V International Conference on Computational Methods in Marine Engineering
MARINE 2013
B. Brinkmann and P. Wriggers (Eds)

GPGPU-ACCELERATED SIMULATION OF WAVE-SHIP INTERACTIONS USING LBM AND A QUATERNION-BASED MOTION MODELER

C. F. JANSSEN, H. NAGRELLI AND T. RUNG

Institute for Fluid Dynamics and Ship Theory
Hamburg University of Technology (TUHH)
Hamburg, Germany
e-mail: {christian.janssen,heinrich.nagrelli,thomas.rung}@tuhh.de
web page: http://www.tuhh.de/fds

Key words: LBM, Free Surface Flow, Fluid-Structure Interaction, GPGPU

Abstract. The paper reports on the applicability of the Lattice Boltzmann based free surface flow solver elbe to the simulation of complex, fluid-structure interaction (FSI) problems in marine engineering. General purpose graphics processing units (GPGPUs) are used to accelerate the numerical calculations. The basic methodology and the initial validation of the solver for three FSI test cases are given in this paper. A more detailed validation and the application of the solver to the numerical simulation of the ditching of a free fall boat will be presented at the *Marine 2013 Conference*.

1 INTRODUCTION

Research on strongly nonlinear fluid-ship interactions has received an ever-increasing interest over the past few years, especially in naval architecture. As experimental studies are afflicted with scale effects which are generally difficult to quantify, efficient, robust and accurate computational approaches towards fluid-structure interaction are highly appreciated, particularly when emphasis is given to violant flow phenomena. In this work, we present a numerical wave tank on the basis of the Lattice Boltzmann Method (LBM) and a quaternion-based floating-body motion modeler. The LBM has recently matured as a viable alternative to classical CFD approaches, i.e. Eulerian Finite-Volume methods or particle-based Lagrangian approaches (e.g. SPH). LBM solves a discretized Boltzmann equation that describes the evolution of particle distribution functions, on Cartesian grids. Whilst modeling essentially similar physics as Navier-Stokes procedures, LBM features a number of performance-related advantages, particularly concerning data locality and parallel computing. The paper addresses the application of a GPGPU-accelerated, VOF-based LBM method [13] for the simulation of fluid-ship interactions. Attention is confined

to efficient models for rigid floating bodies. Such floating-body motions are traditionally performed by using deforming meshes or body-algned rigid grids or a combination of the two. A different option is conceivable in conjunction with LBM, which operates on fixed, equidistant Cartesian grids. Analogue to an immersed boundary approach, a grid update (i.e. remeshing) reduces to the calculation of subgrid distances of the Eulerian lattice nodes to the surface of the structure. The floating body motions themselves are described by a unit-quaternion motion modeler [16] that is coupled to the LBM in a bidirectional, explicit manner. After a brief description of the solver basics and the fluid-structure coupling approach in section 2, three illustrative validation cases are presented in section 3: drag on a sphere and the gravity-driven free fall of a rigid sphere, with and without water entry.

2 NUMERICAL METHOD

elbe is an efficient and accurate toolkit for the numerical simulation of complex two- and three-dimensional flow problems [14]. It considers nonlinear flow behavior with and without a free surface, effects of viscosity and turbulence and is based on a Lattice Boltzmann Method (LBM) on equidistant Cartesian grids. The LBM has become an efficient approach for solving a variety of difficult CFD problems, including those in the field of multiphysics. The numerical approximation is second-order accurate in space and time.

