1.8 yields a fluid viscosity of $\nu \approx 0.01$ m^2/s. The time-dependent offshore wave height (specified along the $x = 0$ boundary) is given as a boundary condition. This test case is especially demanding in terms of bottom elevation, which implies that the forcing terms have to be evaluated for all lattice directions. Moreover, the runup model has to be extended to cope with changes of bottom topography that are not aligned with the grid axes.

Computed time series of surface elevation at three different wave gages are compared in Fig. 7 to experimental data from the scale model experiment. Overall, numerical results reproduce well the salient features of the tsunami-induced flow, particularly for the lower frequency components. The time of initial wave impact and that of arrival of the reflected wave match the experimental values quite well. The maximum water elevation at the numerical probes, however, is approximately 25% less than the experimental values. Attempts to fine-tune the LBM runup model and/or further reduce the fluid viscosity, to better simulate the peaks in water elevation induced instabilities in the runup region. This could be improved by developing a more sophisticated wetting and drying algorithm for the simulation of runup in complex bottom topography, and will be part of future research.

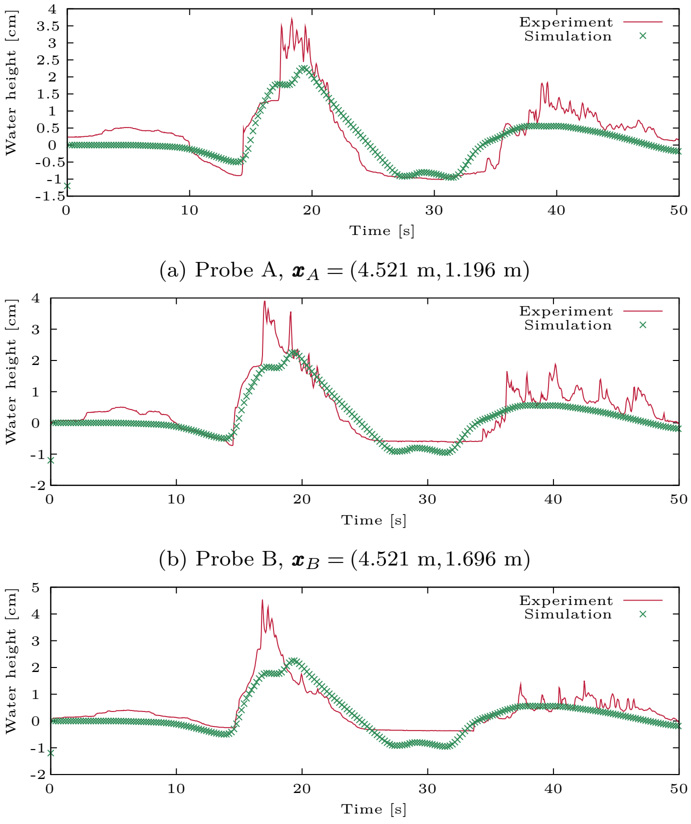
The simulation was carried out on a Tesla C2070 GPGPU, with a maximum performance of 105 MNUPS, and took 240 s of computational time, for 50 s of propagation time (corresponding to 200,000 time steps) on a lattice with 512 x 256 nodes.

Comparison of NSW-LBM to FUNWAVE results

Finally, the simulation results using the NSW-LBM code are compared to results of the FUNWAVE code. FUNWAVE solves the fully nonlinear Boussinesq equations with finite volume and finite difference methods (Wei et al., 1995; Shi et al., 2012). For the flux terms and first-order derivatives, a fourth-order TVD-MUSCL scheme is used, while the higher-order derivatives are discretized with central difference schemes. A third-order Runge-Kutta method is used for the time discretization. FUNWAVE's MPI-parallel code uses a domain decomposition method based on ghost layers (three node rows). For the result comparison, both codes were applied to predict the Monai runup, with the grid resolutions and setups given in Tab. 2.

	Grid	dx	dt	T	Timesteps
Funwave	892x244	1.4 cm	5 ms	50s	10,000
LBM	512x256	1.4 cm	0.25 ms	50s	200,000

Table 2: Simulation setup



(a) Probe A, $\boldsymbol{x}_A = (4.521\ \mathrm{m}, 1.196\ \mathrm{m})$

(b) Probe B, $\boldsymbol{x}_B = (4.521\ \mathrm{m}, 1.696\ \mathrm{m})$

(c) Probe C, $\boldsymbol{x}_C = (4.521\ \mathrm{m}, 2.196\ \mathrm{m})$

Figure 7: Comparison of experimental (Liu et al., 2008) and NSW-LBM results for three different wave gages A, B and C for the Catalina benchmark #2 of the Monai Valley.

