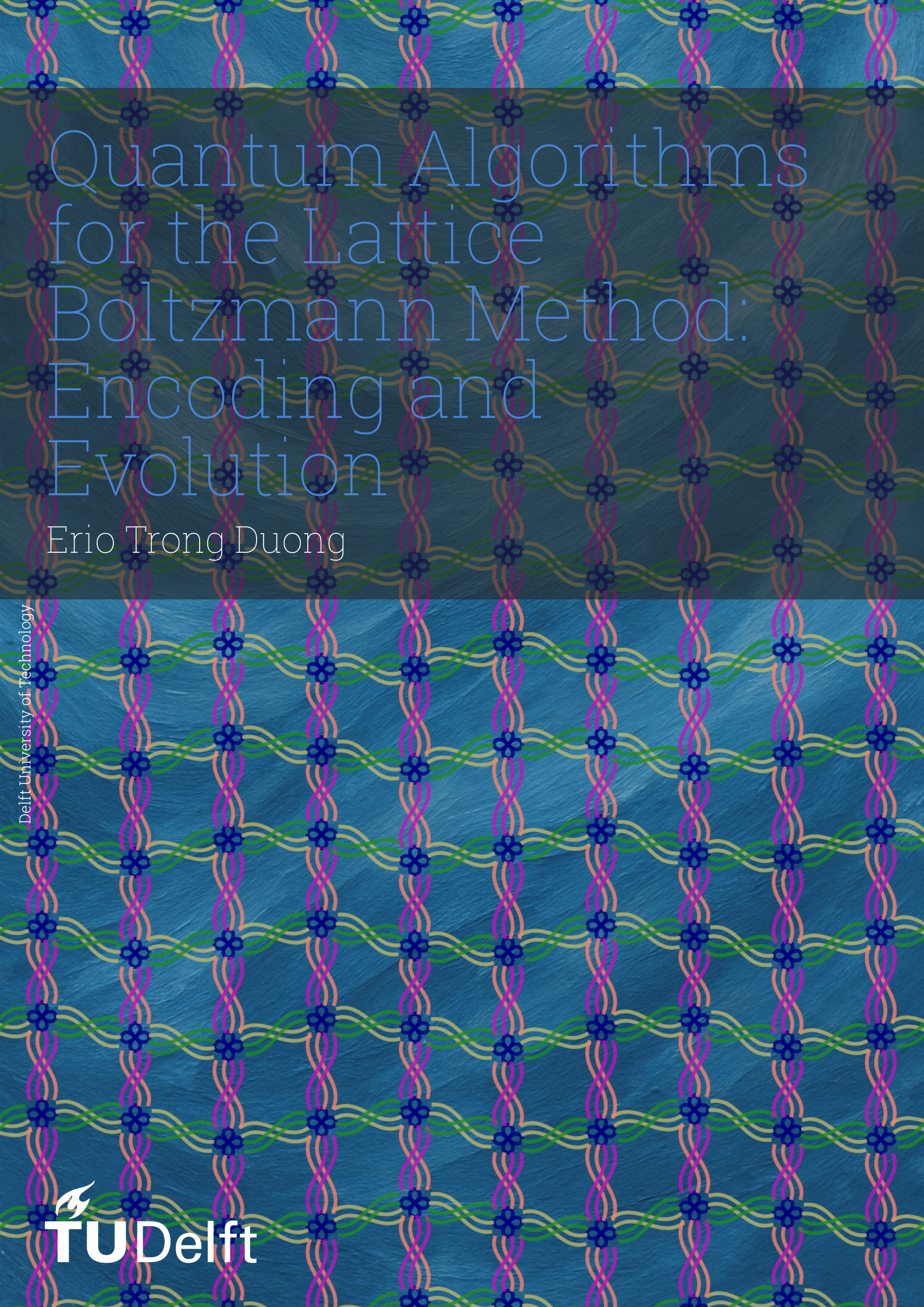




Quantum Algorithms for the Lattice Boltzmann Method: Encoding and Evolution

Erio Trong Duong

Delft University of Technology



TU Delft

Quantum Algorithms for the Lattice Boltzmann Method: Encoding and Evolution

by

Erio Trong Duong

Instructor: Matthias Möller
Project Duration: September, 2024 - June, 2025
Faculty: Institute of Applied Mathematics, TU Delft

Cover: Computer-generated patterns placed on an acrylic paint background created by OpenAI's GPT-Image-1 model
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

This thesis marks the culmination of my Master's journey in Applied Mathematics, and it has been both a deeply challenging and profoundly rewarding experience. I am sincerely grateful to all those who have guided and supported me throughout this process.

First and foremost, I would like to express my heartfelt thanks to my supervisor, Professor Matthias Möller, for introducing me to this fascinating topic at the intersection of quantum computing and fluid dynamics. His vision, encouragement, and unwavering support have been invaluable not only for this thesis but for my broader academic development. I am truly thankful for the freedom he gave me to explore ambitious ideas, while always providing insightful guidance when it was most needed.

A very special thanks goes to Pham Nguyen Tam Minh, my partner in both research and life. Our many in-depth discussions greatly enriched this work. Beyond the academic, his encouragement and care were a constant source of strength throughout this journey.

I am also thankful to my friends and family, who provided support in countless ways. This thesis is as much a product of that environment as it is of solitary thought.

*Erio Trong Duong
Delft, June 2025*

Summary

This thesis explores quantum algorithms for simulating fluid dynamics using the Lattice Boltzmann Method (LBM), with a focus on developing resource-efficient quantum implementations. A central challenge in quantum physical simulation is ensuring that algorithms not only offer computational speedups but also preserve the underlying physical structure of the system. Without this alignment, simulations can become unstable, inaccurate, or theoretically uninformative, especially in fluid dynamics, where conservation laws and symmetries are important. This thesis addresses that challenge by developing quantum versions of the LBM that are constructed with physical interpretability and mathematical consistency at their core.

The main contribution of this thesis is the design and analysis of two quantum encoding strategies for particle distribution functions in LBM: tensor-product encoding and amplitude encoding. Each approach offers different trade-offs between circuit size, precision, and interpretability. The thesis further develops quantum circuits to implement the collision and streaming steps for each of the encoding schemes, with an emphasis on utilizing conservations and symmetries to facilitate the uninterrupted and coherent flow of the multi-round LBM simulations.

To provide a realistic assessment of the methods, the thesis incorporates rigorous error analysis, including the accuracy of each step, post-selection success probabilities, and cumulative errors over multiple simulation steps. Through theoretical analysis and numerical simulations, this thesis demonstrates new physically-informed approaches to quantum LBM, offering scalable and interpretable models for fluid transport on quantum devices.

Contents

Preface	i
Summary	ii
Nomenclature	vi
1 Introduction	1
1.1 Problem Definition	2
1.2 Research Objectives	2
1.3 Thesis Outline	3
2 Background and Fundamentals	5
2.1 Lattice Boltzmann Method	5
2.1.1 Computational Fluid Dynamics and Boltzmann Equation	5
2.1.2 Discrete-velocity Boltzmann equation	7
2.1.3 Lattice Boltzmann equation	9
2.2 Quantum Computation	10
2.2.1 Single-qubit state and Bloch sphere representation	10
2.2.2 Quantum Fourier transform	11
2.2.3 Quantum phase estimation	12
2.2.4 Grover's search algorithm	12
2.2.5 Amplitude amplification	13
2.2.6 Block encoding	13
2.3 Representation Theory	14
2.3.1 Basics	14
2.3.2 Schur's lemma	15
2.3.3 Twirling	17
3 Related work	18
3.1 Quantum Lattice Boltzmann Methods: The First Steps	18
3.2 Dilemma of Encoding Designs	19
4 Methodology	21
4.1 Overview	21
4.2 Tensor-product-encoding-based algorithm	21
4.2.1 Encoding distributions	21
4.2.2 Estimating macroscopic quantities	22
4.2.3 Computing equilibrium distributions	24
4.2.4 Updating post-collision state	29
4.2.5 Propagating distributions	29
4.3 Amplitude-encoding-based algorithm	31
4.3.1 Encoding distributions	31
4.3.2 Feasibility of ideal collision operator for amplitude encoding	32
4.3.3 Symmetry-preserving collision operator	34
4.3.4 Collision ansatz design	36
4.3.5 Unitary streaming	37
4.3.6 Multi-round LBM simulations	38
4.4 Error analysis of quantum Lattice Boltzmann simulations	38
4.4.1 Simulation using tensor-product encoding	38
4.4.2 Simulation using amplitude encoding	39

5	Numerical Simulations and Analysis	41
5.1	Performance of collision operators by optimization on the orthogonal manifold	41
5.2	Performance of symmetry-preserving collision ansatzes	41
5.3	Simulations of the Quantum Lattice Boltzmann Method	42
6	Discussion and Conclusion	44
	References	46
A	Tapered Quantum Phase Estimation	54
B	Approximate streaming	55

List of Figures

2.1	Common DnQm configurations	9
2.2	Bloch sphere representation	11
4.1	Quantum circuit for tensor-product encoding	22
4.2	Quantum circuit for estimating macroscopic quantities	23
4.3	Quantum circuit for the Quantum Fourier Transform	24
4.4	Quantum circuit for the Controlled Phase Shift	25
4.5	Quantum circuits for addition and weighted sum	26
4.6	Quantum circuit for adding a classical number	27
4.7	Quantum circuit for multiplication	27
4.8	Quantum circuit for division	27
4.9	Quantum circuit for creating $\frac{1}{\sqrt{2}}(0\rangle + e^{if_i} 1\rangle)$	29
4.10	Quantum circuit for amplitude encoding	32
4.11	Quantum algorithm for the lattice Boltzmann method using amplitude encoding	38
5.1	Performance and post-selection probability of block-encoded collision operators	41
5.2	Performance of symmetry-preserving collision ansatzes	42
5.3	Simulation of a quantum algorithm for the lattice Boltzmann method	43

Nomenclature

Abbreviations

Abbreviation	Definition
BGK	Bhatnagar-Gross-Krook (collision model)
CFD	Computational Fluid Dynamics
EFTQC	Early Fault-Tolerant Quantum Computing
LBM	Lattice Boltzmann Method
MPS	Matrix Product State
NISQ	Noisy Intermediate-Scale Quantum
QLBM	Quantum Lattice Boltzmann Method
QLGA	Quantum Lattice-Gas Automaton
QFT	Quantum Fourier Transform
QPE	Quantum Phase Estimation
QSP	Quantum Signal Processing
QSVT	Quantum Singular Value Transformation

1

Introduction

The quest to accurately and efficiently simulate fluid dynamics has been a cornerstone of scientific and engineering endeavors, influencing fields as diverse as aerospace engineering, climate modeling, and medical diagnostics. Traditional computational fluid dynamics (CFD) methods, primarily based on solving macroscopic continuum equations such as the Navier-Stokes equations, have achieved significant success in modeling fluid behavior [1]. However, these methods often encounter challenges when dealing with complex boundary conditions [2, 3], multiphase flows [3, 4], and turbulent regimes [5, 6].

The Lattice Boltzmann Method (LBM) [7, 8, 9] has emerged as a promising alternative to conventional CFD approaches. Originating from lattice gas automata [9], LBM operates on a mesoscopic scale, modeling fluids through the evolution of particle distribution functions on a discrete lattice grid using local collision and linear advection. Macroscopic quantities evaluated from moments of the distributions approximate the solution of incompressible Navier-Stokes equations. This approach offers several advantages over traditional methods. The kinetic nature of LBM allows inherent modeling of phase separation and interface dynamics of multiphase flows compared to conventional CFD methods [10, 11, 12]. Another significant advantage of LBM is its ability to handle complex boundary conditions with relative ease. The lattice structure simplifies the representation of irregular geometries, making LBM particularly suitable for simulating flows in intricate domains [13, 14]. Additionally, LBM's localized operations facilitate straightforward parallelization, enhancing computational efficiency and scalability on high-performance computing architectures [15, 16, 17]. However, LBM is not without limitations. Accurately simulating high-speed compressible flows with the standard LBM method remains challenging due to the method's foundation in incompressible or low Mach number assumptions [18, 19]. Implementing precise boundary conditions, especially on curved or moving surfaces, can also be complex and may introduce numerical artifacts if not handled carefully [18]. Moreover, LBM can be more memory-intensive than traditional CFD methods, particularly when modeling three-dimensional or multiphase flows, which necessitates careful consideration of computational resources [20].

Quantum computing leverages the principles of superposition and entanglement to process information in fundamentally new ways, offering the potential for remarkable speedups in solving certain classes of problems. Notable examples include Grover's unstructured search algorithm, which provides a quadratic speedup [21], the quantum Fourier transform (QFT), integral to Shor's algorithm, offering exponential speedup in factoring large integers [22], and Hamiltonian simulation techniques that achieve varying degrees of speedup over classical counterparts [23, 24]. Recently, these major branches of quantum algorithms have been unified under the emerging framework of Quantum Singular Value Transformation (QSVT) [25], which enables polynomial transformations of singular values in linear operators, thereby encompassing a wide array of quantum algorithmic applications. These algorithms and their unifying framework serve as key *subroutines* across various applications in quantum computing, including integer factorization [22], quantum phase estimation [26, 27], quantum walks [28, 29], solving linear systems of equations [30, 31], and ground state preparation and energy estimation [32, 33, 34, 35]. However, these theoretical advantages are contingent upon the realization of *fully fault-*

tolerant quantum computers, which may still require several years to materialize. Over the past decade, substantial research has focused on *Noisy Intermediate-Scale Quantum* (NISQ) algorithms. Characterized by limited qubit counts and error-prone quantum gates, NISQ algorithms often utilize low-depth variational ansatzes and parameter optimization methods on classical computers [36, 37]. However, NISQ devices and algorithms currently offer no provable advantage over classical computing and are increasingly considered impractical for solving problems at a utility scale [38, 39, 40]. Consequently, research interest is shifting toward *Early Fault-Tolerant Quantum Computing* (EFTQC), driven by recent progress in quantum error correction and fault tolerance [41]. Although the specific criteria defining this new era remain under discussion, EFTQC research often employs certain error-corrected sub-routines alongside classical data processing. Several algorithms in this category have demonstrated potential advantages, such as polynomial-scale improvements, in simulated noisy environments [42, 43, 44].

In the context of LBM simulation, quantum computing holds the promise to make efficient simulations as it enables encoding of a large space domain using exponentially few qubits while applying space-localized operators simultaneously across the entire domain. The confluence of LBM and quantum computing is a burgeoning area of research [45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57], with studies indicating that quantum algorithms can be tailored to execute LBM simulations more efficiently. However, none of the existing quantum implementations of the standard LBM achieve a fully coherent evolution. Some encoding strategies enable unitary streaming but are incompatible with implementing non-linear collisions coherently, while others fail to support a unitary streaming step altogether. Furthermore, many approaches rely on measurement and state reinitialization at every time step, leading to excessive resource costs and sub-unity success probabilities, ultimately eliminating any potential quantum advantage.

1.1. Problem Definition

The central challenge addressed in this thesis is the design of a quantum algorithm that enables a purely quantum implementation of the Lattice Boltzmann Method (LBM), one that does not rely on frequent measurements or classical post-processing. Classical LBM operates by evolving discrete distribution functions over a spatial lattice via local collision and streaming steps. Translating this algorithm to a quantum domain involves a fundamental difficulty: while classical distributions are stored as real-valued functions over lattice sites and discrete velocities, quantum states are unit vectors in a Hilbert space. The question then becomes: How can the classical distribution functions $f_i(\mathbf{x}, t)$, which encode the population of particles at site $\mathbf{x}$ with velocity c_i , be faithfully and efficiently represented in a quantum system?

A viable encoding must satisfy multiple criteria. First, it should enable the extraction of macroscopic quantities (e.g., density, velocity) via quantum measurements or state overlaps. Second, it must allow the unitary evolution of the state to reproduce, in the hydrodynamic limit, the behavior of LBM, including correct viscosity and conservation laws. Third, the encoding should be compatible with quantum algorithmic primitives such as Quantum Singular Value Transformation (QSVT), Hamiltonian simulation, or quantum arithmetics, depending on the strategy chosen for evolution.

This thesis aims to explore encodings of particle distribution functions into quantum states that satisfy these physical and computational constraints, with the ultimate goal of formulating a quantum LBM (QLBM) algorithm that is not only implementable on early fault-tolerant devices but also scalable toward quantum advantage.

1.2. Research Objectives

To tackle the overarching goal, this thesis focuses on the following main research question:

- How can classical Lattice Boltzmann distribution functions be efficiently encoded and evolved on a quantum computer in a way that preserves the structure and utility of LBM in simulating fluid dynamics?

