

Article

Validation of the GPU-Accelerated CFD Solver ELBE for Free Surface Flow Problems in Civil and Environmental Engineering

Christian F. Janßen *, Dennis Mierke, Micha Überrück, Silke Gralher and Thomas Rung

Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology,
Am Schwarzenberg-Campus 4, 21073 Hamburg, Germany; E-Mails: dennis.mierke@tuhh.de (D.M.);
micha.ueberrueck@tuhh.de (M.Ü.); silke.gralher@tuhh.de (S.G.); thomas.rung@tuhh.de (T.R.)

* Author to whom correspondence should be addressed; E-Mail: christian.janssen@tuhh.de;
Tel.: +49-40-42878-6040.

Academic Editor: Manfred Krafczyk

Received: 19 January 2015 / Accepted: 15 June 2015 / Published: 7 July 2015

Abstract: This contribution is dedicated to demonstrating the high potential and manifold applications of state-of-the-art computational fluid dynamics (CFD) tools for free-surface flows in civil and environmental engineering. All simulations were performed with the academic research code ELBE (efficient lattice boltzmann environment, <http://www.tuhh.de/elbe>). The ELBE code follows the supercomputing-on-the-desktop paradigm and is especially designed for local supercomputing, without tedious accesses to supercomputers. ELBE uses graphics processing units (GPU) to accelerate the computations and can be used in a single GPU-equipped workstation of, e.g., a design engineer. The code has been successfully validated in very different fields, mostly related to naval architecture and mechanical engineering. In this contribution, we give an overview of past and present applications with practical relevance for civil engineers. The presented applications are grouped into three major categories: (i) tsunami simulations, considering wave propagation, wave runup, inundation and debris flows; (ii) dam break simulations; and (iii) numerical wave tanks for the calculation of hydrodynamic loads on fixed and moving bodies. This broad range of applications in combination with accurate numerical results and very competitive times to solution demonstrates that modern CFD tools in general, and the ELBE code in particular, can be a helpful design tool for civil and environmental engineers.

Keywords: ELBE code; wave propagation; inundation; wave impact; debris flow; wave-current-induced loads; vortex-induced vibrations; lattice Boltzmann method; GPU

1. Introduction

Hydrodynamic free-surface flows with and without fluid-structure interactions (FSI) have received an ever-increasing interest over the past few years (see, e.g., [1]). Applications of such flows range from naval architecture to offshore, marine, coastal and environmental engineering. In the present publication, free surface flows and potential applications of free surface solvers in civil and environmental engineering are addressed. Such flow problems involve an air and a water phase, so that, technically, such flows can be classified as two-phase flows. However, as for most problems of practical relevance, the flow behavior is dominated by the water phase, it is not necessary to resolve the full aerodynamics inside the air phase. Instead, the latter can be represented by proper kinematic and dynamic boundary conditions at the free surface, and the sharp interface is allowed to move freely [2].

Typical free surface flow problems occur in many different fields of civil and environmental engineering. Moreover, large-scale free surface flows, such as breaking waves or tidal currents, are fascinating, as they give us indications of the enormous amount of energy that water can contain, for both good and bad reasons. Recently, natural disasters have attracted public attention to hazardous free surface flow events. Particularly, the 2004 Boxing Day event and the recent 2011 Tohoku tsunami [3] have increased the awareness of the potential threat posed by tsunamis and, consequently, the responsibility of civil engineers to minimize the consequences of such major impacts. Here, numerical tools can help to establish early warning systems and to provide accurate and fast predictions for tsunami severeness and the details of coastal impact [4].

Apart from such challenging, even life-threatening long wave propagation applications, local wave impact and wave slamming are of primary importance in the design process in various kinds of engineering sciences, e.g., for the design of offshore structures, such as offshore wind turbines, which are possibly one answer to the open questions concerning the future of our energy supply, the wave impact force on monopiles is the main load case during the design process and has been part of extensive research [5]. For naval engineers, the flow patterns around and in the wake of a ship hull are important to know, as they highly depend on turbulent effects in the boundary layer. From a numerical point of view, all of those applications require fully three-dimensional, viscous and turbulent free surface flow simulations.

Even in this limited list of applications, the diversity of free surface flow problems and the wide range of length and time scales involved becomes obvious. On top of this, research on free surface flows and efficient numerical tools is by nature multidisciplinary, not only regarding flow theory and conservation laws, but also developing efficient algorithms and high performance computer models that are crucial for the development of accurate and efficient free surface flow solvers for desktop computers.

In this paper, selected applications of our in-house numerical tool ELBE (efficient lattice boltzmann environment, <http://www.tuhh.de/elbe>) for the simulation of free surface flow problems are presented. State-of-the-art high performance graphics processing units will be combined with an efficient numerical method to address various problems in civil and environmental engineering, on different length and time scales. After a quick review of the solver in Section 2, four applications will be addressed: tsunami-related propagation and inundation simulations in Section 3, debris flow in Section 4, dam break scenarios in Section 5 and a numerical wave tank for on- and off-shore structures in Section 6.

2. The ELBE Code

The efficient lattice Boltzmann environment ELBE is a versatile, efficient toolkit for the numerical simulation of complex two- and three-dimensional flow problems. Emphasis is given to marine- and coastal-engineering free-surface flows, such as wave breaking, tank sloshing or tsunami propagation. The model considers non-linear flow behavior with and without a free surface, the effects of viscosity and turbulence. On top of this, the software uses a bidirectional, partitioned, explicit coupling approach for the simulation of fluid-structure interaction problems. The rigid body motion is modeled with a quaternion-based motion modeler or a physics engine. ELBE is based on the lattice Boltzmann method (LBM) and offers various different collision operators, boundary conditions, turbulence models, interface capturing methods and grid refinement techniques. Thanks to the very efficient numerical back-end, three-dimensional simulations of complex flows are possible in a very competitive simulation time, as discussed in the following.

2.1. Lattice Boltzmann Method

While classical CFD solvers are based on the macroscopic Navier–Stokes equations, the LBM handles CFD problems on a microscopic scale. LBM’s fundamental variable is the particle distribution function $f(t, \mathbf{x}, \xi)$, which specifies the probability to meet a fictive particle with velocity ξ at position \mathbf{x} and time t . To obtain a model with reduced computational costs, the velocity space is discretized, and discrete particle velocities e_i are introduced. In the present work, a three-dimensional model with 19 discrete microscopic particle velocities e_i and corresponding particle distribution functions $f_i(t, \mathbf{x})$ is applied. A subsequent standard finite difference discretization of the velocity-discrete Boltzmann equation on an equidistant Cartesian grid then leads to the lattice Boltzmann equation:

$$f_i(t + \Delta t, \mathbf{x} + \Delta t e_i) - f_i(t, \mathbf{x}) = \Omega_i, \quad (1)$$

where the left-hand side is an advection-type expression and the discrete collision operator Ω_i on the right-hand side models the interactions of particles at the microscopic scale. For the latter, the advanced multiple relaxation time (MRT) model [6] is used in this work. The MRT transforms the particle distribution functions into moment space, where they are relaxed to an equilibrium state with several different relaxation rates. The benefits of this operator are an increased stability and the possibility to develop more accurate boundary conditions [7]. The solutions of the lattice Boltzmann equation satisfy the incompressible Navier–Stokes equations up to errors of $\mathcal{O}(\Delta x^2, Ma^2)$ [8]. The first two hydrodynamic moments of the particle distribution functions include the macroscopic values for the density and momentum fluctuation:

$$\rho = \sum_{i=0}^{18} f_i \quad \text{and} \quad \rho_0 \mathbf{u} = \sum_{i=0}^{18} f_i \mathbf{e}_i. \quad (2)$$

A Smagorinsky large eddy model (LES) captures turbulent structures in the flow, including an additional turbulent viscosity ν_T for the effects of unresolved sub-grid eddies [9]. Since after the advection step, the incoming particle distribution functions at the domain boundaries are missing, they are reconstructed with the help of boundary conditions. For no-slip and velocity boundaries, a simple

bounce back scheme is used [10]. At the free surface, the anti-bounce-back rule [11] balances the fluid pressure and the surrounding atmospheric pressure. Volume forces, like gravity, are added directly to the distribution functions at every time step [12].

2.2. Free Surface Model

Free surface flows are immiscible two-phase flows, which are dominated by the denser phase, due to high viscosity and density ratios between the two phases. If capillarity is neglected, the simulation of the denser phase is sufficient for many applications. The influence of the lighter phase on the flow dynamics can be approximated by kinematic and dynamic boundary conditions at the interface. Numerically, the free surface represents a moving boundary, which is allowed to move freely, but at the same time has to be kept sharp. A common and straightforward way to simulate free surface flows in the scope of LBM is the combination of a volume of fluid (VOF) method and a flux-based mesoscopic advection scheme [11]. In contrast to common VOF methods, the flux terms are expressed directly in terms of the distribution functions. The VOF approach captures the interface via the fluid fill level ε of a cell: $\varepsilon = 0.0$ marks an empty cell in the inactive gas domain, and $\varepsilon = 1.0$ corresponds to a filled cell inside the fluid domain. Fluid and gas cells are separated by a closed interface layer with a fill level $\varepsilon \in (0.0; 1.0)$. The new fill level of a cell is calculated by balancing the mass fluxes between the neighboring cells and updating the fill level. The wet area between two cells is calculated on the basis of a simplified surface reconstruction, e.g., as the arithmetic mean of the fill levels of two neighboring cells. Consequently, the lattice Boltzmann VOF advection scheme can be considered as a specialized, geometry-based VOF method on the basis of a mesoscopic advection model. Note that, in contrast to higher order schemes [13], the normal vector information is not considered here. Nevertheless, the model has been shown to perform well for many different free-surface scenarios [14,15] and also in the context of GPU implementations [16].

2.3. Fluid-Structure Interaction

Apart from standard bulk and free surface flows, ELBE also provides a fluid-structure interaction interface. Floating-body motions follow from a motion solver, which converts the external and hydrodynamic forces exerted on the body into the equivalent motions. For the coupling of the explicit LBM and the motion solvers, a bidirectional and explicit coupling approach is used [17,18]. First, the hydrodynamic loads on the rigid bodies are evaluated by integrating the stress tensor over the bodies' surface (stress-integration method (SIM)). Since the rigid bodies are considered to be infinitely stiff and do not allow elastic or plastic deformations, the evaluation of the integral force on each obstacle is sufficient. Once the hydrodynamic loads are calculated, the force and torque information is transferred to the motion solver, and the subsequent time step of rigid body motion is computed. The resulting displacements and velocities are transferred back to ELBE, where the geometries are updated, and a modified bounce back scheme serves to incorporate the rigid body boundary velocity. The bodies are represented by triangulated surface meshes and are mapped to the fluid field by our highly efficient in-house voxelizer [19], so that the grid update is up to two orders of magnitude faster than one flow field update.

For single rigid bodies, we apply either a straightforward spring-damper model or a more advanced quaternion-based motion modeler. For floating and colliding rigid multi-body systems, we recently coupled ELBE to a physics engine, the open-source Open Dynamics Engine (ODE) (<http://www.ode.org>). ODE uses a hard contact model and detects collisions very efficiently, even for triangulated mesh geometries. The equations of motion then are impulse-based solved, yielding a linear complementarity problem (LCP) [20]. If necessary, friction can be included in the simulations by a simplified and efficient Coulomb friction model. The ODE is well known for its high performance, free and easy access and the real-time simulation capabilities. It is mainly used in robotics and vehicle simulations, virtual reality environments, computer games and simulation tools [21–23].

Note that, in the coupling with ELBE, all fluid-related computations (hydrodynamics, force calculation, grid update, boundary conditions) are executed on the GPU device, while the solution of the rigid body equations of motion and/or the full ODE contact problem is done on the CPU.

2.4. Implementation

The main purpose of this paper is the extensive validation of ELBE for large-scale engineering applications. However, several ELBE-specific implementation details will be briefly discussed in the following. For the parallelization, the NVIDIA CUDA toolkit is used, as LBMs are especially well suited for implementation in a GPU context [24,25]. The latter provides a large amount of computing cores (*i.e.*, up to 4992 on a recent NVIDIA Kepler K80 board) and sufficient device memory (24 GB, K80) for typical engineering applications. In 3D, this allows the allocation of up to 120 million lattice nodes per GPU board. The average performance yields more than 500 million node updates per second, for a single precision implementation on a state-of-the-art GPU. Thanks to this very efficient numerical back-end, three-dimensional simulations of complex flows are possible in a very competitive time.

However, many GPU-accelerated solvers (see, e.g., [26]) combine efficient and optimized solver techniques with conventional CPU-based pre- and post-processing routines for the initialization, the transient grid update and the data storage operations (Figure 1). Due to the comparably low bandwidth of the PCI-Express slots that host the GPUs and additional latency effects, this dilutes the overall performance of the algorithms. In ELBE, innovative grid generation and post-processing techniques were developed instead and help to avoid this performance bottleneck completely.

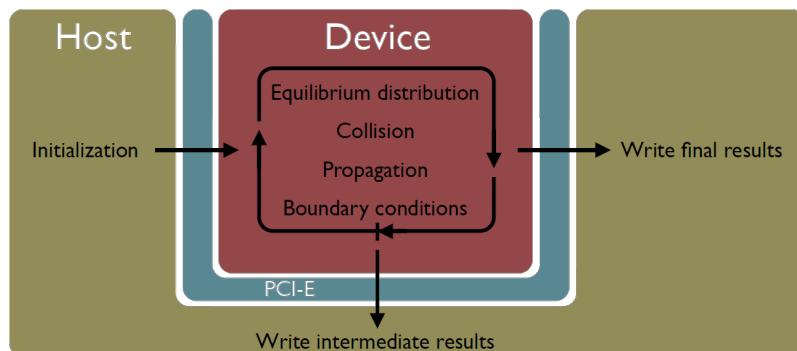


Figure 1. Typical bottlenecks of GPU-accelerated simulations: data transfer from host to device and *vice versa* during initialization and post-processing.

