The Journal of Open Source Software

Musubi: Octree based Lattice-Boltzmann solver for multi-physics

Kannan Masilamani [1], **Gregorio Gerardo Spinelli** [4], **Kartik Jain** [2], **Manuel Hasert**[3], and **Harald Klimach** [1¶]

1 German Aerospace Center, DLR, Germany **2** University of Twente, Netherlands **3** Festo SE & Co. KG, Germany **4** MBDA (Germany), Germany ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 04 December 2025
Published: unpublished

Summary

Musubi is a multi-level, parallel lattice Boltzmann solver and part of the APES suite. It is working on an octree mesh that is linearized by a (Morton) space-filling curve and uses efficient data structures allowing for adaptive, distributed parallel simulations.

Musubi is designed to deal with huge meshes (billions of lattices) and complex geometries on large computing systems efficiently. It can be used for a wide range of application areas from electrodialysis (Masilamani, 2020) over biomedical problems (Jain, 2016) and aero-dynamic setups (Spinelli et al., 2024) to aero-acoustic simulations (Hasert, 2013).

It is written in Fortran, requiring a compiler that provides at least the Fortran 2003 standard.

Statement of need

Highly resolved fluid simulations are an integral part in many scientifc application areas. Due to the nonlinearity and the large amount of degrees of freedom to consider in these problems, these simulations require significant computational resources, which typically are only available in distributed parallel systems. Musubi implements the lattice Boltzmann method (LBM) with a Message Passing Interface (MPI) parallelization and a fully distributed handling of the data, avoiding bottlenecks on individual processors and enabling the scaling of the simulation to hundreds of thousands of MPI processes. Musubi is utilized to solve flow problems in the ultrasound simulator PROTEUS developed at the University of Twente (Blanken et al., 2025).

The lattice Boltzmann method

The lattice Boltzmann method employs ideas of cellular automata and can be represented at its core as a basic two step algorithm. The state of the fluid is represented by particle density functions (PDF) of a discrete velocity field. These PDFs reside on the lattices and are exchanged along the discrete velocity directions. The two steps of the algorithm are the streaming of the PDF information along velocity directions, followed by the so-called collision, computing a new PDF on each lattice. This modeling with discrete velocities also allows for a straight forward handling of complicated wall boundaries, as a simple line intersection with the wall geometry can be used to accurately model the surface. Due to these properties the method has gained popularity in the field of computational fluid dynamics over the last decades.

State of the field

There is a wide range of computational fluid dynamic methods, of which the lattice Boltzmann method represents an attractive option, due to its low number of operations and straight forward explicit implementation.

Other Open Source solvers that utilize this method are, for example, Palabos (Latt et al., 2020), OpenLB (Krause et al., 2021), waLBerla (Bauer et al., 2021) and VirtualFluids (Geier et al., 2025). They all aim at scale-resolved simulations of fluids. Musubi was developed (since 2011) in parallel or predated some of those projects due to a ground-up orientation to make use of octree meshes on massively parallel computing systems with a dedicated mesh format that allows for a distributed parallel reading from the file system. This approach allows for an automated multi-level mesh generation and avoids parallelisation bottlenecks between the mesh generation step and the simulation.

A specific domain that is addressed in Musubi and not covered by other Open Source LBM solvers, is the simulation of the Maxwell-Stefan equation for multiple species (Zudrop et al., 2017) as needed in diffusion processes that appear for example in electrodialysis applications.

Software design

Musubi implements the lattice Boltzmann method in the form of kernels that can be run on individual refinement levels of an octree mesh. It is developed within the APES-Suite (Klimach et al., 2014) of simulation tools revolving around a central Treelm library (Klimach et al., 2012) that provides the handling of this octree mesh on distributed parallel systems. This central part is implemented in a separate library, shared by the different tools in the framework. Though there still is a tight development dependency that is expressed by the use of git submodules.

The dedicated meshing tool Seeder (Harlacher et al., 2012) provides this octree mesh in a format that enables the distributed parallel reading of mesh partitions by all processes. The interpolation and transformation between the involved levels for the local refinement are separated from the kernel, allowing for an implementation of the respective methods without encumberment by the interpolation between the different resolutions. This method was described in detail in (Hasert et al., 2014) and enables the rapid implementation of new numerical kernels. There are various collision schemes implemented (BGK, MRT, HRR, Cumulants) (Spinelli et al., 2023), which can be used on a range of stencil configurations (discrete velocity directions). It is also possible to consider the transport of particles (Vlogman, 2025) and passive scalars in the flow.

The application is designed towards deployment on a wide range of high-performance computing systems. To facilitate this, Musubi is designed with a minimal set of dependencies allowing for deployment on a varity of supercomputing systems ranging from IBM's BlueGene to NEC's SX vector systems (Qi et al., 2016). As user interface in this environment, the scripting language Lua (Ierusalimschy, 2016) is chosen, which allows for a flexible configuration of simulation setups but does not introduce complicated dependencies, as Lua is implemented in standard ANSI C and is compiled along with the project.

