Simulation of liquid-gas-solid flows with the Lattice Boltzmann Method

Simon Bogner, Stefan Donath, Christian Feichtinger, Ulrich Ruede

Abstract: The lattice Boltzmann method (LBM) emerged from the field of computational fluid dynamics as an alternative to Navier-Stokes based approaches. It has since been proven to be applicable also for the simulation of free surface flows (liquid-gas flow) and particulate flows (liquid-solid flow). In the past both algorithmss for free surface flows and particulate flows have been realized that preserve the highly local nature of the LBM, thus allowing parallelization with purely local communication techniques, which is crucial for large-scale computations. Here we present a method that further extends the LBM in order to simulate particulate free surface flows (liquid-gas-solid flows). The method has been integrated into the waLBerla LBM software framework with the goal of achieving a maximal parallel efficiency. As with liquid-gas and liquid-solid lattice Boltzmann, the liquid-gas-solid method has demonstrated its parallel scalability on up to 1536 cores.

Keywords: Free surface; multiphase; complex geometry; parallel; large-scale; lattice Boltzmann; particles; particulate flows; moving objects; moving obstacles; moving boundaries.

1 Introduction

In this paper we present a model for the simulation of liquid-gas-solid flows, i.e., the simulation of particles (rigid bodies) floating in a free surface flow. The proposed model is based on the lattice Boltzmann method which in the past has already been successfully extended to simulate free surface flows (liquid-gas flows) as well as particulate flows (liquid-solid flows). In the following part of the introduction we outline the liquid-gas and the liquid-solid LBM approaches, which our liquid-gas-solid model is based on.

The method as a whole has first been published in [6], there without adressing the aspects of parallel computation. Here we use the same notation, denoting the discrete LBM particle distribution function (PDF) for direction $i \in [1..n]$ by f_i with according lattice velocity vector $\mathbf{v_i}$, and lattice weight w_i. For a given lattice direction i its opposite direction $\bar{i}$ is the one with $\mathbf{v_{\bar{i}}} = -\mathbf{v_i}$.

1.1 Free Surface LBM

A two-phase flow of immiscible fluids, such as a liquid and a gas is called a *free surface* flow. In such a flow the two fluid phases are separated by a two-dimensional boundary layer. In our cell-based lattice Boltzmann approach, this boundary layer is simulated by means of *interface* cells, which form the boundary of the liquid phase towards the gas phase. A more detailed description of the method can be found in [1] (for two-dimensional flows) and in [2] (for three dimensions). This model assumes the dynamics of the gas phase to be of negligible influence on the flow, and therefore only the dynamics of the liquid are simulated. I.e., the gas phase affects the only in terms of gas pressure exerted on the free surface, which is handled by a special boundary condition imposed at the interface cells. In addition to that the effect of surface tension is taken into account as a local modification of the gas pressure.

As the interface layer is not a fixed boundary, it has to be updated dynamically with each lattice Boltzmann step. To allow free movement of the interface layer, the cell volume that is filled with liquid is encoded in an additional local variable φ (the *fill level*). In this *volume of fluid* - approach, if the fill level rises above 1 or drops below 0, then the given interface cell has to be converted into a liquid or gas cell, respectively. The conversion of an interface cell may cause further conversions of neighboring gas or liquid cells into interface cells, as shown

in Fig. 1. In this way it is assured that the lattice always has a valid configuration where all liquid regions are enclosed completely by interface cells (or other boundary cells).

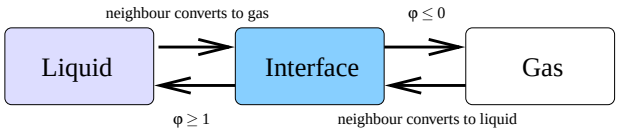

Figure 1: Possible state changes at the interface layer.

1.2 Particulate Flow LBM

For the incorporation of floating objects, the first crucial step is the modeling of moving boundaries. The particles are thus mapped into the lattice, by marking the occupied cells as *obstacle* cells, and applying appropriate boundary conditions around them.

If the local obstacle velocity $\mathbf{u_0}$ is given, then according to [5] the no-slip boundary condition for static walls is modified to

$$f_i(\mathbf{x}, t+\delta_t) = f_{\bar{i}}(\mathbf{x}, t) + \frac{6}{c^2} w_i \rho(\mathbf{x}) \mathbf{v_i} \mathbf{u_0},$$

where ρ is the local fluid density, and $\mathbf{u_0}$ is the obstacle velocity and c is a constant depending on the particular lattice Boltzmann model.