This question can be broken down into several detailed sub-questions:

A. Encoding Layer

- (a) Choice of encoding:
 - How do standard encodings (e.g., amplitude, basis, etc.) perform when applied to LBM distribution functions?
 - Can we encode multiple distribution functions f_i across a lattice into a single quantum state efficiently?
 - How do these encodings affect the complexity of extracting macroscopic observables?
- (b) Preserving physical properties:
 - Can the quantum encoding retain the local and parallel nature of the LBM update rules?
 - What are the trade-offs when coherent quantum operators cannot fully support the conservation laws.
- B. Collision and Streaming Steps
 - (a) Unitary Realizations of Collision Operators
 - Can common LBM collision operators (e.g., BGK model) be implemented via quantum unitaries or block-encodings?
 - What is the cost (in terms of circuit depth and ancilla count) of the implemented collision operators?
 - (b) Streaming as Shift Operator or Quantum Walk
 - What is the role of quantum shift operators and how do they scale with problem size?
 - Is it feasible to implement the streaming step as a quantum walk in the space-velocity domain?
- C. Algorithmic Framework and Performance
 - (a) Simulation Pipeline for QLBM and Emergent Behavior
 - Under realistic implementation and computation conditions, which approaches make the most performing quantum algorithm for multi-round LBM simulations?
 - How do errors in the quantum encoding or evolution propagate through the macroscopic observables?
 - (b) Benchmark of Multi-round Performance
 - How do initial conditions affect the convergence and accuracy of the QLBM simulation in the long run?
 - How good is the output of our QLBM algorithm compared to that of other classical and quantum LBM methods?

1.3. Thesis Outline

This thesis is organized as follows:

- **Chapter 1: Introduction**
Introduces the motivation for combining LBM and quantum computing, highlights classical limitations and quantum opportunities, and frames the central problem and questions.
- **Chapter 2: Background and Fundamentals**
Reviews the classical LBM in detail, including its derivation and discretization, followed by an overview of key quantum computing concepts, with emphasis on common ingredients in popular quantum algorithms.
- **Chapter 3: Related Work**
Surveys existing efforts to simulate fluid dynamics on quantum computers, including quantum lattice gas automata and quantum encodings, together with the respective quantum-encoded implementations of collision and streaming. Our work provide new perspectives on the quantum

encodings and follow-up quantum operations to approach a coherent quantum algorithm for the lattice Boltzmann method.

- **Chapter 4: Methodology**

Outlines the overall workflow of this thesis, from encoding design to LBM simulation flow. Presents the theoretical framework for encoding distribution functions into quantum states, and develops the mathematical tools for implementing quantum collision and streaming steps. This chapter also discusses the theoretical complexity of individual components and the full LBM simulation pipeline using our quantum algorithms.

- **Chapter 5: Numerical Simulation and Analysis**

Presents the results of analytical derivations and numerical simulations. Evaluates the performance of proposed quantum encodings and operators in terms of accuracy, resource requirements, and scalability. Highlights trade-offs between different design choices and benchmarks emergent macroscopic behavior.

- **Chapter 6: Discussion and Conclusion**

Summarizes the main contributions of the thesis and provides answers to the research questions. Reflects on the limitations of the current approach and outlines possible directions for future research, including possible improvements to the methods in this work, as well as actual quantum implementations.

2

Background and Fundamentals

2.1. Lattice Boltzmann Method

We give a brief overview of fluid dynamics and the underlying idea of the lattice Boltzmann method. The interested reader is referred to standard treatments in fluid dynamics [58, 1] and lattice Boltzmann theory [7, 8, 9].

We utilize a mix of vector notation and (Einstein) index notation throughout the thesis. While the former style is prioritized for its brevity, we might elaborate complex expressions using the latter.

2.1.1. Computational Fluid Dynamics and Boltzmann Equation

In traditional fluid dynamics simulation, one typically adopts a macroscopic fluid model governed by the Navier–Stokes equations that describe basic conservation laws (in density, momentum, and energy) applied to a fluid.

The conservation of density, given by the continuity equation below, expresses the fact that the density ρ is locally conserved, with changes in ρ arising only through the advective transport of fluid moving at velocity $\mathbf{u}$,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (\text{general}) \quad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (\text{incompressible}^1) \quad (2.2)$$

The momentum equation represents the balance of forces acting on the fluid encompassing the effects of pressure p , viscous stress tensor $\boldsymbol{\sigma}$, and body force $\mathbf{F}$,

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \mathbf{F} \quad (\text{general}) \quad (2.3)$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \eta \nabla^2 \mathbf{u} + \mathbf{F} \quad (\text{incompressible}) \quad (2.4)$$

where $\boldsymbol{\sigma} = \eta (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + (\eta_B - \frac{2}{3}\eta) (\nabla \cdot \mathbf{u}) \mathbf{I}$. The coefficients η and η_B are called shear viscosity and bulk viscosity, respectively.

The energy equation accounts for the conservation of energy (including both kinetic and internal energies) within the fluid². For many incompressible flow problems, the energy equation is decoupled from the system since temperature variations and viscous heating are often secondary effects. In practice for incompressible flows, the focus is usually on solving the continuity and momentum equations, while the energy equation is only addressed when heat transfer or thermal effects are significant.

¹A fluid incompressible iff $\rho = \text{const}$ or, equivalently, $\nabla \cdot \mathbf{u} = 0$ as given by the continuity equation.

²The general form reads $\frac{\partial(\rho E)}{\partial t} + \nabla \cdot [(\rho E + p)\mathbf{u}] = \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{u}) - \nabla \cdot \mathbf{q} + \mathbf{F} \cdot \mathbf{u}$, where $E = e + \frac{1}{2}|\mathbf{u}|^2$ is the total energy and $\mathbf{q}$ the heat flux. See [59] for a comprehensive treatment of compressible fluid phenomena and applications.

To implement these equations on a computer, one discretizes the spatial and temporal derivatives, arriving at a multitude of possible numerical schemes. A fundamental requirement of such schemes is that they must preserve mass, momentum (and energy for compressible fluids) to a desired accuracy. The development of robust discretization approaches remains a core focus within the field of computational fluid dynamics (CFD).

A notable departure from continuum-based CFD methods emerged in the late 1980s with *lattice gas automata*. Here, one posits discrete particle-like entities that occupy a lattice of discrete nodes. Each node may hold several of these discrete “particles”, which evolve by collision and propagation steps. By construction, local collisions obey strict conservation of mass and momentum, thereby reproducing fluid-like behavior. However, lattice gas automata typically exhibit high statistical noise and may introduce spurious contributions to the macroscopic equations, limiting their effectiveness [9].

Subsequent developments replaced the discrete particle count with smoothly varying *distribution functions*, retaining both the local conservation properties and a discretized velocity space. The new method, called *lattice Boltzmann method* (LBM), has achieved remarkable success in handling complex flows in practical applications. LBM presents a numerical approximate solution to the Boltzmann equation, which is a kinetic equation describing the statistical behavior of a thermodynamic system. The following paragraphs provide a simplified presentation of the method’s foundations, which is adapted from Krüger’s textbook in LBM [7].

The Boltzmann equation can characterize the change of physical quantities like particle density, mean momentum, and energy in the transport of fluid. Let f be the particle distribution at position $\mathbf{x}$, particle velocity $\boldsymbol{\xi}$ and time t . The change in time $\frac{df}{dt}$ is attributed to the external force $\mathbf{F}(\mathbf{x}, \boldsymbol{\xi}, t)$, diffusion of particles, and forces acting on particles during collisions via the differential equation

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f + \mathbf{F} \cdot \nabla_{\boldsymbol{\xi}} f = \left(\frac{\partial f}{\partial t} \right)_{\text{coll}} =: \Omega(f) \quad (2.5)$$

The exact differential term due to collision is extremely complex, even for two-body collision. For solving Boltzmann equation, many attempts to model the collision term have been made, among which the simplest and most popular BGK model, was proposed by Bhatnagar, Gross, and Krook [60]. The authors argue that the collision term effectively brings the distribution function closer to the equilibrium distribution. The equilibrium distribution, which is also called Maxwell–Boltzmann distribution, is a result of the kinetic theory of gases that quantifies the energy distribution of particles in their *thermodynamic equilibrium*. The distribution can be derived from the principle that it maximizes the system’s entropy constrained to a fixed average energy. (See Chapter 4 in [61].) Roughly speaking, minimizing the negative entropy $\mathcal{H} = \int f \ln(f) d^3 \boldsymbol{\xi}$ with Lagrange multipliers yields the expression

$$f^{\text{eq}}(\boldsymbol{\xi}; \rho, \mathbf{u}, \theta) = \frac{\rho}{(2\pi\theta)^{3/2}} e^{-|\boldsymbol{\xi} - \mathbf{u}|^2 / (2\theta)}, \quad (2.6)$$

which is determined by three macroscopic quantities of the fluid, namely density ρ , mean velocity $\mathbf{u}$, and total energy E .

$$\begin{aligned} \rho(\mathbf{x}, t) &= \int f(\mathbf{x}, \boldsymbol{\xi}, t) d^3 \boldsymbol{\xi} \\ \rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) &= \int \boldsymbol{\xi} f(\mathbf{x}, \boldsymbol{\xi}, t) d^3 \boldsymbol{\xi} \\ \rho(\mathbf{x}, t) E(\mathbf{x}, t) &= \int \frac{1}{2} |\boldsymbol{\xi}|^2 f(\mathbf{x}, \boldsymbol{\xi}, t) d^3 \boldsymbol{\xi} \end{aligned} \quad (2.7)$$

The appearance of temperature $\theta = RT$ in the equilibrium distribution is due to the contribution of $RT/2$ to the internal energy density for each degree of freedom, i.e. $\rho E = \frac{1}{2} \rho |\mathbf{u}|^2 + \frac{3}{2} RT$ in the case of mono-atomic gas.

In fact, the central moments of the equilibrium distribution, derived using Gaussian integrals³, give rise to the macroscopic quantities.

$$\begin{aligned}
\rho &= \int f^{\text{eq}} d^3\xi \\
0 &= \int (\xi - \mathbf{u}) f^{\text{eq}} d^3\xi \\
\rho RT \mathbf{I} &= \int (\xi - \mathbf{u}) \otimes (\xi - \mathbf{u}) f^{\text{eq}} d^3\xi \quad (\mathbf{I} \equiv \delta_{\alpha\beta}) \\
0 &= \int (\xi - \mathbf{u})^{\otimes 3} f^{\text{eq}} d^3\xi \\
5\rho(RT)^2 \mathbf{I}^{(4)} &= \int (\xi - \mathbf{u})^{\otimes 4} f^{\text{eq}} d^3\xi \quad (\mathbf{I}^{(4)} \equiv I_{\alpha\beta\gamma\delta}^{(4)} = \delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})
\end{aligned} \tag{2.8}$$

In the BGK model, the distribution f will relax to f^{eq} over a relaxation time τ , hence

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f + \mathbf{F} \cdot \nabla_{\xi} f = \Omega_{\text{BGK}}(f) = -\frac{f - f^{\text{eq}}}{\tau} \tag{2.9}$$

This particle-based description agrees with continuum hydrodynamic equations on the macroscopic scale. In fact, the Navier-Stokes equations can be recovered from either the Boltzmann equation or the lattice Boltzmann equation (introduced later) through a framework called Chapman-Enskog theory (Chapter 3 in [61]). We refer interested readers to the modern approach in [62]. An immediate consequence of the Chapman-Enskog theory in the context of Boltzmann equation (2.5) is that the collision operator must preserve the mean density, momentum, and energy of the particles. The BGK operator Ω_{BGK} , as a valid collision operator, must satisfy:

$$\begin{aligned}
\int f^{\text{eq}} d^3\xi &= \int f d^3\xi &= \rho \\
\int \xi f^{\text{eq}} d^3\xi &= \int \xi f d^3\xi &= \rho \mathbf{u} \\
\int |\xi|^2 f^{\text{eq}} d^3\xi &= \int |\xi|^2 f d^3\xi &= 2\rho E
\end{aligned} \tag{2.10}$$

2.1.2. Discrete-velocity Boltzmann equation

The Boltzmann equation deals with a distribution function $f(\mathbf{x}, \xi, t)$ defined over a continuous velocity space of ξ . Numerically, this is intractable. Hence, one replaces the continuum of possible velocities with a finite, carefully chosen set $\{\xi_i\}$. Each velocity vector ξ_i then carries its own distribution, $f_i(\mathbf{x}, t) \equiv f(\mathbf{x}, \xi_i, t)$. Suitable selection of ξ_i and their weights w_i ensures that the resulting discrete model reproduces the low-order velocity moments (e.g., density, momentum, energy) governing fluid behavior at macroscopic scales.

A pivotal step is specifying the *equilibrium* distribution. In continuous kinetic theory, the Maxwell-Boltzmann form (2.6) is Gaussian. To approximate it in a discrete-velocity setting, one often uses a truncation of the Hermite expansion

$$f(\xi) = w(\xi) \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}_n \mathbf{H}_n(\xi), \quad \mathbf{a}_n = \int f(\xi) \mathbf{H}_n(\xi) d^3\xi \tag{2.11}$$

in which the Hermite polynomials (probabilist's definition) in d -dimensions generated by

$$\mathbf{H}_n(\mathbf{x}) = \frac{(-1)^n}{w(\mathbf{x})} \nabla^{(n)} w(\mathbf{x}), \quad w(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} e^{-|\mathbf{x}|^2/2} \tag{2.12}$$

³It's simpler to compute the integrals using index notation $\int (\xi_{\alpha_1} - u_{\alpha_1}) \dots (\xi_{\alpha_n} - u_{\alpha_n}) f^{\text{eq}} d^3\xi$.

are orthogonal with respect to $w(\mathbf{x})$. The first Hermite polynomials are $\mathbf{H}_0(\mathbf{x}) = 1$, $\mathbf{H}_1(\mathbf{x}) = \mathbf{x}$, and $\mathbf{H}_2(\mathbf{x}) = \mathbf{x} \otimes \mathbf{x} - \mathbf{I}$. The first coefficients for the Hermite expansion of the equilibrium distribution f^{eq} are indeed related to macroscopic quantities [7]

$$\mathbf{a}_0^{\text{eq}} = \rho, \quad \mathbf{a}_1^{\text{eq}} = \rho \mathbf{u}, \quad \mathbf{a}_2^{\text{eq}} = \rho(\mathbf{u} \otimes \mathbf{u} + (\theta - 1)\mathbf{I}) \quad (2.13)$$

As a result, truncating the series at second order fulfills the conservation laws and represents the macroscopic equations. Yet, the inclusion of higher order terms can improve the numerical stability and accuracy. The truncated equilibrium distribution at second order is given by

$$\begin{aligned} f^{\text{eq}}(\rho, \mathbf{u}, \boldsymbol{\xi}) &\approx w(\boldsymbol{\xi})\rho \left[1 + \boldsymbol{\xi} \cdot \mathbf{u} + \frac{1}{2}(\mathbf{u} \otimes \mathbf{u} + (\theta - 1)\mathbf{I}) : (\boldsymbol{\xi} \otimes \boldsymbol{\xi} - \mathbf{I}) \right] \\ &= w(\boldsymbol{\xi})\rho \left[1 + \boldsymbol{\xi} \cdot \mathbf{u} + \frac{1}{2}((\boldsymbol{\xi} \cdot \mathbf{u})^2 - |\mathbf{u}|^2 + (\theta - 1)|\boldsymbol{\xi}|^2 - 3(\theta - 1)) \right] \\ &= w(\boldsymbol{\xi})\rho Q(\boldsymbol{\xi}), \quad Q(\boldsymbol{\xi}) := \left[1 + \boldsymbol{\xi} \cdot \mathbf{u} + \frac{1}{2}((\boldsymbol{\xi} \cdot \mathbf{u})^2 - |\mathbf{u}|^2 + (\theta - 1)|\boldsymbol{\xi}|^2 - 3(\theta - 1)) \right] \end{aligned} \quad (2.14)$$

Discrete velocities $\{\boldsymbol{\xi}_i\}$ and corresponding weights $\{w_i\}$ are selected to compute the leading moments of the truncated equilibrium distribution that corresponds to macroscopic quantities. The moments have the form $\rho \int w(\boldsymbol{\xi})Q(\boldsymbol{\xi})P(\boldsymbol{\xi}) d^3\boldsymbol{\xi}$ where $P(\boldsymbol{\xi})$ is a polynomial with degree ≤ 2 . This integral can be evaluated exactly by summing the integrand over a finite set of velocities $\boldsymbol{\xi}_i$ if and only if $\boldsymbol{\xi}_i$ are *abscisses* of a Gauss-Hermite quadrature of a precision degree $\geq \max_P \deg(Q(\boldsymbol{\xi})P(\boldsymbol{\xi})) = 4$. Namely the discrete velocity set must at least form a 4th-order accurate Hermite quadrature. Such a set of velocities often carry an unhandy factor of $\sqrt{3}$. In practice, one uses normalized velocities $\mathbf{c}_i := c_s \boldsymbol{\xi}_i$, where $c_s = \frac{1}{\sqrt{3}}$ is called the *speed of sound* in lattice Boltzmann theory.

