VI International Conference on Computational Methods in Marine Engineering
MARINE 2015
F. Salvatore, R. Broglia and R. Muscari (Eds)

A NEXT-GENERATION CFD TOOL FOR LARGE-EDDY SIMULATIONS ON THE DESKTOP

CHRISTIAN F. JANSSEN, NILS KOLIHA AND THOMAS RUNG

Institute for Fluid Dynamics and Ship Theory
Hamburg University of Technology (TUHH)
Am Schwarzenberg-Campus 4, 21073 Hamburg, Germany
E-Mail: {christian.janssen,thomas.rung}@tuhh.de
Web page: http://www.tuhh.de/elbe

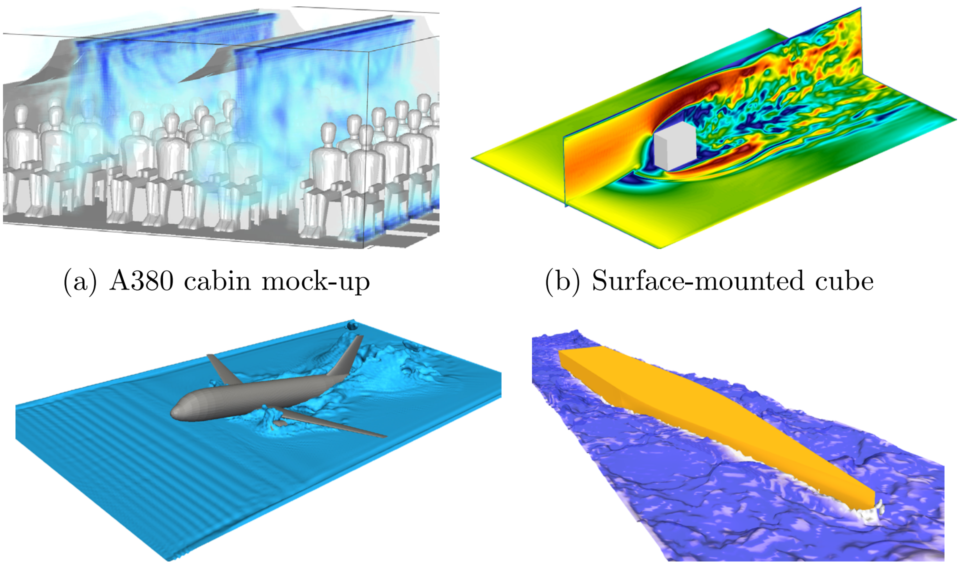
Key words: Lattice Boltzmann Method; ELBE; LES; (near) real-time computing

Abstract. Dive deep into the fascinating world of real-time computational fluid dynamics. We present details of our GPU-accelerated flow solver for the simulation of non-linear violent flows in marine and coastal engineering. The solver, the efficient lattice boltzmann environment ELBE, is accelerated with recent NVIDIA graphics hardware and allows for three-dimensional simulations of complex flows in or near real-time. Details of the very efficient numerical back end, the pre- and postprocessing tools and the integrated OpenGL visualizer tool will be discussed. Moreover, several applications with marine relevance demonstrate that ELBE can be considered as prototype for next-generation CFD tools for simulation-based design (SBD) and interactive flow field monitoring on commodity hardware.

1 INTRODUCTION

Hydrodynamic free-surface flows have received an ever-increasing interest over the past few years. Applications of such flows range from naval architecture to offshore, marine, coastal and environmental engineering [14]. When attention is given to long-term simulations of large-scale problems, the related research and development efforts predominantly use experimental approaches based on scaled models and potential-flow simulations [17]. This preference is motivated by the superior geometric flexibility, the moderate computational efforts or the lack of feasible alternatives. Such methods are afflicted with scale effects or the misrepresentation of important phenomena, e.g. associated to wave breaking, rotational flow and turbulence, and are therefore often supplemented by heuristic corrections. The related uncertainty of the results is generally difficult to quantify, especially for applications beyond the range of experiences. More advanced simulation approaches are associated with prohibitive computational efforts and involve solving the full Navier-Stokes equations with turbulence closure. Accordingly, research is required to develop sound computational procedures for violent, long-term free-surface flow simulations at reasonable time to solution and hardware costs.