Lattice Boltzmann methods, as addressed in the present publication, discretize the governing equations on an equidistant Eulerian grid. Grid points of a computational domain used for lattice Boltzmann CFD implementations can be of a variety of different kinds. Those points representing a fluid particle ensemble are considered fluid particles, whereas other points can represent a slip wall, a no-slip wall, an inflow or even moving velocity boundaries. Since the grid itself is static and does not change over time, it is, apart from proper dynamic boundary conditions at the body surface, the changing character of the points through which a moving body's motion is manifested. Figure 2 (right) shows fluid particles as black circles and solid body particles as red dots. Black circles overlaid with red dots signify solid body particles from previous time steps, where a lighter shade of red indicates solid body particles from iterations further back in time. Our fast surface voxelization technique [19] maps arbitrary, tessellated triangular surface meshes to the LBM grids, directly on the GPU and without access to CPU host memory or CPU resources. The algorithm was optimized for massively parallel execution in a GPU context: its performance is superior to the performance of the flow field computations per time step, which allows efficient fluid-structure interaction simulations, without noticeable performance loss due to dynamic grid updates.

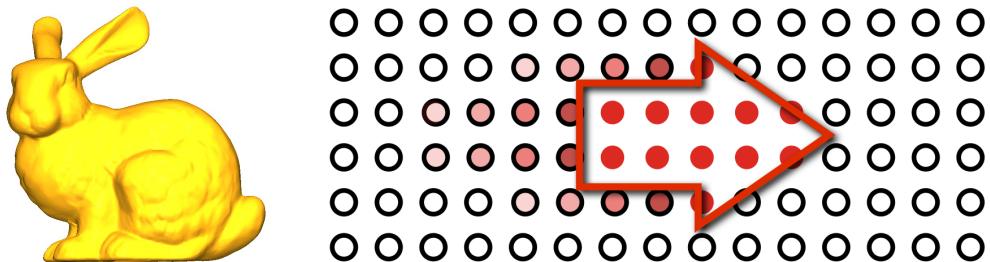


Figure 2. Grid generation in the context of the lattice Boltzmann method (LBM): mapping of discrete triangle mesh representations of geometries (**left**) to the Cartesian grid (**right**, red nodes).

Additionally, an integrated OpenGL-based visualizer tool allows one to visualize the simulation results during runtime, based on innovative, shared CUDA-OpenGL memory spaces residing in GPU memory, and allows for simulations without any CPU-GPU communication at all. ELBEvis is designed to help software developers, scientists and engineers to assess and interact with their computations in real time by providing an efficient and easy to use on-device implementation of several visualization features. After initialization, any time step's data are stored on device memory, so they can be used to compute the next time step's dataset without having to load new data from the host. Thus, the main computation's time loop could almost entirely be carried out on the device. To decouple visualization and computation, they are executed in two separate threads. Figure 3 shows several available filters to process the simulation data, e.g., colored slices through the computational domain, streamlines, glyphs and a volume-rendering representation. Moreover, in the case of, e.g., wave impact simulations, the surface pressure fields can be mapped to the surface of the geometry under consideration.

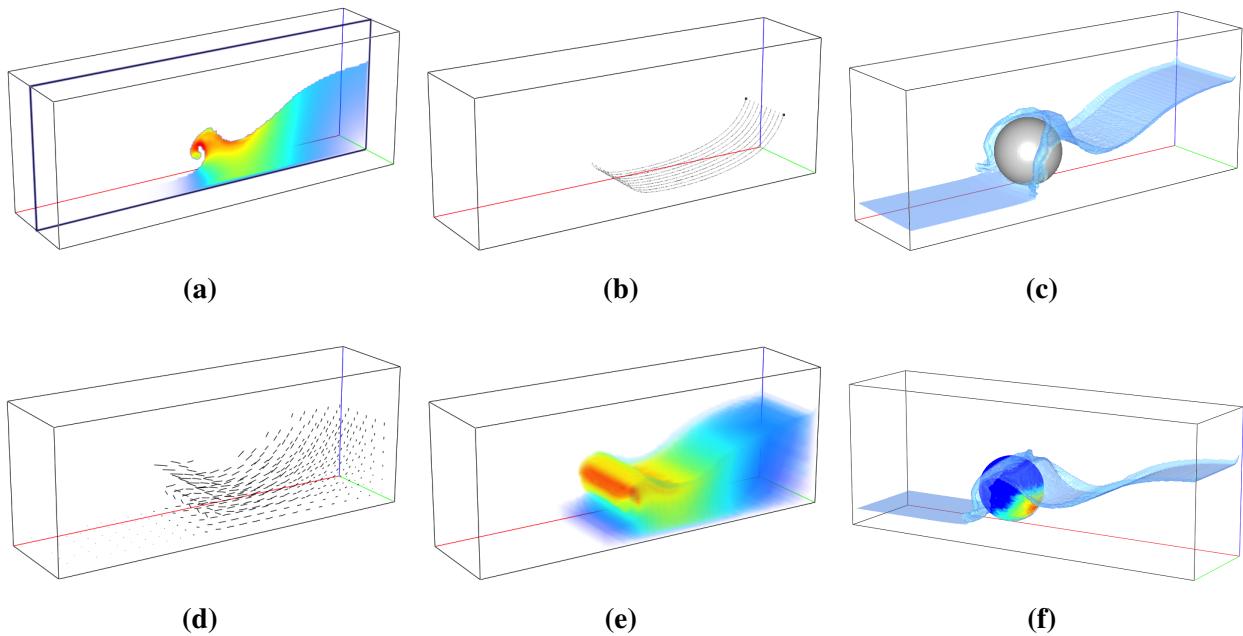


Figure 3. Six different visualizer features of ELBEvis: (a) colored slice, (b) streamlines, (c) geometry rendering, (d) velocity vectors, (e) volume rendering, (f) surface mapping.

On top of this, a graphical user interface (GUI) has been designed to provide a suitable control environment for the user. The GUI is executed as an entirely independent process and communicates with and instructs the computation/visualization via a shared memory segment that is spawned by the ELBEvis process upon its start.

Two different hardware concepts have been used with ELBE so far (Figure 4). Typically, the numerical simulations are run on our local GPU compute server. Moreover, thanks to the reduced use of host-device communication in our method, we recently assembled a configuration based on an external expansion chassis and attached an NVIDIA GTX Titan board to an off-the-shelf laptop. Even though the connection bandwidth was extremely low (PCI-E 1x), no performance degradation was observed thanks to the reduced use of host-device communication in the method.

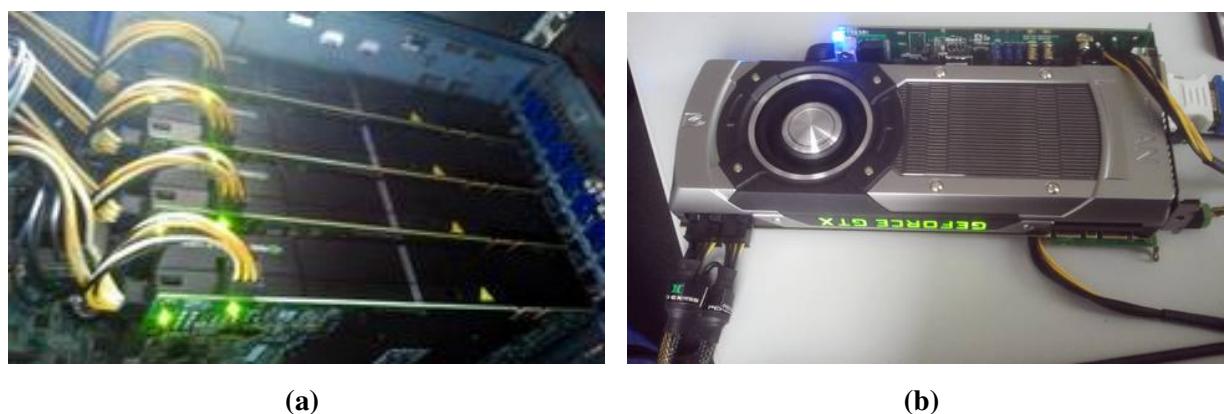


Figure 4. Two complementary ELBE hardware concepts. (a) Multi-GPU compute server; (b) expansion chassis.

2.5. Summary

The ELBE code follows the paradigm of supercomputing on the desktop and aims at running numerical simulations on off-the-shelf commodity hardware to perform a near real-time analysis and evaluation of three-dimensional turbulent flow fields, without tedious access to supercomputers. The recently added proper pre- and post-processing routines reduce the host-to-device communication to further increase performance. We consider ELBE as a prototype for next-generation CFD tools for a near real-time analysis of complex flow fields and simulation-based design (SBD) on commodity hardware. The rich features and the outstanding performance of the code make it a tool that is also applicable to various kinds of challenges in civil and environmental engineering, as discussed in the following sections.

3. Wave Propagation and Inundation Modeling

As the first principal topic of this publication, tsunami-related simulations are addressed. The most recent tsunami with major impact was triggered by a seismic event approximately 40 miles off the coast of Japan, in March 2011. The earthquake triggered powerful tsunami waves that killed nearly 20,000 people. Apart from the massive destruction, long-term harmful environmental impact and the psychological effects, a major nuclear disaster at the Fukushima Daiichi nuclear power plant was triggered, including meltdowns and the release of radioactive material [27]. The degree of damage drastically illustrated how vulnerable modern society is to such natural hazards. The availability of a proper tsunami warning systems (as in Japan) hence is essential. Once earthquakes are detected, tsunami warning centers can predict whether a tsunami was generated or not and, based on additional real-time observations, feed long wave propagation models to simulate the expected tsunami propagation, tsunami impact and inundation. ELBE is already part of a Tsunami related project at Ocean Networks Canada [28] and eventually will be considered as part of the near-shore tsunami early-warning system for British Columbia. However, despite the existence of proper early warning systems, the Tohoku tsunami caused massive destruction. Consequently, apart from early warning systems, such violent flow scenarios also have to be simulated *a priori* in order to assess the hydrodynamic loads on potentially threatened structures and buildings. In this section, the applicability of ELBE to wave propagation and inundation modeling is discussed, consisting of the subsequent simulation of tsunami propagation, near-shore wave impact and debris flow. Ideally, these three components go hand in hand to provide a complete examination of potential wave impacts and serve as a helpful tool for civil and environmental engineers to assess the potential threat to coastal areas and structures.

3.1. Wave Propagation in Shallow Waters

The standard approach in real-time tsunami forecasting is based on models based on nonlinear shallow water (NSW) equations, which are a depth-integrated (hydrostatic) version of Navier–Stokes equations (e.g., [29]). 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). Recently, non-hydrostatic models, such as Boussinesq models (e.g., [27,30,31]), have increasingly been used in research and damage assessment following major tsunami events.

In NSW-LBMs, depth-averaging yields a modified collision operator for subcritical [32] and supercritical [33] flows. Depth-averaged LBM was successfully applied to a variety of illustrative benchmark problems, including wave run-up on a sloping beach [34], applications featuring bed slope and friction terms [35], wind- and density-driven circulation over irregular bathymetry [36] and tank sloshing examples [37]. While the predictive agreement with traditional methods was generally good, LBM run-times were always very competitive. However, instabilities were induced near the wet/dry boundaries. The development of improved wetting/drying models and second-order wet/dry boundary conditions is currently part of active research. In the following, the NSW-LBM module of ELBE is first applied to two benchmark problems, before discussing the application of ELBE to the 2011 Tohoku tsunami event.

3.1.1. Validation with Catalina Benchmark Problems

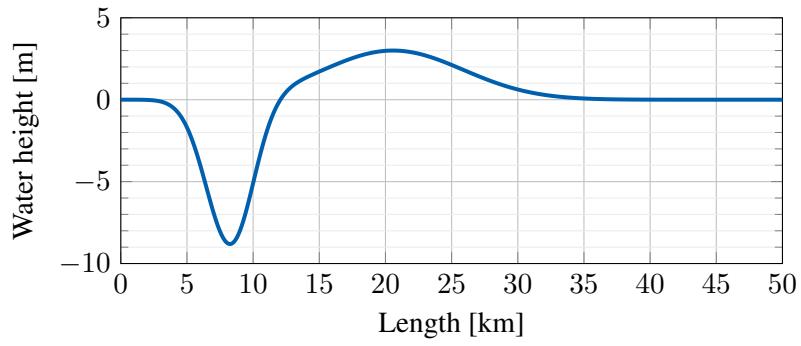
Here, we focus on two standard validation test cases of the tsunami community, which were initially proposed as part of the *Third international workshop on long-wave runup models*, which took place in 2004 on Catalina Island, CA, USA [38]. This and other benchmark test cases can be accessed online [39]. We examined two selected test cases: the 2D tsunami runup over a plane beach and tsunami runup over a complex 3D natural beach (Monai Valley). Note, in those, the latter benchmark problem is a demanding case, with very complex and realistic bottom topography. In both cases, the LBM solution is compared to available analytical, numerical or experimental reference data from the literature. In addition, the accuracy and performance of ELBE is compared to FUNWAVE, a state-of-the-art Boussinesq model used for tsunami propagation [31].

A. 2D Tsunami Runup over a Plane Beach

The benchmark reference data for this 2D runup problem was obtained from the analytical solution of NSW equations of Carrier *et al.* [40], for the initial tsunami wave profile shown in Figure 5 (without initial velocity) and a given beach slope (1:10). The benchmark task was to compute the free surface elevation in the runup region at three points in time ($t = 160, 175, 220$ s) and to compare those to the reference solution. Apart from the offshore wave propagation properties of the solver, the accuracy of the shoreline algorithm for the treatment of the air-water-beach interface at the shoreline is the key point of this benchmark.