Beyond conserving mass and momentum, a critical requirement is to preserve rotational isotropy of the lattice. The specific criterion for a lattice to be “sufficiently isotropic” depends on the underlying physics. In most applications of the LBM to the Navier–Stokes equations, one demands isotropy in all moments of the weight up to fifth order (noting that fifth-order integration via Hermite polynomials is employed). This requirement then leads to the following conditions (Chapter 5 in [9])

$$\begin{aligned} \sum_i w_i &= 1; \\ \sum_i w_i \mathbf{c}_i &= \mathbf{0}; \\ \sum_i w_i \mathbf{c}_i \otimes \mathbf{c}_i &= c_s^2 \mathbf{I}; \\ \sum_i w_i \mathbf{c}_i^{\otimes 3} &= \mathbf{0}^{(3)}; \\ \sum_i w_i \mathbf{c}_i^{\otimes 4} &= c_s^4 \mathbf{I}^{(4)}; \\ \sum_i w_i \mathbf{c}_i^{\otimes 5} &= \mathbf{0}^{(5)} \end{aligned} \quad (2.15)$$

In the LBM community, models are commonly designated by the notation $DnQm$, where n is the spatial dimension, and m is the number of discrete velocity directions. For instance, D2Q9 is a two-dimensional LB scheme with nine discrete velocities, while D3Q15 is a three-dimensional LB scheme comprising fifteen discrete velocities. Figure 2.1 shows the velocity set in three $DnQm$ configurations. The actual values of discrete velocities and their weights can be found in LBM handbooks [7, 8].

The discrete set $\{f_i^{\text{eq}} = f^{\text{eq}}(\mathbf{c}_i)\}$ satisfies the same conservation laws for the first three moments as the continuous Gaussian equilibrium distribution f^{eq} . Under isothermal assumption $\theta = 1$, the final expression of f_i^{eq} reads

$$f_i^{\text{eq}} = w_i \rho \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right] \quad (2.16)$$

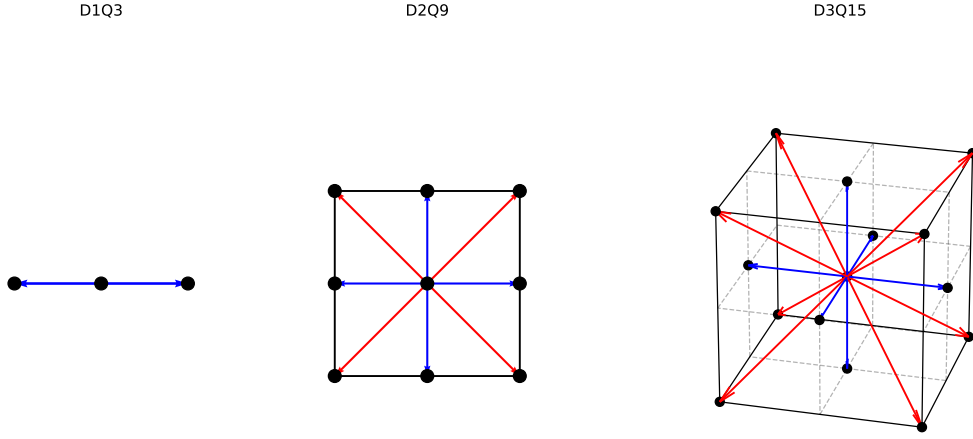


Figure 2.1: Three common DnQm configurations. In each of the settings, the discrete velocity set comprises the colored vectors and the zero vector.

The first moments of the equilibrium distribution in the discretized velocity space have the following explicit expressions [7].

$$\begin{aligned}
 \Pi_0^{\text{eq}} &= \sum_i f_i^{\text{eq}} = \rho \\
 \Pi_1^{\text{eq}} &= \sum_i \mathbf{c}_i f_i^{\text{eq}} = \rho \mathbf{u} \\
 \Pi_2^{\text{eq}} &= \sum_i \mathbf{c}_i \mathbf{c}_i f_i^{\text{eq}} = \rho (\mathbf{u} \mathbf{u} + c_s^2 \mathbf{I}) \\
 \Pi_{3,\alpha\beta\gamma}^{\text{eq}} &= \sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} f_i^{\text{eq}} = \rho c_s^2 (u_\alpha \delta_{\beta\gamma} + u_\beta \delta_{\alpha\gamma} + u_\gamma \delta_{\alpha\beta})
 \end{aligned} \tag{2.17}$$

2.1.3. Lattice Boltzmann equation

The discrete-velocity distributions $f_i(\mathbf{x}, t)$, $i = 1, 2, \dots, N_v$ (N_v is the size of the velocity set) are still continuous functions in space and time. Time-discretizing the Boltzmann equation for each f_i leads to the *lattice Boltzmann equation* (LBE). For the BGK collision model, the LBE is given by

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{\tau} [f_i(\mathbf{x}, t) - f_i^{(\text{eq})}(\rho(\mathbf{x}, t), \mathbf{u}(\mathbf{x}, t))] + \Delta t F_i(\mathbf{x}, t), \quad i = 1, \dots, N_v \tag{2.18}$$

From these equations, the distributions f_i can be updated after every time step Δt . The change depends on the BGK collision term and the forcing term F_i , which can be modelled in many ways as a function of the velocity $\mathbf{c}_i$ and the external force $\mathbf{F}$ [63, 64, 65, 66]. For simplicity, this thesis only addresses the force-free scenario, i.e., $\mathbf{F} = \mathbf{0}$ and $F_i = 0$. Furthermore, the equation can be simulated on a lattice or grid in the space domain. The idea is that the populations f_i always reach neighboring lattice nodes after every time step. Assuming a uniform grid with unit distance Δx , we choose Δx such that components $c_{i,\alpha}$ of $\mathbf{c}_i = (c_{i,x}, c_{i,y}, c_{i,z})$ are integer multiples of $\frac{\Delta x}{\Delta t}$. This is possible as the velocity $\mathbf{c}_i$ have integral components (even $c_{i,\alpha} \in \{-1, 0, +1\}$ in many common configurations). In the unit time Δt , the update rule for a simulation of the LBE becomes

⁴This expression uses index notation for clarity. In vector notation, $\Pi_3^{\text{eq}} = \rho c_s^2 (\mathbf{u} \otimes \mathbf{I} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{u} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{I} \otimes \mathbf{u})_{\text{sym}}$. The subscript “sym” indicates taking all $3! = 6$ permutation of tensor indices α, β, γ , i.e. $(T_{\text{sym}})_{\alpha\beta\gamma} = \frac{1}{6} \sum_{\sigma \in S_3} T_{\sigma(\alpha), \sigma(\beta), \sigma(\gamma)}$.

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) = (1 - \omega)f_i(\mathbf{x}, t) + \omega f_i^{\text{eq}}(\mathbf{x}, t) \quad (2.19)$$

where $\omega = \frac{\Delta t}{\tau}$ is called the *relaxation rate*. The time evolution in a lattice Boltzmann simulation is conveniently divided into two separate sub-steps: collision and streaming.

$$\begin{aligned} f_i^*(\mathbf{x}, t) &= (1 - \omega)f_i(\mathbf{x}, t) + \omega f_i^{\text{eq}}(\mathbf{x}, t) && \text{(collision)} \\ f_i(\mathbf{x} + \mathbf{c}_i, t + 1) &= f_i^*(\mathbf{x}, t) && \text{(streaming)} \end{aligned} \quad (2.20)$$

The explicit discretisation in velocities, space, and time allows for a simple setup of complex boundary conditions. Also, the two sub-step protocol allows for effective parallelisation, in which the collision step can be performed simultaneously over the entire lattice due to its localized nature, and the streaming step is just a memory update.

2.2. Quantum Computation

Quantum computation uses the principles of quantum mechanics, most notably superposition and entanglement, to encode and manipulate information in ways that differ fundamentally from classical methods. Instead of bits, which can be 0 or 1, quantum computers use qubits that can exist in “superpositions” of states, which reads $\alpha|0\rangle + \beta|1\rangle$ in bra-ket notation. This expanded state space can, in principle, allow certain computations to be carried out with exceptionally fewer resources than classical machines.

Modern formulations of quantum computing often follow the “gate model”, in which qubits evolve under a sequence of unitary gates analogous to Boolean logic gates. This gate-based model is often visualized by a quantum circuit, which represent the unitary gates by the order they act on the quantum state. Two landmark results that helped popularize the field are Shor’s quantum algorithm for factoring large integers, potentially undermining many current cryptosystems [22], and Grover’s search algorithm, which achieves a quadratic speedup for unstructured search problems [21]. More broadly, quantum algorithms have been proposed for a variety of tasks in simulation [23, 24, 67, 68], machine learning [69, 70], and data analysis [71, 72], including solving certain linear systems of equations [30, 31].

From an applied-mathematics perspective, quantum computing is especially intriguing because it recasts computational tasks such as those arising in large-scale simulations of physical systems into a quantum framework. Many fluid-dynamics solvers rely heavily on linear algebra operations, and part of the appeal of quantum methods is the possibility (under ideal conditions) of exponential acceleration in handling very large system sizes. Of course, achieving such speedups demands not just suitable hardware but also carefully designed quantum algorithms that mitigate error rates and match well with the problem’s structure.

While today’s quantum devices remain limited (the so-called “Noisy Intermediate-Scale Quantum” or NISQ era), research on quantum algorithms for numerical methods, including those relevant to the Lattice Boltzmann Method, points towards long-term prospects in accelerating fluid simulations and other heavy computational tasks in scientific computing. For deeper treatments of the foundations of quantum computing and well-known quantum algorithms, see Nielsen & Chuang’s standard textbook in quantum computation [73]. We present a number of concepts and techniques that often serve as building blocks in advanced quantum algorithms but are not always covered in basic introductions to quantum computing. We decide to focus on their mathematical aspect and defer the respective implementation on quantum circuits to later chapters.

2.2.1. Single-qubit state and Bloch sphere representation

A single-qubit pure state may always be written in the form

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle, \quad (2.21)$$

where the polar angle $0 \leq \theta \leq \pi$ and the azimuthal angle $0 \leq \phi < 2\pi$. This representation shows how the relative amplitude and phase between $|0\rangle$ and $|1\rangle$ completely specify the qubit’s state. The

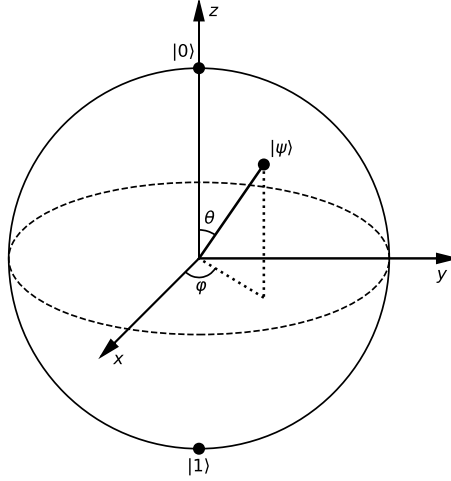


Figure 2.2: Bloch sphere representation for $|\psi\rangle = \cos(\theta/2) |0\rangle + e^{i\phi} \sin(\theta/2) |1\rangle$.

conventional choice of θ and ϕ removes any physically irrelevant global phase and ensures that only the two degrees of freedom unique to a single qubit are retained.

A geometric interpretation places every valid single-qubit state on the Bloch sphere, a unit sphere whose coordinates correspond to the qubit's amplitudes in a real, three-dimensional space. Here, θ and ϕ define the Bloch vector

$$\mathbf{n}_\psi = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta) \quad (2.22)$$

As shown in [Figure 2.2](#), a qubit state thus corresponds to a single point on this sphere, with its latitude and longitude set by θ and ϕ . Rotations around the Bloch sphere can be carried out by unitary gates, reflecting how quantum gates transform the amplitudes and phases of the qubit's state.

2.2.2. Quantum Fourier transform

The quantum Fourier transform (QFT) is a foundational block in quantum algorithms, first introduced as part of Shor's algorithm [\[22\]](#). It acts as the quantum analogue of the discrete Fourier transform and is defined on an n -qubit basis state $|x\rangle$ by the operation

$$\text{QFT} : |x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(\frac{2\pi i x k}{2^n}\right) |k\rangle. \quad (2.23)$$

We label the qubits so that an integer $x = x_{n-1}x_{n-2} \dots x_0$ has its least significant bit (LSB) on the right. In that notation, the action of the QFT can be written as a tensor product

$$\text{QFT} |x_{n-1}x_{n-2} \dots x_0\rangle = \bigotimes_{m=0}^{n-1} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i \phi_m(x)} |1\rangle \right) \quad (2.24)$$

where the phase $\phi_m(x)$ depends on the bits $x_0, \dots, x_m$. In binary-fraction form,

$$\phi_m(x) = 0.x_m x_{m-1} \dots x_0 = x_m 2^{-1} + x_{m-1} 2^{-2} + \dots + x_0 2^{-(m+1)} \quad (2.25)$$

Unlike a classical Fourier transform that processes 2^n points in roughly $O(2^n \log(2^n))$ time, implementing QFT in the tensor form (2.24) requires only $O(n^2)$ quantum gates (or $O(n \log(n))$ for an approximation), giving it a central role in highly efficient quantum algorithms. A prominent application is in extracting periodic information for problems like integer factoring, most famously in Shor's algorithm [22].

A more specialized usage of the QFT appears in the design of Draper's adder [74], a quantum circuit to perform addition of binary numbers. When two numbers $|x\rangle$ and $|y\rangle$ are encoded in separate n -qubit registers, there is a way to add x to y by transforming $|y\rangle$ into the Fourier domain, applying controlled phase operations conditioned on $|x\rangle$, and then inverting the QFT. Because addition by x in the computational basis corresponds to multiplication by phase factors $\exp(2\pi i k x / 2^n)$ in the Fourier basis, the net effect is to shift $|y\rangle$ to $|y + x\rangle$.

2.2.3. Quantum phase estimation

Quantum phase estimation (QPE) is an algorithm that uses the quantum Fourier transform (QFT) to extract the eigenvalues of a given unitary operator U . In more concrete terms, if $|\psi\rangle$ is an eigenstate of U with eigenvalue $e^{2\pi i \theta}$, $0 \leq \theta < 1$, then QPE can approximate θ to a chosen number of bits of precision. This ability to read off the eigenphase of a quantum operator forms the foundation of many more complex quantum algorithms, including quantum walks [29] and certain linear-systems solvers [30].

The standard approach to QPE starts by preparing an ancillary register of n qubits in the state $|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle$ and a system register in the eigenstate $|\psi\rangle$. Applying the controlled- U gate

$$U_C = \sum_{k=0}^{2^n-1} |k\rangle \langle k| \otimes U^k \quad (2.26)$$

on the system evolves the state into $\frac{1}{\sqrt{2^n}} \left(\sum_{k=0}^{2^n-1} \exp(2\pi i k \theta) |k\rangle \right) |\psi\rangle$. The inverse QFT on the ancilla then coherently recovers the binary fraction of θ , an n -bit representation of θ . With probability at least $\frac{4}{\pi^2}$, the value obtained is the nearest n -bit representation of θ [75]. A recent work proposes a method to use additional $O(\log(\log(1/\varepsilon)))$ ancillary qubits to increase the probability to $1 - \varepsilon$ [27]. The performance of the original QPE algorithm can be formally described as follows.

Theorem 2.1: Quantum phase estimation [26]

Let U be a unitary operator with eigenvectors $|\psi_j\rangle$ and eigenvalues $e^{2\pi i \theta_j}$ for $\theta_j \in [0, 1]$, $j = 1, \dots, N$. For a precision parameter $\varepsilon > 0$, there exists a quantum algorithm that runs in time $O\left(\frac{\log(N)}{\varepsilon}\right)$ and with probability $1 - \frac{1}{\text{poly}(N)}$ maps a state $|\psi\rangle = \sum_j \alpha_j |\psi_j\rangle$ to the state $|\psi\rangle = \sum_j \alpha_j |\psi_j\rangle |\bar{\theta}_j\rangle$ such that $|\bar{\theta}_j - \theta_j| < \varepsilon$ for all $j = 1, \dots, N$.

