A Parallel Free Surface Lattice Boltzmann Method for Large-Scale Applications

Stefan Donath, Christian Feichtinger, Thomas Pohl, Jan Götz, Ulrich Rüde

Chair for System Simulation, University of Erlangen, 91058 Erlangen, Germany
{donath, feichtinger, pohl, goetz, ruede}@informatik.uni-erlangen.de

Abstract: In recent years the lattice Boltzmann method (LBM) has been established as a popular alternative to conventional computational fluid dynamics. With a free surface extension to the method, simulation of bubbly flows and even metal foaming processes is possible. The extension is based on a volume of fluid approach and an explicit tracking of the interface, including a reconstruction of the curvature to model surface tension. In order to accomplish realistic engineering applications, large domain sizes are required, and thus efficient parallelization for several thousand processes is inevitable. Our previous implementation of the parallel free surface algorithm used all-to-all communication schemes resulting in only moderate parallel efficiency when using more than hundred processes. Therefore, the algorithm has been adapted to communicate updates only locally in a restricted neighborhood, which complicates data exchange between processes, in particular when bubbles extend across several subdomains and in case topological changes occur through the coalescence of bubbles. The novel algorithm increases parallel efficiency and enables usage of several thousand processors, rendering large-scale engineering applications like simulation of liquid water in a fuel cell possible. It has been integrated into the waLBerla LBM framework, which features basic tools for communication and data management, designed for massively parallel flow simulations. With this implementation, free surface simulations exhibit parallel efficiency of 90% on up to 4 080 cores.

Keywords: free surface; massively parallel merge algorithm; lattice Boltzmann method.

1. INTRODUCTION

Free surface flow is omnipresent in nature and everyday life. Examples are river flow, oceanic waves, rain, or sparkling water. However, it also plays a crucial role in technological processes. Simulation of many bubbles or of a highly resolved free surface can assist understanding of metal foaming processes (Fig. 1, [8]) or can be used for shape optimization of boats. In order to accomplish such engineering applications, large domain sizes are required, and thus efficient parallelization for several thousand processes is inevitable. Within our seven years of experience in using the lattice Boltzmann method (LBM) with a free surface extension [8, 13], Pohl [9] parallelized the algorithm by means of MPI. Since his method uses all-to-all communication schemes, it achieves only moderate scaling efficiency with more than hundred processes. Therefore, the algorithm has been adapted to avoid all-to-all communication schemes, which complicates handling of topological changes like regions of gas (bubbles) crossing process boundaries or coalescing with each other. The novel algorithm increases parallel efficiency and enables usage of several thousand processors, rendering large-scale engineering applications possible. It has been integrated into the waLBerla framework [2], which is a software suite uniting different LBM flavors with basic tools for communication and data management, designed for massively parallel flow simulations. It is used as platform for sophisticated extensions to the LBM. Besides blood flow and Brownian motion, most prominent examples are multi-component flows, multi-phase or free-surface flows, and flows with moving objects. Most recently, large-scale simulation of fluid-structure interaction (FSI) has been accomplished (Fig. 2, [4]). With the successful integration of large-scale free surface method, simulation in fuel cells (see Sec. 3) is feasible, and possibly the combination with FSI will be achieved in future.

1.1 Related Work

Apart from the LBM, there is a variety of other computational fluid dynamics techniques to study multiphase flow phenomena, such as the volume of fluid method, level sets, and boundary integral methods. However, the advantage of LBM lies in its suitability for flows in complex geometries like porous media, since the mapping to the lattice is

Fig. 1: Simulation of foams with free-surface extension.

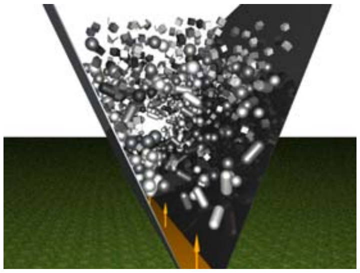

Fig. 2: Parallel simulation of fluidstructure interaction.