Domain	$51.2 \text{ km} \times 6.25 \text{ m}$
Lattice	8192×4
Δx	6.25 m
c	1000 m/s
Δt	6.25 ms
$1/\tau$	1.95
δ	0.0001

(a)



(b)

Figure 5. Initial wave profile. (a) Parameters; (b) tsunami runup onto a plane beach.

The LBM simulation parameters are summarized in Figure 5a. Figure 6 compares the resulting free surface profiles in the runup region to the benchmark reference data. A very good agreement can be seen, even at a relatively low grid resolution in ELBE. The larger differences for the last few nodes in the runup region are likely due to the employed first-order runup model, which does not use extrapolation of the water levels or complex reconstruction of the bottom elevation. Higher-order runup models would further improve the runup behavior, while the overall flow is already well-represented in the present simulation. 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 Figure 5a still is several orders of magnitude higher than the viscosity of water ($\nu \approx 25 \text{ m}^2/\text{s}$), but a further decrease does not significantly change the results.

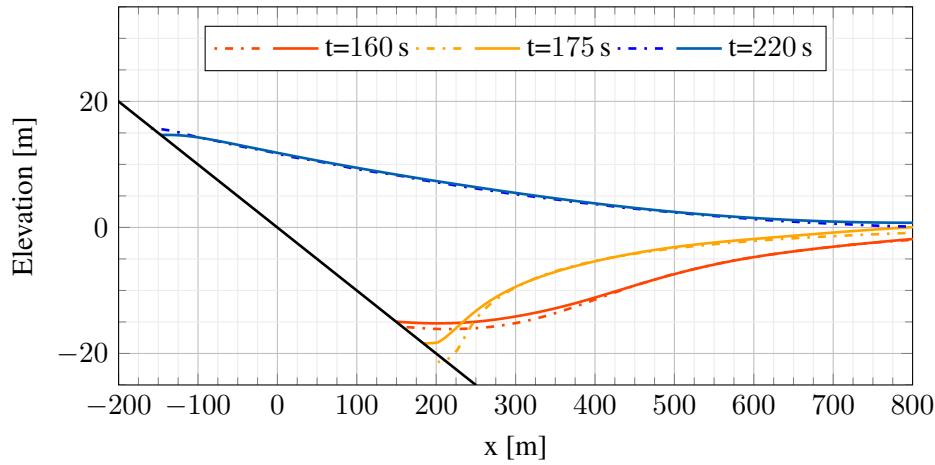


Figure 6. Free surface in the runup region. ELBE results (solid lines) in comparison to the reference nonlinear shallow water (NSW) solution of [40] (alternating dots and dashes) for three points in time.

The simulation was run on an NVIDIA Tesla C2070 GPU in about 30 s, nearly ten-times faster than real time. Frandsen [41] computed the same test case with a D1Q3 LB model on a scalar processor, using a lattice of 100,000 nodes, yielding a grid spacing $\Delta x = 0.5 \text{ m}$ and a time step $\Delta t = 0.002 \text{ s}$.

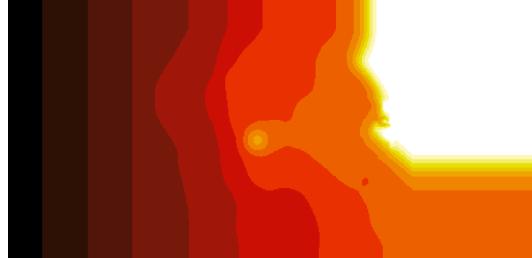
The CPU time for the whole D1Q3 simulation amounts to 6700–11,000 s for results with a very similar numerical accuracy in comparison to our results for the free surface elevation in the runup region. We found that a further grid refinement does not significantly change the results, so that we decided to keep our comparably low grid resolution. From a technical point of view, ELBE would have been able to increase the grid resolution up to a total of 50 million lattice nodes on the C2070 device. Despite the different grid resolutions, the comparison of times-to-solution of the two models is valid, as ELBE employs a 2D numerical model with nine degrees of freedom per lattice node (instead of three in the D1Q3 model) and a grid resolution of four lattice nodes in lateral direction, yielding an algorithmic overhead of approximately a factor of 20.

B. Tsunami Runup over a Complex 3D Natural Beach

After the theoretical benchmark case, ELBE is tested against experimental field data for the Okushiri 1993 tsunami event, which was also part of the Catalina set of benchmark problems. 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. The corresponding 1/400 scale laboratory experiments of runup in the Monai Valley were conducted in order to reproduce and better understand the coastal impact of the Okushiri tsunami.

The scale model dimensions and parameters are given in Figure 7. The grid spacing was chosen as recommended in the benchmark setup information, and the given time-dependent offshore wave height (specified along the $x = 0$ boundary) is used. The benchmark is especially demanding in terms of bottom elevation. Moreover, the runup model had to be extended to cope with changes of bottom topography that are not aligned with the grid axes.

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



(a)

(b)

Figure 7. Tsunami runup onto a complex three-dimensional beach. (a) Parameters; (b) coastal topography.

The numerical results for the surface elevation at three different wave gauges are compared to experimental data from the scale model experiment in Figure 8. All in all, the ELBE results reproduce the salient features of the tsunami-induced flow quite well, particularly for the lower frequency components. The time of the initial wave impact and the arrival of the reflected wave match the experimental values very well for all three probes. 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 obtain a more realistic (and higher) water elevation induced instabilities in the runup region. This can potentially be improved by more sophisticated wetting/drying algorithms for the simulation of runup in complex bottom topography. As these run-up algorithms operate on a very small subset of lattice nodes only, the global workload per time step and the performance of the code are not affected significantly.

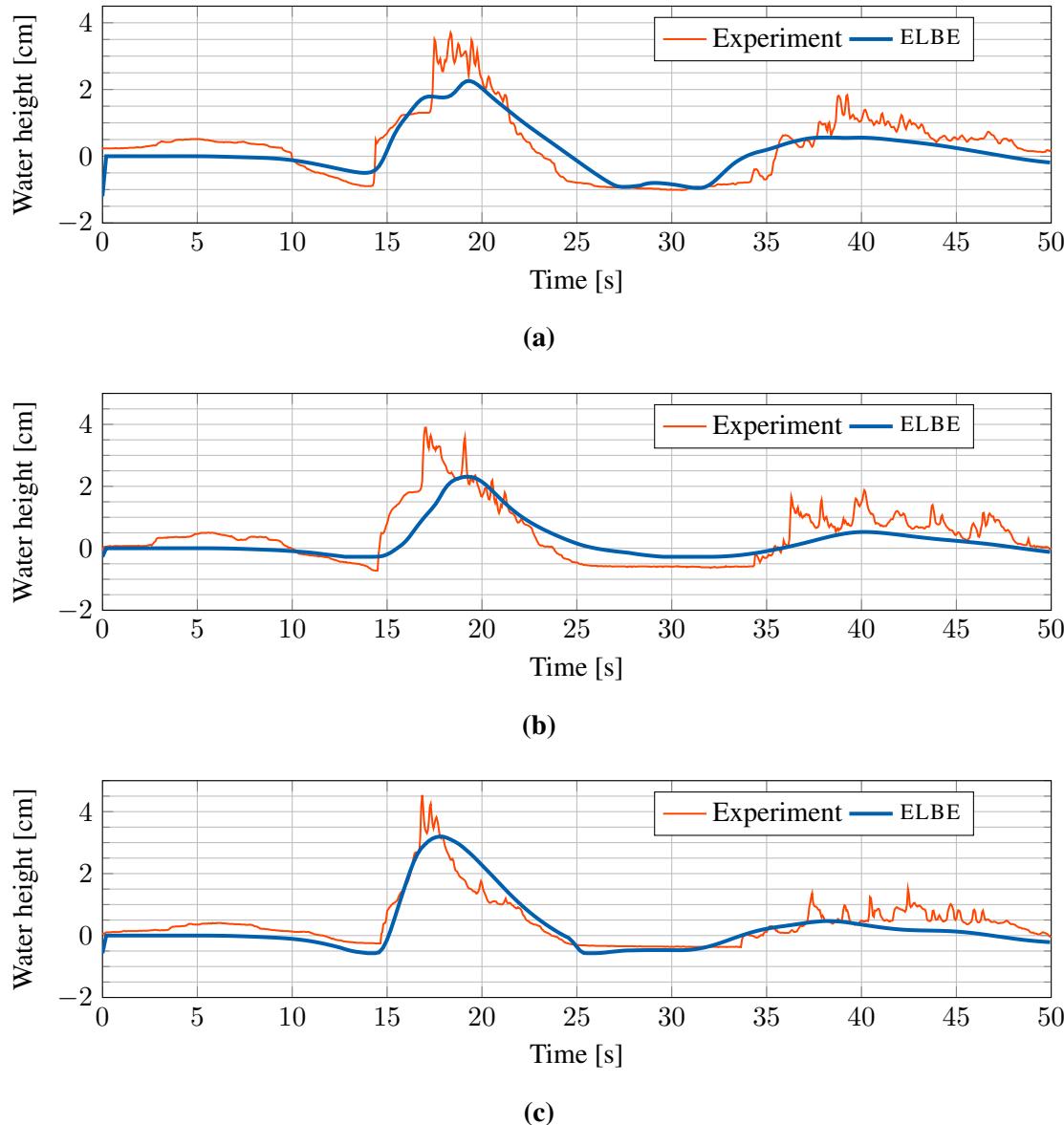


Figure 8. Comparison of experimental [38] and NSW-LBM results for three different wave gauges, A, B and C, for the Catalina Benchmark 2 of the Monai Valley. **(a)** Probe A, $\vec{x}_A = (4.521 \text{ m}, 1.196 \text{ m})$; **(b)** Probe B, $\vec{x}_B = (4.521 \text{ m}, 1.696 \text{ m})$; **(c)** Probe C, $\vec{x}_C = (4.521 \text{ m}, 2.196 \text{ m})$.

C. Performance Considerations

The simulation of the Monai Valley case took 240 s of computational time, for 50 s of propagation time (corresponding to 200,000 time steps) on a lattice with 512×256 nodes. For a proper performance rating, the performance is compared to the results of the FUNWAVE code. FUNWAVE solves the fully-nonlinear Boussinesq equations with finite volume and finite difference methods [30,31]. The

flux terms and first-order derivatives are discretized with a fourth-order scheme, while for the higher order derivatives, central difference schemes are used. A third-order Runge–Kutta method is used for the time discretization. FUNWAVE’s parallel code uses a domain decomposition method based on three rows of ghost layers and the message passing interface MPI. Grid resolutions and numerical setups for the simulations in ELBE and FUNWAVE are given in Table 1.

Table 1. Simulation setup and 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).

Solver	Grid	Δx	Δt	Time steps	Duration	NUP	Core-h	MNUPS	KNUPS/Core
FUNWAVE	892×244	1.4 cm	5 ms	10,000	42 min	2.1×10^9	11 h	0.86	54
LBM	512×256	1.4 cm	0.25 ms	200,000	4 min	2.6×10^{10}	30 h	105	234

FUNWAVE’s simulation results for the three wave gauges of the Monai Valley test case can be found in [42], and a very good agreement of numerical and experimental data, also for the higher frequency variations, can be observed. The benchmark simulations were run on a Mac desktop with two quad-core CPUs (in hyper-threading mode), with a core clock of 2.93 GHz each; the ELBE simulations were run on a Tesla C2070 GPU board with 448 cores running at a clock speed of 1.15 GHz. Table 1 summarizes the resulting performance for the two simulations: with 2.6×10^{10} node updates, due to the smaller time step, the NSW-LBM model needs 20-times more iterations than the Boussinesq solver. In the end, however, ELBE still runs about ten-times faster than FUNWAVE, due to the higher number of cores on the GPU. Moreover, the number of node updates per core also is a factor of four higher than in FUNWAVE. All of this shows the competitiveness of the ELBE solver for tsunami-related simulations. In combination with the currently-developed new runup models, the solver has the potential to become a competitive, fast and efficient tool for wave propagation and inundation simulations in the medium term.

3.1.2. Tohoku Tsunami

After the initial validation with universally-accepted benchmarks, the ELBE solver is applied to the most recent tsunami event, the 2011 Tohoku tsunami. The bathymetry is extracted from the ETOPO1 NOAA database [43], for a simulation domain of $1000 \text{ km} \times 1000 \text{ km}$. The initial free surface elevation is generated by an analytical model [44], which is used to simulate ground deformation produced by, e.g., tectonic faults, such as earthquakes. For a given rectangular fault geometry, the free-surface elevation can be computed. The case is addressed with four consecutively-refined ELBE grids, as specified in Figure 9. In this one-directional grid coupling, the simulation results are interpolated from the coarser grids and serve as initial conditions for the fine grids. A bidirectional coupling approach has successfully been implemented in ELBE for the single-phase solver and is currently adapted to the shallow water cases.

Figure 10 shows selected snapshots of the free surface elevation on three of the four grids, as the wave approaches the shoreline. In future work, the computed free surface elevations will be compared to available field data, and maximum water levels and inundation lengths will be verified with observations and numerical reference data. In any case, thanks to the recently-implemented interface to the ETOPO1

database and the Okada scripts, the ELBE code is applicable to almost any part of the ocean subject to tsunami generation, propagation and impact. As soon as reliable information on the earthquake source regions are available, initial conditions can be generated to feed the actual tsunami propagation simulations. Moreover, thanks to the efficient implementation, different input parameter sets can be simultaneously tested and simulated in *a priori* simulations, e.g., to identify the worst-case scenarios in civil engineering design.