2.2.4. Grover's search algorithm

Grover's search algorithm addresses the problem of locating a marked element in an unstructured database of size N in approximately $O(\sqrt{N})$ operations, improving on the classical $O(N)$ approach. The algorithm starts with the uniform superposition $|s\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle$, which equally weights all N possible database entries. An oracle unitary O marks a special solution $|x_0\rangle$ by performing

$$O : |x\rangle \mapsto \begin{cases} -|x\rangle, & \text{if } x = x_0 \\ |x\rangle, & \text{otherwise} \end{cases} \quad (2.27)$$

This phase flip alone does not yet isolate $|x_0\rangle$. Grover's algorithm applies O followed by a diffusion operator D that reflects states about the average amplitude. Formally,

$$D = 2|s\rangle\langle s| - I \quad (2.28)$$

Each Grover iteration G is the composition $G = DO$. Letting G act on $|s\rangle$ repeatedly rotates the state vector in a two-dimensional subspace spanned by $|s\rangle$ and $|x_0\rangle$. We summarize the main result on the algorithm's complexity in the following theorem.

Theorem 2.2: Grover's algorithm [21]

Given access to the quantum oracle O to a vector $\mathbf{x} = \{0, 1\}^N$, the Grover's algorithm, with high probability, finds the index of the marked element using $O(\sqrt{N})$ applications of O .

2.2.5. Amplitude amplification

Amplitude amplification generalizes Grover's method to scenarios where one starts with a procedure A that prepares a "good" state $|\psi_{\text{good}}\rangle$ with some probability p , embedded in a superposition

$$|\psi\rangle = A|0^{\otimes n}\rangle = \sqrt{p}|\psi_{\text{good}}\rangle + \sqrt{1-p}|\psi_{\text{bad}}\rangle \quad (2.29)$$

In place of Grover's oracle, one uses a reflection S_0 that flips the phase of the component $|\psi_{\text{good}}\rangle$, and a second reflection S_ψ about the overall initial superposition. If $Q = S_\psi S_0$ is the amplification operator (analogous to $G = DO$ in Grover's algorithm), repeated applications of Q systematically increase $\sqrt{p}$ to near unity, so that a final measurement reveals $|\psi_{\text{good}}\rangle$ with high probability. The complexity scales as $O(\frac{1}{\sqrt{p}})$, mirroring the $O(\sqrt{N})$ speedup in the case where $p = \frac{1}{N}$. This method is thus a potent subroutine in quantum algorithms that begin with a known but imperfect preparation step and wish to amplify its success amplitude. The result is summarized as follows.

Theorem 2.3: Amplitude amplification [76]

Given one copy of the quantum state $|\psi\rangle$, unitary $S_0 = I - 2|\psi_{\text{good}}\rangle\langle\psi_{\text{good}}|$, and unitary $S_\psi = 2|\psi\rangle\langle\psi| - I = 2A|0^{\otimes n}\rangle\langle 0^{\otimes n}|A^\dagger - I$. There exists a quantum algorithm that, with high probability, produces the state $|\psi_{\text{good}}\rangle$ using $O(\frac{1}{\sqrt{p}})$ applications of S_0 and S_ψ . (Equivalently, $O(\frac{1}{\sqrt{p}})$ applications of A and $A^\dagger$.)

Important variants of amplitude amplification generalizes the algorithm to the scenarios when p is unknown (fix-point search [77]), A is a block-encoding unitary acting on a non-trivial initial state (oblivious amplitude amplification [78]), or both [79].

2.2.6. Block encoding

Block encoding is a powerful framework that allows embedding arbitrary (non-unitary) linear operators into larger unitary matrices, enabling quantum algorithms to access and manipulate such operators efficiently within the constraints of quantum mechanics [80]. Suppose A is an $2^n \times 2^n$ matrix (not necessarily unitary) with operator norm $\|A\| \leq \alpha$. A unitary matrix U acting on a larger space (typically $n + a$ qubits) is said to be an (α, a, ε) -*block encoding* of A if

$$\|A - \alpha(|0^{\otimes a}\rangle\langle 0^{\otimes a}| \otimes I_s) U (|0^{\otimes a}\rangle\langle 0^{\otimes a}| \otimes I_s)\|_2 \leq \varepsilon \quad (2.30)$$

This means that the top-left block of U , in the basis where the first a qubits are in the state $|0\rangle^{\otimes a}$, approximates the scaled matrix A to error ε in operator norm.

$$U = \begin{bmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{bmatrix} \quad (2.31)$$

The encoding of U in quantum circuits can be realized by one of the following techniques: linear combination of unitaries (LCU) [81, 82], density matrix block encoding [83], sparse access oracles [84]. The preferable choice of encoding depends on sparsity and specific properties of A [85].

Block encodings enable quantum access to dense or structured matrices that appear in linear algebra problems. Many important matrices (Hamiltonians, density operators, Jacobians, etc.) can be embedded this way using ancillary qubits and controlled unitaries. Once A is block encoded, quantum algorithms can perform operations such as phase estimation, matrix inversion, exponentiation, or singular value transformation via techniques like quantum signal processing (QSP) [24] and quantum singular value transformation (QSVT) [80].

The strength of block encoding lies in its modularity: it abstracts non-unitary operators into a unitary-compatible form, allowing powerful primitives like Hamiltonian simulation, quantum linear system solvers, or amplitude estimation to be generalized beyond sparse or diagonal matrices. This makes it a key unifying tool in modern quantum algorithm design.

The quantum algorithms introduced earlier (subsections 2.2.2 to 2.2.6) belong to the class of fault-tolerant quantum algorithms. They come with strong theoretical backing and are widely believed to offer quantum speedups for certain problems. That said, turning these algorithms into actual quantum circuits composed of native gates is far from straightforward. Many protocols, such as the Quantum Fourier Transform, require gate operations with very high precision, which remains a significant challenge for current hardware. As of now, there is no experimental evidence that demonstrates any of these algorithms working reliably at a practical scale.

2.3. Representation Theory

Representation theory plays a foundational role in the analysis of many symmetry-involving algorithms. In quantum information and quantum computation, the theory finds practical applications in quantum error characterization and error mitigation strategies [86, 87, 88, 89]. In recent years, representation theoretic methods also have gained prominence in the design of machine learning models aimed at solving symmetry-constrained problems in both classical [90, 91, 92, 93, 94, 95] and quantum computing [96, 97, 98, 99], giving rise to the field known as *geometric machine learning*.

2.3.1. Basics

This section provides a brief overview of the essential concepts, with a focus on equivalent representations and the technique of twirling, both of which are particularly relevant to one of our main techniques.

Definition 2.1: Group

A **group** G is a set equipped with a binary operation $\cdot : G \times G \rightarrow G$ satisfying:

1. **Associativity:** $g_1 \cdot (g_2 \cdot g_3) = (g_1 \cdot g_2) \cdot g_3$ for all $g_1, g_2, g_3 \in G$
 2. **Identity:** There exists an element $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$
 3. **Inverses:** For each $g \in G$, there exists $g^{-1} \in G$ such that $g \cdot g^{-1} = g^{-1} \cdot g = e$
- (The symbol $\cdot$ is usually omitted for quick writing, i.e. $g_1 g_2 \equiv g_1 \cdot g_2$)

If G and H are groups and $\phi : G \rightarrow H$ is a map, ϕ is called a group homomorphism if it satisfies $\phi(g_1 g_2) = \phi(g_1) \phi(g_2)$ for all $g_1, g_2 \in G$. That is, ϕ respects the group operation: the image of a product is the product of the images.

Definition 2.2: Representation

A **representation** of a group G on a vector space V is a homomorphism $\tau : G \rightarrow \text{GL}(V)$, where $\text{GL}(V)$ denotes the group of invertible linear operators on V . The vector space V is called the representation space of τ . We often call the pair (τ, V) a representation of G .

Representations can also be defined on infinite dimensional vector spaces or on vector spaces over fields other than $\mathbb{C}$, but in the scope of our work, we mainly consider representations on finite dimensional complex vector spaces. For example, for finite-dimensional quantum systems on $\mathcal{H} \cong \mathbb{C}^d$, we

typically consider $\tau(g) \in \mathcal{U}(\mathcal{H})$, the group of unitary operators on the Hilbert space. In that case, τ is also called a unitary representation.

Definition 2.3: Irreducible representation

A subspace W of V is called invariant under τ if $\tau(g)W \subseteq W$ for all $g \in G$. The representation τ is called *irreducible* or *simple* if V and $\{0\}$ are the only invariant subspaces.

A group can have many representations, but in many cases, we might consider some of them as “the same” representation. The following definition tells when two representations are the same.

Definition 2.4: Equivariance

Let (τ_1, V_1) and (τ_2, V_2) be two representations of a group G . A linear map $T : V_1 \rightarrow V_2$ is called an *intertwining map* or *equivariant map* if

$$T\tau_1(g) = \tau_2(g)T, \text{ for all } g \in G \quad (2.32)$$

The representations τ_1 and τ_2 are also equivariant. Furthermore, when T is bijective, the representations are *equivalent*, denoted by $\tau_1 \cong \tau_2$.

The notion of equivariance is the preservation of structural properties of representations. The intertwining map T transfers the group’s action on V_1 to V_2 .

One can combine different representations to make new representations. A possible construction is through a direct sum defined as follows.

Definition 2.5: Direct sum of representations

Let (τ_1, V_1) and (τ_2, V_2) be two representations of a group G . The direct sum of τ_1 and τ_2 is the representation $\tau_1 \oplus \tau_2$ on $V_1 \oplus V_2$ defined by

$$[(\tau_1 \oplus \tau_2)(g)](v_1, v_2) = (\tau_1(g)v_1, \tau_2(g)v_2), \text{ for all } g \in G, v_i \in V_i \quad (2.33)$$

Definition 2.6: Completely reducible representation

A representation is *completely reducible* or *semisimple* if it is equivalent to a direct sum of irreducible representations.

Theorem 2.4: Complete reducibility of unitary representations

A unitary representation τ on a finite-dimensional Hilbert space $\mathcal{H}$ is completely reducible.

Proof. Let $W \subseteq \mathcal{H}$ be an invariant subspace of τ and $W^\perp$ be its orthogonal complement in $\mathcal{H}$. Using the unitarity of τ , it is straightforward to see $W^\perp$ is also an invariant subspace and $\mathcal{H} = W \oplus W^\perp$. We prove the theorem by induction on $d = \dim \mathcal{H}$. If $d = 1$, τ must be irreducible. Now, by recursion on dimension, we can decompose $\mathcal{H} = W_1 \oplus W_2 \oplus \dots \oplus W_k$, where each $\tau|_{W_i}$ is irreducible. $\square$

2.3.2. Schur's lemma

Next, we present Schur’s lemma and its consequences in the setting of finite-dimensional vector spaces. These are elementary yet extremely useful tools in representation theory.

Theorem 2.5: Schur's lemma

Let (τ_1, V_1) and (τ_2, V_2) be finite-dimensional complex irreducible representations of a group G , and let $T : V_1 \rightarrow V_2$ be an intertwining map. The following statements hold:

1. If V_1 and V_2 are not isomorphic, $T = 0$.
2. If $V_1 \cong V_2$, T is bijective.

Proof. Suppose $T \neq 0$. We show that V_1 and V_2 are isomorphic. The kernel of T , $\ker(T) \subseteq V_1$, is an invariant subspace. The irreducibility of τ_1 implies $\ker(T) = 0$. Similarly, the image $\text{Im}(T) \subseteq V_2$ is also an invariant subspace, so we have $\text{Im}(T) = V_2$ by the irreducibility of τ_2 . The intertwining map T is injective and surjective, hence bijective. $\square$

Corollary 2.1

Let (τ, V) be a finite dimensional irreducible complex representation of a group G . Let $T : V \rightarrow V$ be an intertwining map of τ with itself, $T\tau(g) = \tau(g)T$ for all $g \in G$. Then $T = \lambda I$ for some scalar $\lambda \in \mathbb{C}$.

Proof. Fix an eigenvalue λ of T . The operator $T - \lambda I$ is also a self-intertwining map which is not injective. Then it must be zero, i.e. $T = \lambda I$. $\square$

Corollary 2.2

Let (τ, V) be a finite dimensional completely reducible complex representation of a group G . Suppose $\tau = \bigoplus_i m_i \tau_i$ be a decomposition of τ into irreducible subrepresentations, in which τ_i is representative of a distinct isomorphism class with multiplicity m_i . Then the commutant algebra

$$\text{End}_G(V) := \{T \in \text{End}(V) \mid T\tau(g) = \tau(g)T \text{ for all } g \in G\} \quad (2.34)$$

is isomorphic to the direct sum of matrix algebras

$$\text{End}_G(V) \cong \bigoplus_i \text{Mat}_{m_i}(\mathbb{C}) \quad (2.35)$$

i.e. one copy of $m_i \times m_i$ matrices for each irreducible type τ_i .

Proof. Let W_i be the image of the irreducible subrepresentation τ_i . The decomposition of τ implies a decomposition of V :

$$V \cong \bigoplus_i (W_i \otimes \mathbb{C}^{m_i}) \quad (2.36)$$

Fix an isomorphism $V_i \cong W_i \otimes \mathbb{C}^{m_i}$ in which $\tau(g)|_{V_i} = \tau_i(g) \otimes I_{m_i}$. Suppose $T : V \rightarrow V$ is an intertwining map. Since V_i as the image of isomorphic irreducible subrepresentations are G -invariant, T maps V_i to V_j if and only if $W_i \cong W_j$. By Schur's lemma, any intertwining map between $W_i \not\cong W_j$ is zero, T must have no component mapping between V_i and V_j . The commutant is decomposed as

$$\text{End}_G(V) \cong \bigoplus_i \text{End}_G(V_i) \quad (2.37)$$

By our choice of V_i such that $\tau(g)|_{V_i} = \tau_i(g) \otimes I_{m_i}$, any operator T_i that commutes with $\tau(g)|_{V_i}$ must act trivially on W_i , i.e. it has the form $T_i = I_{W_i} \otimes A$ for any matrix $A \in \text{Mat}_{m_i}(\mathbb{C})$. Hence $\text{End}_G(V_i) \cong \text{Mat}_{m_i}(\mathbb{C})$. $\square$

Intuitively, a self-intertwining map T of a completely reducible representation acts as the identity on each irreducible subspace W_i but mixes the multiplicities copies via arbitrary linear maps. The map

has the following structure

$$T = \bigoplus_i (I_{W_i} \otimes A_i), \quad A_i \in \text{Mat}_{m_i}(\mathbb{C}) \quad (2.38)$$

2.3.3. Twirling

Schur's lemma underpins the uniqueness (up to scalar) of intertwining maps between irreducible representations and is foundational for the concept of *twirling*. Twirling is a general averaging procedure associated with group actions, used to extract the invariant part of a linear transformation under a given symmetry.

Let G be a compact group with a unitary representation $\pi : G \rightarrow \mathcal{U}(V)$, where V is a finite-dimensional complex Hilbert space. Given a linear operator $T : V \rightarrow V$, the twirled version of T , denoted $\mathcal{T}_G(T)$, is defined by averaging over the group action

$$\mathcal{T}_G(T) := \int_G \pi(g) T \pi(g)^\dagger d\mu(g), \quad (2.39)$$

where $d\mu$ denotes the normalized Haar measure on G . If G is a finite group, this expression simplifies to:

$$\mathcal{T}_G(T) = \frac{1}{|G|} \sum_{g \in G} \pi(g) T \pi(g)^\dagger \quad (2.40)$$

This construction satisfies several important properties:

- **Equivariance:** The operator $\mathcal{T}_G(T)$ commutes with the group action:

$$\pi(h) \mathcal{T}_G(T) = \mathcal{T}_G(T) \pi(h), \quad \forall h \in G.$$

Hence, $\mathcal{T}_G(T) \in \text{End}_G(V)$, the commutant of the representation π .

- **Projection property:** If $T \in \text{End}_G(V)$, then $\mathcal{T}_G(T) = T$. In this sense, twirling acts as a projection from $\text{End}(V)$ onto $\text{End}_G(V)$.

By Schur's lemma and its generalizations, this means that twirling effectively removes all components of T that are not symmetric under the group action. In other words, it isolates the *group-invariant* part of an operator.

Example: Pauli Twirling In practice, one often uses the Pauli group $\mathcal{P}_n$ for n -qubit systems. The Pauli twirl of noise channel $\mathcal{E}$ transforms it into a Pauli channel, i.e., one that applies Pauli errors with certain probabilities. This simplification is crucial for designing efficient error correction codes and for benchmarking quantum devices [86, 87, 88, 89].

3

Related work

Quantum algorithms for simulating physical systems have progressed rapidly in recent decades, fueled by the promise of quantum speedup in solving high-dimensional, structured problems such as partial differential equations (PDEs). Among these, the LBM as a mesoscopic model for fluid dynamics has emerged as a natural candidate for quantum implementation due to its local, rule-based structure and parallelizability. The idea of leveraging quantum systems to simulate LBM originated in the late 1990s, with foundational contributions from Yepez and others. In what follows, we review the evolution of quantum lattice Boltzmann methods (QLBM), beginning with these early works and continuing through to recent developments that incorporate advances in quantum hardware and algorithmic design.