These boundary conditions will induce fluid flow according to obstacle movement. Conversely, fluid pressure and momentum impose stresses at the obstacle surface resulting in movement of the object. This force is computed according to

$$\mathbf{F} = \sum_{\mathbf{x} \in N} \sum_{i=1}^{n} \mathbf{v_i} \left(f_{\bar{i}}(\mathbf{x}, t+\delta_t) + f_i(\mathbf{x}, t) \right) \cdot w_o(\mathbf{x} + \delta_t \mathbf{v_i}),$$

where N denotes the set of all fluid cells in the neighborhood of a given obstacle cell, $w_o(\mathbf{x})$ is an indicator function, that is 1 if $\mathbf{x}$ is an obstacle cell, and 0 otherwise. This way of calculating the forces on an object from the reflected PDFs is sometimes called the *momentum exchange method* [4].

The forces calculated with the momentum exchange method are used to compute the movement of the objects. In order to correctly apply boundary conditions, the lattice has to be updated dynamically, since the object positions and therefore the set of obstacle cells may change during the simulation. Fig. 2 shows the mapping of a spherical objects at different positions.

1.3 Large Scale Computations

Both the free surface flow method from Sec. 1.1 and the particulate flow model from Sec. 1.2 are integrated in the waLBerla LBM software framework — a lattice Boltzmann framework that provides the necessary tools for communication and data management for massively parallel computations [9]. In [7] large-scale particulate flows have been realized using waLBerla together with the a rigid body physics engine, that handles the particle dynamics. Also, massively parallel free surface flow computations are possible with waLBerla [3]. In the latter case feasibility of large-scale simulations is improved mainly by resorting to local operations wherever possible and avoiding global communication in spite of non-local information like volume and pressure of the gas phase.

2 Liquid-Gas-Solid LBM

2.1 Method Outline

The free surface LBM, Sec. 1.1, relies on a closed boundary around the liquid. This boundary is realized by a closed layer of interface cells towards the gas phase that is dynamically updated according to fluid flow. Similarly, moving obstacles as described in Sec. 1.2 represent an additional type of dynamic boundary for the LBM. With obstacles moving freely within the simulation domain of a free surface flow the correct handling of cell conversions becomes a critical task which is illustrated in Fig. 3. It is clear that for a moving particle that changes its

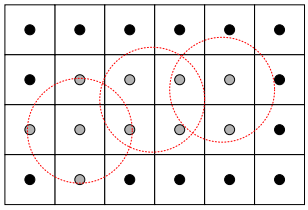

Figure 2: A spherical particle covering different cells according to its position.

position prior fluid cells may have to be converted into obstacle cells, because they are now covered by the object. Conversions of this type are considered as uncritical, because they cannot lead to invalid lattice configurations, i.e., configurations where the interface layer between gas and liquid is no longer closed. The inverse transitions (red arrows in Fig. 3) from obstacle to a fluid cell type however are critical: If a prior obstacle cell is no longer occupied by the obstacle, it has to be converted into either liquid, interface or gas cell in a way that ensures a closed interface layer in between liquid and gas phase. For a prior obstacle cell $\mathbf{x}$ to be converted into a fluid state, we define the neighborhood of non-obstacle cells $N := \{\mathbf{x} + \delta_t \mathbf{v_i} \mid i \in [1..n], \mathbf{x} + \delta_t \mathbf{v_i} \text{ is no obstacle}\}$.

- If N contains only liquid cells, then $\mathbf{x}$ is converted into liquid.
- If N contains only gas cells, then $\mathbf{x}$ is converted into gas.
- If N contains more then one cell type and $n_i \geq 0$ interface cells:
  - If N contains liquid but no gas, then $\mathbf{x}$ is converted to liquid, as the cell can be assumed to lie "under" the free surface.
  - If N contains gas but no liquid, then $\mathbf{x}$ is converted to gas.
  - If N contains liquid cells as well as gas cells, then $\mathbf{x}$ is set to interface.

After the correct cell type has been determined, the cell has to be initialized with correct values: In the case of a liquid or interface cell, it is initialized with an equilibrium set of PDFs, parameterized with the obstacle velocity. Density is interpolated from the neighborhood N. For further details, see [6].