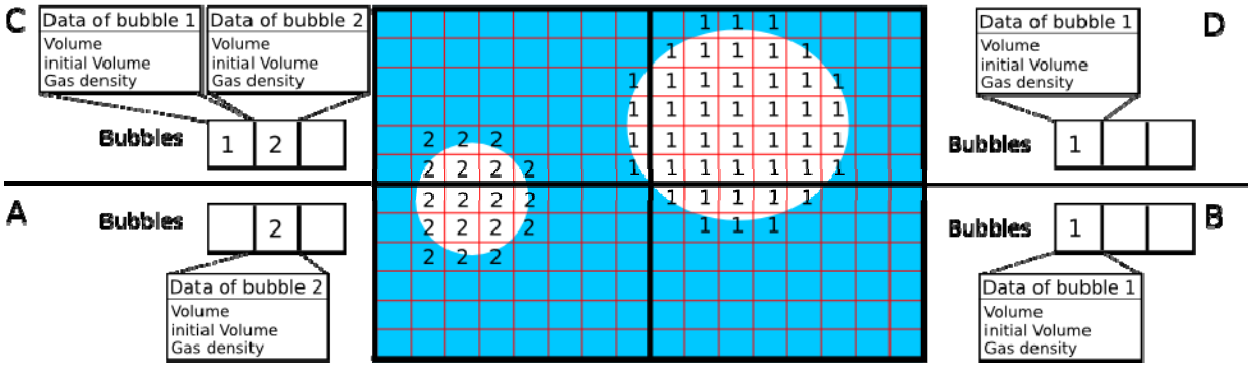

Fig. 3: Data representation of a bubble in free-surface LBM: Gas and interface cells store an ID of the bubble. Additionally, each process stores a list of bubble entries with volume data.

very flexible and involves little computational time. Gunstensen [5] and Tölke [14] presented enhanced variations based on the two-phase approach of Rothmann and Keller [10], which models capillary pressure resulting from the color gradient of a recoloring scheme. Shan and Chen [11] proposed to use an interaction potential representing molecular forces to simulate multiphase and multicomponent fluid flow. There are also methods based on free-energy approach and level sets. Besides the multiphase and multicomponent models, a couple of free-surface models arose in the past decade for simulating moving interfaces between immiscible gas and liquids. The method proposed by Ginzburg and Steiner [3] is the foundation of the variant by Körner et al. [7] which is based on the assumption that the influence of gas phase on liquid phase can be reduced to the force exerted by its pressure and the surface tension. Since only the liquid phase is computed and the interface is a more complex boundary condition, this method saves computation time compared to the two-phase methods mentioned before. Pohl [9] implemented Körner's method for three dimensions and altered the algorithm to enable efficient MPI parallelization.

1.2 Outline of Free Surface Lattice Boltzmann Method

The LBM is a cell-based, local update scheme that performs collision operations on so-called particle distribution functions (PDFs) which represent the fraction of the total mass in each lattice site moving in a discrete direction. Its theory [12] has been proven to resemble a solution of Navier-Stokes equations solving time-dependent, quasi-incompressible flows in continuum mechanics.

The free surface extension introduces different cell types for gas, liquid and the interface in between. While gas cells are omitted, in liquid cells standard LBM is performed. A fill value specifies the quantitiy of liquid in a cell. At the interface, missing PDFs from the gas phase are reconstructed such that velocities of gas and liquid equal, and forces by the liquid, gas pressure, and the surface tension are balanced. Complex calculations compute the pressure of the bubbles from their volume and curvature. This implies, that additionally to cell-based data, each process holds volume information on the bubbles (see Fig. 3).

For details on the LBM in general see [12]. Details on the free surface extension can be found in [7, 9].

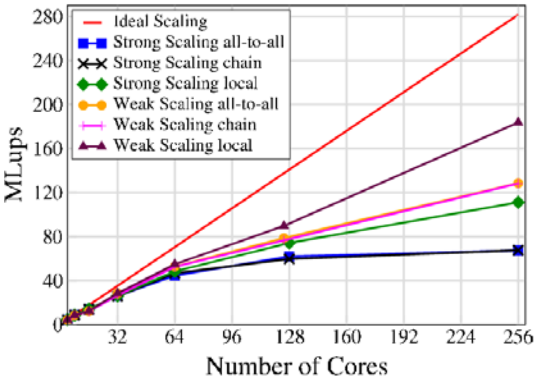

Fig. 4: *Weak and strong scaling on a Woodcrest platform [15] simulating a single small rising bubble in the center of the domain.*