At the same time, the rapid growth of high-performance computing capabilities has triggered the idea of a seamless integration of numerical simulations into the design process. Due to the complex nature of the applications, sophisticated numerical methods and hardware concepts have to be harmonized to achieve the required response times, desirably in (or near to) real-time. In this context, the Lattice Boltzmann Method (LBM) has recently matured as a viable alternative to conventional CFD approximations. Whilst modeling essentially similar physics as classical continuum mechanics Navier-Stokes procedures (e.g. using FVM, FDM or FEM) [9], LBM features a number of performance-related advantages, particularly concerning data locality and parallel computing.



(a) A380 cabin mock-up

(b) Surface-mounted cube

(c) Ditching of a generic aircraft

(d) Numerical towing tank

Figure 1: Sample applications of ELBE

In this contribution, we present our in-house Lattice Boltzmann based solver ELBE, a versatile, efficient toolkit for the numerical simulation of complex two- and three-dimensional flow problems[1]. The model considers non-linear free surface flow behavior, effects of viscosity and turbulence. On top, the software uses a bidirectional, partitioned, explicit coupling approach for the simulation of fluid-structure interaction problems. The very efficient numerical back end allows for three-dimensional simulations of complex flows in very competitive simulation time and on local commodity hardware. We consider ELBE as a prototype for next-generation CFD tools that support for a near real-time analysis of complex flow fields and simulation-based design (SBD). The rich features and the outstanding performance of the code make it a tool that is also applicable to various kinds

[1] http://www.tuhh.de/elbe

of challenging applications in marine engineering. A brief description of the LBM and the related free surface models is given in section 2, before turning to the recent code extensions for efficient pre- and postprocessing that allow for (near) real-time simulations of complex flows. Finally, four selected applications are presented (section 4), conclusions are drawn and future perspectives are discussed (section 5).

2 NUMERICAL METHOD

While classical CFD solvers are based on the macroscopic Navier-Stokes equations, LBMs handle CFD problems on a microscopic scale. LBM's fundamental variable is the particle distribution function $f(t, \vec{x}, \vec{\xi})$, which specifies the probability to meet a fictive particle with velocity $\vec{\xi}$ at position $\vec{x}$ and time t. To obtain a model with reduced computational costs, the velocity space is discretized and discrete particle velocities $\vec{e}_i$ are introduced. In the present work, the D3Q19 model with 19 discrete microscopic particle velocities $\vec{e}_i$ and corresponding particle distribution functions $f_i(t, \vec{x})$ 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

$$f_i(t + \Delta t, \vec{x} + \Delta t\, \vec{e}_i) - f_i(t, \vec{x}) = \Omega_i \ , \tag{1}$$

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 on the microscopic scale. For the latter, the advanced multiple relaxation time (MRT) model [1] 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 [2]. The solutions of the lattice Boltzmann equation satisfy the incompressible Navier-Stokes equations up to errors of $\mathcal{O}(\Delta x^2, \mathrm{Ma}^2)$ [10]. The first two hydrodynamic moments of the particle distribution functions include the macroscopic values for the density and momentum fluctuation:

$$\rho = \sum_{i=0}^{18} f_i \qquad \text{and} \qquad \rho_0\, \vec{u} = \sum_{i=0}^{18} f_i\, \vec{e}_i \ . \tag{2}$$

A Smagorinsky large eddy model (LES) captures turbulent structures in the flow, including an additional turbulent viscosity ν_{T} for effects of unresolved sub-grid eddies [12]. 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 [5]. Since sub-grid wall distances are not taken into account in this model, the scheme is only second-order accurate for boundaries which are exactly located in the middle of two lattice nodes. 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 in every time step [4].

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 LBM 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 and 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. In contrast to higher-order schemes [6], the normal vector information is not considered here.

3 IMPLEMENTATION

For the implementation and parallelization, the NVIDIA CUDA toolkit is used, as LBM methods are especially well suited for implementation in a General Purpose Graphical Processing Unit (GPU) context [6]. The latter provides a large amount of compute cores (i.e. up to 4,992 on a recent NVIDIA Kepler K80 board) and sufficiently enough device memory (24 GB, K80) for typical engineering applications. 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 very competitive time.