2.1 LBM basics

In contrast to classical CFD solvers which are dealing with the macroscopic Navier-Stokes equations, the LBM regards CFD problems on a microscopic scale. The primary variable of microscopic approaches is the particle distribution function $f(t, \boldsymbol{x}, \boldsymbol{\xi})$, which specifies the probability to encounter a particle at position $\boldsymbol{x}$ at time t with velocity $\boldsymbol{\xi}$. In order to obtain a model with reduced computational costs, the velocity space is discretized and discrete velocities $\boldsymbol{e}_i$ are introduced. In this work, the D3Q19 model [23] with 19 discrete microscopic particle velocities $\boldsymbol{e}_i$ and corresponding particle distribution functions $f_i(t, \boldsymbol{x})$ is used. The evolution of these discrete distribution functions is described by the discrete Boltzmann equation. A standard finite difference discretization on an equidistant grid finally leads to the *lattice* Boltzmann equation,

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

The left-hand side of this equation is an advection-type expression,while the discrete *collision operator* Ω_i models the interactions of particles on the microscopic scale. Collision operators Ω_i of different complexity can be used. In a single relaxation time (SRT) model [1], the particle distribution functions are driven to an equilibrium state with a single relaxation rate (which relates to the kinematic viscosity of the fluid ν). In the more advanced MRT model [3] used in the present work, the particle distribution functions are transformed into moment space, where they are relaxed with several different relaxation

rates. This increases the stability and at the same time enables the development of more accurate boundary conditions [5]. The solutions of the lattice Boltzmann equation satisfy the incompressible Navier-Stokes equations up to errors of $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\mathrm{Ma}^2)$ [15]. The macroscopic values for density fluctuation ρ and momentum $\rho_0 \boldsymbol{u}$ are the first two hydrodynamic moments of the particle distribution functions:

$$\rho = \sum_{i=0}^{18} f_i \quad \text{and} \quad \rho_0 \boldsymbol{u} = \sum_{i=0}^{18} \boldsymbol{e}_i f_i \tag{2}$$

As free surface flows usually occur at very high Reynolds numbers in the turbulent regime, a Smagorinsky large eddy model (LES) [18] is used to capture turbulent structures in the flow. The effect of the small sub-grid eddies on the large-scale flow structures is modeled through an additional turbulent viscosity ν_T, which - in a Smagorinsky model - depends on the local strain rate, $\nu_\mathrm{T} = (C_S \Delta x)^2 \|\mathbf{S}\|$, with Smagorinsky constant C_S and strain rate tensor $S_{\alpha\beta}$. For further details, see [18].

At the domain boundaries, the incoming particle distribution functions are missing after the advection step and are reconstructed with the help of boundary conditions. For no-slip and velocity boundaries, a simple bounce back scheme is used. The missing particle distribution function f_I is reconstructed as

$$f_I^{t+1}(\boldsymbol{x}) = f_i^t(\boldsymbol{x}) + 2\rho_0 w_i \frac{\boldsymbol{e}_i \bar{\boldsymbol{u}}}{c_s^2} \tag{3}$$

where i is the inverse direction to I, $\bar{\boldsymbol{u}}$ denotes the prescribed boundary velocity [2] and w_i are model-specific, constant weighting factors [11]. The subgrid wall distance is not taken into account in this model, so that 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 [17] balances the fluid pressure and the surrounding athmospheric pressure p_B:

$$f_I^{t+1} = -f_i^t + f_I^{eq}(\rho_B, \boldsymbol{u}_B) + f_i^{eq}(\rho_B, \boldsymbol{u}_B) \tag{4}$$

where $f_{i,I}^{eq}(\rho_B, \boldsymbol{u}(t_B, \boldsymbol{x}_B))$ are Maxwellian equilibrium distribution functions and ρ_B is related to the surrounding pressure by $\rho_B = p_B c_s^{-2}$. Gravity and other volume forces $\boldsymbol{F}$ are added directly to the distribution functions f_i in every time step [8]:

$$\Delta f_i = 3\omega_i \rho \boldsymbol{e}_i \cdot \boldsymbol{F}. \tag{5}$$

2.2 Free surface model

Free-surface flows are two-phase flows with high viscosity ratios and high density ratios between two immiscible phases. The flow is dominated by the denser phase and the interface is allowed to move freely. If capillary forces are neglected, the simulation of the denser and more viscous phase is sufficient and the influence of the second (less dense) phase on the flow dynamics can be represented by appropriate 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 couple of approaches have been developed to use the LBM for free surface flow simulations [6, 7, 12, 20]. Apart from the aforementioned methods, Koerner and Thuerey [17, 25] combine LBM with a VOF method and a flux-based advection scheme. Their algorithm initially was developed for the simulation of metal foams, but is capable of handling free-surface flow simulations as well, and was found to be robust and stable while keeping the interface sharp. Opposite to common VOF methods, the flux terms are expressed directly in terms of LBM distribution functions. In a VOF interface capturing approach, the interface is captured via the *fill level* of a cell, which qualifies the amount of a cell which is filled with fluid:

$$\varepsilon = \frac{V_{fluid}}{V_{cell}} \tag{6}$$

A fill level of 0.0 marks an empty cell in the inactive gas domain, a fill level of 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 between 0.0 and 1.0. The fill level changes in time, and the new fill level of a cell at time step $n+1$ is calculated via balancing the mass fluxes between the neighboring cells and updating the fill level via

$$\varepsilon^{n+1} = \frac{m^{n+1}}{\rho^{n+1}} = \frac{\rho^n \varepsilon^n + \sum_i \Delta m_i}{\rho^{n+1}}, \tag{7}$$

where $\rho^{n/n+1}$ is the fluid density at time step n resp. $n+1$ according to Eq. (2) and ε^n is the fill level at time step n [17, 25]. The mass flux terms m_i between neighboring cells are expressed in terms of particle distribution functions:

$$\Delta m_i = \left[f_I\left(\boldsymbol{x}, t\right) - f_i\left(\boldsymbol{x}, t\right) \right] \cdot A_i \cdot \Delta t \tag{8}$$

with the two antiparallel particle distribution functions $f_{i,I}$ entering or leaving the corresponding cell. A_i denotes the wet area between two cells and is calculated on the basis of a simplified surface reconstruction. It can be estimated e.g. as the arithmetic mean of the fill levels of two neighboring cells. Opposite to higher-order schemes (such as presented in [12]), the normal vector information is not considered here. To sum up, the Lattice Boltzmann VOF advection scheme can be considered as a specialized, geometry-based VOF method on the basis of a mesoscopic advection model.

2.3 Fluid-structure interaction

Floating-body motions follow from a motion modeler which converts the external and hydrodynamic forces exerted on the body into its motion. The latter involves appropriate descriptions of the spatial position and angular orientation. The most common way to describe the angular orientation of a rigid body is the use of Euler angles. They represent three composed elementary rotations of a body-fixed local coordinate system referring to a global system. In such an arrangement the effect of a Gimbal Lock can occur, when two axes are driven into a parallel configuration, which "locks" the system into a rotation in a degenerate two-dimensional space. To avoid this singularity and ensure that each motion is uniquely defined, unit-quaternions - also known as Euler parameters - are employed for the motion modeler [22].

For the coupling to the explicit LB method, a bidirectional, explicit coupling approach is used, which has been proposed and validated in [4]. First, the force acting on the rigid bodies is evaluated by means of the momentum exchange method [19]. The force $\boldsymbol{F}$ acting on an obstacle in the flow results from the momentum of the particles hitting the boundary. It can be computed by balancing the particle momentum before and after hitting the boundary:

$$\boldsymbol{F} = \sum_{i \in \Gamma} \boldsymbol{F}_i = -\frac{V}{\Delta t} \boldsymbol{e}_i \left(f_i(t + \Delta t, \boldsymbol{x}) + f_I(t, \boldsymbol{x}) \right) \tag{9}$$

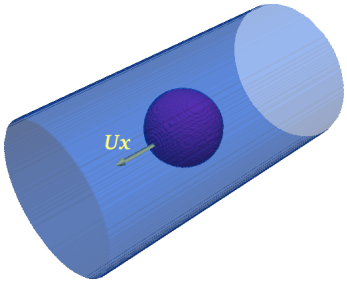
for all links i that are cut by the obstacle [21]. Since the rigid bodies do not allow elastic or plastic deformations, the evaluation of the integral force on the whole rigid body is sufficient. After the calculation of fluid loads, the force information is transferred to the structural solver and one time step of rigid body motion is computed. The resulting displacements and velocities are passed to the fluid solver, where the geometry is updated and the modified bounce back scheme (Eq. (3)) serves to incorporate the rigid body boundary velocity.