3.1. Quantum Lattice Boltzmann Methods: The First Steps

The earliest proposals for simulating classical fluid dynamics using quantum algorithms were introduced by Jeffrey Yepez in a series of seminal works spanning from the late 1990s to early 2000s. Yepez’s 1998 work laid the theoretical foundation for what would later be termed the quantum lattice-gas automaton (QLGA), where qubit ensembles represent occupation probabilities in a discretized lattice space, and local quantum gates mimic collision dynamics between virtual fluid particles [100].

In his 2001 and 2002 papers [101, 102], Yepez presented increasingly refined models where unitary evolution, interleaved with ensemble measurements and classical communication, yields a mesoscopic-scale finite-difference lattice-Boltzmann equation. Particularly in his 2002 paper, Yepez demonstrates the emergence of the nonlinear Burgers equation from a type-II quantum computer architecture, a lattice of small quantum processors with communication. This work explicitly bridges the microscopic (quantum) and macroscopic (fluid) scales, pioneering the factorized quantum lattice-gas (QLG) algorithm and introducing the notion of encoding fluid fields via occupation probabilities of local qubits.

Yepez’s algorithm is based on the repeated application of three key steps: state preparation, local unitary collision, and streaming by measurements and classical communication. Notably, by choosing a $U(2)$ collision operator that preserves local occupancy (i.e., particle number), Yepez derives an emergent quantum lattice-Boltzmann equation capable of capturing nonlinear phenomena such as shock formation. His numerical simulations of Burgers turbulence validated the quantum algorithm against exact solutions, with the simulation showing second-order spatial convergence and first-order temporal accuracy [102].

However, a fundamental limitation of Yepez’s approach is its reliance on a type-II quantum computing architecture, where every node of the lattice is equipped with its own quantum processor. These processors are restricted to performing local operations, while the streaming step is implemented non-unitarily via collapse of quantum states, classical communication of measurement outcomes, and reinitialization at each time step. Although this enables a clean separation between quantum and classical layers, the large system size introduces a significant scalability bottleneck for large-scale simulations and vastly diverges from existing NISQ architectures.

3.2. Dilemma of Encoding Designs

Following Yepez's early contributions, the field of quantum computational fluid dynamics saw limited progress for over a decade, until a recent wave of renewed interest—particularly in the development of quantum lattice Boltzmann methods. These efforts have led to a variety of algorithmic strategies [46, 47, 48, 49, 50, 57, 51, 52, 53, 54, 55], yet all face a fundamental design choice: how to encode fluid distribution functions onto quantum registers in a manner that facilitates an implementation of collision and streaming. The two prevailing options are amplitude encoding and product state encoding (also known as basis state encoding), each with inherent limitations.

Amplitude encoding stores the entire distribution functions in the amplitudes of a quantum state in the form $\sum_{\mathbf{x}, i} f_i(\mathbf{x}) |\mathbf{x}\rangle |i\rangle$ or $\sum_{\mathbf{x}, i} \sqrt{f_i(\mathbf{x})} |\mathbf{x}\rangle |i\rangle$. This is highly efficient in terms of qubit usage and allows for streaming operations to be implemented unitarily as velocity-controlled shift operators in Hilbert space. However, as Schalkers shows, this encoding is fundamentally incompatible with direct implementation of non-linear collision operators [45]. Because amplitudes evolve linearly under unitary operations, encoding non-linear fluid dynamics requires an indirect strategy.

To address this, several methods have been proposed to block-encode the non-linear collision operator into a larger unitary, enabling it to be simulated within the constraints of quantum gates. Currently, three main approaches exist:

- Equilibrium distribution encoding, where a first-order expansion of the equilibrium function (e.g., $f_i^{\text{eq}} = w_i \rho (1 + \mathbf{c}_i \cdot \mathbf{u} / c_s^2)$) is block-encoded and used to generate update amplitudes [46, 47, 48, 49, 50, 57]. An implicit assumption in this approach is the uniformity of the velocity field over the domain, which makes the approach suitable for simulating laminar flows.
- Carleman-linearized collision, where the non-linear collision operator is approximated by a truncated Carleman linearization, which can then also be embedded in a block-encoded unitary [51, 52, 56]. Both streaming and collision steps are achieved via unitary operators in this approach at the cost of a complex quantum circuit.
- Controlled rotation schemes, where refined controlled rotations are used to prepare amplitudes matching the desired post-collision state, followed by conditional branching [54] or post-selection to eliminate undesired parts [53].

All three methods introduce overhead in the form of ancillary qubits and probabilistic success. All of the three approaches require full-state tomography, which provides an estimate of the post-collision amplitudes only to be re-prepared at every time step, resulting in a "evolve-then-reset" algorithmic flow that is resource-intensive and difficult to scale.

Product state encoding offers an alternative by assigning each momentum population its own qubit or register. This allows for local, possibly non-linear collision operators to be implemented directly and deterministically. However, as Schalkers also proves, this comes at a different cost: streaming between lattice sites generally becomes non-unitary [45]. While this approach offers more flexibility in implementing non-linear dynamics, it does not scale easily under current quantum hardware due to the demand for high-precision arithmetics. A notable work by Steijl [55], who proposes the use of a quantum arithmetic logic unit to perform the non-linear collision operator coherently. Fluid quantities are encoded into quantum registers and the update rule is evaluated via reversible arithmetic circuits. This avoids measurement altogether and enables a fully unitary implementation. Acknowledging different advantages of amplitude-based and basis-state-based encodings, Steijl calls for further investigation on efficient transformations between the two representation types [103, 104] as basis-state-based algorithms can also have quantum advantage, including Fourier transform [105]. Meanwhile, Schalkers suggested a basis state encoding scheme that takes into account the distributions within the vicinity of $\mathbf{x}$. If that vicinity contains every possible position particles can move to after T streaming steps, the encoding scheme ensures a coherent quantum LBM/LGA simulation within a time duration T .

While the question of optimal encoding remains unresolved, progress has continued on other important fronts. Notably, several efforts have focused on efficient techniques for implementing the streaming step, aiming to reduce circuit depth and qubit overhead [47]. Boundary conditions, especially specular reflections, have also received attention. Approaches have been designed to encode solid boundaries and physically correct particle flow behavior at walls [106]. In parallel, some researchers have explored

the potential of collisionless QLBM, applying the framework to regimes where transport phenomena dominate, such as rarefied flows [107] and self-gravitating systems [108].

4

Methodology

4.1. Overview

This chapter presents the methodology employed in this master's thesis to address the research questions introduced in [chapter 1](#). The primary objective of this work is to investigate which encoding strategies for particle distributions enable an effective quantum implementation of the lattice Boltzmann method (LBM). In particular, we introduce two quantum LBM algorithms that differ in their underlying data-encoding strategies.

Before we describe these methods in detail, we first introduce the general encoding structure common to both schemes. A choice of data encoding comes with its own set of operations that can be conveniently executed by the algorithm. Here, our consideration is restricted to quantum computers with sequential unitary operators (quantum gates) acting on two-level systems (qubits).

Let the lattice nodes be located on integral coordinates in a grid of dimension $L_x \times L_y \times L_z$, and let each node be labeled by the corresponding coordinate vector $\mathbf{x} = (x, y, z)$. We denote the distribution at each node by $f_{\mathbf{x},i} = f_i(\mathbf{x})$. Because the total number of nodes, $L = L_x L_y L_z$, can be very large, an efficient encoding strategy for the spatial component is to use a superposition of the form

$$|\psi\rangle = \sum_{\mathbf{x}} |\mathbf{x}\rangle |\psi_{\mathbf{x}}\rangle, \quad (\text{normalization omitted}) \quad (4.1)$$

where $|\psi_{\mathbf{x}}\rangle$ encodes all the fictitious velocities (i.e., $f_{\mathbf{x},i}$ for every i) at position $\mathbf{x}$, whose form is determined by the choice of encoding. This representation requires $O(\log_2(L))$ qubits for the spatial-coordinate register.

With this general framework in mind, we turn to the two main methods developed in this work. In particular, [section 4.2](#) introduces the first method, which utilizes a *tensor-product encoding* scheme. We detail its algorithmic formulation and theoretical underpinnings. Subsequently, [section 4.3](#) describes the second method, based on *amplitude encoding*, following the same pattern of algorithmic description. [section 4.4](#) is dedicated for the analysis on the accuracy and performance of quantum algorithms associated to the encoding methods. By covering both methods within a unified framework, this chapter offers a perspective on the range of encoding strategies for quantum LBM. Some detailed refinements and numerical simulation results are delayed until [chapter 5](#).

4.2. Tensor-product-encoding-based algorithm

4.2.1. Encoding distributions

We propose a product-state encoding method that encodes N_v distributions $f_{\mathbf{x},1}, \dots, f_{\mathbf{x},N_v}$ employing N_v qubits in the state

$$|\psi_{\mathbf{x}}\rangle = \bigotimes_{i=1}^{N_v} \frac{1}{\sqrt{2}} (|0\rangle + e^{if_{\mathbf{x},i}} |1\rangle) \quad (4.2)$$

This embedding has several advantages:

- First, it allocates a single separate qubit to “store” each distribution, which has the Bloch sphere coordinates $\vec{n}_{\mathbf{x},i} = (\cos(f_{\mathbf{x},i}), \sin(f_{\mathbf{x},i}), 0)$.
- Second, the state $|\psi\rangle$ can be easily prepared from either classical data or quantum data by applying controlled rotations on the uniform superposition $\sum_{\mathbf{x}} |\mathbf{x}\rangle |+\rangle^{\otimes N_v}$. If one needs to prepare the state from classical data, the rotations controlled by the lattice node $|\mathbf{x}\rangle$ rotate by a predefined set of angles. When the data distribution values $|f_{\mathbf{x},i}\rangle$ are available as a computational basis state, the rotations are controlled by both the lattice node and the basis state.
- Third, it is also fairly simple to discard the distribution register in state $|\psi_{\mathbf{x}}\rangle$ without affecting other parts of the quantum state $|\psi\rangle$, except for possibly introducing new relative phases. One can reset the i -th qubit in the register by measuring it in the standard basis and apply a NOT gate when the measurement outcome is 1, which will also leave a relative phase $e^{if_{\mathbf{x},i}}$.

Fig. 4.1 illustrates a generic quantum circuit that prepares this encoding from classical data.

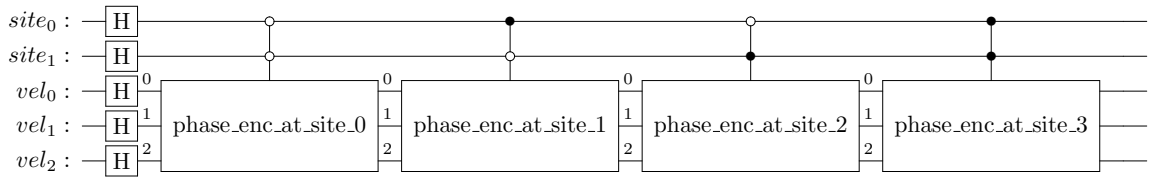


Figure 4.1: Encoding circuit of the distribution for a D1Q3 scheme with 4 sites. The “phase_enc_at_site_k” unitaries are tensor products of single-qubit phase gates across the “vel” (velocity) register.

4.2.2. Estimating macroscopic quantities

The described encoding strategy enables estimating macroscopic quantities, such as density and momentum, using purely quantum operations. Observe that $|\psi_{\mathbf{x},i}\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{if_{\mathbf{x},i}} |1\rangle)$ is the eigenstate of the rotation $R_{\mathbf{x},i} = \exp(if_{\mathbf{x},i}(\vec{n}_{\mathbf{x},i} \cdot \vec{\sigma}))$ with eigenvalue $e^{if_{\mathbf{x},i}}$, where $\vec{\sigma} = (X, Y, Z)$ is the vector of Pauli operators. That being said, applying quantum phase estimation (QPE) would return the value of (a multiple of) $f_{\mathbf{x},i}$ stored in some ancillary qubits. The QPE protocol requires the implementation of $R_{\mathbf{x},i}$, which is available without explicit knowledge of $f_{\mathbf{x},i}$. Let $A_{\mathbf{x},i}$ be the unitary that prepares the state $|\psi_{\mathbf{x},i}\rangle$. The rotation $R_{\mathbf{x},i} = A \exp(if_{\mathbf{x},i}Z) A^\dagger$ can be implemented using $A, A^\dagger$ and the rotation about Z axis,

$$\begin{aligned} \exp(if_{\mathbf{x},i}Z) &= (2|+\rangle\langle+| - I)(2|\psi_{\mathbf{x},i}\rangle\langle\psi_{\mathbf{x},i}| - I) \\ &= XA(2|0\rangle\langle 0| - I)A^\dagger \\ &= XAZA^\dagger \end{aligned} \quad (4.3)$$

A straightforward preparation of $A_{\mathbf{x},i}$ involves a Hadamard gate H followed by a phase gate $P(f_{\mathbf{x},i}) = \text{diag}(1, e^{if_{\mathbf{x},i}})$. Denote $P_{\mathbf{x},i} = P(f_{\mathbf{x},i})$ for short. The rotation is realized by the gate sequence

$$R_{\mathbf{x},i} = P_{\mathbf{x},i} H X P_{\mathbf{x},i} X P_{\mathbf{x},i}^\dagger H P_{\mathbf{x},i}^\dagger \quad (4.4)$$

We consider the entire state $|\psi\rangle$. This state can be evolved from $\sum_{\mathbf{k}} |\mathbf{x}\rangle \otimes |+\rangle^{\otimes N_v}$ by the unitary $U = \sum_{\mathbf{x}} |\mathbf{x}\rangle \langle \mathbf{x}| \otimes P_{\mathbf{x},1} \otimes \dots \otimes P_{\mathbf{x},N_v}$, i.e. phase gates controlled by $\mathbf{x}$. We also introduce the $\mathbf{x}$ -controlled rotation unitary

$$\begin{aligned} R &= \sum_{\mathbf{x}} |\mathbf{x}\rangle \langle \mathbf{x}| \otimes R_{\mathbf{x},1} \otimes \dots \otimes R_{\mathbf{x},N_v} \\ &= U \cdot (I^{\log_2(L)} \otimes (HX)^{\otimes N_v}) \cdot U \cdot (I^{\log_2(L)} \otimes X^{\otimes N_v}) \cdot U^\dagger \cdot (I^{\log_2(L)} \otimes H^{\otimes N_v}) \cdot U^\dagger \end{aligned} \quad (4.5)$$

Applying the QPE protocol for the unitary R on the state $|\psi\rangle$ essentially encodes the macroscopic density $\rho_{\mathbf{x}} = \sum_i f_{\mathbf{x},i}$ as binary numbers in an ancillary register.

$$\text{QPE}_R |\psi\rangle |0^{\otimes r}\rangle = \sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}\rangle \otimes \left| \frac{\rho_{\mathbf{x}} 2^r}{2\pi} \right\rangle \quad (4.6)$$

The same procedure also leads to the encoding of the macroscopic momentum. Common velocity configurations have discrete velocities $\mathbf{c}_i \in \{0, +1, -1\}^3$. By applying other three separate QPE protocols, we can estimate and store $\sum_i c_{i,\alpha} f_{\mathbf{x},i}$, $\alpha = x, y, z$ in ancillary registers for momentum. Note that the macroscopic density is equivalent to the case when $c_{i,\alpha} = +1$ for all i . We might capture possible minus signs by the eigenphase $e^{-if_{\mathbf{x},i}}$ associated to the eigenstate $|\psi_{\mathbf{x},i}\rangle$ of the reverse rotation

$$R_{\mathbf{x},i}^\dagger = P_{\mathbf{x},i} H P_{\mathbf{x},i} X P_{\mathbf{x},i}^\dagger X H P_{\mathbf{x},i}^\dagger \quad (4.7)$$

Let $\{1, \dots, N_v\}$ be partitioned into $I_{0,\alpha}$, $I_{+1,\alpha}$, and $I_{-1,\alpha}$ by the value of $c_{i,\alpha}$. Phase estimation of the mixed rotation

$$R_\alpha = \sum_{\mathbf{x}} |\mathbf{x}\rangle \langle \mathbf{x}| \otimes \left[\bigotimes_{i \in I_{+1,\alpha}} R_{\mathbf{x},i} \right] \otimes \left[\bigotimes_{i \in I_{0,\alpha}} I \right] \otimes \left[\bigotimes_{i \in I_{-1,\alpha}} R_{\mathbf{x},i}^\dagger \right], \quad \alpha \in \{x, y, z\} \quad (4.8)$$

yields the α -component of the momentum $p_{\mathbf{x},\alpha} = \sum_i c_{i,\alpha} f_{\mathbf{x},i}$. The estimation of macroscopic density and momentum include $1 + 3 = 4$ QPE protocols as shown in Fig. 4.2. At the end of the QPE protocols, the quantum state below carries all necessary variables required to compute the collision process

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}\rangle \otimes \left| \frac{2^r \rho_{\mathbf{x}}}{2\pi} \right\rangle \left| \frac{2^r p_{\mathbf{x},x}}{2\pi} \right\rangle \left| \frac{2^r p_{\mathbf{x},y}}{2\pi} \right\rangle \left| \frac{2^r p_{\mathbf{x},z}}{2\pi} \right\rangle \quad (4.9)$$

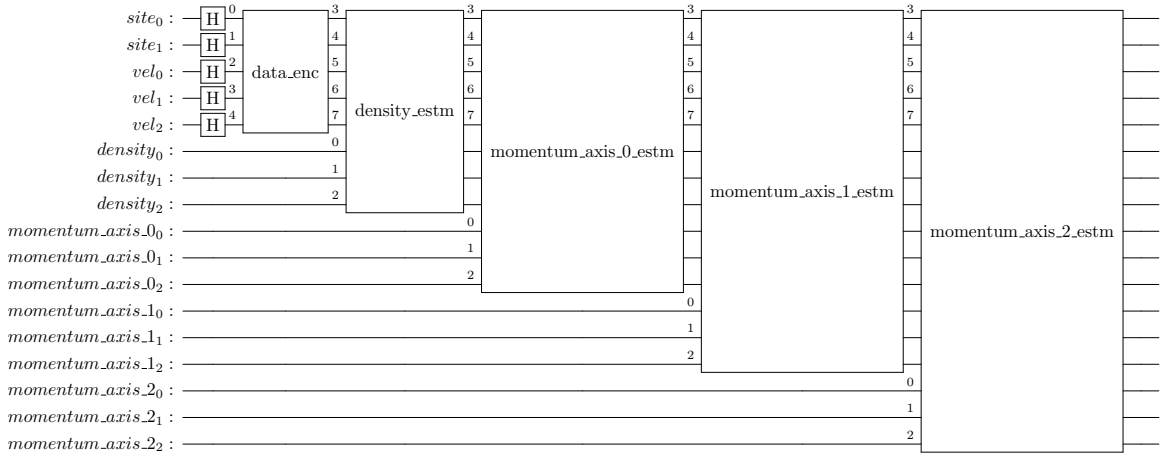


Figure 4.2: Quantum circuit that creates the state (4.9) for an imaginary D3Q3 scheme. The binary-fraction representations of density and momentum are stored in lower registers.