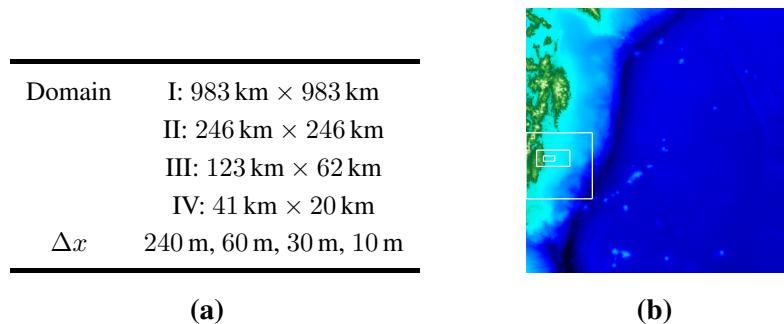


Figure 9. Tohoku tsunami. (a) Parameters; (b) bathymetry.

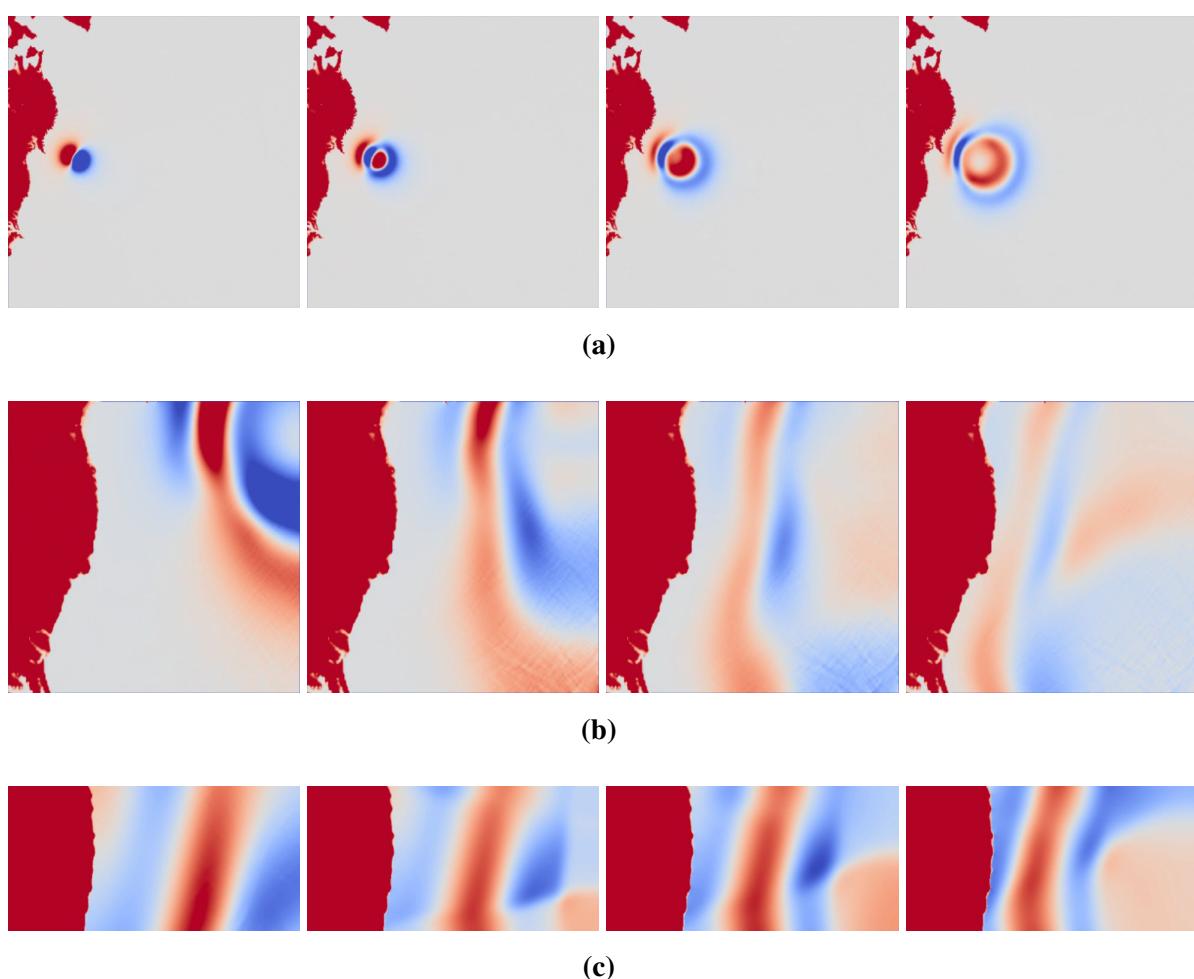


Figure 10. Results on three of the four consecutively-refined grids. (a) Grid Level I; (b) Grid Level II; (c) Grid Level III.

3.2. Near-Field Wave Impact

Once the Tsunami propagation and inundation is successfully modeled, attention can be devoted to the near-field wave impact. Wave impact on engineering structures is one of the most popular applications of free surface flow solvers. Apart from single obstacles in the flow, which could also be analyzed in experimental setups and have been intensely investigated numerically, scenarios that involve more complete and complex geometries are still of great interest. From the numerical point of view, these test cases are demanding due to high Reynolds numbers, large computational domains and distinct geometrical features, which ask for massively parallel computing to tackle the simulations. Here, we exemplarily show the impact of a wave on South Manhattan to demonstrate the capability of the ELBE solver to deal with complex geometries of a large scale. The wave is generated by a breaking dam scenario. Snapshots of the simulation for selected time steps and camera angles are given in Figure 11. The simulation was run on one single NVIDIA Tesla card using the maximum available device memory on a $512 \times 512 \times 80$ grid. Even with a full-scale grid spacing of $\Delta x = 2.3$ m, detailed information on the liquid entry after the initial wave impact can be obtained.

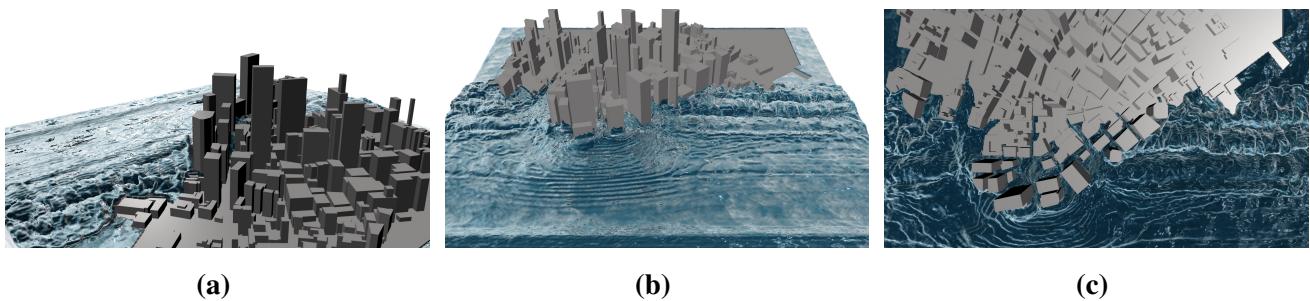


Figure 11. Wave impact on South Manhattan: results. **(a)** Northwest view; **(b)** south view; **(c)** top view.

4. Debris Flow: Coupling to a Physics Engine

Wave impact simulations on complex geometries provide detailed information on liquid entry and impact pressures. With the recent advances in high performance computing and GPU development, it has even become possible to go beyond the scope of static geometries and grid layouts and to resolve structural dynamics in the context of hydrodynamic simulation.

4.1. Validation of the Coupling Methodology

Before addressing more complex wave-multibody interactions, the coupling of the explicit LBM and the motion solver is examined, and a standard validation example is given. Here, a freely falling cube is analyzed, in line with the experimental work of [45] (Figure 12).

In Figure 13, the results of ELBE (left), our in-house finite-volume code FreSCo⁺ (middle) and the experiment (right) are compared, and a very good agreement can be seen. Exemplarily, the Euler angles are plotted against FreSCo⁺ reference data in Figure 14, and a reasonable agreement is found. At the

same time, the validity of both ELBE motion modelers is shown, as both the ODE and our in-house quaternion-based motion modeler produce almost identical results.

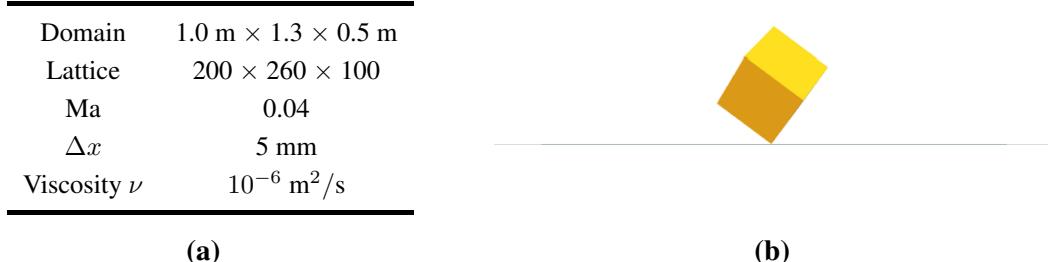


Figure 12. Freely falling cube: parameters and initial configuration. (a) Parameters; (b) initial condition.

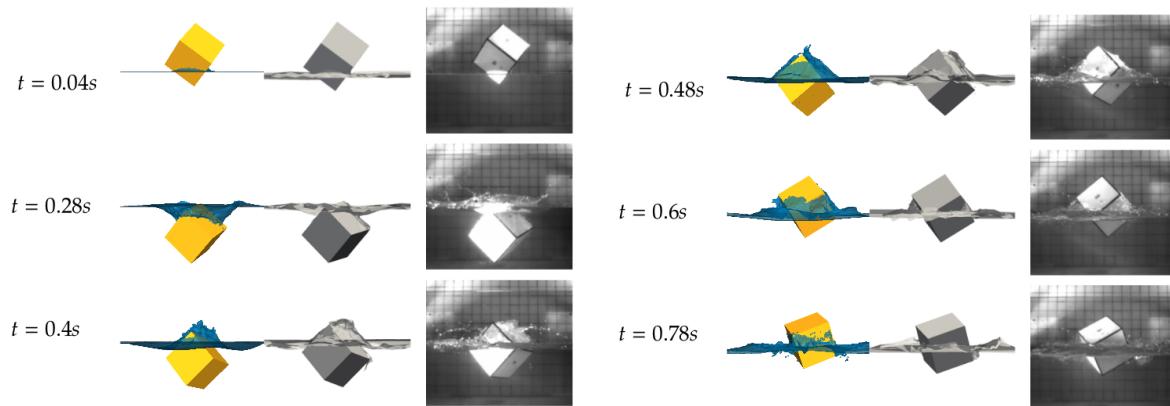


Figure 13. Freely falling cube: results for six selected time steps.

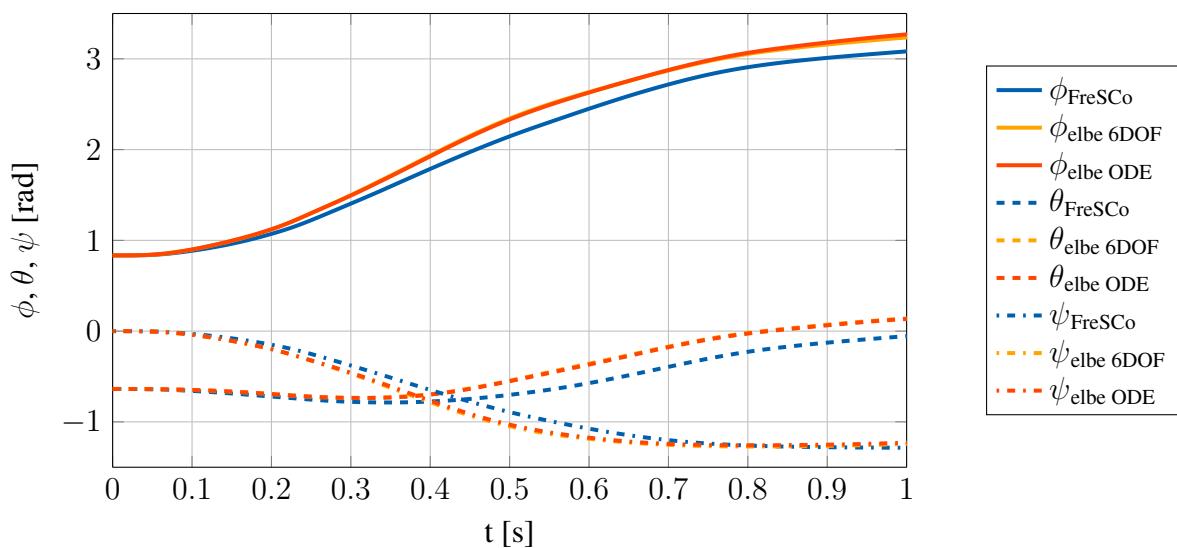


Figure 14. Freely falling cube: Euler angles. Comparison of FreSCo⁺ results with ELBE-ODE (Open Dynamics Engine) and ELBE-6DOF (quaternion-based motion modeler).

The numerical simulation of the twp-second event took approximately 5.4 h on a 10-core CPU machine using FreSCo⁺, while the ELBE simulation lasted less than 10 min on a recent NVIDIA GTX Titan gaming board.

4.2. ODE Performance Considerations

For enhanced debris flow simulations, we recently coupled ELBE to a physics engine, the open-source Open Dynamics Engine (ODE) (<http://www.ode.org>), to model floating and colliding rigid multi-body systems. After the initial physical validations, the second test case serves to investigate the calculation costs of ODE's collision method, as the collision detection/handling for triangulated surface mesh geometries was not investigated by other groups in detail yet and is examined in the following. ODE splits the collision handling into two separate parts, the collision detection and the collision handling. To accelerate the collision detection, it is split up into two phases. First, it is tested if the axis-aligned bounding boxes (AABB) of the geometries under consideration overlap (broad phase). If this is the case, the selected collision detection library tests for actual intersections of the two putatively colliding geometries (narrow phase). In certain situations, further efficiency improvements can be gained by using one of four different hierarchical bounding spaces or an arbitrary combination of them.