3 RESULTS

In the following, the numerical results for three validation cases are given. Since LBM usually operates on a finite difference grid, is explicit in time and requires only next neighbor interaction, it is very efficient in combination with GPGPUs and massively parallel hardware. In recent years, several authors accelerated their (LBM) computations on general-purpose graphics hardware, see e.g. the pioneering work of [28, 30, 29]. Later on, Toelke implemented two- and three-dimensional LB bulk flow models on nVIDIA GPGPUs [26, 27] and showed an efficiency gain of up to two orders of magnitude compared to a single-core CPU code. With the development of software development kits (SDKs), in combination with the recent hardware improvements, even complex flow simulations including free surfaces and fluid-structure interaction can be efficiently addressed. In this work, the GPGPU-accelerated elbe code is used [14]. The solver is based on the NVIDIA CUDA framework and basically follows the implementation strategies discussed in [12].

3.1 Drag on a sphere

The first test case is concerned with the drag force on a sphere moving with constant velocity in a cylindrical pipe. Fig. 1 shows the computational domain and the main simulation parameters. The simulation is carried out in the local sphere frame of reference: the sphere is fixed, and velocity boundary conditions are used at the inlet, outlet and the cylinder surface. Fig. 2 shows the converged velocity field in x direction.



Pipe length L_x $[m]$	16.0
Pipe diameter D $[m]$	4.0
Sphere diameter d $[m]$	2.0
Ratio $\lambda = \frac{d}{D}$ $[-]$	0.5
Initial sphere COG $[m]$	[16.0 4.0 4.0]
Re numbers $[-]$	10, 25, 50
Ma numbers $[-]$	0.02, 0.04, 0.08, 0.1

Figure 1: Sphere drag setup & calculation parameters

The force on the sphere is evaluated and validated in terms of the dimensionless drag coefficient C_d,

$$C_d = \frac{8\ F_x}{\rho u^2 \cdot d^2 \pi}, \tag{10}$$

for a sphere with diameter d moving with velocity u in a fluid with density ρ. To approximate C_d analytically, the extended Stokes solution by Schiller and Naumann [24] is used. The equation approximates the C_d coefficient for spheres in an infinite fluid with constant velocity. The influence of pipe walls surrounding the sphere is considered with an additional correction term [9], leading to the following final expression for the drag:

$$C_{d,\bigcirc} = \frac{24}{Re}\{1 + 0.15 \cdot Re^{0.687} + \frac{1 - 0.75857\lambda^5}{12.1050\lambda + 2.0865\lambda^3 1.7068\lambda^5 + 0.72603\lambda^6}\}, \tag{11}$$

depending on the ratio $\lambda = d/D$ of the diameters of the sphere and the pipe. This approximation leads to an accuracy of 95% for $\lambda < 0.6$ and $\mathrm{Re} < 50$. In Fig. 3, the results of a grid convergence study for the $\mathrm{Re} = 10$ case are shown for selected Mach numbers. It can be observed that the numerical results converge with increasing grid resolution, but do not converge to the analytical reference value for C_d based on Eq. (11). First studies suggest that this discrepancy depends on (i) the exact location of the sphere in the grid (due to the first-order simple bounce back rule), and (ii) the channel length (due to the disturbance of the flow pattern by the inlet and outlet boundary conditions). Both effects will be studied more carefully in future work. However, the numerical errors are in the same order of magnitude as the predicted accuracy of Eq. (11).