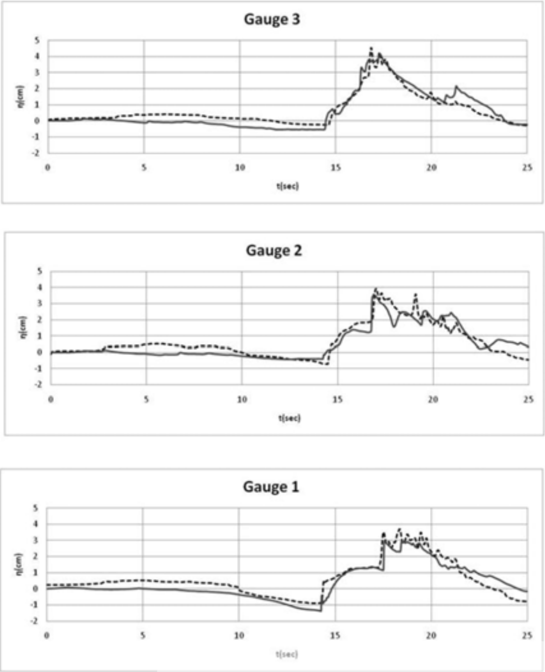


Figure 8: Comparison of experimental (Liu et al., 2008) and FUNWAVE's results for wave gages A, B and C for the Catalina benchmark #2 of the Monai Valley.

	Duration	NUP	Core-h	MNUPS	KNUPS/C.
FUNWAVE	42 min	2.1E09	11 h	0.86	54
NSW-LBM	4 min	2.6E10	30 h	105	234

Table 3: Performance results for FUNWAVE in comparison to NSW-LBM. Total number of node updates (NUP) and core hours (Core-h); resulting performance in million node updates per second (MNUPS) and thousand node updates per computational core (KNUPS/C)

Result accuracy Fig. 8 shows FUNWAVE's simulation results for the three wave gages in Fig. 7, compared to the experimental reference data. A very good agreement can be seen. Compared to Fig. 7, the higher frequency variations in surface elevation are better predicted than in results of the NSW-LBM simulation, which additionally underpredict maximum elevations and, hence, seem to be too "viscous" and lacking dispersive effects. The times of initial wave impacts matches the experimental values quite well in both models.

Code performance FUNWAVE simulations were run on a MAC desktop with two quadcore CPUs (in Hyperthreading mode), with a core clock of 2.93 GHz each; the LBM simulation was run on a Tesla C2070 GPGPU board with 448 cores running at a clock speed of 1.15 GHz. Tab. 3 summarizes the resulting performance for the two simulations that are depicted in Fig. 7 and Fig. 8, respectively. With 2.6E10 node updates, due to the smaller time step, the NSW-LBM model needs 20 times more iterations than FUNWAVE. In the end, however, the LBM code implemented still runs about ten times faster than FUNWAVE, due to the higher number of cores on the GPGPU. Moreover, the number of node updates per core also is a factor of 4 higher than in FUNWAVE.

CONCLUSIONS AND OUTLOOK

We presented an efficient GPGPU implementation of a Lattice Boltzmann model for the solution of the non-linear shallow water (NSW) equations, including a simple model for the treatment of the wet-dry interface. Both SRT and MRT collision operators were implemented, validated, and applied to several benchmark problems. The initial validation shows a good performance of the code, and a good agreement with the reference data in most cases. The application to the more complex standard Tsunami benchmark cases, however, indicates that further model improvements are necessary. In particular, the Monai Valley test case clearly demonstrates that FUNWAVE produces better results, which may be in part due to the inclusion of frequency dispersion effects in the latter model (this could be verified by turning off dispersion in FUNWAVE and re-running results).

However, the performance comparison between the proposed NSW-LBM code and FUNWAVE indicates that the faster LBM solver could become a valuable component of realtime/early warning systems, along with standard tsunami propagation models, once it is equipped with a better runup model and has an improved stability.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support for this research from the NVIDIA Academic Partnership Program (APP). SG and CJ acknowledge support of Grant OCE-09-27014 from the US National Science Foundation. SG also acknowledges support of Grant EAR-09-11499 from the US National Science Foundation.