Two complex triangulated mesh geometries are considered [46]. One geometry is fixed, and the second geometry is freely falling and accelerated towards the first one by gravitational forces. To achieve as much contacts as possible in time, the restitution coefficient is set to $k = 0.01$. The time step size is set to $\Delta t = 10^{-5}$ s. A total of five different surface mesh sizes were tested. In the first simulation, both surface meshes consists of 10.960 triangles. In the following runs, the triangle count is increased quadratically by the factors 2^2 , 3^2 , 4^2 and 5^2 . Three time-consuming operations have to be considered: the collision detection, the collision handling and the overhead of ELBE to finish the current time step and to initialize the collision detection of the next time step.

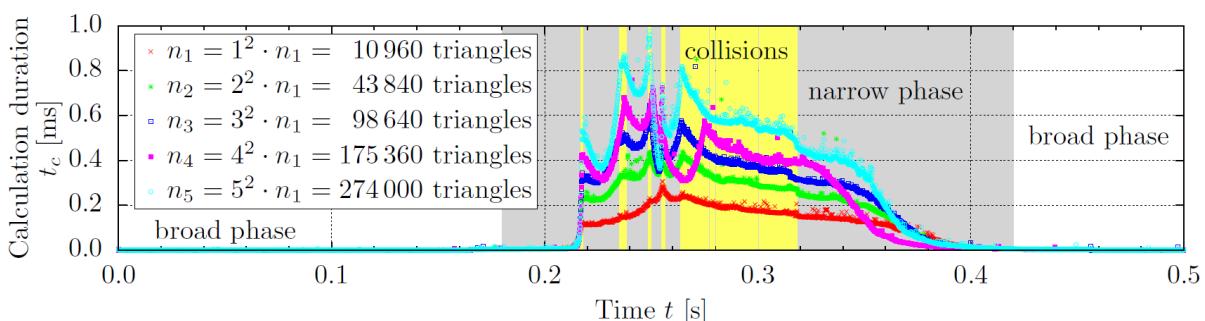


Figure 15. Performance of ODE's collision detection method, coupled to ELBE, $\Delta t = 10^{-5}$ s.

Figure 15 shows the detailed time measurements of the collision detection part of the computation, plotted against the total simulation time t . In the broad phase, the computational costs are almost zero (beginning and end of the simulation), while the computational costs in the narrow phase (once the AABBs intersect) are significantly higher (middle part, in gray, $0.18\text{ s} < t < 0.42\text{ s}$). When the actual collisions take place, the calculation time rises again significantly (yellow). Two fundamental

conclusions can be drawn here: First, since the calculation times of the five simulations nearly increase linearly, while the triangle count n increases quadratically, the computational cost of the collision detection is approximately of $\mathcal{O}(\sqrt{n})$. Second, as a triangle mesh refinement has no direct impact on the size of the LCP itself, the computational cost of the collision handling remains constant for all five cases, $t \approx 0.01$ ms. The remaining part of the time loop, *i.e.*, the force calculation and the re-gridding step, is much more expensive. A linear increase with the number of triangles (and consequently, the number of rigid bodies) is observed. To sum up, the ODE can efficiently address complex multibody problems, also for tessellated surface meshes, as commonly used to describe and model complex engineering geometries.

4.3. Debris Flow Application

To demonstrate the applicability of the coupled ELBE-ODE solver, complex fluid-multi-body interactions of an incoming wave over a beach-section with obstacles are shown (Figure 16). The incoming wave is generated by a breaking dam scenario. The obstacles are two pairs of containers and a duck, being characterized by surface meshes, as specified in Table 2. Two-point-five seconds of real-time flow were simulated within 4 h of computation, yielding an average node update rate of 40 million node updates per second (MNUPS). Further optimizations of the GPU-accelerated coupling algorithms, including an optimized force calculation based on k-d trees and axis-aligned bounding boxes, are currently part of active research. At the same time, in the scope of naval architecture, several scenarios of complex fluid-multi-body interactions, e.g., of incoming waves that hit a section of a shipping harbor and ship-ice interactions, are currently investigated. Nevertheless, the debris flow simulation nicely demonstrates the capability of the coupled ELBE-ODE solver to simulate fluid-structure interactions in civil and environmental engineering.

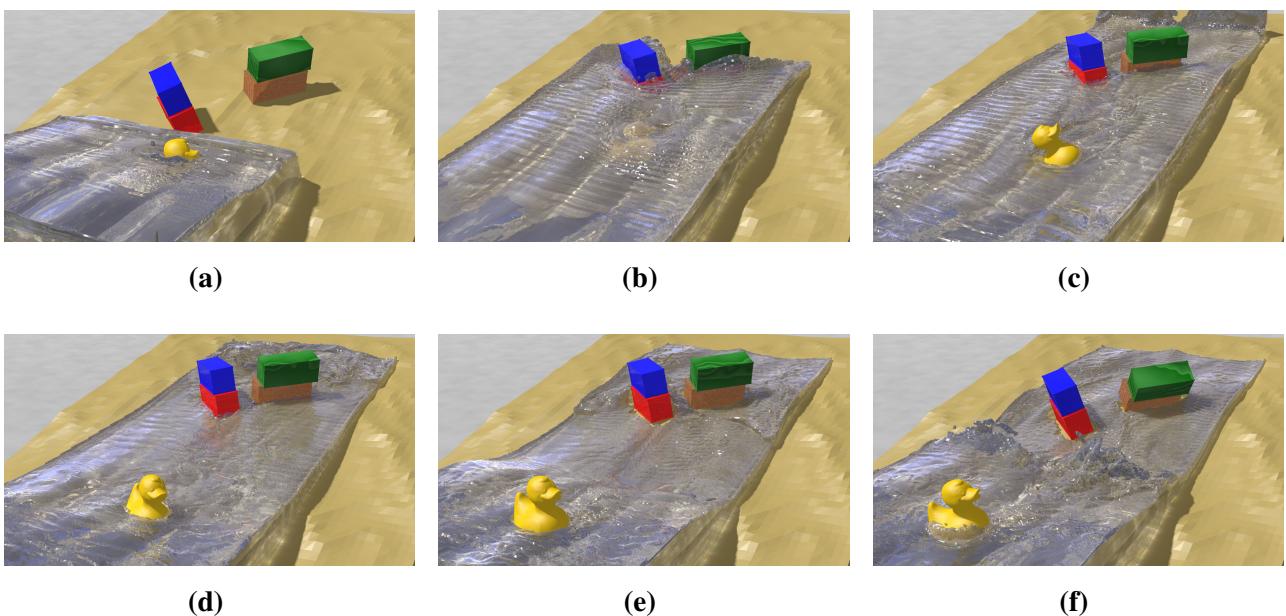


Figure 16. Rendered screenshots of the debris flow simulation. (a) $t = 0.3$ s; (b) $t = 0.75$ s; (c) $t = 1.2$ s; (d) $t = 1.65$ s; (e) $t = 2.1$ s; (f) $t = 2.5$ s.

Table 2. Simulation parameters of the debris flow test case. (a) ELBE; (b) ODE.

	(a)	(b)	
Domain	2.0 m × 0.8 m × 0.5 m (1:10)	Restitution coefficient	0.5
Lattice	400 × 160 × 100	Friction coefficient	0.5
Δx	0.005 m	Number of triangles	
Δt	3E–5 s	Bathymetry	8958
Re	750,000	Containers	4 × 1730
Fr	5.3	Duck	10,960
Ma	0.05		

5. Dam Break Simulations

The second principal application for CFD problems in civil engineering that is addressed in this contribution is related to dam break scenarios. In these applications, a column of a fluid is freed at Time 0, to collapse and flow under gravity, yielding a surge wave of considerable height that potentially interacts with structures in the flow. Dam breaks have been examined in many publications, starting with the pioneering work of Martin and Moyce [47] in the early 1950s. Here, two more sophisticated dam breaks with practical relevance for civil engineers are addressed: (i) a bund wall overtopping case; and (ii) a dam break with surge wave impact on an obstacle in the flow.

5.1. Bund Wall Overtopping

The first breaking dam test case is related to the safety of cylindrical reservoirs containing dangerous chemicals. In case of catastrophic failure, fluid would flow from the reservoirs into a retaining trench encircled by a vertical containment wall. The question is how much fluid would overtop the wall for certain failure types and configurations. The problem can be studied in a radial direction (*i.e.*, 2D vertical slice), and 2D laboratory experimental data are available for validation. The final goal of such simulations is, *e.g.*, to develop predictions for the overtopping characteristics and the dynamic pressures on the bund walls post catastrophic failure of the storage vessels and, if possible, to propose methods of mitigation.

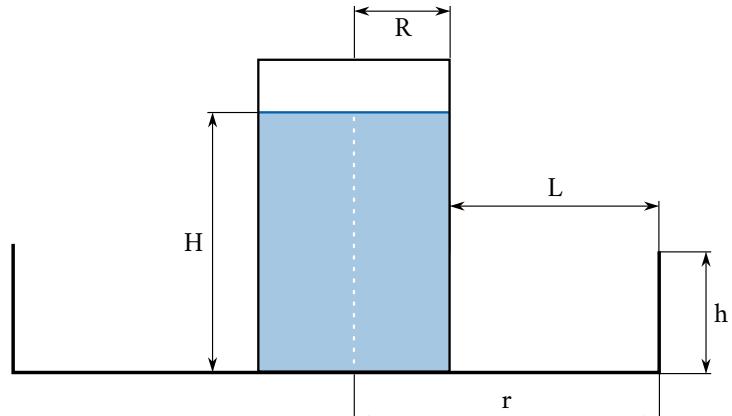
For the purpose of numerical modeling, a 3D tank rupture is simulated as a 2D dam break problem (Figure 17b). The fluid volume of height H and radius R in the tank is released at time $t = 0$. The fluid then propagates towards and hits a bund of height h (modeled as a stationary wall), whose toe is located at a distance L from the tank. The modeling will continue until the fluid is stationary, and then, the amount of fluid that overtops the wall is calculated.

The 12 investigated cases are given in Table 3a, in accordance with the work of [48]. Resulting free surface locations and wave patterns for four selected cases are depicted in Figure 18. Entire flow patterns and overtopping volumes for different tank and bund wall configurations can be observed. In Table 3b, numerical results for the overtopping volume Q (relative to the total fluid volume Q_0) and the bund wall impact pressure p' (relative to the hydrostatic pressure p_0) are compared to the experimental reference

data for all 12 cases. A very good agreement can be seen for both parameters: the relative error is below 16% (for the overtopping volume) and 17% (for the impact pressure), emphasizing the predictive accuracy of ELBE even for complex engineering applications.

Radius R	300 mm
Height H	{120, 300, 600} mm
Radius r	{520, 574, 775, 822} mm
Height h	{0.2 – 0.4} H

(a)



(b)

Figure 17. Specification of overtopping test cases. (a) Parameters; (b) tank and bund nomenclature [48].

Table 3. Simulation parameters and results. (a) Simulation parameters in accordance with the tank spill experiments of [48]; (b) results of simulations in comparison to the tank spill experiments of [48].

Case	Tank		Bund					
	R (mm)	H (mm)	r (mm)	h (mm)	exp.	num.	exp.	num.
1	300	120	574	36	1	32.00	30.99	1.32
2	300	300	574	90	2	38.00	36.87	1.97
3	300	600	574	180	3	34.00	28.95	1.79
4	300	120	520	48	4	22.00	24.88	2.01
5	300	300	520	120	5	26.00	29.90	1.65
6	300	600	520	240	6	21.00	24.37	1.82
7	300	120	822	24	7	28.00	30.65	4.12
8	300	300	822	60	8	45.00	38.26	2.80
9	300	600	822	120	9	43.00	36.87	2.22
10	300	120	775	36	10	15.00	16.68	2.09
11	300	300	775	90	11	24.00	25.22	1.35
12	300	600	775	180	12	24.00	27.10	1.23

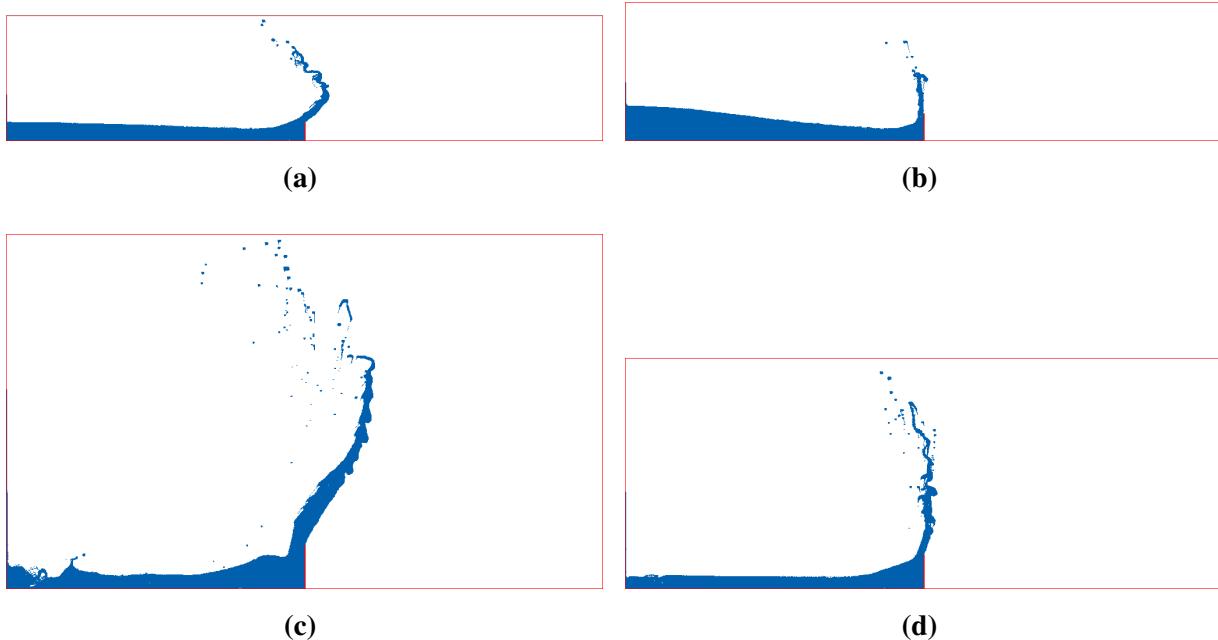


Figure 18. Results for the overtopping test case of Atherton [48] for four different cases. (a) Case 1, $t = 0.5$ s; (b) Case 4, $t = 0.35$ s; (c) Case 9, $t = 0.6$ s; (d) Case 11, $t = 0.6$ s.

