

A Practical Implementation of Quantum Lattice Boltzmann Algorithms

Călin A. Georgescu^{a,*}, Matthias Möller^a

^aDelft University of Technology, Department of Applied Mathematics, Mekelweg 4, Delft, The Netherlands

ARTICLE INFO[†]

Keywords:

Quantum Computing;
Quantum Algorithm;
Lattice Boltzmann;
Python

ABSTRACT

We introduce QLBM, a software package that aims to unify the fragmented landscape of quantum Lattice Boltzmann Methods (QBM) and facilitate future work. We highlight several nuances of competing approaches and introduce the user- and researcher-facing sides of QLBM. We demonstrate how QLBM provides an end-to-end environment for developing and analyzing QBMs for both simulators and quantum hardware. QLBM is publicly available at <https://github.com/QCFD-Lab/qlbm> under an MPL-2.0 license.

1. Introduction

The field of Quantum Computing (QC) [1] has recently received a substantial amount of attention from both academia and industry. QC carries the potential to augment the current hybridized computational landscape with a drastically different yet complementary archetype. The exponential memory efficiency and quantum parallelism traits of the quantum computing paradigm makes it especially attractive for the increasingly demanding large-scale CFD applications of today.

The current state of QC hardware has been undergoing rapid development and is currently in the so-called *Noisy Intermediate-Scale Quantum* phase [2]. While quantum computers available today showcase some of the core advantages that theoretical physics promises, they are limited in both the number of qubits available and the time span that qubits can retain coherent states. These constraints impose great limitations on the applications that quantum computers can presently carry out.

To facilitate the research of quantum algorithms in an era without *Fault-Tolerant* QCs, scientists have turned to simulation methods instead. Recently, increasingly many software frameworks have emerged to bridge the gap between theoretical advances in algorithmics and hardware availability. These range from general purpose simulation tools [3, 4, 5] to specialized packages aimed at machine learning [6, 7] and material simulation [8]. The current state of *quantum software* intersects QC theory and available hardware such that researchers can leverage classical

hardware to verify large-scale algorithmic prototypes while quantum counterparts edge closer to fault-tolerance. At the same time, available software retains compatibility with the circuit model of universal quantum computers.

In this work, we introduce QLBM, a software package for developing quantum Lattice Boltzmann Methods. We first briefly introduce the Lattice Boltzmann Method (LBM) and its quantum counterparts that motivate the importance of our work in Sec. 1.1 and Sec. 1.2, respectively. Section 2 describes the QLBM software package and its features.

1.1. Lattice Boltzmann methods

The Boltzmann Equation (BE) describes the kinetic behavior of fluid at the mesoscopic scale, nestled between microscopic Newtonian dynamics and macroscopic Navier-Stokes continua. The BE models the state of populations of fluid particles as a statistical distribution function over physical space, velocity, and time. The discretization of the BE along phase space and time yields the Lattice Boltzmann Equation, which can in turn be solved numerically by the Lattice Boltzmann Method (LBM). Each LBM time step can be conceptually broken down into three subroutines: streaming through physical space, reflection at the boundaries of the fluid domain, and (non-linear) particle collision.

In recent years, the LBM has become a more and more popular option to solving fluid flow problems for several reasons. From a theoretical standpoint, it allows for the computation of macroscopic quantities such as mass and momentum density [9]. From a practical standpoint, the LBM lends itself well to massively parallel computing paradigms [10, 9]. Over the years, several parallel software implementations of the LBM have emerged, including OPENLB [11], WALBERLA [12], LBMPY [13], and PYLBM [14], which are able to carry out distributed simulations on hundreds of heterogeneous compute nodes. Simultaneously, researchers have been investigating formulations that allow the parallelism of the LBM to be exploited through paradigm of quantum computing.

[†]This paper is part of the ParCFD 2024 Proceedings. A recording of the presentation is available on YouTube. The DOI of this document is 10.34734/FZJ-2025-02446 and of the Proceedings 10.34734/FZJ-2025-02175.