REFERENCES

Balzano, A. (1998). Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. *Coastal Engng.*, 34(1-2):83 – 107.

Bhatnagar, P. L., Gross, E. P., and Krook, M. (1954). A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.*, 94(3):511–525.

Carrier, G. F., Wu, T., and Yeh, H. (2003). Tsunami run-up and draw-down on a plane beach. *J. Fluid Mech.*, 475:79–99.

d'Humieres, D., Ginzburg, I., Krafczyk, M., Lallemand, P., and Luo, L.-S. (2002). Multiple Relaxation-Time Lattice Boltzmann models in three-dimensions. *Royal Society of London Philosophical Transactions Series A*, 360:437–451.

Frandsen, J. B. (2006). Investigations of wave runup using a LBGK modeling approach. In *CMWR XVI Proc. XVI Intl. Conf. Comp. Meth. in Water Resoures, Copenhagen, Denmark.*

Frandsen, J. B. (2008). A simple LBE wave runup model. *Progress in Comp. Fluid Dyn.*, 8:222–232.

Geller, S., Krafczyk, M., Tölke, J., Turek, S. and Hron, J. (2006). Benchmark computations based on Lattice-Boltzmann, Finite Element and Finite volume Methods for laminar Flows. *Comp. and Fluids*, 35:888–897

Ginzburg, I. and D'Humieres, D. (2003). Multireflection boundary conditions for lattice Boltzmann models. *Phys. Rev. E*, 68(6):066614.1–066614.30.

Grilli, S.T. and Subramanya, R. (1996). Numerical Modeling of Wave Breaking Induced by Fixed or Moving Boundaries. *Comp. Mech.*, 17(6):374–391.

Grilli, S.T., J.C. Harris, T. Tajalibakhsh, J.T. Kirby, F. Shi., T.L. Masterlark and C. Kyriakopoulos, (2012). Numerical simulation of the 2011 Tohoku tsunami: Comparison with field observations and sensitivity to model parameters. *Proc. 22nd Offshore and Polar Engng. Conf. (ISOPE12, Rodos, Greece, June 2012)* (submitted).

Harvey, M. J., De Fabritiis, G., and Giupponi, G. (2008). Accuracy of the lattice-Boltzmann method using the Cell processor. *Phys. Rev. E*, 78(5):056702.

Körner, C., Thies, M., Hofmann, T., Thürey, N., and Rüde, U. (2005). Lattice Boltzmann Model for Free Surface Flow for Modeling Foaming. *J. Stat. Phys.*, 121(18):179–196.

Kowalik, Z. and Murty, T. S. (1993). *Numerical modeling of ocean dynamics.* World Scientific Pub.

Ladd, A. J. (1994). Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *J. Fluid Mech.*, 271:285–309.

Lallemand, P. and Luo, L.-S. (2000). Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61:6546–6562.

Liu P.L.-F., Yeh, H., and Synolakis C. (2008). Advanced numerical models for simulating tsunami waves and runup (Edited Proc. of the 2004 Catalina Island Workshop). *World Scientific Publishing.*

Lynett, P. J., Wu, T.-R., and Liu, P. L. F. (2002). Modeling wave runup with depth-integrated equations. *Coastal Engng.*, 46(2):89 – 107.

Schönherr, M., Kucher, K., Geier, M., Stiebler, M., Freudiger, S., and Krafczyk, M. (2011). Multi-thread implementations of the lattice Boltzmann method on non-uniform grids for CPUs and GPUs. *Comp. and Math. with Appl.*, 61(12).

Shi, F., Kirby, J. T., Harris, J. C., Geiman, J. D., and Grilli, S. T. (2012). A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation. *Ocean Modeling*, 43-44:36–51.

Thömmes, G., Seaid, M., and Banda, M. K. (2007). Lattice boltzmann methods for shallow water flow applications. *Int. J. Numer. Meth. Fluids*, 55:673–692.

Tubbs, K. (2010). *Lattice Boltzmann Modeling for shallow water equations using high performance computing.* PhD thesis, Lousiana State University.

Wei, G., Kirby, J. T., Grilli, S. T., and Subramanya, R. (1995). A fully nonlinear Boussinesq model for surface waves. I. Highly nonlinear, unsteady waves. *J. Fluid Mech.*, 294:71–92.

Zhou, J. (2003). *Lattice Boltzmann Methods for Shallow Water Flows.* Springer. ISBN 3540407464.