The Journal of Open Source Software

MARBLES: Multi-scale Adaptively Refined Boltzmann LatticE Solver

Marc T. Henry de Frahan [1], **Nilesh Sawant** [1], **Ethan Young** [1], **Hariswaran Sitaraman** [1], **Ross E. Larsen** [1,2], **and Jon Rood** [1]

1 Scalable Algorithms, Modeling and Simulation Group, National Renewable Energy Laboratory, USA **2** Renewable and Sustainable Energy Institute, University of Colorado Boulder, Boulder, Colorado 80309

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Philip Cardiff

Reviewers:

- @pkkolluru
- @WeiqunZhang

Submitted: 29 August 2025
Published: unpublished

Summary

MARBLES is a computational fluid dynamics solver that leverages the lattice Boltzmann method (LBM) and block-structured adaptive mesh refinement (AMR) to simulate flows in complex media. The solver leverages the AMReX (Zhang et al., 2019) library, a library that provides underlying data structures, parallel paradigms, grids, domain decomposition, and portability programming models to run on massively parallel computing architectures. All major GPU architectures (e.g., NVIDIA, AMD, Intel) are supported through the use of performance portability functionalities implemented in AMReX. The MARBLES software is released in NREL Software Record SWR-23-37 "MARBLES (Multi-scale Adaptively Refined Boltzmann LatticE Solver)" (Henry de Frahan et al., 2023).

MARBLES implements LBM, a mesoscopic approach to computational fluid dynamics in which the computation of the Boltzmann transport equation by advection and collision of fictitious particles results into a solution of the Navier Stokes equation at the macroscale. The evolution equation of LBM proceeds in two parts: a "streaming" step akin to space discretization in which particles representing probability distribution functions at grid cells transfer information about their current state to their nearest neighbors and a "collision" step akin to time discretization in which these advected quantities give rise to new state variables by subsequently relaxing toward their equilibrium distributions over some characteristic time. Macroscopic state variables such as density, momentum and total energy are obtained by locally computing moments of the probability distribution functions. In MARBLES, these steps are carried out on parallelized AMReX blocks in which the phase space is discretized using 27 grid neighbors in 3D (D3Q27) which provides high accuracy and slightly increased memory consumption compared to other LBM stencils. An equivalent stencil is implemented for 2D simulations. The stencil can be sub-divided and applied locally in a region with half the grid spacing using an implementation of the explode and coalesce algorithm (Chen et al., 2006), where refined cells process two time steps per every step of the parent grid assuming a factor of two grid refinement ratio. This cycling between refinement levels and interpolation between grids is managed by AMReX functionality and minimizes communication between parallel blocks to enable excellent scaling performance.

MARBLES uses the Embedded Boundary (EB) formulation in AMReX to represent complex geometries. In this approach, an arbitrary surface is defined by the user, using either compositions of simple shapes or an STL file. This is used to intersect the Cartesian mesh and define cells that are covered by the geometry (i.e., inside the body). Bounce back conditions are imposed on these surfaces as part of the streaming step to capture the effect of the geometry on the fluid flow. This leads to a very robust handling of very complex geometry, including flow through porous media.

MARBLES is written in C++ and uses an MPI+X approach: the AMR grid patches are distributed on different CPU ranks using MPI. Each grid can be either (i) logically tiled and solved on different threads on a multi-core CPU machine using OpenMP, or (ii) solved on GPU threads on GPU nodes using CUDA, HIP, and SYCL.

Statement of Need

Several software tools that implement the Lattice Boltzmann method can found online, including a PyTorch based solver, Lettuce (Bedrunka et al., n.d.), the commonly used OpenLB (Krause et al., 2021), Palabos (Latt et al., 2021), waLBerla (Bauer et al., 2021; Godenschwager et al., 2013), and others (Mora et al., 2020; Pastewka & Greiner, 2019; Schmieschek et al., 2017).

In contrast with these solvers, MARBLES is the only code to the authors' knowledge to be built on AMReX data structures which allows us to tackle exascale problems and take advantage of both CPU and GPU hardware with little change to the underlying code. Because of this, MARBLES is easily extensible to other capabilities and has the potential to perform well on emerging architectures. MARBLES, through its use of vendor-agnostic programming models implemented in AMReX, achieves high performance from a small desktop station to the world's largest supercomputer. Additionally, AMReX provides tools to naturally track regions of the fluid in need of refinement (e.g., around an evolving fluid-vapor interface or near the surface of a deforming solid obstacle) and communicate variables between grid refinement levels in a highly performant manner.