5.2. Wave Impact

Breaking dam scenarios are standard validation cases for free surface flow solvers. Such cases do not require the use of elaborate boundary conditions, so that here, a more complex experiment including wave gauges and pressure probes is selected for validation purposes; see Figure 19.

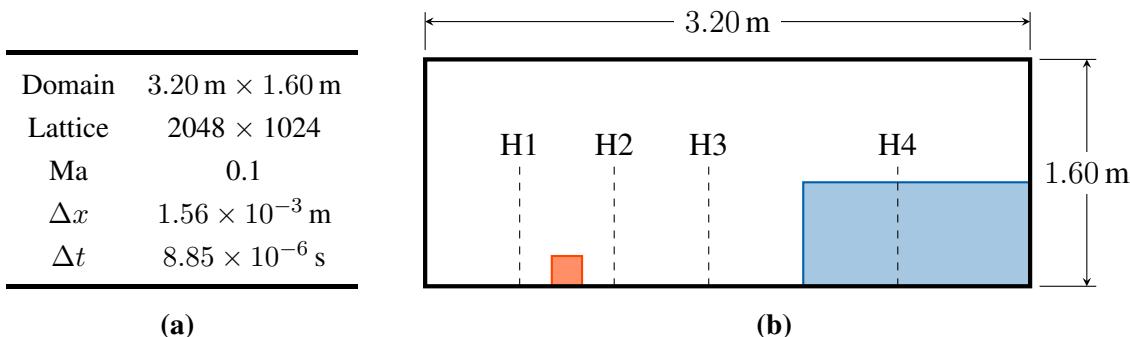


Figure 19. Specification of the wave impact test case. (a) Parameters; (b) test case setup [49].

The experiments were conducted in a laboratory tank at the Maritime Research Institute Netherlands (MARIN) [49,50]. A box was placed in the tank and was covered by eight pressure sensors, four on the front of the box at height $z = \{0.025, 0.063, 0.099, 0.136\}$ m and four on the top of the box at $x = \{0.806, 0.769, 0.733, 0.696\}$ m. In the back part of the tank, a water column is constrained by a door, which is pulled up by releasing a weight at the initial time and is made to collapse. The water elevation then was measured in the tank at four locations (H1–H4, in the reservoir and in the tank, at $x_i = \{0.5, 1.0, 1.5, 2.66\}$ m). The experimental results for the height probes H2 and H4 and the pressure probes P1 and P7 are public domain.

The problem is addressed with a fully two-dimensional model, as the main physical effects in this case are basically two-dimensional. No periodic boundary conditions were employed. In Figure 20, the ELBE results are plotted against the experimental reference data for two selected pressure probes at the front surface (P1) and the upper surface (P7) of the box. The plotted LBM pressure signals were processed with a moving average low-pass filter with a reasonable filter width to remove the high-frequency pressure noise. The experimental data are reproduced very well, both in terms of amplitude and frequency. On top of the accurate prediction of the initial impact, even the arrival time of the reflected wave at $t \approx 4.7$ s is reproduced very well. The numerical simulation for this test case took approximately 60 min, on the two million node grid, including a full post-processing. In combination with the high result accuracy, this emphasizes the applicability of ELBE to civil-engineering-related wave impact scenarios.

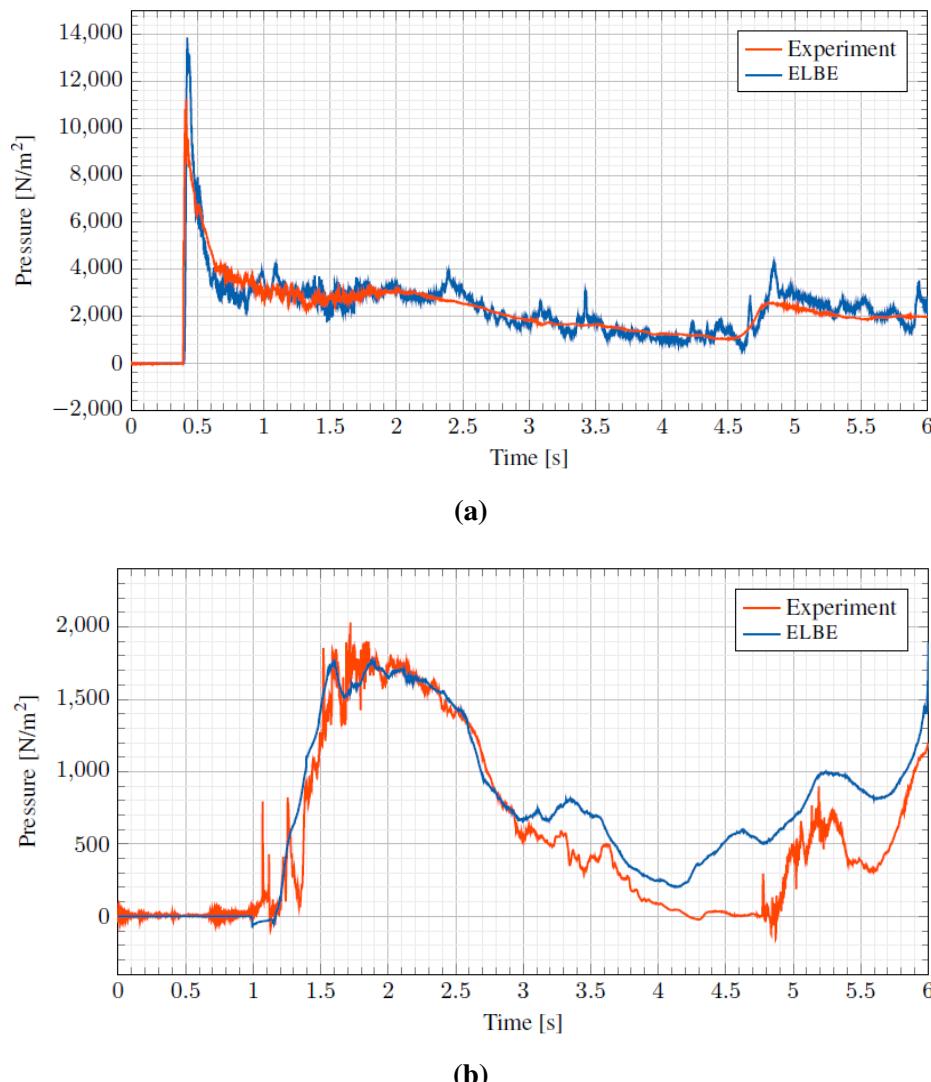


Figure 20. ELBE results for two selected pressure probes, high-resolution simulation with $\Delta x \approx 1.56$ mm and Mach number $Ma = 0.1$. (a) Pressure probe P1 (low-pass filtered); (b) pressure probe P7 (low-pass filtered).

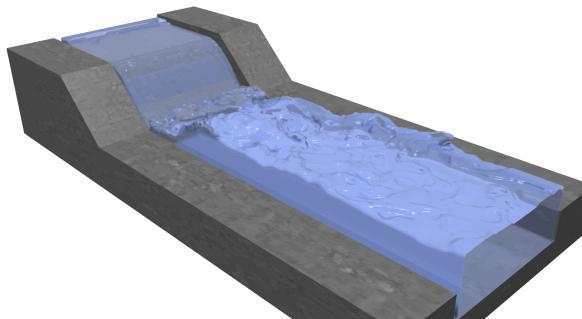
6. Numerical Wave Tank for On- and Off-Shore Structures

Finally, a more classic application of CFD in engineering is addressed. While the first two scenarios demonstrated the applicability of the ELBE code for violent flow situations, such as tsunamis and wave impact, the evaluation of hydrodynamic forces for non-violent cases is one of the main and very basic applications for CFD techniques in civil and environmental engineering. Three application examples are discussed in the following.

6.1. River Engineering

As a first application, the flow past a weir is examined, as detailed in Figure 21. The parameters for our simulation are chosen in such a way that the subcritical inflow switches to a supercritical state while or shortly after passing the weir and jumps back to the subcritical state after passing the weir. In this hydraulic jump, energy is dissipated, so that the downstream flow velocity is lowered and erosion effects in the river bed are reduced.

Domain	$38.4 \text{ m} \times 6.4 \text{ m} \times 4.8 \text{ m}$
Flux	$q = 2.8 \text{ m}^2/\text{s}$
Inflow velocity	$v_1 = 0.70 \text{ m/s}$
Gravity	$g = 9.81 \text{ m/s}^2$
Viscosity	$\nu = 1 \times 10^{-6} \text{ m}^2/\text{s}$
Lattice	$192 \times 32 \times 24, \Delta x = 0.2 \text{ m}$ $384 \times 64 \times 48, \Delta x = 0.1 \text{ m}$ $576 \times 96 \times 72, \Delta x = 0.067 \text{ m}$ $768 \times 128 \times 96, \Delta x = 0.05 \text{ m}$



(a)

(b)

Figure 21. Flow past a weir. (a) Parameters; (b) snapshot at the end of the simulation at $t = 60 \text{ s}$.

The corresponding flow parameters can be estimated via Bernoulli's equation for the inflow (1), the area directly behind the weir (2) and the outflow region (3), yielding analytical values of the Froude number, flow velocity and water heights (Figure 22a). The Froude number Fr_2 behind the weir also determines the characteristics of the hydraulic jump [51]. The estimated Froude number around $\text{Fr}_2 \approx 4.5$ is generally considered to be sufficiently high for a stable, oscillating and wavy hydraulic jump. The weir geometry has to be chosen carefully, to guarantee the required performance and to avoid negative pressures and the risk of cavitation. On top a step in the rear part of the domain prevents the hydraulic jump from moving downstream and leaving the domain.

A rendered snapshot of the simulation after $t = 60 \text{ s}$ is shown in Figure 21b. Moreover, our numerical results for flow Sections 2 and 3 are compared to the Bernoulli estimates in Figure 22a, for the four consecutively-refined grids. In general, a fairly good agreement can be observed. However, in the supercritical flow section on the back of the weir, the numerical predicted water level on the coarsest grid is three-times higher than the Bernoulli prediction. Grid refinement slightly improves (*i.e.*, lowers) the value of water height Probe 2, but still, the numerical prediction is two-times higher. This discrepancy

might occur due to two reasons: (i) The no-slip boundary condition on the weir surface decelerates the flow, which in combination with the fairly low grid resolution in this area, yields an increased water level and a reduced flow velocity in Section 2. (ii) Bernoulli's equation is valid for inviscid, irrotational flows only and does not consider viscous and/or turbulent dissipation, so that the total energy of the fluid in Section 3 is overestimated, yielding a lower theoretical water level than observed numerically.

In Figure 22b, the x-component of the flow velocity in the hydraulic jump is shown. The maximum flow velocity does match the predicted value of $v_2 = 8.51$ m/s, and the expected flow behavior in the hydraulic jump can be observed, including turbulent structures and local backflow. Moreover, the near-wall slowdown on the back of the weir can be identified. Finally, the Smagorinsky LES model tends to overestimate the turbulent effects in the near-wall region, which should be tackled by the use of dynamic Smagorinsky models or advanced turbulence models. The simulations on the coarsest grid ($\Delta x = 0.1$ m, $\Delta t = 1.4 \times 10^{-3}$ s) are in the same order of magnitude as the time scale of the actual fluid event.

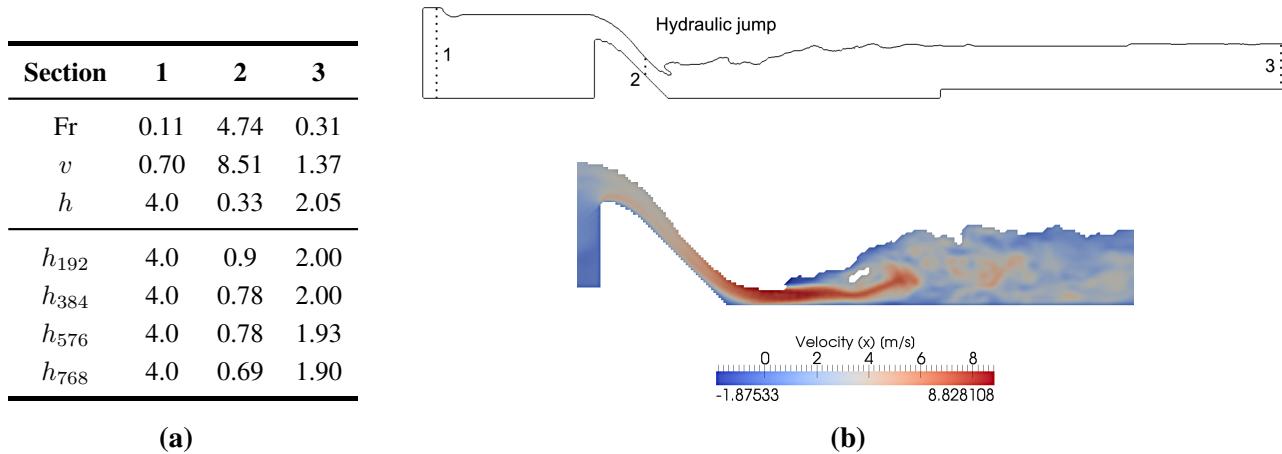


Figure 22. Results for the flow past a weir (GPU simulation). (a) Bernoulli predictions (top) vs. ELBE results (bottom); (b) slices with Sections 1–3 and the velocity profile for a grid spacing $\Delta x = 0.067$ m.