This paragraph discusses an issue and the solution regarding the accuracy of using QPE in the phase estimation problem. The standard QPE protocol of a unitary U using r precision qubits is introduced in chapter 2, which involves applying the unitary

$$\sum_{k=0}^{2^r-1} |k\rangle \langle k| \otimes U^k \quad (4.10)$$

on the initial state $|+\rangle^{\otimes r}$ of the precision qubits. However, the probability of getting the closest binary representation $\bar{\theta}$ to the true phase θ (i.e., the error $|\theta - \bar{\theta}|$ is below the threshold $\delta := 2^{-(r+1)}$) can be as low as $\frac{4}{\pi^2} \approx 0.4$. The work in [27] introduces a method called Tapered Quantum Phase Estimation (tQPE) and shows that using $r_{\text{taper}} = 6$ additional precision qubits can increase the probability of getting δ -bounded errors to $1 - \varepsilon$ for $\varepsilon \leq 10^{-11}$. We describe the implementation of the tQPE protocol on quantum circuits in [Appendix A](#).

4.2.3. Computing equilibrium distributions

We have applied a QPE-based protocol to estimate macroscopic quantities and store the density and the momentum vector as binary values. It is then natural to compute the equilibrium distributions using the expression (2.16). As the values are stored in a quantum superposition, the computation involves quantum arithmetic logic units (qALU) that perform elementary arithmetics for binary inputs. We design the quantum subroutines for computing addition, subtraction, multiplication, and division of signed numbers, whose constructions rely on two building blocks called Quantum Fourier Transform and Controlled Phase Shift. We will explain the notions and circuit implementations of the building blocks and basic arithmetics on binary numbers.

Quantum Fourier Transform The definition of quantum Fourier transform (QFT) is already introduced in (2.23). A typical implementation of the transform realizes the product-form expression in (2.24) using Hadamard gates and the controlled version of the phase gate $P_k := \text{diag}(1, \exp(2\pi/2^k))$. The desired output can be obtained by applying $\lfloor n/2 \rfloor$ SWAP gates from the output of the circuit depicted in [Figure 4.3](#) that effectively reverses the order of qubits. The SWAP gates are often omitted to save gate counts. In that case, operations succeeding the QFT circuit must also respect the reversed qubit order.

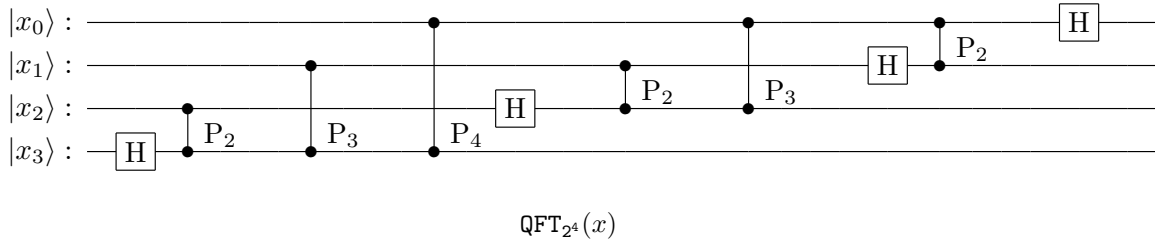


Figure 4.3: Quantum circuit for the Quantum Fourier Transform on 4 qubits (without final SWAP gates).

Controlled Phase Shift Controlled phase shift (CPS) multiplies each computational basis state $|x\rangle |y\rangle$ by the phase $e^{\frac{2\pi i}{2^n} xy}$, i.e. it implements $|x\rangle |y\rangle \mapsto e^{\frac{2\pi i}{2^n} xy} |x\rangle |y\rangle$. Expanding $xy = \left(\sum_{k=0}^{n-1} x_k 2^k\right) \left(\sum_{l=0}^{n-1} y_l 2^l\right)$ in the phase factor yields

$$e^{\frac{2\pi i}{2^n} xy} = \prod_{k=0}^{n-1} \prod_{l=0}^{n-1} \exp(2\pi i \cdot x_k y_l \cdot 2^{k+l-n}) \quad (4.11)$$

Only terms with $k + l \geq n$ contribute nontrivially to the product. Hence the CPS can be realized by the commuting gate sequence

$$U = \prod_{k=0}^{n-1} \prod_{l=0}^{n-1-k} CP_{n-k-l} \quad (4.12)$$

where CP_{n-k-l} denotes the controlled version of the P_{n-k-l} gate. (Control-target direction does not matter as the gate is symmetric.) The circuit in [Figure 4.4](#) visualizes an arrangement of the gate sequence that facilitates parallel execution of every CP_k gate for each $k = 1, 2, \dots$. The circuit can also be modified to handle the case when y has a different bit length $m < n$. The idea is using the standard CPS circuit for x and $\tilde{y}$, a zero-padded bitstring of y , and removing CP gates that act on a padded zero as those gates always act as identity.

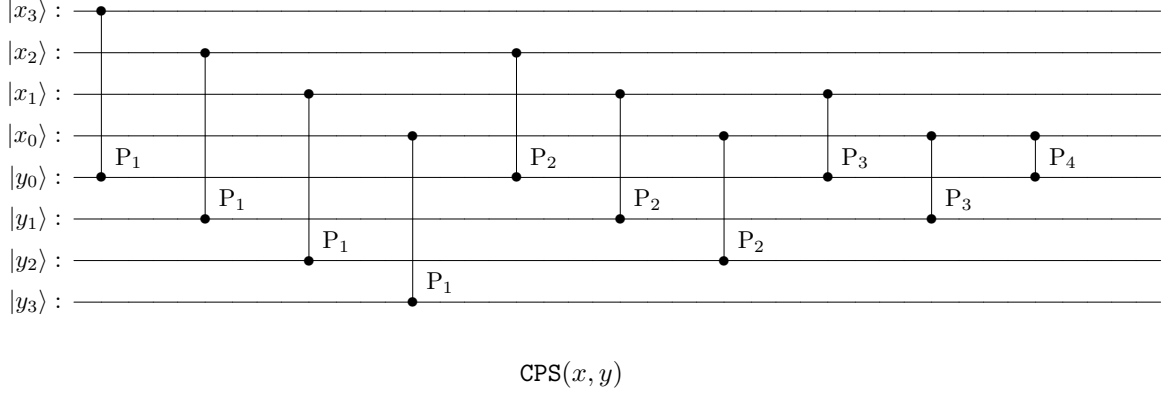


Figure 4.4: Quantum circuit for the Controlled Phase Shift on a pair of 4-bit numbers. The qubits of x is shown in reversed order for consistency with the reversal of bits as a result of the SWAP-less QFT circuit (Figure 4.3) applied on $|x\rangle$.

Addition, Subtraction, Weighted Sum Addition of signed integers lies at the heart of computational arithmetics, on which other operations such as subtraction, weighted sum, multiplication, and division are built. Draper’s adder is a seminal quantum algorithm for performing modular addition efficiently in the quantum setting [74]. The algorithm transforms a number to the Fourier basis and performs addition on the phase space as follows

$$|x\rangle |y\rangle \xrightarrow{\text{QFT} \otimes I} \sum_k e^{\frac{2\pi i}{2^n} xk} |k\rangle |y\rangle \xrightarrow{\text{CPS}} \sum_k e^{\frac{2\pi i}{2^n} (x+y)k} |k\rangle |y\rangle \xrightarrow{\text{QFT}^\dagger \otimes I} |(x+y) \bmod 2^n\rangle |y\rangle \quad (4.13)$$

Phase space computation also provides ample flexibility for computing general expressions of the form $x + \alpha y$, $\alpha \in \mathbb{Z}$ [109]. The expression can be computed by using $(\text{CPS})^\alpha$, the α -exponent of the controlled phase shift operator. We note that $(\text{CPS})^\alpha$ can be implemented using an identical circuit to the CPS operator, with the angle of every phase gate multiplied by α . This generalization also enables computing the weighted sum $\sum_{i=1}^m \alpha_i x_i$ with m application of the CPS operator:

$$\begin{aligned} (\text{QFT } |0\rangle)_0 |x_1\rangle_1 \dots |x_m\rangle_m &\xrightarrow{\text{CPS}_{0,1}} \sum_k e^{\frac{2\pi i}{2^n} (0 + \alpha_1 x_1)k} |k\rangle_0 |x_1\rangle_1 \dots |x_m\rangle_m \\ &\dots \\ &\xrightarrow{\text{CPS}_{m-1,m}} \sum_k e^{\frac{2\pi i}{2^n} (0 + \alpha_1 x_1 + \dots + \alpha_m x_m)k} |k\rangle_0 |x_1\rangle_1 \dots |x_m\rangle_m \\ &\xrightarrow{\text{QFT}_0^\dagger} |(\alpha_1 x_1 + \dots + \alpha_m x_m) \bmod 2^n\rangle_0 |x_1\rangle_1 \dots |x_m\rangle_m \end{aligned} \quad (4.14)$$

Draper’s adder and its generalizations perform *modular* arithmetics on *unsigned* integers. A modular expression is effectively *nonmodular* when the desired output does not overflow the output register’s dimension. This can be achieved by supplementing extra “carry” qubits to the output register. Suppose addends $x_1, \dots, x_m$ are n -bitstrings, i.e. $0 \leq x_i < 2^n$ if x_i is an unsigned binary number or $0 \leq |x_i| < 2^{n-1}$ if x_i is a signed binary number encoded in *two’s complement* (explained in the next paragraph). The weighted sum $\alpha_1 x_1 + \dots + \alpha_m x_m$ with bounded weights $|\alpha_i| \leq A$ is then bounded by $R = mA \cdot 2^n$ for unsigned inputs or $R = mA \cdot 2^{n-1}$ for signed inputs. Assigning $\lceil \log_2 R \rceil$ qubits for the output space ensures a nonmodular outcome. Standard addition and subtraction are simple cases when one extra carry qubit suffices, i.e we allocate $n + 1$ qubits for the register with $|x\rangle$ and n qubits for the register with $|y\rangle$.

To enable both signed and unsigned arithmetic in a uniform manner, we adopt the two’s complement representation for integers. In classical digital logic, the two’s complement of an n -bit number x is constructed by first computing the bitwise complement of x (often called “one’s complement”) and then

adding 1. More concretely, if x is an integer in the range $-2^{n-1} \leq x \leq 2^{n-1} - 1$, we represent x in binary by

$$\bar{x} = \begin{cases} x, & \text{if } x \geq 0 \\ 2^n + x, & \text{if } x < 0 \end{cases}$$

whose most significant bit indicates the sign of x ,

$$\text{MSB}(\bar{x}) = \begin{cases} 0, & \text{if } x \geq 0 \\ 1, & \text{if } x < 0 \end{cases}$$

The computation of arithmetics described earlier also needs correction in the output's sign. We consider the nonmodular Draper's adder in which we encode $|\bar{x}\rangle$ and $|\bar{y}\rangle$ using n qubits each. A carry qubit initialized in $|0\rangle$ is placed as the most significant bit in the $\bar{x}$ -register. Without modification, the standard algorithm sends $|0\bar{x}\rangle |\bar{y}\rangle$ to $|(\bar{x} + \bar{y}) \bmod 2^{n+1}\rangle |\bar{y}\rangle$, where the output is

$$(\bar{x} + \bar{y}) \bmod 2^{n+1} = \begin{cases} (x + y) \bmod 2^{n+1}, & \text{if } x, y \geq 0 \\ (x + y + 2^n) \bmod 2^{n+1}, & \text{if } x \geq 0, y < 0 \text{ or } x < 0, y \geq 0 \\ (x + y + 2^{n+1}) \bmod 2^{n+1}, & \text{if } x, y < 0 \end{cases} \quad (4.15)$$

The desired outcome, presumably in two's complement, is $(x + y + 2^{n+1}) \bmod 2^{n+1}$ regardless of inputs' sign. The algorithm fails to produce this output when exactly one input is negative. We modify the algorithm by initializing the carry bit in $|\text{MSB}(\bar{x}) \oplus \text{MSB}(\bar{y})\rangle$, which can be obtained by two CNOTs gate from the most significant bit of the two's complements to the carry bit *prior* to the main algorithm. This ensures the final output is correctly encoded in two's complement with the sign of $x + y$ stored in the carry bit. This correction rule generalizes to the integer-weighted sum $\sum_{i=1}^m \alpha_i x_i$ when we initialize the sign bit in $\sum_i \alpha_i \cdot \text{MSB}(\bar{x}_i)$ modulo 2. The circuit constructions for addition and weighted sum are presented in [Figure 4.5](#).

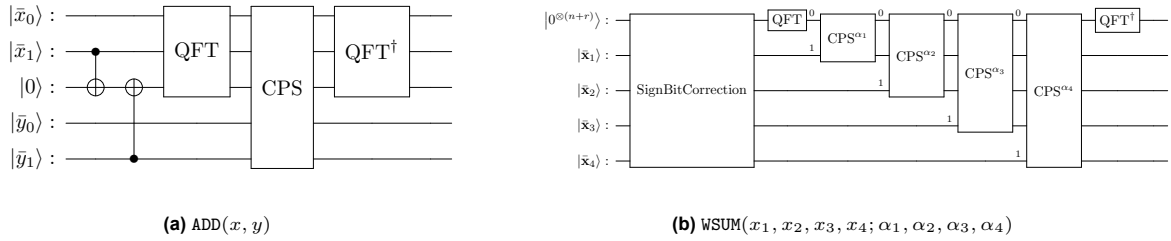


Figure 4.5: (a) Quantum circuit for adding two 2-bit signed integers x and y . The CNOT gates correct the sign qubit that starts in $|0\rangle$. The sum is stored in the top three qubits with MSB being the sign qubit. (b) Quantum circuit for computing the weighted sum $\sum_{i=1}^4 \alpha_i x_i$. If every $\bar{x}_i$ is encoded using n bits, a nonmodular computation requires at least $r = \lceil \log_2(|\alpha_1| + |\alpha_2| + |\alpha_3| + |\alpha_4|) \rceil$ extra qubits.