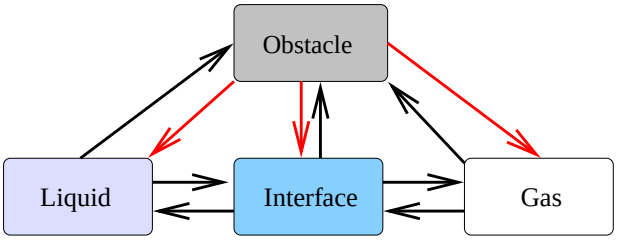

Figure 3: Possible cell type conversions in presence of moving obstacles.

2.2 Parallel Computation with Moving Obstacles

The cell conversion algorithm from the previous section is based on local operations. I.e., for the conversion of a cell only the direct cell neighbor information is needed. This means that a single ghost layer is sufficient for the distribution of information at process boundaries. The method has been tested on a scenario with particles floating on the free surface of a liquid. Fig. 4 shows the scaling results of strong and weak scaling over a varying number of processes. In either case the performance was measured in *MLups* (Millions of Lattice Updates per second). In the first case, strong scaling was performed with a fixed domain size of 512x512x64 cells. For a weak scaling experiment, we started on 24 processes with 4096 particles in a domain of size 320x240x300. The largest run so far was on 1536 processes, with 262144 particles in a domain of 2560x1920x300 cells. The performance was 998 MLups (ca. 86 % compared to ideal scaling). The test runs were executed on RRZE's LIMA cluster [8] which has a total number of 6000 cores. We are going to present the scaling results on 5952 cores soon.

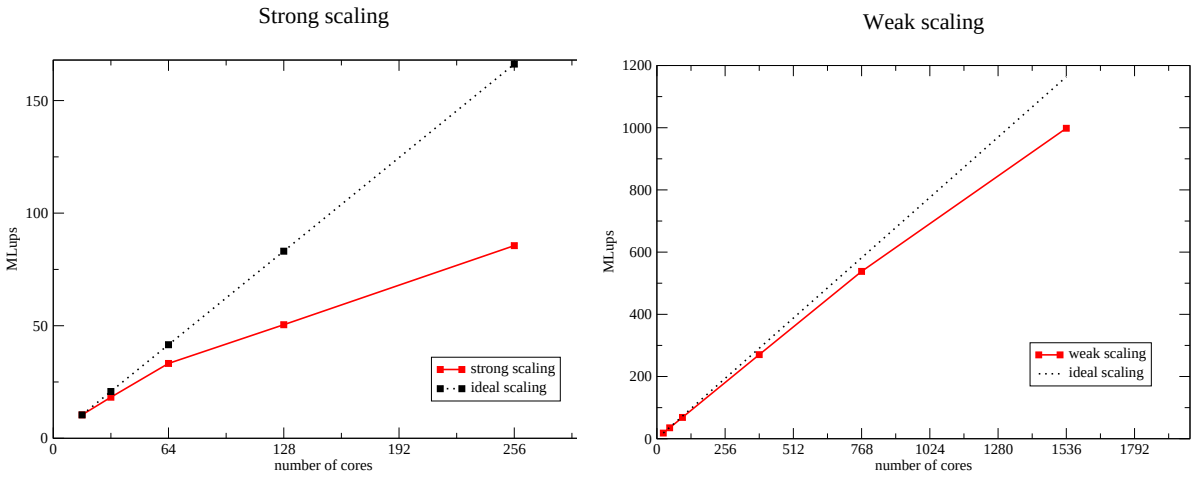

Figure 4: Strong and weak scaling performed on RRZE's LIMA Cluster [8].

3 Conclusion and future lines

The presented model for liquid-gas-solid flows has been implemented and tested within the waLBerla framework. Until now, most of the test cases consisted of some basin filled with liquid like in Fig. 5. This kind of flow is (relatively) simple in regards to the gas-liquid and gas-solid interaction, since there exists only one connected region of gas with a constant "atmospheric" pressure. However, for the future the goal is to extend our approach on more complex scenarios and arbitrary liquid-gas-solid suspensions which consist of a large number of independent bubbles interacting with the liquid and the particles. Since bubbly flows have already been realized already using the method from Sec. 1.1 [3], we expect to reach a similar goal with bubble-particle flows within the next years.

References

[1] Carolin Körner and Michael Thies. Lattice Boltzmann model for free surface flow. Material Science and Technology WTM, Universität Erlangen-Nürnberg, 2004.

[2] Thomas Pohl. High Performance Simulation of Free Surface Flows Using the Lattice Boltzmann Method. PhD thesis, Universität Erlangen-Nürnberg, 2007.