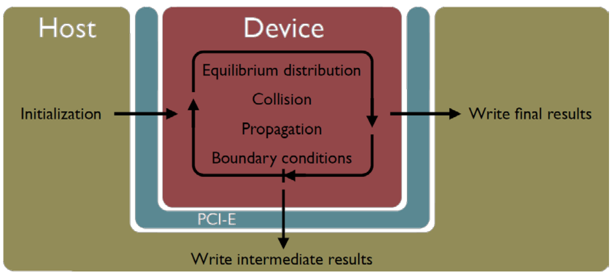


Figure 2: Typical bottlenecks of GPU-accelerated simulations: data transfer from host to device and vice versa during initialization and postprocessing

However, many GPU-accelerated solvers combine very efficient and optimized techniques with conventional pre- and postprocessing routines for the initialization, the transient grid update and the data storage operations (Figure 2). 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, so that innovative grid generation and postprocessing techniques had to be developed.

The *ePiP* fast surface voxelization technique maps arbitrary, tessellated triangular surface meshes to the LBM grids. The algorithm is optimized for massively parallel execution and 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. Further details are discussed in [7].

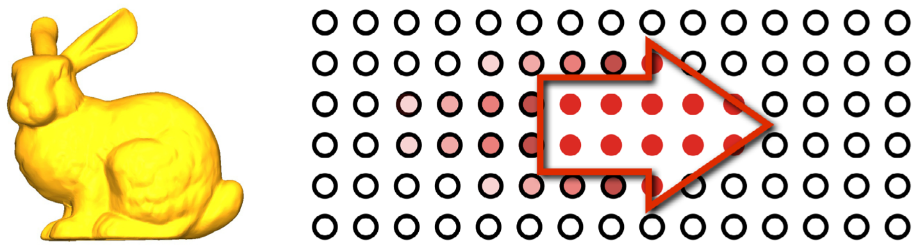


Figure 3: Grid generation in the context of LBM: mapping of discrete triangle mesh representations of geometries (left) to the Cartesian grid (right, red nodes).

Additionally, an integrated OpenGL-based visualizer tool allows to visualize the simulation results during runtime, based on innovative, shared CUDA-OpenGL memory spaces. The elbeVIS integrated visualization tool is designed to help software developers, scientists, and engineers to asses and interact with their executed 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 is stored on device memory so it can be used to compute the next time step's data set 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. The portable operating system interface for unix (POSIX) threads, or short *pthreads*, are used to realize the multithreaded design. Figure 4 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 case of, e.g., wave impact simulations, the surface pressure fields can be mapped to the surface of the geometry under consideration.

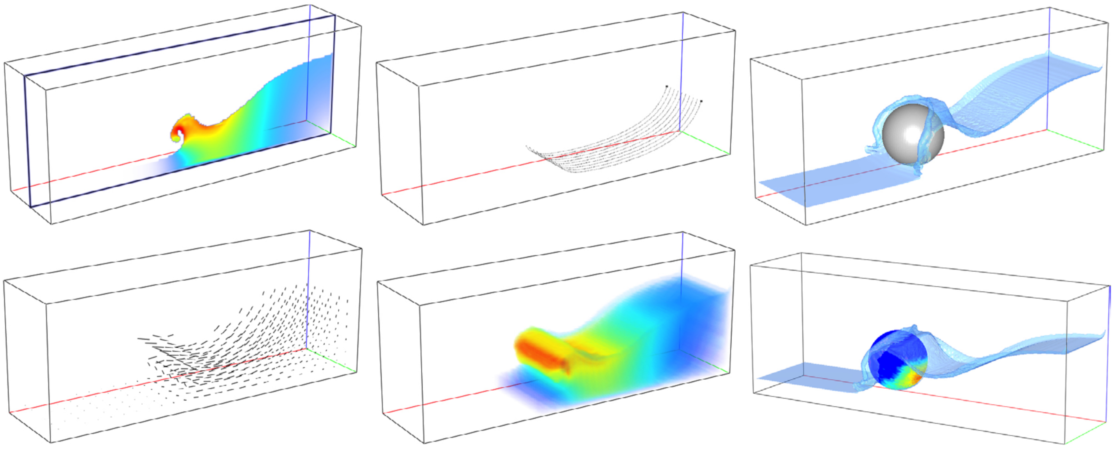


Figure 4: elbeVIS in action

On top, a graphical user interface (GUI) has been designed, in order 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. 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.