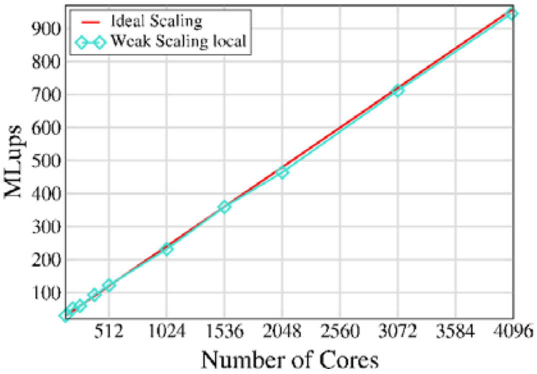

Fig. 5: *Weak scaling on an Itanium 2-based SGI Altix [6].*

2. LOCALIZED BUBBLE MERGE ALGORITHM

In previous implementations, handling of bubble data was not localized. Every process stored volume data of every bubble, irrespective where the bubble was located, and volume changes were communicated in each time step among all processes. Pohl [9] implemented two techniques to achieve this information exchange: Using all-to-all communication primitives of MPI library turned out to be less efficient than arranging all processes in a chain, where the two ends start to send packets upwards and downwards, respectively, and every participant in between merges the own information to the message which is passed on. The latter communication scheme results in *N-1* iteration steps for *N* processes, but involves only direct neighbor communication, minimizing messages over slow connections if processes are properly placed in the network.

In order to enable massively parallel free-surface simulations, the novel algorithm avoids global communication by storing bubble data only on processes that require it. As a consequence, bubbles crossing the boundary of a process have to be sent to the neighbor, and leaving of a bubble has to be recognized to allow deletion of volume data. Moreover, bubble coalescence has to be handled locally among the involved processes of the bubbles, which may become complicated if several bubble volumes have to be merged, possibly with data unknown to a process. The key of the concept is to store a handle for each process a bubble resides on, and communicate this data among all processes of this bubble. Additionally, information on coalescence with other bubbles, including the name of the process that is responsible for the fusion, is stored in the bubble data. With this information distributed, the requirements can be fulfilled: For proper detection whether a bubble crosses the boundary, the cell-based bubble ID (see Fig. 3) is used. If an unknown ID appears in the halo, data for a bubble has to be received. Likewise, if an ID touches the border to a neighbor that is not contained in the process list of this bubble, it has to be sent. At the same time, all other processes knowing this bubble have to be informed on the change. Handling merges is more complicated. The novel algorithm works with variable number of iterations, depending on the complexity of situation. Higher bubble IDs are merged to the lower, and a process responsible to perform a certain merge pair has to wait until all higher merges are done. If a merge is performed, all processes that knew one of the involved bubbles will be informed on the change. Conflicts can arise if bubble merge information is invalidated by another merge. In this case, the algorithm ensures restoration of consistency by rippling messages through the corresponding processes. Since MPI library requires an explicit receive issued in order to establish exchange of information, processes only passively involved in the merge action (i.e. know bubbles but not perform the merge themselves) have to communicate in each iteration until the expected data arrives. More details on this algorithm, including a description of its implementation in waLBerla, can be found in [1].

Consequently, the novel algorithm possibly means more communication in case of merges. However, this communication occurs on a more local vicinity, i.e. the neighborhood of processes harboring the bubbles involved. Since merges of bubbles occur rarely (a bubble usually covers the distance of one cell in approx. 1 000 time steps),