Research impact statement

Musubi has been successfully deployed in the simulation of scale-resolved fluid problems for a variety of problems. Originally, a main funding source for the software was the German HISEEM project that aimed at the investigation of effective electrodialysis processes for seawater desalination (Johannink et al., 2015). Another main focus in the development is put on biomedical application (Jain et al., 2016), which was originally supported by the European THROMBUS project (Zimny et al., 2013). In the biodmedical domain Musubi now is also used as the fluid simulation tool in the ultrasound simulator PROTEUS developed at the University of Twente (Blanken et al., 2025).

While the two aforementioned application domains primarily are concerned with the simulation of liquids, there are also applications to gaseous fluids, for example in the simulation of human upper airway aerodynamics (Hebbink et al., 2022). Musubi's application extends beyond these domains and has been used in general aerodynamic simulations (Spinelli et al., 2024), aswell as in the domain of aero-acoustics (Qi et al., 2015), where the resolution of multiple spatial scales plays an important role. The distributed handling of octree meshes is well suited in these settings as it allows for the resolution of the scales that need to be resolved.

AI usage disclosure

No generative AI tools were used in the development of this software, the writing of this manuscript, or the preparation of supporting materials.

Acknowledgements

This software has been written by many people over the years. The individual authors can be found in each file with the respective copyright statement. Not appearing in those lists of authors is Sabine Roller, who enabled the development of this software in the first place and we are very grateful for this possibility. We especially thank our fellow contributors to this code basis Jiaxing Qi (Qi, 2017), Jens Zudrop (Zudrop, 2015), Simon Zimny (Zimny, 2015), Peter Vitt, Jana Gericke, Tristan Vlogman, Mengyu Wang and many students. Further, we thank Christian Siebert, who advised us on parallel algorithms and data structures. The development of Musubi was partially funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) in the framework of the HPC software initiative in the project HISEEM, by the European Commission in the Seventh Framework Programme in the area of Virtual Physiological Human (THROMBUS project, ICT-2009.5.3, project reference 269966), by the German Research School of Simulation Sciences, the University of Siegen, the University of Twente and the German Aerospace Center, DLR. We are grateful for the computing time provided by LRZ in Munich and by HLRS in Stuttgart who also contributed performance evaluations within the POP project.

References

Bauer, M., Eibl, S., Godenschwager, C., Kohl, N., Kuron, M., Rettinger, C., Schornbaum, F., Schwarzmeier, C., Thönnes, D., Köstler, H., & Rüde, U. (2021). waLBerla: A block-structured high-performance framework for multiphysics simulations. *Computers & Mathematics with Applications*, *81*, 478–501. https://doi.org/10.1016/j.camwa.2020.01.007

Blanken, N., Heiles, B., Kuliesh, A., Versluis, M., Jain, K., Maresca, D., & Lajoinie, G. (2025). PROTEUS: A physically realistic contrast-enhanced ultrasound simulator—part i: Numerical methods. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *72*(7), 848–865. https://doi.org/10.1109/TUFFC.2024.3427850

Geier, M., Kutscher, K., Schönherr, M., Wellmann, A., Peters, S., Alihussein, H., Linxweiler, J., & Krafczyk, M. (2025). VirtualFluids – open source parallel LBM solver. *Computer Physics Communications*, *317*, 109810. https://doi.org/10.1016/j.cpc.2025.109810

Harlacher, D. F., Hasert, M., Klimach, H., Zimny, S., & Roller, S. (2012). Tree based voxelization of STL data. In M. Resch, X. Wang, W. Bez, E. Focht, H. Kobayashi, & S. Roller (Eds.), *High performance computing on vector systems 2011* (pp. 81–92). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22244-3_6

Hasert, M. (2013). *Multi-scale lattice boltzmann simulations on distributed octrees* [PhD thesis]. RWTH Aachen University.

Hasert, M., Masilamani, K., Zimny, S., Klimach, H., Qi, J., Bernsdorf, J., & Roller, S. (2014). Complex fluid simulations with the parallel tree-based Lattice Boltzmann solver Musubi. *Journal of Computational Science*, *5*(5), 784–794. https://doi.org/10.1016/j.jocs.2013.11.001

Hebbink, R. H. J., Wessels, B. J., Hagmeijer, R., & Jain, K. (2022). Computational analysis of human upper airway aerodynamics. *Medical &Amp; Biological Engineering &Amp; Computing*, *61*(2), 541–553. https://doi.org/10.1007/s11517-022-02716-8

Ierusalimschy, R. (2016). *Programming in lua* (Fourth edition). Lua.org. ISBN: 9788590379867

Jain, K. (2016). *Transition to turbulence in physiological flows: Direct numerical simulation of hemodynamics in intracranial aneurysms and cerebrospinal fluid hydrodynamics in the spinal canal* [PhD thesis]. Universität Siegen. ISBN: 978-3-936533-83-5

Jain, K., Roller, S., & Mardal, K.-A. (2016). Transitional flow in intracranial aneurysms–a space and time refinement study below the kolmogorov scales using lattice boltzmann method. *Computers & Fluids*, *127*, 36–46. https://doi.org/doi:10.1016/j.compfluid.2015.12.011

Johannink, M., Masilamani, K., Mhamdi, A., Roller, S., & Marquardt, W. (2015). Predictive pressure drop models for membrane channels with non-woven and woven spacers. *Desalination*, *376*, 41–54. https://doi.org/10.1016/j.desal.2015.07.024

Klimach, H., Hasert, M., Zudrop, J., & Roller, S. (2012). Distributed Octree Mesh Infrastructure for Flow Simulations. *European Congress on Computational Methods in Applied Sciences and Engineering*, 1–15.