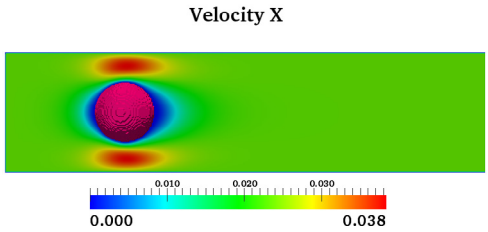


Figure 2: $v_{x,LB}$ for $Re = 10$

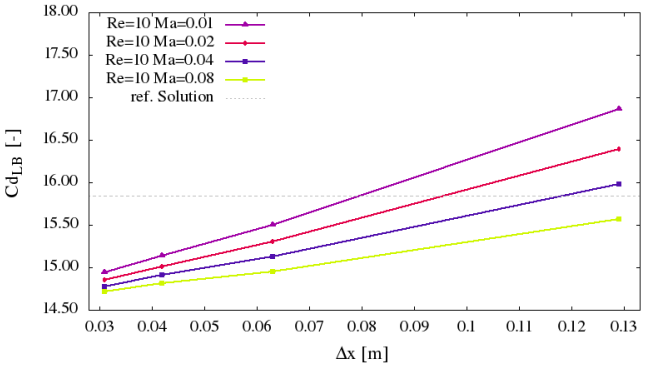
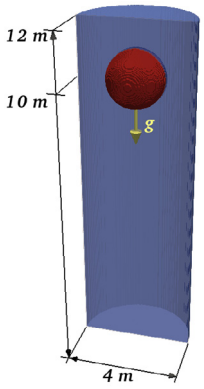


Figure 3: C_d grid convergence study for $Re = 10$

3.2 Falling sphere

For the validation of the explicit coupling of LBM and quaternion-based motion modeler, the gravity-driven sphere motion in a fluid at rest is examined, see Fig. 4. The simulation is started from a state of rest. During the initial stages of the simulation, the sphere is accelerated by the gravitational force. With increasing sphere velocity, the drag force increases, too, so that finally a terminal velocity v_z is reached.



Pipe height $[m]$	12.0
Initial sphere height $[m]$	10.0
Sphere diameter d $[m]$	2.0
Ratio $\lambda = \frac{d}{D}$ $[-]$	0.5
Re numbers $[-]$	10, 25, 50
Ma numbers $[-]$	0.02, 0.04, 0.08, 0.1

Figure 4: Rigid sphere in a fluid at rest - setup and simulation parameters

In our simulation, the ratio of fluid and sphere density is set to a constant value of $\rho_{Sph}/\rho_{Fluid} = 1.5$ and gravity is set to $g = 9.81[m/s^2]$. The fluid viscosity is artificially adjusted so that the final terminal velocity matches the given Reynolds number Re. For this purpose, the force balance on the sphere is considered. The force in z direction consists of two parts: the hydrostatic fluid pressure that causes buoyancy forces, and the viscous drag, which balances the effective sphere weight, $(\rho_{Sph} - \rho_{Fluid})$, so that

$$(\rho_{Sph} - \rho_{Fluid}) \cdot V_{Sph} \cdot g + \frac{1}{2}\rho_{Fluid} v_Z^2 \cdot C_{d,\bigcirc} \cdot A_{Sph} = 0 \tag{12}$$

holds, with sphere cross section A_{Sph} and sphere volume V_{Sph}. For given values of Re and λ, the drag coefficient $C_{d,\bigcirc}$ is obtained by evaluating Eq. (11)). The terminal velocity can be calculated by rearranging Eq. (12),

$$v_z = \sqrt{\frac{(\rho_{Sph} - \rho_{Fluid}) g \frac{4}{3} d}{\rho_{Fluid} \cdot C_{d,\bigcirc}}}, \tag{13}$$

and the corresponding (artificial) fluid viscosity ν yields $\nu = v_z d/\mathrm{Re}$. Fig. 5 shows the velocity field for selected time steps for the Re = 10 case.