6.2. Wave-Current-Induced Loads on Submerged Bodies

As a second validation example and potential application of the ELBE wave tank, wave-current-induced loads on submerged bodies are addressed, for the case of a generic gravity foundation. In the scope of the development of renewable energies, tidal stream generators are getting more and more popular. While there exist several different concepts to install tidal stream generators in the open sea, one of the most popular ones is the gravity foundation. During the structural design process, it is important to know the hydrodynamic loads on the foundation caused by current and waves. In addition to simplified hydrodynamic models, which estimate the structural loads based on inviscid and/or irrotational flows, fully-resolved CFD tools, including wave and current boundary conditions, can give insight into the detailed hydrodynamic loads. In this context and to validate the employed CFD tools, a numerical and experimental benchmark for hydrodynamic wave-current loads acting on generic gravity foundations was recently published by Markus *et al.* [52]. Five different geometries

are analyzed experimentally and numerically (Figure 23b) for scenarios with currents, waves and combined wave-current loads. Here, we present ELBE results for two selected scenarios, to show the validity of the hydrodynamic approach, the force calculation methods and wave-current inlet/outlet boundary conditions.

Domain	$3 \text{ m} \times 0.55 \text{ m}$
Lattice	512×94
Ma	0.025
Δx	$5.86 \times 10^{-3} \text{ m}$
Δt_{C3}	$2.23 \times 10^{-4} \text{ s}$
Δt_{H3}	$1.7746 \times 10^{-4} \text{ s}$
Viscosity	$\nu = 1 \times 10^{-6} \text{ m}^2/\text{s}$

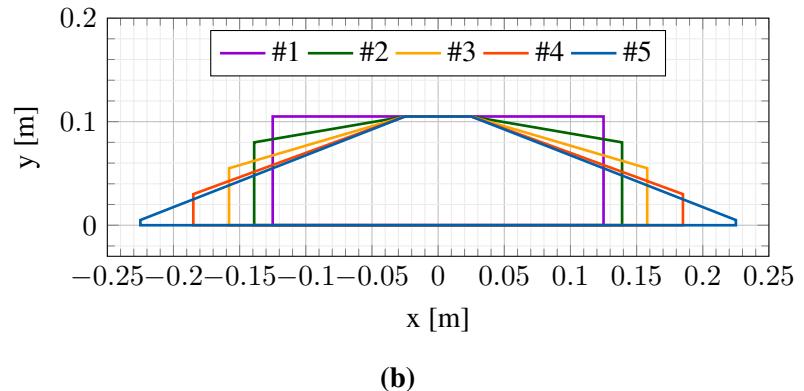


Figure 23. Flow past a generic gravity foundation model. (a) Parameters; (b) experimentally-investigated geometries.

The five geometries have a cross-sectional area of 250 cm^2 , a height from the ground of 10.5 cm and a top base dimension of at least 5 cm. The load conditions are split up into three groups: current loads, wave-induced loads and wave-current loads. To demonstrate the applicability of ELBE to such scenarios, two cases are selected: a current load (case C3 in [52]) with a flow velocity of $v = 0.38 \text{ m/s}$ and a wave load (case H3 in [52]) with a wave height $h = 0.12 \text{ m}$ and a wave period of $T = 1.9 \text{ s}$. The initial water height above the structure is chosen to be $h_0 \approx 0.35 \text{ m}$, in accordance with the experimental data.

The resulting horizontal force results are summarized in Table 4. Note that the given current loads are mean force values over a time span of 5 s, and the given wave loads are the average value of the horizontal force maxima.

Table 4. Comparison of numerical results from the ELBE wave tank to experimental data [52]: horizontal forces, given in (N/m).

Geometry	Current C3		Waves H3		
	#	Exp.	ELBE	Exp.	ELBE
1		14.39	13.92	80.45	78.67
2		3.69	5.16	60.64	63.64
3		8.07	8.97	59.79	63.44
5		9.53	16.39	51.44	61.45

In Figure 24, ELBE results are plotted against the foundation inclination angle, again in comparison to the experimental data for both load scenarios. Very good agreement can be seen for smaller inclination angles, while the ELBE solution differs with increasing inclination angle. This is due to the geometry representation in ELBE, which was restricted to first order for the selected cases. Hence, the geometry with a higher inclination angle is represented as worse than rectangular, axis-parallel geometries.

Consequently, and as expected, the best results are obtained for Geometry 1, where the numerical and experimental results differ by only about 3%. Finally, the transient force signals are compared to the experimental data; see Figure 25. A very good agreement both for the maximum amplitude and the period of the force signal can be observed.

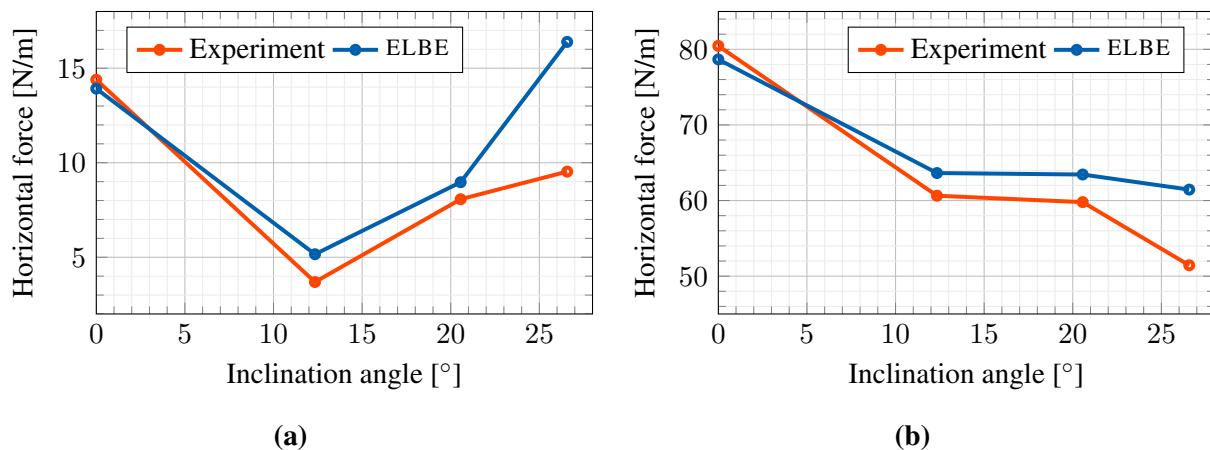


Figure 24. Comparison of horizontal forces in experimental and numerical wave tanks for two different scenarios, as a function of inclination angle. (a) Current condition C3; (b) wave condition H3.

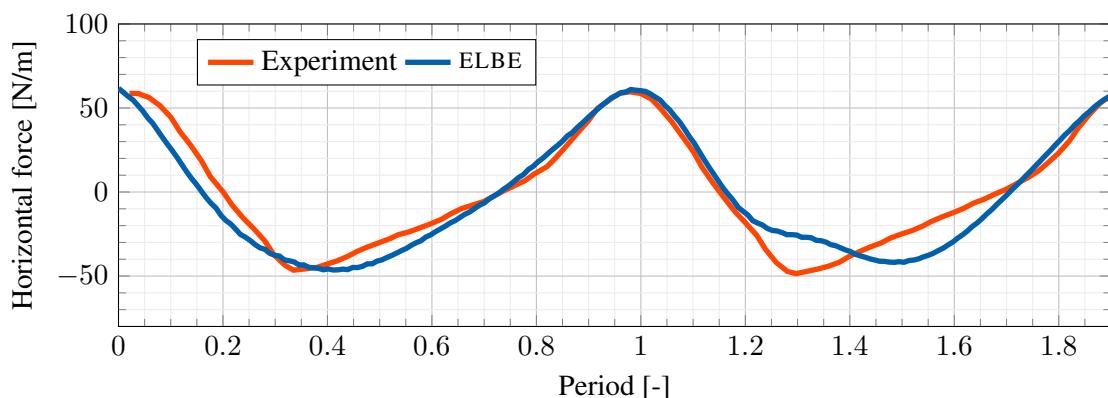


Figure 25. Comparison of transient experimental and numerical results for the horizontal hydrodynamic forces; two selected wave periods, Geometry 3.

To conclude, it could be shown that ELBE is applicable to the numerical simulation of wave/current-induced loads on submerged structures and can be a helpful tool in the structural design process. Thanks to the efficient implementation and the innovative GPU hardware that is used by the code, the presented simulations could be run in less than one minute, on a Kepler K40M board. Hence, such simulations could easily be included into the foundation design process and serve as a basis for a proper shape optimization.

6.3. Vortex-Induced Vibrations

Last, but not least, a very recent and up-to-date phenomenon is addressed: structural vibrations that are induced by transient, turbulent flows. Detailed investigations of such vortex-induced vibrations were, e.g., made by Mittal and Kumar [53]. They examined a circular cylinder in a uniform flow at $Re = 325$ with two translational degrees of freedom using a stabilized space-time finite element formulation. Results were presented for various values of structural frequency of the one-mass oscillator. We repeated these test cases with ELBE (see parameters in Table 5) and achieved results with a quality comparable to those of Mittal and Kumar [53] and Geller [17]. Figure 26 shows the trajectories for eight different dimensionless eigenfrequencies F_s . Note that $F_s = 0.21$ represents the resonance case with the highest amplitude. Even here, a very good agreement can be seen. Further information on the test case can be found in [17,53].

Table 5. Vortex-induced vibrations test case parameters.

Domain	3.20 m \times 1.60 m
Lattice	512 \times 256
Δx	0.47 m, 0.23 m
Reynolds number Re	325
Strouhal number F_s	0.18–0.53
Dimensionless mass \bar{m}	4.7273
Lehr's damping ratio	0.00033

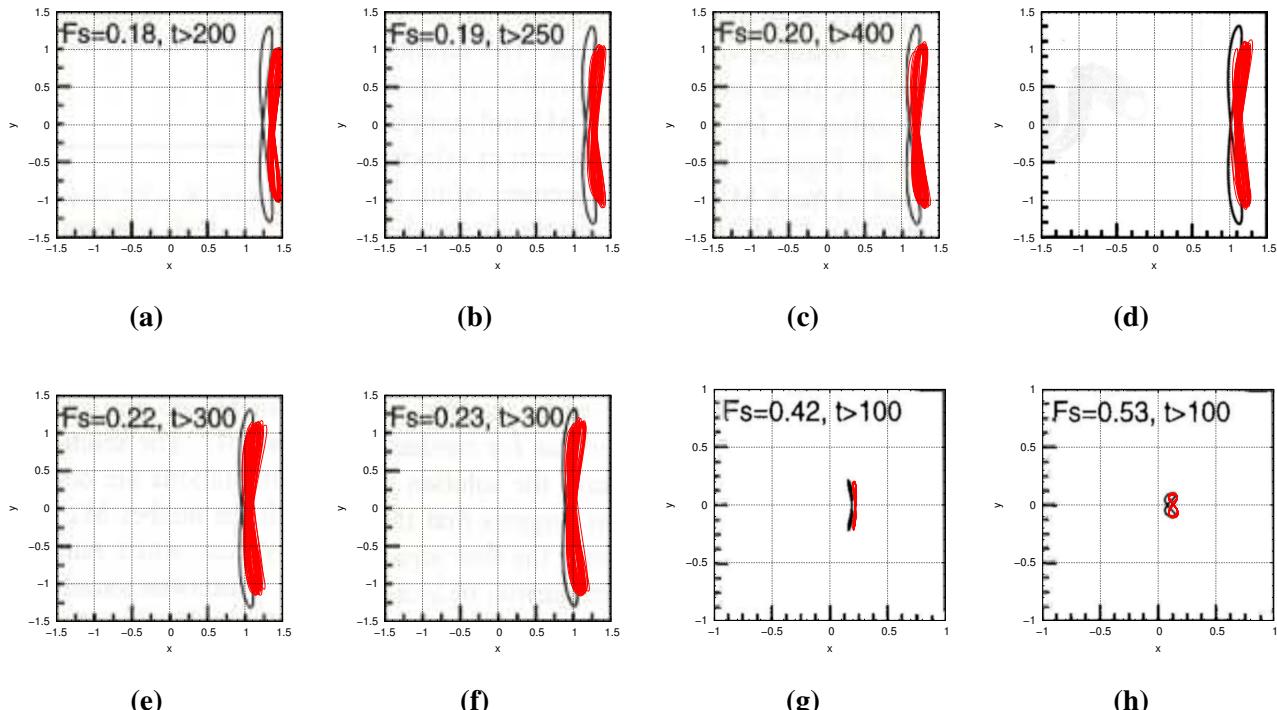


Figure 26. Results for the oscillating cylinder test case for varying Strouhal numbers F_s (red), in comparison to reference data of [53] (black). (a) $F_s = 0.18$; (b) $F_s = 0.19$; (c) $F_s = 0.20$; (d) $F_s = 0.21$; (e) $F_s = 0.22$; (f) $F_s = 0.23$; (g) $F_s = 0.42$; (h) $F_s = 0.53$.