(a) Expansion chassis

(b) Multi-GPU compute server

Figure 5: Two complementary ELBE hardware concepts

4 APPLICATIONS

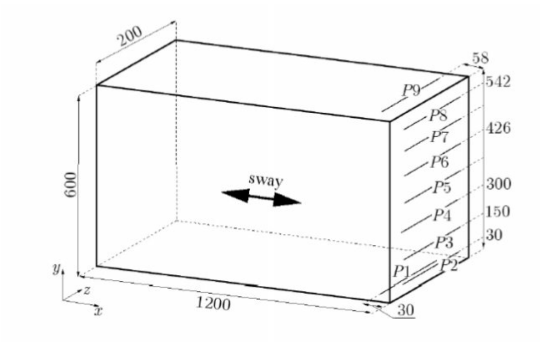
In this section, four applications of ELBE are presented. Emphasis is given to marine and coastal engineering free-surface flows. Alternative applications and validation examples, e.g. in the field of civil engineering, were recently published elsewhere [8].

4.1 Tank sloshing

First, horizontal sloshing in a rectangular tank is investigated [18]. A two-dimensional D2Q9 model is applied (Figure 6) and the sloshing mode is excited with a pulsating horizontal body force component, while the actual tank geometry remains fixed.

Parameter	Value
Domain	1.2m x 0.6m
Lattice	512 x 256
Δx	0.00235m
Δt	0.0001s
Viscosity ν	$10^{-6}\ m^2/s$
Amplitude	0.0605m
Period T	1.74s
Fill level	20 %

(a) Parameters



(b) Experimental setup (taken from [18])

Figure 6: Tank sloshing test case: simulation parameters and experimental setup.

The impact pressures are tracked at several probe locations. In Figure 7, the results for two selected probes are compared to the experimental reference data of [18] and numerical results that were obtained with OpenFOAM [3]. An excellent agreement can be observed, both for the impact points (in time) and the peak pressures. The ELBE simulation of 10 seconds of tank sloshing took only 90 seconds, on a recent K40 GPU. Comparable simulation times of conventional Navier-Stokes solvers usually accumulate up to several hours (e.g., the OpenFOAM simulation lasts 6 hours [3]).