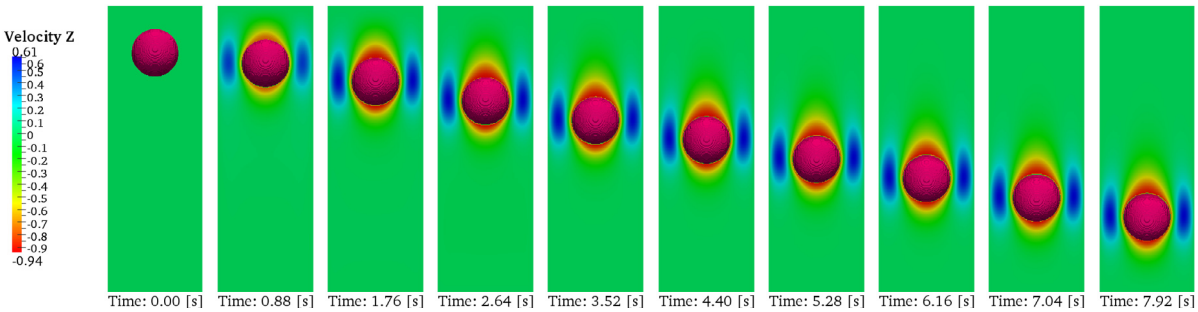
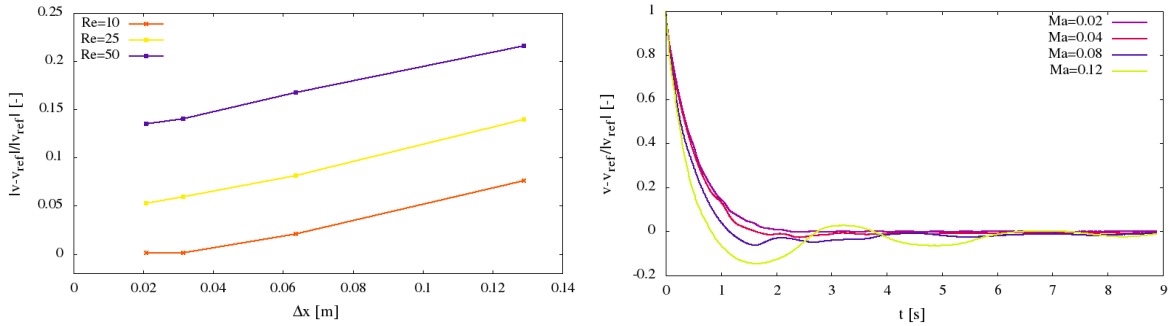


Figure 5: Falling sphere in a pipe, geometry and flow velocity in z direction$[\frac{m}{s}]$ for selected time steps (Ma = 0.04, Re = 10)

In Fig. 6a, the relative error of the terminal sphere velocity v_z is plotted against the grid resolution Δx for three selected Reynolds numbers. Convergence can clearly be observed. The numerical error increases with increasing Reynolds number, due to the limited accuracy of the C_d approximation (Eq. (11)). In Fig. 6b, the signed relative error in v_z is depicted for selected Mach numbers. For lower Mach numbers (corresponding to smaller time steps), less oscillations and a faster convergence is found. Note that the resulting terminal velocity is predicted very well, also for Re = 10. This supports the previous attempts to explain the error in C_d: here, the sphere moves (balancing the grid mapping errors), and no-slip boundary conditions are used at the top and bottom.



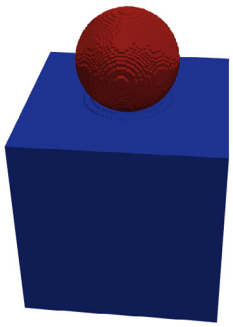
(a) Relative error of v_z as a function of lattice spacing Δx (Re = 10, 25, 50, Ma = 0.04)

(b) Relative error of v_z over time (Ma = 0.02, 0.04, 0.08, 0.12, Re = 10)

Figure 6: Results for the falling sphere test case

3.3 Sphere drop

The third and last test case deals with the water entry of a rigid sphere. As the sphere penetrates the water surface, its movement is damped due to buoyancy effects and additional contributions of the dynamic pressure gradient, i.e. drag. The sphere motion is damped, and the position of the sphere finally converges to the hydrostatic equilibrium.