Multiplication Multiplying an integer with another can be carried out in the same manner as computing a weighted sum with the “weights” being the other’s bit. Concretely, the multiplication of x with an unsigned binary $y = y_{m-1} \dots y_0$ can be written as $xy = \sum_{i=0}^{m-1} y_i 2^i x$. One can construct the multiplication circuit from the weighted sum circuit which gets its weights from qubits of y . In general, when y is also a signed integer, we multiply $x|y|$ and correct the product sign if $y < 0$. The negation in two’s complement is formally expressed as a *complement* $y \mapsto (2^m - y)$. Taking absolute value of y , however, only involves a negation of $m - 1$ value bits if the sign bit is 1. The reversible circuits to compute the complement and the absolute value are depicted in [Figure 4.6](#). Given these ingredients, we construct a circuit in [Figure 4.7](#) to multiply an n -bit x and an m -bit y and store the product using $n + m - 1$ bits.

Division Division is a fundamental arithmetic operation with challenging implementation as the operation produces an output pair, quotient and remainder. Unlike multiplication, which can be efficiently performed using QFT and addition in the phase domain, division requires iterative approaches that mimic classical division techniques while maintaining reversibility. We adopt a quantum circuit for division based on a non-restoring division algorithm [110]. Given two non-negative n -bit integers as input

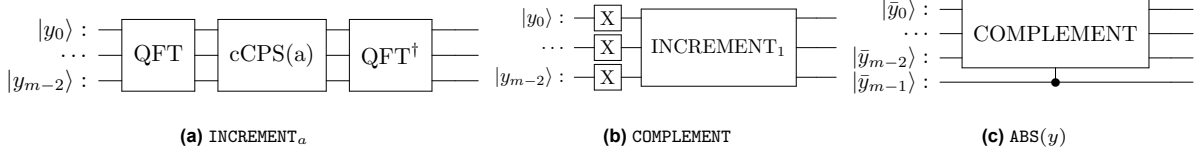


Figure 4.6: (a) Quantum circuit for adding a classical number a to an $(m - 1)$ -bit y . This resembles the function $\text{ADD}(y, a)$ in which the Controlled Phase Shift is replaced by a *classically* Controlled Phase Shift (cCPS) that performs classical *if*-conditions instead of CNOT gates. (b) Quantum circuit for taking the complement $2^{m-1} - y$. (c) Quantum circuit for taking the absolute value of y . We denote the initial state $\bar{y}_{m-1}\bar{y}_{m-2} \dots \bar{y}_0$ to emphasize that this circuit applies for two's complement encoding. The unsigned binary number $|y|$ is stored in the first $m - 1$ qubits, while the original sign in the last qubit is unchanged.

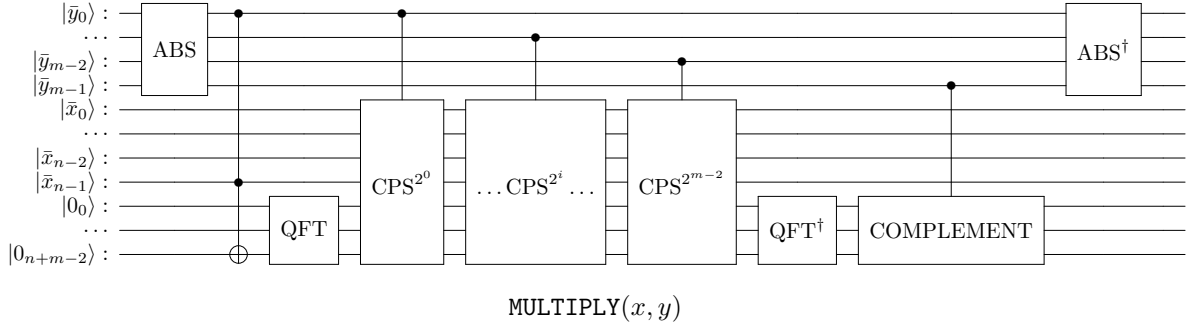


Figure 4.7: Quantum circuit for multiplying two signed numbers x and y . The Toffoli gate corrects the sign bit of the product. At the end of the circuit, the inputs $|\bar{x}\rangle$ and $|\bar{y}\rangle$ are unchanged, and the product xy in two's complement is stored in the last $n + m - 1$ qubits.

(dividend and divisor), the circuit outputs the quotient and remainder through conditional addition and subtraction. For signed inputs, the quotient and remainder are negated accordingly. The circuit in [Figure 4.8](#) implements a division between two non-negative integers.

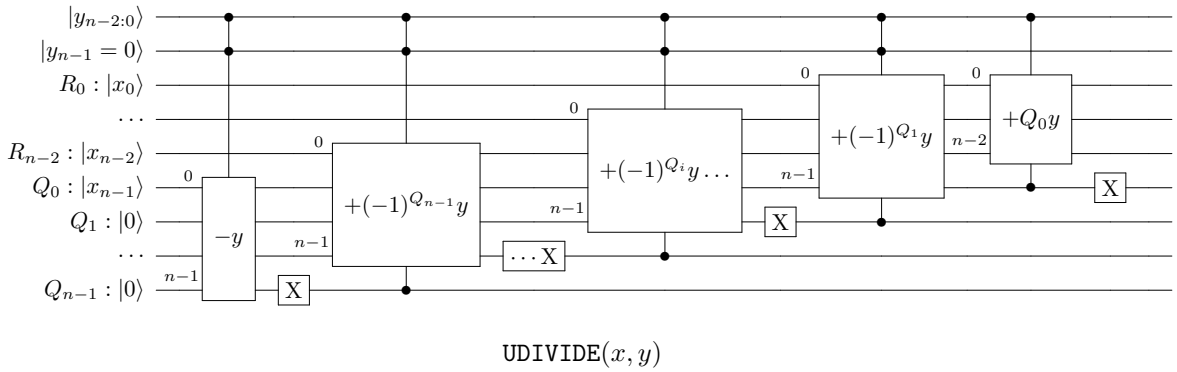


Figure 4.8: Quantum circuit for dividing x by y . The inputs are non-negative and encoded in n -bit two's complement, i.e. $x = 0x_{n-2} \dots x_0$ and $y = 0y_{n-2} \dots y_0$. The quotient and the remainder can be found in the qubits $|Q\rangle = |Q_{n-1} \dots Q_0\rangle$ and $|R\rangle = |R_{n-2} \dots R_0\rangle$, respectively.

Process to evaluate equilibrium distributions Given quantum arithmetical instruments, one can effectively compute the equilibrium distribution. However, efficiently grouping operations leads to not only less computational expenses but also better numerical stability. Substituting the lattice's speed of sound $c_s = \frac{1}{\sqrt{3}}$ into (2.16), we get

$$\begin{aligned}
f_i^{\text{eq}} &= w_i \rho \left[1 + 3\mathbf{c}_i \cdot \mathbf{u} + \frac{9}{2}(\mathbf{c}_i \cdot \mathbf{u})^2 - \frac{3}{2}|\mathbf{u}|^2 \right] \\
&= w_i \left[\rho + 3\mathbf{c}_i \cdot \mathbf{p} + \frac{(3\mathbf{c}_i \cdot \mathbf{p})^2 - 3|\mathbf{p}|^2}{2\rho} \right]
\end{aligned} \tag{4.16}$$

The second expression, as a function of density ρ and momentum $\mathbf{p} = \rho\mathbf{u}$, accomodates arithmetics from the macroscopic quantities given in the quantum state (4.9). Let the rational-valued weights $w_i = \frac{\alpha_i}{\beta_i}$, where α_i and β_i are coprimes. We can further rewrite the expression as a fraction.

$$f_i^{\text{eq}} = \frac{\alpha_i [2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p}) + (3\mathbf{c}_i \cdot \mathbf{p})^2 - 3|\mathbf{p}|^2]}{2\rho\beta_i} =: \frac{\text{Nu.}}{\text{De.}} \tag{4.17}$$

Evaluating the equilibrium distribution using this expression induces minimal truncation error as it only uses a single integer division. We suggest a qubit-saving strategy to compute f_i^{eq} in a reversible manner through the following steps

1. The initial macroscopic quantities are stored in binary-fraction form $|\rho\rangle |p_x, p_y, p_z\rangle$.
2. Make another copy of each momentum component using CNOT gates: $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2}$.
3. Compute the sum $(3\mathbf{c}_i) \cdot \mathbf{p} = \sum_{\alpha} (3c_{i,\alpha}) p_{\alpha}$ and store it in a register: $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle$.
4. Make another copy of the previous result: $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle^{\otimes 2}$
5. Compute $\rho + 3\mathbf{c}_i \cdot \mathbf{p}$, $(3\mathbf{c}_i \cdot \mathbf{p})^2$, and $3\mathbf{p} \cdot \mathbf{p}$ and store them in different registers:
 $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle^{\otimes 2} |\rho + 3\mathbf{c}_i \cdot \mathbf{p}\rangle |(3\mathbf{c}_i \cdot \mathbf{p})^2\rangle |3\mathbf{p} \cdot \mathbf{p}\rangle$.
6. Compute $2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p})$ and store it in a register:
 $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle^{\otimes 2} |\rho + 3\mathbf{c}_i \cdot \mathbf{p}\rangle |(3\mathbf{c}_i \cdot \mathbf{p})^2\rangle |3\mathbf{p} \cdot \mathbf{p}\rangle |2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p})\rangle$
7. Compute the numerator and denominator in (4.17) and store them in two registers:
 $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle^{\otimes 2} |\rho + 3\mathbf{c}_i \cdot \mathbf{p}\rangle |(3\mathbf{c}_i \cdot \mathbf{p})^2\rangle |3\mathbf{p} \cdot \mathbf{p}\rangle |2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p})\rangle |\text{Nu.}\rangle |\text{De.}\rangle$
8. Copy the numerator state to another register:
 $|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle^{\otimes 2} |\rho + 3\mathbf{c}_i \cdot \mathbf{p}\rangle |(3\mathbf{c}_i \cdot \mathbf{p})^2\rangle |3\mathbf{p} \cdot \mathbf{p}\rangle |2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p})\rangle |\text{Nu.}\rangle^{\otimes 2} |\text{De.}\rangle$
9. Compute the division, which should output the quotient $\lfloor f_i^{\text{eq}} \rfloor$ in one of the numerator registers and a truncated remainder $\{f_i^{\text{eq}}\} := f_i^{\text{eq}} - \lfloor f_i^{\text{eq}} \rfloor$ in a new register.

The final state after the arithmetical process has the form

$$|\rho\rangle |p_x, p_y, p_z\rangle^{\otimes 2} |3\mathbf{c}_i \cdot \mathbf{p}\rangle^{\otimes 2} |\rho + 3\mathbf{c}_i \cdot \mathbf{p}\rangle |(3\mathbf{c}_i \cdot \mathbf{p})^2\rangle |3\mathbf{p} \cdot \mathbf{p}\rangle |2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p})\rangle |\text{Nu.}\rangle |\text{De.}\rangle |\lfloor f_i^{\text{eq}} \rfloor\rangle |\{f_i^{\text{eq}}\}\rangle \tag{4.18}$$

One might notice that we have access to the binary values of ρ and $\mathbf{p}$ in the unit of $\frac{2^r}{2\pi}$ from the quantum state (4.9). Performing the same procedure on the provided binary numbers still gives the the same equilibrium distribution value, except that it is also scaled by $\frac{2^r}{2\pi}$. We will later show this factor can be absorbed by the structure of the circuit that implements the streaming part.

The process to compute the non-linear equilibrium distribution outlined above is expensive in terms of qubit counts and high-fidelity quantum gates. We enumerate the number of qubits to perform each step of the computation. Assume (i) the lattice is 3-dimensional (ii) every number is stored in the two's complement representation (iii) each of the values ρ, p_x, p_y, p_z are stored using r qubits. We bound intermediate values using $\rho, |p_x|, |p_y|, |p_z| \leq 2^{r-1}$, from which suitable number of qubits can be allocated to store such values. Since common DnQm configurations have discrete velocities with $c_{i,\alpha} \in \{-1, 0, 1\}$, we can compute and store $3\mathbf{c}_i \cdot \mathbf{p}$ using $r+4$ qubits for $|3\mathbf{c}_i \cdot \mathbf{p}| \leq 9 \cdot 2^{r-1} < 2^{r+3}$. By bounding their values in a similar manner, we can store $\rho + 3\mathbf{c}_i \cdot \mathbf{p}$, $(3\mathbf{c}_i \cdot \mathbf{p})^2$, $3|\mathbf{p}|^2$, and $2\rho(\rho + 3\mathbf{c}_i \cdot \mathbf{p})$ using $r+4$, $2r+6$, $2r+3$, and $2r+4$ qubits, respectively. Estimating this number for the numerator and denominator requires the exact values of discrete weights w_i . We take the D3Q27 configuration as an example in which $\alpha_i \leq 8$ and $\beta_i \leq 256$. In that case, the numerator and the denominator are bounded by $832 \cdot 2^{2r-2}$ and $512 \cdot 2^{r-1}$, hence their binary forms requires $2r+9$ and $r+9$ qubits for storing, respectively. As a result, one also needs $2r+9$ qubits for the quotient and $r+9$ qubits for the remainder. To summarize, the

final state (4.18) consumes a total of $22r + 61$ qubits for storing the binary values. Note that the process would also need a couple of ancillary qubits to store sign-related factors during the computation. The ancillas, however, are reusable as they are always returned to $|0\rangle$ after each step.

4.2.4. Updating post-collision state

This part describes how to lift the post-collision distribution $f_i^* = (1 - \omega)f_i + \omega f_i^{\text{eq}}$, $\omega = \Delta t / \tau$, to the phase, creating new quantum states $\frac{1}{\sqrt{2}}(|0\rangle + e^{if_i^*}|1\rangle)$. These states will replace $\frac{1}{\sqrt{2}}(|0\rangle + e^{if_i}|1\rangle)$ as the initial data-encoded states in the succeeding simulation time. In order to ensure the reversibility of the QPE-based macroscopic quantity estimation and equilibrium distribution evaluation, the initial state $\frac{1}{\sqrt{2}}(|0\rangle + e^{if_i}|1\rangle)$ must stay intact after the update step. As the no-cloning theorem inhibits making copies of the initial state, we might be only able to encode $f_i^* = f_i^{\text{eq}}$ i.e. the post-collision distribution corresponding to the full relaxation $\omega = 1$. The new encoding can be carried out by controlled rotations whose angles are determined by the binary-fraction representation $|f_i^{\text{eq}}\rangle$. We formally describe the state update of one discrete velocity as the mapping

$$|+\rangle \left| \frac{2^r f_i^{\text{eq}}}{2\pi} \right\rangle \mapsto |\psi_i^*\rangle \left| \frac{2^r f_i^{\text{eq}}}{2\pi} \right\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{if_i^{\text{eq}}}|1\rangle) \left| \frac{2^r f_i^{\text{eq}}}{2\pi} \right\rangle \quad (4.19)$$

We illustrate an implementation of this mapping knowing no explicit value but the qubits that store $\frac{2^r f_i^{\text{eq}}}{2\pi}$ in [Figure 4.9](#).

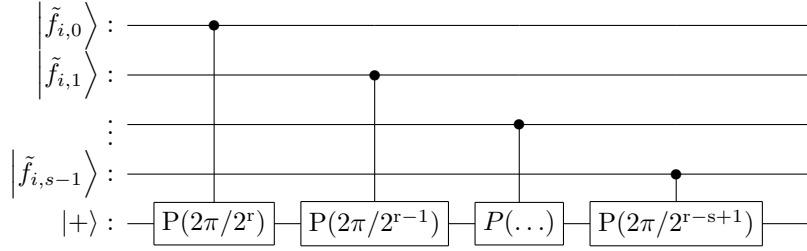


Figure 4.9: Quantum circuit that prepares the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{if_i}|1\rangle)$ from the s -bit binary-fraction $|\tilde{f}_i\rangle = \left| \frac{2^r f_i}{2\pi} \right\rangle$.

After creating the new post-collision state without destroying the initial state, we can free up the working space by uncomputing the phase estimations and arithmetic computations in reverse order. This would return the working-space qubits to $|0\rangle$. Omitting the free qubits, we have the entire system in the state

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}^*\rangle \quad (4.20)$$

Lastly, we can reset the N_v qubits carrying the initial state to $|0\rangle$. This requires measuring the qubits in the standard basis and apply a corrective NOT gate to qubits that have the measurement outcome 1. In that case, the correction also leaves a relative phase of $e^{if_{\mathbf{x},i}}$, yet they are irrelevant to our algorithm. The final state after collision update has the same form as the initial superposition, except measurement-dependent phases $\varphi_{\mathbf{x}}$.

$$\sum_{\mathbf{x}} e^{i\varphi_{\mathbf{x}}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}^*\rangle \quad (4.21)$$

4.2.5. Propagating distributions

Propagation, or streaming, is an essential component of any LBM algorithm. In the classical LBM setting, the streaming operation adjust the “position” of the distributions

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}\rangle \mapsto \sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x}+\mathbf{c}_i,i}\rangle \quad (4.22)$$

It was proved that basis encoding does not allow a unitary streaming operation [45]. Even though our tensor-product encoding is slightly different, the same problem persists. As we might consider distribution-encoded states $|\psi_{\mathbf{x},i}\rangle$ as arbitrary states on the Bloch sphere's equator, we can only transform the superposition in a meaningful way by applying gates on the coordinate register. In fact, we can stream every distribution along a fixed velocity $\mathbf{c}$ by adding $-\mathbf{c}$ to the coordinates.