[3] Stefan Donath and Christian Feichtinger and Thomas Pohl and Jan Götz and Ulrich Rüde. A Parallel Free Surface Lattice Boltzmann Method for Large-Scale Applications. 21st International Conference onf Parallel Computational Fluid Dynamics, 2009

[4] D. Yu and R. Mei and L.-S. Luo and W. Shyy. Viscous Flow computation with the method of lattice Boltzmann equation. Prog. Aero. Sci. 39(5):329-367, 2003

[5] A.J.C. Ladd. Numerical simulations of particle suspensions via a discretized Boltzmann equation part i. theoretical foundation. Lawrence Livermore National Laboratory, 2007.

[6] Simon Bogner. Simulation of Floating Objects in Free-Surface Flows. Diploma Thesis. Lehrstuhl für Informatik 10, University of Erlangen-Nürnberg, 2009.

[7] Jan Götz and Christian Feichtinger and Klaus Iglberger and Stefan Donath and Ulrich Rüde Large scale simulation of fluid structure interaction using Lattice Boltzmann methods and the "physics engine". Proceedings of the 14th Biennial Computational Techniques and Applications Conference, CTAC-2008 (Canberra, AU, 13.07./ 16.07.2008) . 2008, S. C166–C188 (ANZIAM Journal)

[8] Lima Cluster, Regionales Rechenzentrum Erlangen (RRZE), http://www.rrze.uni-erlangen.de/dienste/arbeiten-rechnen/hpc/systeme/lima-cluster.shtml.

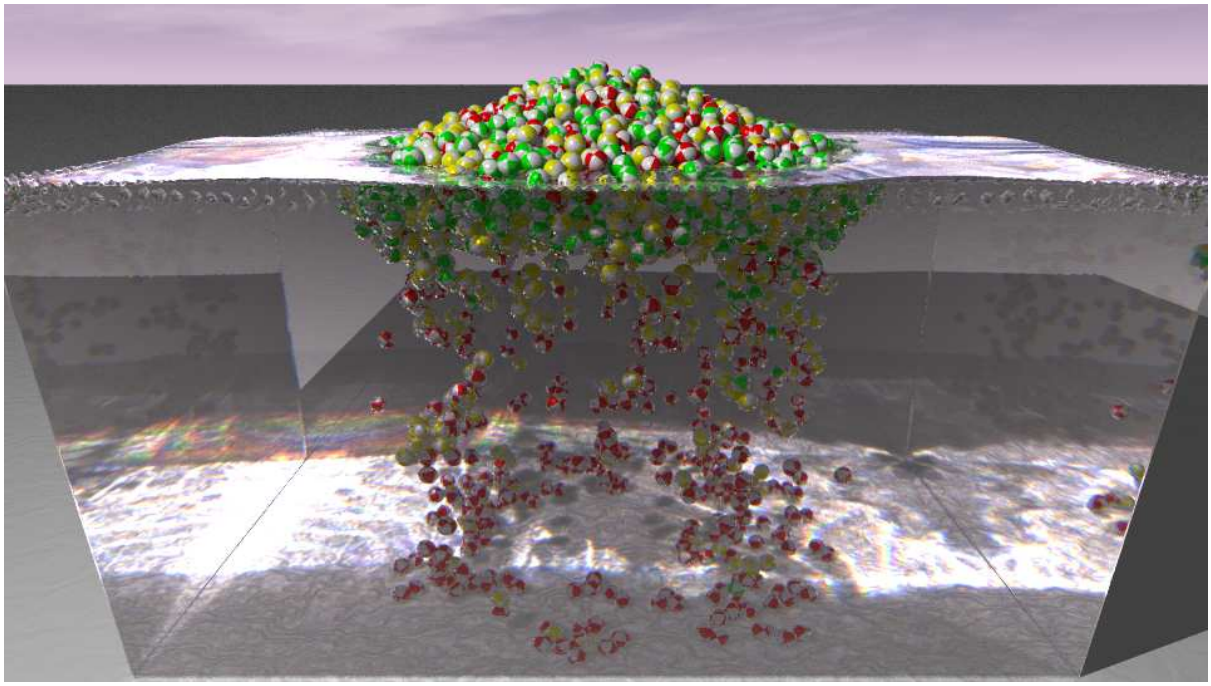

Figure 5: Visualization of a test case of 4096 particles being dropped into a basin filled with liquid.

[9] C. Feichtinger and S. Donath and H. Köstler and J. Götz and U. Rüde: WaLBerla: HPC Software Design for Computational Engineering Simulations, Accepted for publication in Journal of Computational Science, 2011