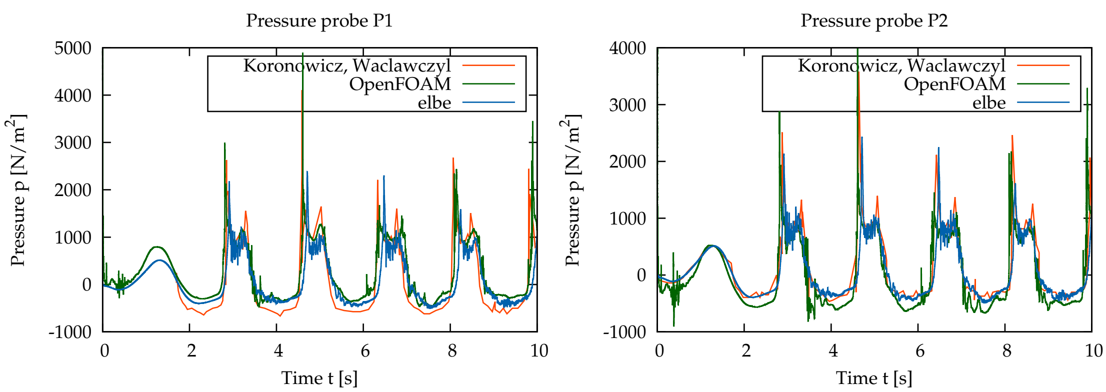


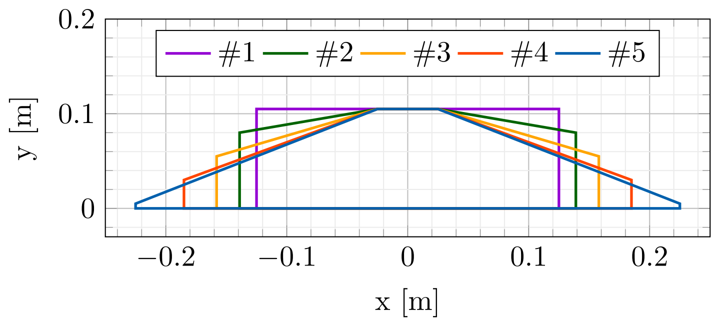
Figure 7: ELBE results in comparison to experimental pressure signals [18] and OpenFOAM [3]

4.2 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. During the structural design process of such foundations, it is desirable to know the precise hydrodynamic loads that will be caused by current and waves. In addition to simplified hydrodynamic models, that 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 [15]. Five different geometries are analyzed experimentally and numerically (Figure 8b), for scenarios with currents, waves, and combined wave-current loads. Here, we present ELBE results for two selected scenarios, to shoe the validity of the hydrodynamic approach, the force calculation techniques, and wave-current inlet/outlet boundary conditions.

Parameter	Value
Domain	3m x 0.55m
Lattice	512 x 94
Ma	0.025
Δx	5.86E-3m
Δt_{C3}	2.23E-4s
Δt_{H3}	1.7746E-4s
Viscosity ν	$1 \cdot$ E-6 m^2/s

(a) Parameters

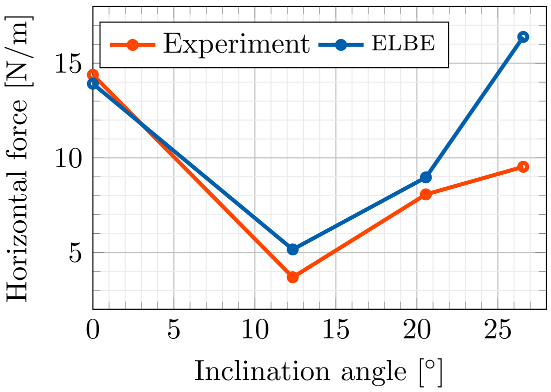


(b) Experimentally investigated geometries

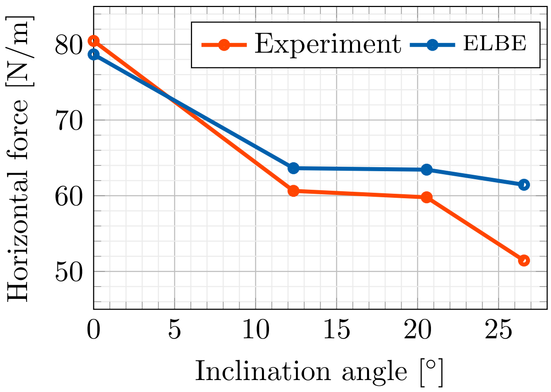
Figure 8: Flow past a generic gravity foundation model

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 of [15]) with a flow velocity of $v = 0.38$ m/s and a wave load (case H3 of [15]) with a wave height $h = 0.12$ m and a wave period of $T = 1.9$ s. The initial water height above the structure is chosen to be $h_0 \approx 0.35$ m, in accordance with the experimental data.

In Figure 9, ELBE results are plotted against the foundation inclination angle, again in comparison to the experimental data for both load scenarios. 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. Very good agreement can be seen for smaller inclination angles, while the ELBE solution differs with increasing geometry 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 worse than rectangular, axis-parallel geometries. Consequently, and as expected, best results are obtained for geometry #1, where the numerical and experimental results differ only about 3 %. Further details, including the transient force signals in comparison to the experimental data can be found in [8].



(a) Current condition C3



(b) Wave condition H3

Figure 9: Comparison of horizontal forces in experimental and numerical wave tanks for two different scenarios, as a function of inclination angle

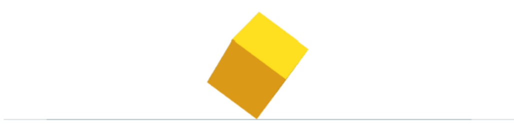
Again, the performance of the solver is outstanding. The simulations were run in less than one minute, on a Kepler K40M board, and could easily be included into the foundation design process to serve as a basis for a proper shape optimization.