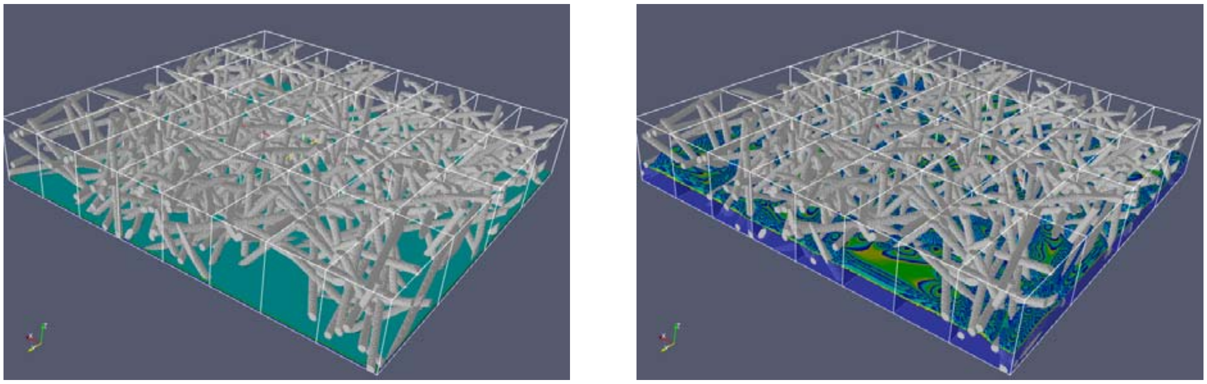

Fig. 6: *Small extract of simulation in fiber geometry similar to gas diffusion layer of a fuel cell. Colors on liquid surface depict fill values.*

the more complicated merge algorithm is paid off by the performance gain due to the saved all-to-all messages. Thus, parallel efficiency benefits by the locality of volume data exchange compared to the all-to-all communication in each time step. Figure 4 gives an impression of performance improvement, while Fig. 5 proves that this algorithm is suitable for large-scale parallel runs on up to 4 080 cores.

3. LARGE-SCALE FREE-SURFACE APPLICATIONS

A recent project with other universities, research institutes, and industry involves the simulation of liquid water in a polymer-electrolyte fuel cell (PEFC) with a proton exchange membrane. On the cathode side of the membrane, reaction of protons, electrons, and oxygen results in liquid water, which is to be evacuated from the reaction zone to sustain electrical performance. Hence, optimization of the structure and properties of the porous membrane is of particular interest. Since experiments cannot accomplish reliable quantification of water throughput in relation to material parameters due to the micron scales, simulation will assist improvement of process. The gas-diffusion layer (GDL) of a PEFC is characterized by a porous structure consisting of many thin fibers. Water generated in the layer below this structure has to evacuate towards a flow channel above. Fig. 6 shows a flooding scenario in a similar geometry, indicating the suitability of the novel algorithm for large-scale simulations. The simulation shown in Fig. 6 was carried out on a domain of $2.7 \cdot 10^7$ lattice cells. Final evaluations will be extended to domain sizes of $2.5 \cdot 10^9$ lattice cells. Due to nondisclosure agreements, Fig. 6 shows a synthetically generated geometry by random placement of fibers, not resembling any characteristics of a true GDL.

Figure 7 shows that the novel algorithm enables finely resolved simulations of many bubbles: A domain of $7.7 \cdot 10^8$ lattice cells contains 3 000 rising bubbles of different diameters, consuming approx. 400 GB of memory.

ACKNOWLEDGEMENTS

This work is carried out within the framework of DECODE, funded by the European Commision (CORDIS project 213295). Parts of this work are funded by the Bundesministerium für Bildung und Forschung under SKALB (grant 01IH08003A) as well as by the „Kompetenznetzwerk für Technisch-Wissenschaftliches Hoch- und Höchstleistungsrechnen in Bayern" (KONWIHR) via waLBerlaMC. Many thanks to the cxHPC group at the Regional Computing Center Erlangen (RRZE) for their valued support.

REFERENCES

1. Donath, S., Feichtinger, C., Götz, J., Deserno, F., Iglberger, K., and Rüde, U., *waLBerla: On implementation details of a localized parallel algorithm for bubble coalescence*, Tech. Report 09-3, Chair for System Simulation, Univ. of Erlangen, March 2009.
2. Feichtinger, C., Götz, J., Donath, S., Iglberger, K., and Rüde, U., *Concepts of waLBerla Prototype 0.1*, Tech. Report 07-10, Chair for System Simulation, Univ. of Erlangen, 2007.
3. Ginzburg, I., and Steiner, K., Lattice Boltzmann model for free-surface flow and its application to filling process in casting, *J. Comp. Phys. 185* (2003), 61-99.