*Corresponding author

 c.a.georgescu@tudelft.nl (C.A. Georgescu); m.moller@tudelft.nl (M. Möller)

ORCID(s): 0000-0002-8102-6389 (C.A. Georgescu);
0000-0003-0802-945X (M. Möller)

High-Level Specification

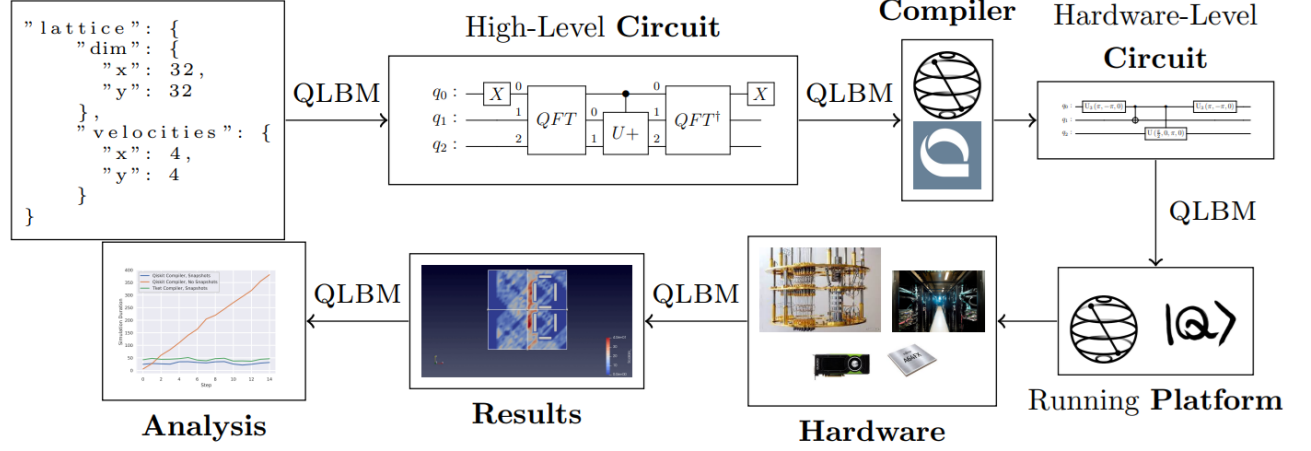


Figure 1: Overview of the QLBM workflow.

1.2. Quantum algorithms for Lattice Boltzmann methods

The initial wave of research into Quantum Computational Fluid Dynamics (QCFD) occurred between 2001 and 2003 and largely focused on extending the lattice-gas model to distributed quantum devices [15, 16, 17, 18]. This work tailors quantum lattice-gas solvers to a decentralized system of quantum computers with limited number of qubits per device, linked together through classical communication channels. Though this approach enables the balancing of the computational workload through horizontal scaling, it requires a number of qubits that grows linearly with the number of grid points of the lattice.

Recently, Quantum Lattice Boltzmann Methods (which we abbreviate as QBMs, to distinguish from the software) have emerged as promising candidates for the future direction of QCFD. Research surrounding QBMs has largely focused on the development of quantum primitives that implement (parts of) the LBM time-marching loop. These initiatives have given rise to several techniques that accommodate specific subroutines of the LBM, imposing trade-offs between scalability and versatility. One way to categorize existing QBMs is by how they address the inherent nonlinearity of collision.

Todorova and Steijl [19] and Schalkers and Möller [20] propose *collisionless* methods that include primitives for particle streaming and boundary conditions, but omit the collision operator entirely. Steijl and co-workers [21, 22] alternatively propose a method in which quantum primitives that implement floating point arithmetic can compute nonlinear terms, but require a reversible conversion between the encoding of the quantum state used to perform streaming and

the encoding that enables the computation of the nonlinear velocity terms at each time step. Succi and collaborators [23, 24] adopt an approach based on truncated Carleman linearization, that approximates the non-linear LBE by a finite-dimensional linear system of equations that can be expressed in terms of (unitary) quantum operators. Budinski [25, 26] further developed an approach that enables both streaming and collision but that incurs a probability of measuring an orthogonal (irrelevant) quantum state after each time-step. Finally, Schalkers and Möller [27] extended a previously developed encoding equipped it with a collision operator at the cost of requiring a number of qubits that scales with the number of simulated time steps.