MARBLE's unique features consist in implementing the two-population compressible thermal LBM (Sawant et al., 2022) in a highly scalable and performance portable framework. The LBM formulation implemented in MARBLES uses one lattice for mass and momentum equations and another lattice for the energy equation, keeping the whole solver within the lattice Boltzmann framework without the need for a hybrid discretization approach. A fully lattice Boltzmann framework combines exact conservation as in finite volume solvers with the flexibility of a structured grid like in immersed boundary finite difference solvers, without their respective drawbacks of grid generation and conservation errors, respectively. Using a nearest-neighbor based correction to equilibrium pressure, the model maintains Galilean invariance up to third order in velocity space, sufficient to faithfully simulate the correct viscosity and thermal diffusivity over a wide range of temperatures, while still using the standard D3Q27 lattice.

MARBLES is intended for students, researchers and engineers interested in simulating mesoscale and macroscale flows with the Lattice Boltzmann method on modern high performance computing hardware. With performance portability and being vendor agnostic foremost considerations in the design of the code, MARBLES can leverage the computational resources available on the latest heterogeneous exascale platforms, including GPUs. In this context, MARBLES is a valuable tool for studying fluid dynamics and heat transfer scientific and engineering problems involving complex geometries. Its applications include flows in porous media such as electrodes and geological microstructures, aerodynamic calculations such as forces on turbines blades and structures subject to external flows, and non-equilibrium flows such as those involved in semiconductor manufacturing.

Acknowledgments

This work was authored by the National Renewable Energy Laboratory (NREL), operated under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

Bauer, M., Eibl, S., Godenschwager, C., Kohl, N., Kuron, M., Rettinger, C., Schornbaum, F., Schwarzmeier, C., Thönnes, D., Köstler, H., & others. (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

Bedrunka, M. C., Wilde, D., Kliemank, M., Reith, D., Foysi, H., & Krämer, A. (n.d.). Lettuce: PyTorch-based lattice boltzmann framework. *High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt Am Main, Germany, June 24–July 2, 2021, Revised Selected Papers*, 40. https://doi.org/10.1007/978-3-030-90539-2_3

Chen, H., Filippova, O., Hoch, J., Molvig, K., Shock, R., Teixeira, C., & Zhang, R. (2006). Grid refinement in lattice boltzmann methods based on volumetric formulation. *Physica A: Statistical Mechanics and Its Applications*, *362*(1), 158–167. https://doi.org/10.1016/j.physa.2005.09.036

Godenschwager, C., Schornbaum, F., Bauer, M., Köstler, H., & Rüde, U. (2013). A framework for hybrid parallel flow simulations with a trillion cells in complex geometries. *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 1–12. https://doi.org/10.1145/2503210.2503273

Henry de Frahan, M., Young, E., Sitaraman, H., Rood, J., & Larsen, R. (2023). *MARBLES (multi-scale adaptively refined boltzmann LatticE solver) [SWR-23-37]*. [Computer Software] https://doi.org/10.11578/dc.20231009.2. https://doi.org/10.11578/dc.20231009.2

Krause, M. J., Kummerländer, A., Avis, S. J., Kusumaatmaja, H., Dapelo, D., Klemens, F., Gaedtke, M., Hafen, N., Mink, A., Trunk, R., & others. (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., & others. (2021). Palabos: Parallel lattice boltzmann solver. *Computers & Mathematics with Applications*, *81*, 334–350. https://doi.org/10.1016/j.camwa.2020.03.022

Mora, P., Morra, G., & Yuen, D. A. (2020). A concise python implementation of the lattice boltzmann method on HPC for geo-fluid flow. *Geophysical Journal International*, *220*(1), 682–702. https://doi.org/10.1093/gji/ggz423

Pastewka, L., & Greiner, A. (2019). *HPC with python: An MPI-parallel implementation of the lattice boltzmann method*. https://doi.org/10.15496/publikation-29062

Sawant, N., Dorschner, B., & Karlin, I. V. (2022). Consistent lattice Boltzmann model for reactive mixtures. *Journal of Fluid Mechanics*, *941*, A62. https://doi.org/10.1017/jfm.2022.345

Schmieschek, S., Shamardin, L., Frijters, S., Krüger, T., Schiller, U. D., Harting, J., & Coveney, P. V. (2017). LB3D: A parallel implementation of the lattice-boltzmann method for simulation of interacting amphiphilic fluids. *Computer Physics Communications*, *217*, 149–161. https://doi.org/10.1016/j.cpc.2017.03.013

Zhang, W., Almgren, A., Beckner, V., Bell, J., Blaschke, J., Chan, C., Day, M., Friesen, B., Gott, K., Graves, D., Katz, M., Myers, A., Nguyen, T., Nonaka, A., Rosso, M., Williams, S., & Zingale, M. (2019). AMReX: A framework for block-structured adaptive mesh refinement. *Journal of Open Source Software*, *4*(37), 1370. https://doi.org/10.21105/joss.01370