7. Summary and Conclusions

We presented several applications of the efficient lattice Boltzmann environment ELBE with practical relevance for various civil engineering applications. Wave propagation, wave impact and less violent hydrodynamic load cases were addressed. The validations show a good performance of the code and a good agreement of the numerical results with the reference data in most of the cases. In combination with the recently implemented OpenGL-based visualizer tool, which allows one to visualize the simulation results during runtime [54], the incorporation of this high-end numerical tool into a simulation-based design (SBD) process of civil and environmental engineers has become more than possible. In future work, several model aspects will be improved, particularly concerning wetting/drying models in the shallow water simulations, higher order boundary conditions near the solid surfaces, grid refinement and a coupling to inviscid methods.

Acknowledgments

The authors gratefully acknowledge the support of NVIDIA, in the scope of their Academic Partnership Program, for proposals on tsunami warning systems (2011, University of Rhode Island, Kingston, RI, USA) and violent flow modeling (2012, Hamburg University of Technology, Hamburg, Germany), as well as NVIDIA GPU Research Center (Hamburg University of Technology, Hamburg, Germany, since 2014). The authors thank Alexander Steinert for the data preparation and numerical simulation of the tsunami example in Section 3.1.2. The authors acknowledge the support of Stephan T. Grilli, Arash Bigdeli and Amir Banari for the simulations of bund wall overtopping in Section 5.1, which have been conducted with ELBE at the University of Rhode Island in the scope of an industrial research project.

Author Contributions

Christian F. Janßen is chief developer of the ELBE code, did the simulations in Sections 3.1.1, 3.2 and 6.1, supervised the authors working on all other simulations and wrote most of the paper. Dennis Mierke implemented the coupling of ELBE to the ODE rigid body engine. He simulated the debris flow case in Section 4 and the vortex-induced vibration example in Section 6.3. Micha Überrück implemented the first preliminary free surface grid refinement schemes in ELBE and developed the numerical wave tank used for the simulation in Section 6.2. He also assisted in preparing most of the figures. Silke Gralher worked on the two-dimensional slamming simulations and the FFT filtering of pressure signals discussed in Section 5.2. Thomas Rung leads the Computational Fluid Dynamics group and supervises all projects.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Le Touzé, D.; Grenier, N.; Barcarolo, D.A. *2nd International Conference on Violent Flows*. Available online: <http://www.publibook.com/librairie/livre.php?isbn=9782748390353> (accessed on 7 July 2015).
2. Dean, R.G.; Dalrymple, R.A. *Water Wave Mechanics for Engineers and Scientists*; World Scientific Publishing: Singapore, 1991.
3. Dunbar, P.; McCullough, H.; Mungov, G.; Varner, J.; Stroker, K. 2011 Tohoku earthquake and tsunami data available from the National Oceanic and Atmospheric Administration/National Geophysical Data Center. *Geomat. Nat. Hazards Risk* **2011**, *2*, 305–323.
4. Synolakis, C.; Bernard, E. Tsunami science before and beyond Boxing Day 2004. *Philos. Trans. A Math. Phys. Eng. Sci.* **2006**, *364*, 2231–2265.
5. Wienke, J.; Oumeraci, H. Breaking wave impact force on a vertical and inclined slender pile—Theoretical and large-scale model investigations. *Coast. Eng.* **2005**, *52*, 435–462.
6. D’Humieres, D.; Ginzburg, I.; Krafczyk, M.; Lallemand, P.; Luo, L.S. Multiple-Relaxation-Time Lattice Boltzmann models in three dimensions. *R. Soc. Lond. Philos. Trans. Ser. A* **2002**, *360*, 437–451.
7. Ginzburg, I.; Steiner, K. Lattice Boltzmann model for free-surface flow and its application to filling process in casting. *J. Comput. Phys.* **2003**, *185*, 61–99.
8. Junk, M.; Yang, Z. Pressure boundary condition for the lattice Boltzmann method. *Comput. Math. Appl.* **2009**, *58*, 922–929.
9. Krafczyk, M.; Tölke, J.; Luo, L.S. Large-eddy simulations with a multiple-relaxation-time LBE model. *Int. J. Mod. Phys. B* **2003**, *17*, 33–39.
10. He, X.; Luo, L.S. Lattice Boltzmann model for the incompressible Navier-Stokes equation. *J. Stat. Phys.* **1997**, *88*, 927–944.
11. Körner, C.; Thies, M.; Hofmann, T.; Thürey, N.; Rüde, U. Lattice Boltzmann Model for Free Surface Flow for Modeling Foaming. *J. Stat. Phys.* **2005**, *121*, 179–196.
12. Guo, Z.; Zheng, C.; Shi, B. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E* **2002**, *65*, 046308.
13. Janßen, C.; Krafczyk, M. A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids. *Comput. Math. Appl.* **2010**, *59*, 2215–2235.
14. Donath, S.; Feichtinger, C.; Pohl, T.; Götz, J.; Rüde, U. A Parallel Free Surface Lattice Boltzmann Method for Large-Scale Applications. In *Parallel Computational Fluid Dynamics: Recent Advances and Future Directions*; DEStech Publications: Lancaster, PA, USA, 2010; pp. 318–330.
15. Anderl, D.; Bogner, S.; Rauh, C.; Rüde, U.; Delgado, A. Free surface lattice Boltzmann with enhanced bubble model. *Comput. Math. Appl.* **2014**, *67*, 331–339.
16. Janßen, C.; Krafczyk, M. Free surface flow simulations on GPGPUs using LBM. *Comput. Math. Appl.* **2011**, *61*, 3549–3563.
17. Geller, S. Ein explizites Modell für die Fluid-Struktur-Interaktion basierend auf LBM und p-FEM. Ph.D. Thesis, TU Carolo-Wilhelmina zu Braunschweig, Braunschweig, Germany, 2010. (In German)

18. Geller, S.; Kollmannsberger, S.; Bettah, M.; Scholz, D.; Düster, A.; Krafczyk, M.; Rank, E. An explicit model for three dimensional fluid structure-interaction using LBM and p -FEM. In *Fluid-Structure Interaction II*; Bungartz, H.J.; Schäfer, M., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2010; pp. 285–325.
19. Janßen, C.F.; Koliha, N.; Rung, T. A fast and rigorously parallel surface voxelization technique for GPGPU-accelerated CFD simulations. *Commun. Comput. Phys.* **2015**, doi:10.4208/cicp.2014.m414.
20. Stewart, D.E.; Trinkle, J.C. An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Inelastic Collisions and Coulomb Friction. *Int. J. Numer. Methods Eng.* **1996**, *39*, 2673–2691.
21. Boeing, A.; Bräunl, T. Evaluation of Real-Time Physics Simulation Systems. In Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia, Perth, Australia, 1–4 December 2007; pp. 281–288.
22. Hummel, J.; Wolff, R.; Stein, T.; Gerndt, A.; Kuhlen, T. An Evaluation of Open Source Physics Engines for Use in Virtual Reality Assembly Simulations. *Adv. Vis. Comput.* **2012**, *7432*, 346–357.
23. Metrikine, I.; Borzov, A.; Lubbad, R.; Løset, S. Numerical Simulation of a Floater in a Broken-Ice Field: Part II—Comparative Study of Physics Engines. In Proceedings of the 31st International Conference on Ocean, Offshore and Arctic Engineering (OMAE2012), Rio de Janeiro, Brazil, 1–6 July 2012; Volume 6, pp. 477–486.
24. Tölke, J.; Krafczyk, M. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. *Comput. Vis. Sci.* **2008**, *1*, 29–39.
25. Tölke, J.; Krafczyk, M. TeraFLOP computing on a desktop PC with GPUs for 3D CFD. *Int. J. Comput. Fluid Dyn.* **2008**, *22*, 443–456.
26. NVIDIA. List of GPU-Accelerated Applications. 2015. Available online: <http://www.nvidia.com/object/gpu-applications.html> (accessed on 15 January 2015).
27. Grilli, S.; Harris, J.; Tajalibakhsh, T.; Kirby, J.; Shi, F.; Masterlark, T.; Kyriakopoulos, C. Numerical simulation of the 2011 Tohoku tsunami: Comparison with field observations and sensitivity to model parameter. In Proceedings of the 22nd Offshore and Polar Engineering Conference, ISOPE12, Rodos, Greece, 17–23 June 2012.
28. Ocean Networks Canada. *A Tsunami Detection Initiative for British Columbia*. Workshop Report; Port Alberni (Canada), 27–28 March 2014; Ocean Networks Canada, University of Victoria; Victoria, BC, Canada, 2014.
29. Kowalik, Z.; Murty, T.S. *Numerical Modeling of Ocean Dynamics*; World Scientific Publishing: Singapore, 1993.
30. Wei, G.; Kirby, J.; Grilli, S.; Subramanya, R. A fully nonlinear Boussinesq model for surface waves. I. Highly nonlinear, unsteady waves. *J. Fluid Mech.* **1995**, *294*, 71–92.
31. Shi, F.; Kirby, J.; Harris, J.; Geiman, J.; Grilli, S. A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation. *Ocean Model.* **2012**, *43–44*, 36–51.
32. Zhou, J. *Lattice Boltzmann Methods for Shallow Water Flows*; Springer: Berlin/Heidelberg, Germany, 2003.

33. Chopard, B.; Pham, V.; Lefevre, L. Asymmetric lattice Boltzmann model for shallow water flows. *Comput. Fluids* **2013**, *88*, 225–231.
34. Frandsen, J. Investigations of wave runup using a LBGK modeling approach. In Proceedings of the XVI International Conference on Computational Methods in Water Resources, Copenhagen, Denmark, 18–22 June 2006.
35. Thömmes, G.; Seaid, M.; Banda, M.K. Lattice Boltzmann methods for shallow water flow applications. *Int. J. Numer. Meth. Fluids* **2007**, *55*, 673–692.
36. Tubbs, K. Lattice Boltzmann Modeling for Shallow Water Equations Using High Performance Computing. Ph.D. Thesis, Louisiana State University, Baton Rouge, LA, USA, May 2010.
37. Janßen, C.; Bengel, S.; Rung, T.; Dankowski, H. A fast numerical method for internal flood water dynamics to simulate water on deck and flooding scenarios of ships. In Proceedings of the 32nd International Conference on Ocean, Offshore and Arctic Engineering (OMAE), Nantes, France, 9–14 June 2013.
38. Liu, P.F.; Yeh, H.; Synolakis, C. *Advanced Numerical Models for Simulating Tsunami Waves and Runup*; World Scientific Publishing: Singapore, 2008.
39. Catalina 2004: The third international workshop on long-wave runup models. Benchmark problems #1 – #4. Available online: http://isec.nacse.org/workshop/2004_cornell/benchmark.html (accessed on 15 January 2015).
40. Carrier, G.F.; Wu, T.T.; Yeh, H. Tsunami run-up and draw-down on a plane beach. *J. Fluid Mech.* **2003**, *475*, 79–99.
41. Frandsen, J. A simple LBE wave runup model. *Prog. Comput. Fluid Dyn.* **2008**, *8*, 222–232.
42. Shi, F.; Kirby, J.T.; Tehranirad, B.; Harris, J.C. *FUNWAVE-TVD, Documentation and Users' Manual*; Research Report, cacr-11-04; University of Delaware: Newark, DE, USA, 2011.
43. Amante, C.; Eakins, B.W. ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis. NOAA Technical Memorandum NESDIS NGDC-24. Available online: <https://www.ngdc.noaa.gov/mgg/global/relief/ETOPO1/docs/ETOPO1.pdf> (accessed on 7 July 2015).
44. Okada, Y. Surface deformation due to shear and tensile faults in a half-space. *Bull. Seismol. Soc. Am.* **1985**, *75*, 1135–1154.
45. Kraskowski, M. Validation of the RANSE Rigid Body Motion Computations. In Proceedings of the 12th Numerical Towing Tank Symposium, Cortona, Italy, 4–6 October 2009; Volume 6, pp. 99–104.
46. Willie. Rubber Duck. 2013. Available online: <http://www.thingiverse.com/thing:139894> (accessed on 15 January 2015).
47. Martin, J.C.; Moyce, W.J. Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane. *Philos. Trans. R. Soc. Lond. A* **1952**, *244*, 312–324.
48. Atherton, W.; Ash, J.W.; Alkhaddar, R.M. *An Experimental Investigation of Bund Wall Overtopping and Dynamic Pressures on the Bund Wall Following Catastrophic Failure of a Storage Vessel*; Research Report 333; Liverpool John Moores University: Liverpool, UK, 2005.
49. Kleefsman, K.M.T. Water Impact Loading on Offshore Structures—A Numerical Study. Ph.D. Thesis, University of Groningen, Groningen, The Netherlands, 18 November 2005.

50. Wemmenhove, R. Numerical Simulation of Two-Phase Flow in Offshore Environments. Ph.D. Thesis, Rijksuniversiteit Groningen, Groningen, The Netherlands, 16 May 2008.
51. Chow, V.T. *Open-Channel Hydraulics*; MC Graw-Hill: New York, NY, USA, 1959.
52. Markus, D.; Ferri, F.; Wüchner, R.; Frigaard, P.; Bletzinger, K.-U. Complementary numerical-experimental benchmarking for shape optimization and validation of structures subjected to wave and current forces. *Comput. Fluids* **2014**, *118*, 69–88.
53. Mittal, S.; Kumar, V. Finite element study of vortex-induced cross-flow and in-line oscillations of a circular cylinder at low Reynolds numbers. *Int. J. Numer. Methods Fluids* **1999**, *31*, 1087–1120.
54. Janßen, C.F.; Koliha, N.; Rung, T. Online visualization and interactive monitoring of real-time simulations of complex turbulent flows on off-the-shelf commodity hardware. *Comput. Vis. Sci.* **2015**, submitted.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).