4.3 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. The rigid body motion is modeled with a quaternion-based motion modeler and a physics engine. For the coupling of the explicit LBM and the motion solver, a bidirectional and

Parameter	Value
Domain	1.0m x 1.3 x 0.5m
Lattice	200 x 260 x 100
Ma	0.04
Δx	5mm
Viscosity ν	$10^{-6} m^2/s$

(a) Parameters



(b) Initial condition

Figure 10: Freely falling cube: parameters and initial configuration

explicit coupling approach is used, as further discussed in [16]. Here, a standard validation example is given, following the experimental work of [13] in which a freely falling cube is entering a flat water surface. In Figure 11, the results of ELBE, our in-house Finite-Volume code FreSCo+ and the experiment are compared, and a very good agreement can be seen. More detailed comparisons (in terms of displacements and Euler's angles) will be presented at the conference.

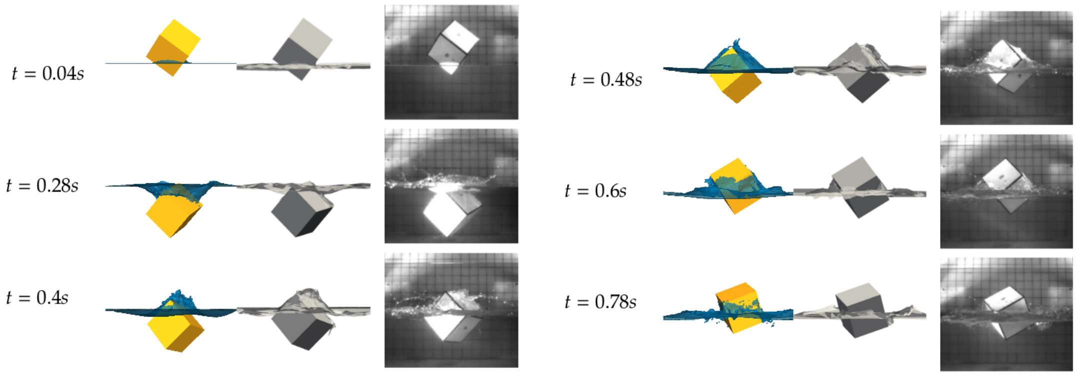


Figure 11: Freely falling cube: results for six selected time steps

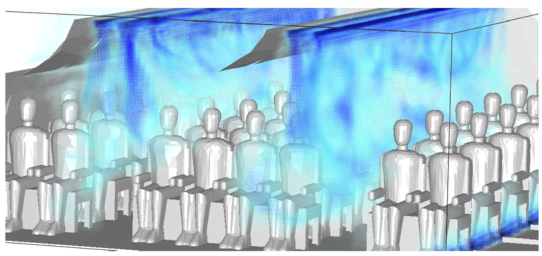
The numerical simulation of the 2 second event took approximately 5.4 hours on a 10-core CPU machine using FreSCo+, while the ELBE simulation lasted only 12 mins on a recent NVIDIA GTX Titan gaming board.

4.4 Towards real-time CFD: cabin aerodynamics

The previously presented test cases show the applicability of ELBE to complex large-scale free surface flows. The computational times are comparably short, but still higher than real-time. To demonstrate the real-time capability of the solver, a large-scale flow problem without free surface is addressed and the aerodynamics inside an A380 cabin mock-up is analyzed. Such flow simulations are of great interest in the design process of the cabin ventilation system, in order to assess the thermal comfort of the passengers during flight.

Parameter	Value
Domain	7.02m x 5.10m x 2.18m
Lattice	352 x 256 x 109
Ma	0.1
Δx	0.02m
Viscosity ν	$10^{-6} m^2/s$

(a) Parameters



(b) Velocity field

Figure 12: Cabin aerodynamics inside an Airbus A380 cabin mockup

Due to the relatively small grid resolution and the low inflow velocity, the numerical simulation of internal aerodynamics could be realized in real-time, on a ten million node grid and while watching the transient evolution of flow patterns in our integrated visualizer tool elbeVIS. The simulation of 10 minutes of real-time hence took approximately 10 minutes. The computational setup with an external GTX Titan board in a ViDock expansion chassis was used.