The current state of QBMs is fragmented between several approaches that each present different strengths and weaknesses. This poses several challenges for researchers seeking to advance the field. First, the many nuances of present QBM techniques make the comparison of the performance and scalability difficult. From various quantum state encodings to the decomposition of exponentially sized matrices into quantum gates, QBMs build on top of extensive knowledge and technology stacks that make implementation a significant challenge. Second, the fractured nature of the field poses challenges for techniques that augment existing work. Third, the scarce availability of QBM implementations detracts from the reproducibility of the field.

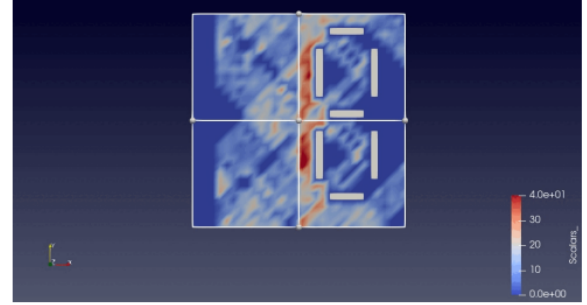
2. The QLBM package

In this talk we present QLBM, an open-source software package that aims to accelerate QBM research. With this

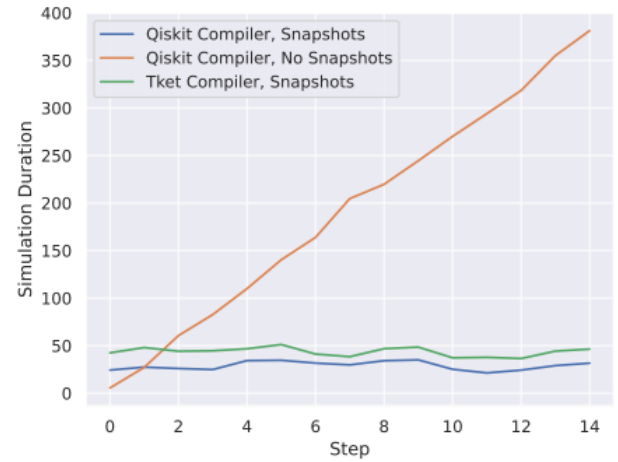
initiative we seek to lay the foundation of a unified end-to-end development environment for quantum Lattice Boltzmann algorithms. QLBM aims to facilitate both the seamless development and comparison of existing methods and their integration into the broader quantum software technology stack, reducing redundant effort for researchers and practitioners alike.

Figure 1 provides a high-level schematic of the end-to-end workflow that QLBM enables. In the first part of the talk, we focus on the application of QBMs already implemented in the software. This includes a demonstration of how QLBM converts human-readable JSON specifications of 2D and 3D systems into quantum circuits, and how to convert the generated circuits to specific platforms. We also showcase the integration of QLBM with PARAVIEW [28] for flow field and geometry visualization. Figure 2a displays an example of the supported visualization techniques. Finally, we show several concrete simulation use cases that users can implement with just a few lines of Python code.