Sphere diameter $[m]$	2.0
Initial water height $[m]$	4.0
Initial $COG[m]$	2.0, 2.0, 6.0
Domain size $[m]$	4.0, 4.0, 8.0
Density $\rho_{Sph}[\frac{kg}{m^3}]$	100, 200, 300, 400, 500, 600
Draft $T_{calc}[m]$	0.392, 0.574, 0.727, 0.866, 1.000, 1.134
Final $COG_{Zcalc}[m]$	4.634, 4.478, 4.352, 4.239, 4.131, 4.023

Figure 7: Water entry of the sphere at $0.5s$ & calculation parameters

The final sphere position depends on the ratio of sphere density and fluid density (ρ_{Sph}/ρ_{Fluid}) and is estimated with the principle of buoyancy, which requires the balance of buoyant force and sphere weight,

$$\rho_{Fluid} V_C = \rho_{Sph} V_{Sph} \tag{14}$$

where V_C denotes the volume of the submerged spherical calotte of height T,

$$V_C = \frac{T^2}{3}(3r - T). \tag{15}$$

Assuming $d = 1$, the draft T can be calculated by solving

$$0 = T^3 - 1.5T^2 + 0.5\frac{\rho_{Sph}}{\rho_{Fluid}} \quad with \quad T \leq 1.0 \quad \frac{\rho_{Sph}}{\rho_{Fluid}} \leq 1.0 \tag{16}$$

for T. For other sphere diameters d, the draft T can be scaled linearly. In our simulations, the fluid density is set to $\rho_{Fluid} = 1000\frac{kg}{m^3}$ and spheres of diameter $d = 2m$ and densities from 100 to 600 $[\frac{kg}{m^3}]$ are examined. The corresponding final positions of the sphere center of gravity (COG) are summarized in Fig. 7. In Fig. 8, the transient sphere trajectories are depicted and the convergence to the static swimming position can be seen. As expected, the higher the sphere density, the longer the oscillations last.

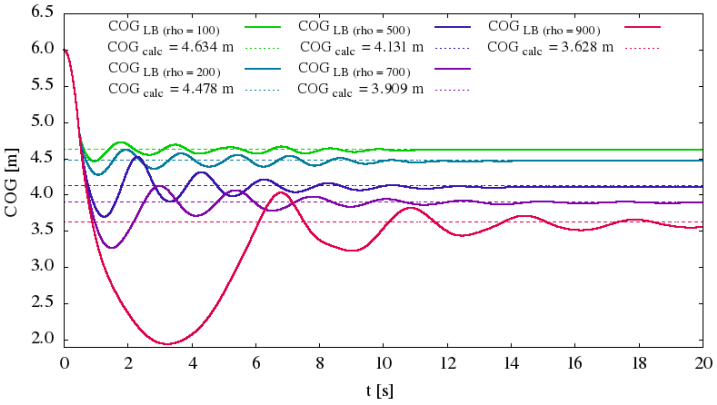


Figure 8: Sphere trajectories for different densities

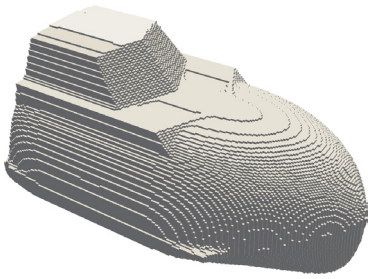


Figure 9: Voxel geometry of the Hatecke GFF(-T) 5.5 free fall boat

4 APPLICATION: FREEFALL LIFEBOAT

The validated code finally will be applied to the free fall and ditching of the freefall lifeboat Hatecke GFF(-T) 5.5M. A voxelized representation of geometry is depicted in Fig. 9. The numerical results and a comparison to reference data [10] will be presented at the conference.

5 CONCLUSIONS

In this paper, the validation of the Lattice Boltzmann based solver elbe for free surface flow problems involving nonlinear interactions of fluid and structure was presented. The model uses a VOF interface capturing approach and is coupled to a quaternion-based motion modeler in a bidirectional, explicit manner.

Three validation test cases with rather simple geometries were addressed. The results demonstrate that the proposed numerical methodology is generally able to produce accurate results for three-dimensional FSI applications, such as the water entry of a rigid sphere. Due to the straightforward grid generation, elbe is also applicable to more complex, three-dimensional geometries, as present in large-scale applications in marine engineering. The numerical results for the free fall and subsequent ditching of a free-fall rescue boat will be presented at the conference as an application example.