5 CONCLUSIONS

In this work, we presented the efficient lattice boltzmann environment ELBE, a highly efficient numerical wave tank on the basis of the Lattice Boltzmann Method. The proposed numerical methodology is able to reproduce accurate results for complex, three-dimensional test cases in a very competitive computational time. Hence, ELBE can be considered as a prototype for next-generation CFD tools that will allow for a real-time analysis of complex flow fields and simulation-based design in the near future.

Future work will address non-local grid refinement for a better resolution of near-wall effects, dynamic Smagorinsky turbulence models and wall functions for a better representation of turbulence, and a multi-GPU implementation of the code for large-scale applications.

REFERENCES

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

[2] I. Ginzburg and K. Steiner. Lattice Boltzmann model for free-surface flow and its application to filling process in casting. *J. Comput. Phys.*, 185(1):61–99, 2003.

[3] T. Grahs and C. Janßen. GPU-beschleunigte Mehrphasensimulationen für komplexe maritime Anwendungen. In *Proc. NAFEMS 2014 (submitted)*, Bamberg, May 2014.

[4] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E*, 65(4), April 2002.

[5] X. He and L.-S. Luo. Lattice Boltzmann model for the incompressible Navier-Stokes equation. *Journal of Statistical Physics*, 88:927–944, 1997.

[6] C. Janßen and M. Krafczyk. A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids. *Computers and Mathematics with Applications*, 59(7):2215–2235, 2010.

[7] C. F. Janßen, N. Koliha, and T. Rung. A fast and rigorously parallel surface voxelization technique for GPGPU-accelerated CFD simulations. *Communications in Computational Physics (accepted for publication)*, 2014.

[8] Christian F. Janßen, Silke Gralher, Dennis Mierke, Micha Überrück, and Thomas Rung. GPU-accelerated CFD for large-scale free surface flow problems in civil and environmental engineering. *Computation (submitted)*, 2015.

[9] M. Junk, A. Klar, and L.-S. Luo. Asymptotic analysis of the lattice Boltzmann equations. *J. Comput. Phys.*, 210(2):676–704, 2005.

[10] M. Junk and Z. Yang. Pressure boundary condition for the lattice Boltzmann method. *Comput. Math. Appl.*, 58(5):922–929, 2009.

[11] C. Körner, M. Thies, T. Hofmann, N. Thürey, and U. Rüde. Lattice Boltzmann Model for Free Surface Flow for Modeling Foaming. *Journal of Statistical Physics*, 121(18):179–196, October 2005.

[12] M. Krafczyk, J. Tölke, and L.-S. Luo. Large-eddy simulations with a multiple-relaxation-time LBE model. *Int. J. Mod. Phys. B*, 17:33–39, 2003.

[13] Marek Kraskowski. Validation of the ranse rigid body motion computations. In *Proc. 12th Numerical Towing Tank Symposium*, volume 6, pages 99–104, 2009.

[14] D. Le Touzé. *2nd International Conference on Violent Flows.* Publibook/Société des écrivains, 2012.

[15] D. Markus, F. Ferri, R. Wchner, P.B. Frigaard, and K.-U.Bletzinger. Complementary numerical-experimental benchmarking for shape optimization and validation of structures subjected to wave and current forces. *Computers and Fluids*, preprint, 2014.

[16] D. Mierke, C.F. Janßen, and T. Rung. GPU-accelerated large-eddy simulation of ship-ice interactions. In *Proc. 6th Int. Conf. on Comp. Methods in Marine Engng.*, pages 229–240, 2015.

[17] J.N. Newman. Trapping of water waves by moored bodies. *Journal of Engineering Mathematics*, 62:303–314, 2008.

[18] Tomasz Waclawczyk and Tadeusz Koronowiczy. Comparison of CICSAM and HRIC high-resolution schemes for interface capturing. *Journal of Theoretical and Applied Mechanics*, 46(2):325–345, 2008.