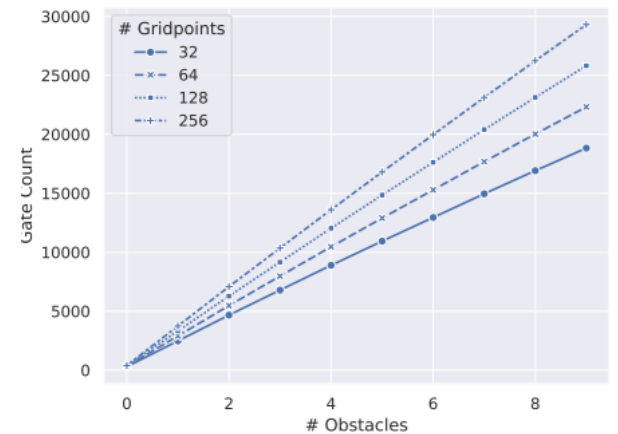
In the second part of the talk, we concentrate on QLBM as a tool for developing novel QBMs. To this end, we highlight the internal structure of the framework, the segregation of quantum circuit primitives into a complexity hierarchy, and the extensibility of already implemented methods. We also spotlight QLBM's out-of-the-box integration with state-of-the-art quantum circuit compilers and simulators that support a broad range of CPU and GPU architectures. To complement the research-oriented side of QLBM, we showcase mechanisms for exploiting potential speedups in quantum simulators and an accompanying set of analysis tools that allow for fair and replicable comparisons between competing methods. Figure 2b shows one such mechanism, which reduces the runtime of simulating time steps of the collisionless QBM (CQBM) developed by Schalkers and Möller [20] from linear to constant. QLBM additionally contains tools meant to ease the transition from simulators to (possibly fault-tolerant) quantum hardware. We demonstrate QLBM's capacity to accurately estimate critical properties of transpiled quantum circuits for a variety of different quantum hardware vendors. Such properties include circuit depth, and the number exact of hardware-native gates. QLBM relies on QISKIT to enable flexible analyses with regard to qubit connectivity and gate sets, which come at no additional cost to the software's internal representation. Figure 2c depicts how the number of gates of circuits implementing CQBM [20] scales with the number of obstacles for 4 different grid refinement levels. We conclude the talk by discussing prospects of QCfD software and QLBM's place within the broader domain.



(a) 2D flow field visualization around boundary-conditioned geometry.



(b) Run time reduction for simulating one time step using QLBM exploits.



(c) Number of high-level gates for 2D CQBM algorithm for different grids.

Figure 2: Example of visualization and analysis options built in QLBM.