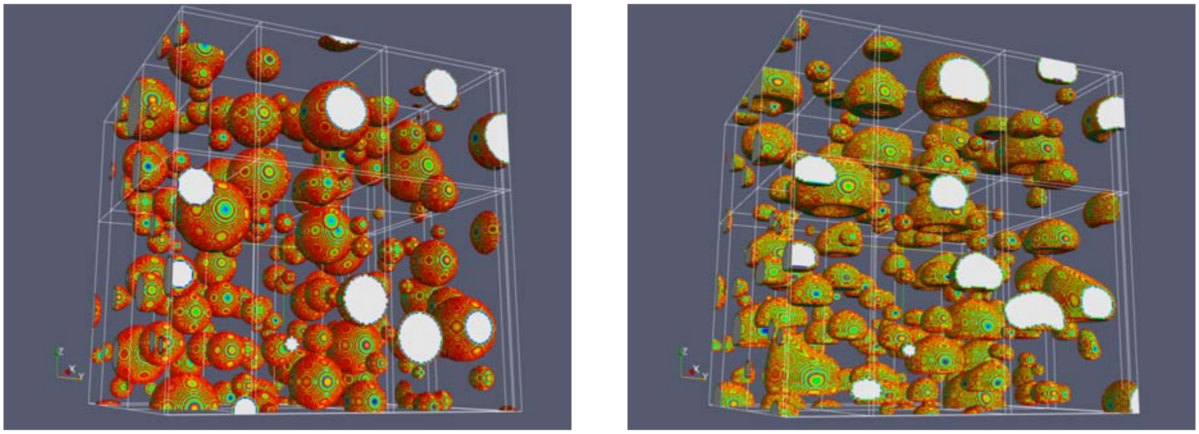

Fig. 7: *Small extract from simulation of 3 000 bubbles at time steps 1 (left) and 15 000 (right) showing only gas cells (white) and interface cells (colored by fill value).*

4. Götz, J., Feichtinger, C., Iglberger, K., Donath, S., and Rüde, U., Large scale simulation of fluid structure interaction using lattice Boltzmann methods and the 'physics engine', *Proceedings of CTAC 2008 (Geory N. Mercer and A. J. Roberts, eds.)*, *ANZIAM J.*, *vol. 50*, 2008, pp. C166-C188.

5. Gunstensen, A.K., Rothman, D.H., Zaleski, S., and Zanetti, G., Lattice Boltzmann model of immiscible fluids, *Phys. Rev. A 43* (1991), no. 8, 4320-4327.

6. HLRB2, http://www.lrz-muenchen.de/services/compute/hlrb/.

7. Körner, C., Thies, M., Hofmann, T., Thürey, N., and Rüde, U., Lattice Boltzmann model for free surface flow for modeling foaming, *J. Stat. Phys. 121* (2005), no. 1-2, 179-196.

8. Körner, C., Thies, M., and Singer, R.F., Modeling of metal foaming with lattice Boltzmann automata, *Adv. Eng. Mat. 4* (2002), 765-769.

9. Pohl, T., *High Performance Simulation of Free Surface Flows Using the Lattice Boltzmann Method*, Ph.D. thesis, Univ. of Erlangen, 2008.

10. Rothmann, D., and Keller, J.M., Immiscible cellular automaton fluids, *J. Stat. Phys. 52* (1988), 1119-1127.

11. Shan, X., and Chen, H., Lattice Boltzmann model for simulating flows with multiple phases and components, *Phys. Rev. E 47* (1993), no. 3, 1815-1820.

12. Succi, S., *The lattice Boltzmann equation for fluid dynamics and beyond*, Oxford University Press, 2001.

13. Thürey, N., and Rüde, U., *Free surface lattice-Boltzmann fluid simulations with and without level sets*, VMV, IOS Press, 2004, pp. 199-208.

14. Tölke, J., *Gitter-Boltzmann-Verfahren zur Simulation von Zweiphasenströmungen*, Ph.D. thesis, Technical University of Munich, 2001.

15. Woody, http://www.rrze.de/dienste/arbeiten-rechnen/hpc/systeme/woodcrest-cluster.shtml.