The authors gratefully acknowledge support for this research from the nVIDIA Academic Partnership Program (APP).

REFERENCES

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

[2] M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a lattice-Boltzmann fluid with boundaries. Physics of Fluids, 13:3452–3459, 2001.

[3] 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.

[4] S. Geller. Ein explizites Modell für die Fluid-Struktur-Interaktion basierend auf LBM und p-FEM. PhD thesis, Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften der Technischen Universität Carolo-Wilhelmina zu Braunschweig, 2010.

[5] I. Ginzburg and D. d'Humieres. Multireflection boundary conditions for lattice Boltzmann models. Physical Review E, 68(6):066614.1–066614.30, December 2003.

[6] 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.

[7] A. K. Gunstensen, D. H. Rothman, S. Zaleski, and G. Zanetti. Lattice Boltzmann model of immiscible fluids. Phys. Rev. A, 43(8):4320–4327, Apr 1991.

[8] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. Physical Review E, 65(4):046308.1 – 046308.6, April 2002.

[9] W. L. Haberman and R. M. Sayre. Motion of rigid and fluid spheres in stationary and moving liquids inside cylindrical tubes. Technical Report 1143, Dept. of the Navy, David Taylor Model Basin, Hydromechanics Laboratory, October 1958.

[10] J. Hatecke. Design Study Freefall Lifeboat GFF(-T) 5.5M. Technical Report Report No. 2006-20, Sachverständigenbüro Dipl.-Ing. Jan Hatecke, 2007.

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

[12] 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.

[13] C. Janßen and M. Krafczyk. Free surface flow simulations on GPGPUs using LBM. Computers and Mathematics with Applications, 61(12):3549–3563, June 2011.

[14] C. F. Janßen. elbe - efficient lattice boltzmann environment. http://www.tuhh.de/fds/research/current-projects/elbe.html.

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

[16] N. Koliha. Development of a quaternion-based 6-DOF model for the simulation of floating bodies. Bachelor thesis, Schriftenreihe Schiffbau, TU Hamburg-Harburg, 2011.

[17] 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.

[18] 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.

[19] A.J.C. Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. Journal of Fluid Mechanics, 271:285–309, 1994.

[20] P. Lallemand, L.-S. Luo, and Y. Peng. A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two dimensions. Journal of Computational Physics, 226(2):1367–1384, 2007.

[21] N.Q. Nguyen and A.J.C. Ladd. Sedimentation of hard-sphere suspensions at low Reynolds number. J. Fluid Mech, 535:73 – 104, 2004.

[22] P.E. Nikravesh. Computer-Aided Analysis of Mechanical Systems. Prentice-Hall, 1988.

[23] Y. H. Quian, D. d'Humieres, and P. Lallemand. Lattice BGK models for Navier Stokes equations. Europhysics Letters, 17:479–484, 1992.

[24] L. Schiller and A.Z. Naumann. Ueber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung. Zeitschrift des Vereines Deutscher Ingenieure, 77(12):318–320, 1933.

[25] N. Thürey and U. Rüde. Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids. Computing and Visualization in Science, page ., 2008.

[26] J. Tölke and M. Krafczyk. Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA. Computing and Visualization in Science, 1:29–39, 2008.

[27] J. Tölke and M. Krafczyk. TeraFLOP computing on a desktop PC with GPUs for 3D CFD. International Journal of Computational Fluid Dynamics, 22:443–456, 2008.

[28] X. Wei, Y. Zhao, Z. Fan, W. Li, F. Qiu, S. Yoakum-Stover, and A. Kaufman. Lattice-based flow field modeling. IEEE Transactions on Visualization and Computer Graphics, 10(6):719–729, 2004.

[29] Y. Zhao. Melting and flowing in multiphase environments. Computers & Graphics, 30(4):519–528, 2006.

[30] H. Zhu, X. Liu, Lui Y., and E. Wu. Simulation of miscible binary mixtures based on lattice Boltzmann method. Computer Animation and Virtual Worlds, 17:403–410, 2006.