Klimach, H., Jain, K., & Roller, S. (2014). End-to-end parallel simulations with APES. In M. Bader, A. Bode, H.-J. Bungartz, M. Gerndt, G. R. Joubert, & F. Peters (Eds.), *Parallel computing: Accelerating computational science and engineering (CSE)* (Vol. 25, pp. 703–711). IOS Press. https://doi.org/10.3233/978-1-61499-381-0-703

Krause, M. J., Kummerländer, A., Avis, S. J., Kusumaatmaja, H., Dapelo, D., Klemens, F., Gaedtke, M., Hafen, N., Mink, A., Trunk, R., Marquardt, J. E., Maier, M. L., Haussmann, M., & Simonis, S. (2021). OpenLB–Open source lattice Boltzmann code. *Computers & Mathematics with Applications*, *81*, 258–288. https://doi.org/10.1016/j.camwa.2020.04.033

Latt, J., Malaspinas, O., Kontaxakis, D., Parmigiani, A., Lagrava, D., Brogi, F., Belgacem, M. B., Thorimbert, Y., Leclaire, S., Li, S., Marson, F., Lemus, J., Kotsalos, C., Conradin, R., Coreixas, C., Petkantchin, R., Raynaud, F., Beny, J., & Chopard, B. (2020). Palabos: Parallel lattice boltzmann solver. *Computers & Mathematics with Applications*. https://doi.org/10.1016/j.camwa.2020.03.022

Masilamani, K. (2020). *Framework for coupled simulation of electrodialysis processes* [PhD thesis, RWTH Aachen University]. https://doi.org/10.18154/RWTH-2021-02437

Qi, J. (2017). *Efficient lattice boltzmann simulations on large scale high performance computing systems* [PhD thesis, RWTH Aachen University]. https://doi.org/10.18154/RWTH-2017-04121

Qi, J., Hasert, M., Klimach, H., & Roller, S. (2015). Aeroacoustic simulation of flow through porous media based on lattice boltzmann method. In M. Resch, W. Bez, E. Focht, H. Kobayashi, J. Qi, & S. Roller (Eds.), *Sustained simulation performance 2015* (pp. 195–204). Springer International Publishing. https://doi.org/10.1007/978-3-319-20340-9_16

Qi, J., Jain, K., Klimach, H., & Roller, S. (2016). Performance evaluation of the LBM solver musubi on various HPC architectures. In G. R. Joubert, H. Leather, M. Parsons, F. Peters, & M. Sawyer (Eds.), *Parallel computing: On the road to exascale* (Vol. 27, pp. 807–816).

https://doi.org/10.3233/978-1-61499-621-7-807

Spinelli, G. G., Gericke, J., Masilamani, K., & Klimach, H. G. (2024). Key ingredients for wall-modeled LES with the lattice boltzmann method: Systematic comparison of collision schemes, SGS models, and wall functions on simulation accuracy and efficiency for turbulent channel flow. In *Discrete and Continuous Dynamical Systems - S* (No. 11; Vol. 17, pp. 3224–3251). https://doi.org/10.3934/dcdss.2023212

Spinelli, G. G., Horstmann, T., Masilamani, K., Soni, M. M., Klimach, H., Stück, A., & Roller, S. (2023). HPC performance study of different collision models using the lattice boltzmann solver musubi. *Computers & Fluids*, *255*, 105833. https://doi.org/10.1016/j.compfluid.2023.105833

Vlogman, T. (2025). *Coupled lattice boltzmann and discrete element method simulations with applications to liver radioembolization* [PhD Thesis - Research UT, graduation UT, University of Twente; University of Twente]. https://doi.org/10.3990/1.9789036567046

Zimny, S. (2015). *Hemodynamic flow simulation in patient specific cerebral aneurysms* [PhD thesis]. RWTH Aachen University.

Zimny, S., Chopard, B., Malaspinas, O., Lorenz, E., Jain, K., Roller, S., & Bernsdorf, J. (2013). A multiscale approach for the coupled simulation of blood flow and thrombus formation in intracranial aneurysms. *Procedia Computer Science*, *18*, 1006–1015. https://doi.org/10.1016/j.procs.2013.05.266

Zudrop, J. (2015). *Efficient numerical methods for fluid- and electrodynamics on massively parallel systems* [PhD thesis]. RWTH Aachen University.

Zudrop, J., Masilamani, K., Roller, Sabine, & Asinari, P. (2017). A robust lattice boltzmann method for parallel simulations of multicomponent flows in complex geometries. *Computers and Fluids*, *153*, 20–33. https://doi.org/10.1016/j.compfluid.2017.04.021