References

- [1] M. A. Nielsen, I. L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2010. doi:10.1017/CBO9780511976667.
- [2] J. Preskill, Quantum Computing in the NISQ era and beyond, *Quantum* 2 (2018) 79. doi:10.22331/q-2018-08-06-79.
- [3] Qiskit contributors, *Qiskit: An open-source framework for quantum computing* (2023). doi:10.5281/zenodo.2573505.
- [4] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, S. Tamiya, T. Yamamoto, T. Yan, T. Kawakubo, Y. O. Nakagawa, Y. Ibe, Y. Zhang, H. Yamashita, H. Yoshimura, A. Hayashi, K. Fujii, Qulacs: a fast and versatile quantum circuit simulator for research purpose, *Quantum* 5 (2021) 559. doi:10.22331/q-2021-10-06-559.
- [5] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, R. Duncan, tket: a retargetable compiler for nisq devices, *Quantum Science and Technology* 6 (1) (2020) 014003. doi:10.1088/2058-9565/ab8e92.
- [6] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. Di Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isaacson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O’Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, N. Killoran, PennyLane: Automatic differentiation of hybrid quantum-classical computations (2018). arXiv:1811.04968.
- [7] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa, E. Peters, O. Lockwood, A. Skolik, S. Jerbi, V. Dunjko, M. Leib, M. Streif, D. Von Dollen, H. Chen, S. Cao, R. Wiersema, H.-Y. Huang, J. R. McClean, R. Babush, S. Boixo, D. Bacon, A. K. Ho, H. Neven, M. Mohseni, TensorFlow Quantum: A Software Framework for Quantum Machine Learning (2020). arXiv:2003.02989.
- [8] L. Bassman Oftelie, C. Powers, W. A. De Jong, Arqtic: A full-stack software package for simulating materials on quantum computers, *ACM Transactions on Quantum Computing* 3 (3) (2022) 1–17. doi:10.1145/3511715.
- [9] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Viggien, The lattice boltzmann method, *Springer International Publishing* 10 (978-3) (2017) 4–15. doi:10.1103/PhysRev.94.511.
- [10] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Oxford University Press, 2001. doi:10.1093/oso/9780198503989.001.0001.
- [11] M. J. Krause, A. Kummerländer, S. J. Avis, H. Kusumaatmaja, D. Dapelo, F. Klemens, M. Gaedtke, N. Hafen, A. Mink, R. Trunk, J. E. Marquardt, M.-L. Maier, M. Haussmann, S. Simonis, OpenLB—Open source lattice Boltzmann code, *Computers & Mathematics with Applications* 81 (2021) 258–288. doi:10.1016/j.camwa.2020.04.033.
- [12] M. Bauer, S. Eibl, C. Godenschwager, N. Kohl, M. Kuron, C. Rettinger, F. Schornbaum, C. Schwarzmeier, D. Thönnies, H. Köstler, U. Rude, waLBerla: A block-structured high-performance framework for multiphysics simulations, *Computers & Mathematics with Applications* 81 (2021) 478–501. doi:10.1016/j.camwa.2020.01.007.
- [13] M. Bauer, H. Köstler, U. Rude, lbmpy: Automatic code generation for efficient parallel lattice Boltzmann methods, *Journal of Computational Science* 49 (2021) 101269. doi:10.1016/j.jocs.2020.101269.
- [14] Pylbm contributors, *Pylbm: an all-in-one package for numerical simulations using lattice boltzmann solvers*. (2023). URL <https://github.com/pylbm/pylbm>
- [15] J. Yepez, Quantum lattice-gas model for computational fluid dynamics, *Physical Review E* 63 (4) (2001) 046702. doi:10.1103/PhysRevE.63.046702.
- [16] J. Yepez, B. Boghosian, An efficient and accurate quantum lattice-gas model for the many-body Schrödinger wave equation, *Computer Physics Communications* 146 (3) (2002) 280–294. doi:10.1016/S0010-4655(02)00419-8.
- [17] J. Yepez, Quantum lattice-gas model for the Burgers equation, *Journal of Statistical Physics* 107 (2002) 203–224. doi:10.1023/A:1014514805610.
- [18] M. A. Pravia, Z. Chen, J. Yepez, D. G. Cory, Experimental demonstration of quantum lattice gas computation, *Quantum Information Processing* 2 (2003) 97–116. doi:10.1023/A:1025835216975.
- [19] B. N. Todorova, R. Steijl, Quantum algorithm for the collisionless Boltzmann equation, *Journal of Computational Physics* 409 (2020) 109347. doi:10.1016/j.jcp.2020.109347.
- [20] M. A. Schalkers, M. Möller, Efficient and fail-safe quantum algorithm for the transport equation, *Journal of Computational Physics* 502 (2024) 112816. doi:10.1016/j.jcp.2024.112816.
- [21] R. Steijl, Quantum Algorithms for Nonlinear Equations in Fluid Mechanics, in: *Quantum Computing and Communications*, IntechOpen, 2022. doi:10.5772/intechopen.95023.
- [22] Y. Moawad, W. Vanderbauwhede, R. Steijl, Investigating hardware acceleration for simulation of CFD quantum circuits, *Frontiers in Mechanical Engineering* 8 (2022) 925637. doi:10.3389/fmech.2022.925637.
- [23] W. Itani, S. Succi, Analysis of Carleman linearization of lattice Boltzmann, *Fluids* 7 (1) (2022) 24. doi:10.3390/fluids7010024.
- [24] C. Sanavio, S. Succi, Lattice Boltzmann–Carleman quantum algorithm and circuit for fluid flows at moderate Reynolds number, *AVS Quantum Science* 6 (2) (2024). doi:10.1116/5.0195549.

- [25] L. Budinski, Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method, *Quantum Information Processing* 20 (2) (2021) 57. doi:10.1007/s11128-021-02996-3.
- [26] L. Budinski, Quantum algorithm for the Navier–Stokes equations by using the streamfunction-vorticity formulation and the lattice Boltzmann method, *International Journal of Quantum Information* 20 (02) (2022) 2150039. doi:10.1142/S0219749921500398.
- [27] M. A. Schalkers, M. Möller, On the importance of data encoding in quantum Boltzmann methods, *Quantum Information Processing* 23 (1) (2024) 20. doi:10.1007/s11128-023-04216-6.
- [28] J. Ahrens, B. Geveci, C. Law, 36 - ParaView: An End-User Tool for Large-Data Visualization, in: *Visualization Handbook*, Elsevier, 2005, pp. 717–731. doi:10.1016/B978-012387582-2/50038-1.