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}\rangle \mapsto \sum_{\mathbf{x}} |\mathbf{x} - \mathbf{c}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}\rangle \equiv \sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x}+\mathbf{c},i}\rangle \quad (4.23)$$

It is clear that streaming every distribution along the same direction fails to satisfy the conservation of density and momentum. We propose two solutions to performing the propagation, each of which has its own limitation. Before diving into the solutions, we rewrite the quantum states as time-dependent.

Assume the current time is t . The ideal quantum state at time $t + 1$ is

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}^{(t+1)}\rangle = \sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle \quad (4.24)$$

The streaming quantum operator should be able to send the current post-collision states in their respective directions

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}^{*(t)}\rangle \xrightarrow{\text{streaming}} \sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle \quad (4.25)$$

First approach to streaming: space-time encoding The presence of every state $|\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle$ in the entanglement with $|\mathbf{x}\rangle$ is important for the simulation at time $t + 1$. The first solution ensures such presences with additional distribution-encoded qubits and related arithmetics. This approach is motivated by the fact that for every i , $|\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle$ is prepared using initial states $|\psi_{\mathbf{x}-\mathbf{c}_i,j}^{(t)}\rangle$ for $j = 1, \dots, N_v$. The initial states are also a streamed post-collision state at the previous time, $|\psi_{\mathbf{x}-\mathbf{c}_i,j}^{(t)}\rangle = |\psi_{\mathbf{x}-\mathbf{c}_i-\mathbf{c}_j,j}^{*(t-1)}\rangle$. Induced by the set $S = \{\mathbf{c}_i : i = 1, \dots, N_v\}$ of displacement vectors, the light cone of all possible positions up to time T is defined as

$$\mathcal{L}_T(S) = \bigcup_{t=0}^T \left\{ \sum_{j=1}^t \mathbf{c}_{i_j} : \mathbf{c}_{i_j} \in S \right\}, \quad (4.26)$$

where for $T = 0$ the sum is interpreted as the singleton set $\{\mathbf{0}\}$. The approach assumes the initial encoding

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{\mathbf{c} \in \mathcal{L}_T(S)} \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x}-\mathbf{c},i}^{(0)}\rangle \quad (4.27)$$

Then our original method can be adapted to perform the LBM simulation up to time T , i.e. to create the quantum state

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle \otimes \bigotimes_{i=1}^{N_v} |\psi_{\mathbf{x},i}^{(T)}\rangle \quad (4.28)$$

The fundamental problem with this approach is the number of qubits we need to store and process the initial distributions. As the size of the light cone $\mathcal{L}_T(S)$ can be as large as $O((N_v)^T)$, the encoding (4.27) requires $O((N_v)^{T+1})$ qubits in the velocity register initially. Preparing this state and applying any operation on this large number of qubits are generally impractical, unless we only simulate the system in a very short window T .

Second approach to streaming: approximating nearby distributions Recall that, at every time step, the quantum states $|\psi_{\mathbf{x}-\mathbf{c}_i,i}^*\rangle$ are encodings of the post-collision distributions $f_{\mathbf{x}-\mathbf{c}_i,i}^* = f_{\mathbf{x}-\mathbf{c}_i,i}^{\text{eq}}$. It is possible to approximate the equilibrium distribution using macroscopic information and equilibrium distribution at node $\mathbf{x}$ as given by the binary-fraction values stored in the registers $|\rho_{\mathbf{x}}\rangle$ $|p_{\mathbf{x},x}\rangle$ $|p_{\mathbf{x},y}\rangle$ $|p_{\mathbf{x},z}\rangle$ and $|f_{\mathbf{x},i}^{\text{eq}}\rangle$, respectively. We use the first-order Taylor expansion

$$f_i^{\text{eq}}(\mathbf{x} - \mathbf{c}_i) = f_i^{\text{eq}}(\mathbf{x}) - \mathbf{c}_i \cdot \nabla f_i^{\text{eq}}(\mathbf{x}) \quad (4.29)$$

The spatial gradient of the equilibrium distribution is evaluated by

$$\nabla f_i^{\text{eq}}(\mathbf{x}) = \frac{\partial f_i^{\text{eq}}}{\partial \rho} \nabla \rho + \frac{\partial f_i^{\text{eq}}}{\partial \mathbf{u}} \nabla \mathbf{u} \quad (4.30)$$

The gradients of density and velocity can be solved by two equations that involve the zero- and first-order moment of the *non-equilibrium part* $f_i^{\text{neq}} = f_i - f_i^{\text{eq}}$. By conservation of density and momentum,

$$\sum_i f_i^{\text{neq}} = 0, \quad \sum_i \mathbf{c}_i f_i^{\text{neq}} = \mathbf{0} \quad (4.31)$$

Alternatively, we can evaluate the moments using the first-order approximation of the non-equilibrium part in the Chapman-Enskog expansion,

$$f_i^{\text{neq}} = -\tau(\partial_t + \mathbf{c}_i \cdot \nabla) f_i^{\text{eq}} \quad (4.32)$$

Using the standard expression for evaluating f_i^{eq} (2.16) and rotational isotropy conditions (2.15), we estimate $\sum_i f_i^{\text{neq}}$ and $\sum_i \mathbf{c}_i f_i^{\text{neq}}$ as functions of $\nabla \rho$ and $\nabla \mathbf{u}$. The roots of these functions are approximate values of $\nabla \rho$ and $\nabla \mathbf{u}$. Substituting their values to the first-order Taylor expansion (4.29), we successfully approximate the post-collision state of a nearby neighbor, e.g. we obtain the binary representation of $|f_{\mathbf{x}-\mathbf{c}_i,i}^*\rangle$. Note that we have discussed previously that the state $|\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle$ can be prepared using the binary value $|f_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle$ with a number of controlled rotations. Since $|\psi_{\mathbf{x},i}^{(t+1)}\rangle = |\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle$, this procedure also replaces the usual streaming step for every state $|\psi_{\mathbf{x}-\mathbf{c}_i,i}^{*(t)}\rangle$ with $\mathbf{c}_i \neq \mathbf{0}$.

It remains a challenge to solve for the post-collision state using coherent quantum operators. We demonstrate in Appendix B that the approximate equations are linear in $\nabla \rho$ and $\nabla \mathbf{u}$, but solving the system of equations and computing follow-up quantities require fault-tolerant quantum linear solver. This approach requires intensive management and calibrations of arithmetic components that are beyond the scope of our work.

4.3. Amplitude-encoding-based algorithm

4.3.1. Encoding distributions

The particle distributions $f_{\mathbf{x},1}, \dots, f_{\mathbf{x},N_v}$ with a normalization constraint $\sum_i f_{\mathbf{x},i} = 1$ can be encoded using N_v qubits in the state

$$|\psi_{\mathbf{x}}\rangle = \sum_{i=1}^{N_v} \sqrt{f_{\mathbf{x},i}} |e_i\rangle \quad (4.33)$$

where $e_i \in \{0, 1\}^{N_v}$ is a one-hot vector with 1 at position i and 0 everywhere else. This embedding identifies every velocity or mode in the LBM configuration to a qubit and offer several benefits.

- First, it can be prepared using N_v rotations and controlled rotations. We present the preparation procedure in the next paragraph.
- Second, there are unitary operators called *excitation gates* that preserve the form of the state upon acting on the encoded state. That is, such operators map a state in $\mathcal{R} := \{\sum_i \alpha_i |e_i\rangle : \alpha_i \in \mathbb{R}\}$ to another state in $\mathcal{R}$. This is particularly helpful for designing a collision operator that does not alter the form of the encoding.

- Third, this encoding is equivariant to permutations of distributions. Suppose $(f_1, \dots, f_{N_v}) \mapsto (f_{\sigma(1)}, \dots, f_{\sigma(N_v)})$ for some permutation $\sigma \in S_{N_v}$. The action of the permutation on the Hilbert space can be represented by a unitary $W(\sigma)$ comprising of SWAP gates among qubits that map $|e_i\rangle \mapsto |e_{\sigma^{-1}(i)}\rangle$. The action of it on the encoded state is

$$W(\sigma) \left(\sum_{i=1}^{N_v} \sqrt{f_{\mathbf{x},i}} |e_i\rangle \right) = \sum_{i=1}^{N_v} \sqrt{f_{\mathbf{x},i}} |e_{\sigma^{-1}(i)}\rangle, \quad (4.34)$$

which is identical to $\sum_{i=1}^{N_v} \sqrt{f_{\mathbf{x},\sigma(i)}} |e_i\rangle$ upon a re-indexing of qubits. This symmetry-preserving property of the amplitude encoding is desirable as the collision operator in common LBM configurations also has certain permutation symmetries.

We describe the preparation of the quantum state $\sum_i \sqrt{f_i} |e_i\rangle$ from the initial basis state $|0\rangle^{\otimes N_v}$, where each $|e_i\rangle$ corresponds to a one-hot vector having exactly one qubit in the state $|1\rangle$.

The preparation proceeds iteratively. First, we apply a single-qubit rotation $R_y(\theta_1)$ on the first qubit, with the rotation angle given by $\theta_1 = 2 \arcsin(\sqrt{f_1})$, yielding the superposition state

$$\sqrt{f_1} |10 \dots 0\rangle + \sqrt{1-f_1} |00 \dots 0\rangle \quad (4.35)$$

Subsequently, for each qubit k (with $2 \leq k \leq N_v - 1$), we apply a multi-controlled $R_y(\theta_k)$ rotation conditioned on ever previous qubit being in state $|0\rangle$, where

$$\theta_k = 2 \arcsin \left(\sqrt{\frac{f_k}{1 - \sum_{j=1}^{k-1} f_j}} \right) \quad (4.36)$$

Each iteration transfers the appropriate amplitude from the residual state $|0 \dots 0\rangle$ to the state where only the k -th qubit is flipped to $|1\rangle$. After performing this controlled-rotation process sequentially on all $N_v - 1$ qubits, the final quantum state accurately encodes each amplitude $\sqrt{f_i}$ into the corresponding computational basis state $|e_i\rangle$. (The general preparation circuit is shown in [Figure 4.10](#).) This state then serves as the starting point for subsequent symmetry-respecting transformations in the quantum operator for collision.

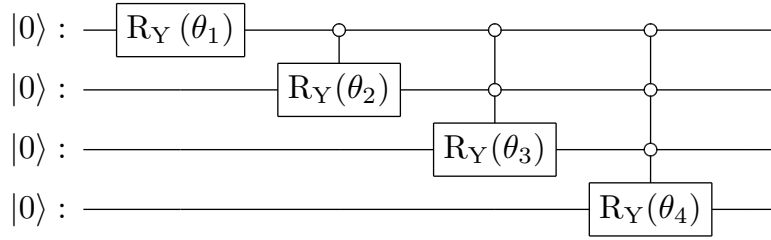


Figure 4.10: Quantum circuit to prepare states of the form $\sum_{i=1}^4 \sqrt{f_i} |e_i\rangle$. Rotation angles are computed by (4.36).

4.3.2. Feasibility of ideal collision operator for amplitude encoding

Collision as a fully unitary operator We examine the feasibility of simulating the collision phenomenon using a unitary operator that acts on amplitude-encoded states. Let $\mathcal{S} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^S$ be a dataset of involving the initial distributions and post-collision distributions. In particular, $\mathbf{x}^{(i)} = \left(\sqrt{f_1^{(i)}}, \dots, \sqrt{f_{N_v}^{(i)}} \right)$ with $\|\mathbf{x}^{(i)}\|_2 = 1$ encode the square root of initial distributions and $\mathbf{y}^{(i)} := \left(\sqrt{f_1^{\text{eq}(i)}}, \dots, \sqrt{f_{N_v}^{\text{eq}(i)}} \right)$ the corresponding equilibrium distributions. Suppose $U_{\text{col}} \in \mathbb{C}^{N_v \times N_v}$ be a unitary collision matrix, hence $U_{\text{col}} \mathbf{x}^{(i)} \approx \mathbf{y}^{(i)}$. It is natural to assume U_{col} to be the solution to optimization problem

$$\min_{\mathbf{U}: \mathbf{U}\mathbf{U}^\dagger = \mathbf{I}} \sum_{i=1}^S 1 - \left| \left(\mathbf{y}^{(i)} \right)^\dagger \mathbf{U} \mathbf{x}^{(i)} \right| \quad (4.37)$$

We will prove later that the solution must be a real unitary, or *orthogonal*, matrix up to a global phase. Ignoring the phase, the objective function can be rewritten in the least-square form using the identity $1 - |(\mathbf{y}^{(i)})^\dagger \mathbf{U} \mathbf{x}^{(i)}| = \frac{1}{2} \|\mathbf{U} \mathbf{x}^{(i)} - \mathbf{y}^{(i)}\|_2^2$. Let $\mathbf{X} = [\mathbf{x}^{(1)} \dots \mathbf{x}^{(S)}]$ and $\mathbf{Y} = [\mathbf{y}^{(1)} \dots \mathbf{y}^{(S)}]$ be data matrices in $\mathbb{R}^{N_v \times S}$. The optimization (4.37) is equivalent to

$$\min_{\mathbf{U}: \mathbf{U}\mathbf{U}^\dagger = \mathbf{I}} \frac{1}{2} \|\mathbf{U}\mathbf{X} - \mathbf{Y}\|_F^2 \quad (4.38)$$

This optimization problem is known as the *Orthogonal Procrustes Problem*. Upon expanding the expression, the solution can be found by maximizing $\text{Re Tr}(\mathbf{U}\mathbf{X}\mathbf{Y}^\dagger)$. Let $\mathbf{X}\mathbf{Y}^\dagger$ have the singular value decomposition $\mathbf{X}\mathbf{Y}^\dagger = \mathbf{R}\mathbf{\Sigma}\mathbf{Q}^\dagger$. The solution to the problem (4.37) is $S - \text{Tr}(\mathbf{\Sigma})$, which is achieved at $\mathbf{U}^* = \mathbf{Q}\mathbf{R}^\dagger$. As $\mathbf{X}$ and $\mathbf{Y}$ are real matrices, the left and right singular vectors of $\mathbf{X}\mathbf{Y}^\dagger$ also have real entries up to a global phase. The solution $\mathbf{U}^*$ is then a real unitary matrix. This unitary matrix can be made compatible with the amplitude encoding (4.33) by identifying standard basis vectors $\mathbf{e}_i \in \mathbb{R}^n$ with the corresponding one-hot binary vectors $|e_i\rangle$ as quantum states. For example,

$$\sum_{i=1}^{N_v} \sqrt{f_i} \mathbf{e}_i \mapsto \sum_{i=1}^{N_v} \sqrt{f_i} |e_i\rangle \quad (4.39)$$

We find the average fidelity error $1 - |(\mathbf{y}^{(i)})^\dagger \mathbf{U}^* \mathbf{x}^{(i)}|$ from numerical simulations to be $1 - \text{Tr}(\mathbf{\Sigma})/S \approx 0.04$. This error is well above numerical precision, hence we might argue that there is no unitary operator to perform the collision ideally.

Collision as a block-encoded operator Some papers, on the other hand, utilize block-encoding techniques to embed a non-unitary collision operator in a larger unitary [46, 47, 48, 49, 50]. We will examine whether this approach is possible to create an ideal collision operator. For the next paragraphs, we write $|x^{(i)}\rangle$ and $|y^{(i)}\rangle$ instead of $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ for clarity, even though they are not necessarily quantum states in a two-level system. The collision operator is block-encoded in a unitary that acts on $\mathbb{C}^{N_v \cdot 2^r}$, where r is the number of ancillary qubits. Let the large unitary act on $|a\rangle |x_i\rangle$, where $|a\rangle$ is some initial state of the ancillas. Assume the output state $|\hat{y}_i\rangle$ of the collision operator is obtained in the subspace $|0\rangle^{\otimes r}$ with the corresponding measurement probability λ_i . The entire output state can be written as

$$U |a\rangle |x_i\rangle = \sqrt{\lambda_i} |0\rangle^{\otimes r} |\hat{y}_i\rangle + \sqrt{1 - \lambda_i} |\varphi\rangle \quad (4.40)$$

where $|\varphi\rangle$ has the support on the orthogonal complement of $|0\rangle^{\otimes r}$. We introduce a similar optimization to (4.37) for the block-encoding case,

$$\begin{aligned} \min_{U: U U^\dagger = I} \sum_{i=1}^S 1 - |\langle y_i | \hat{y}_i \rangle| &= \min_{U: U U^\dagger = I} \sum_{i=1}^S 1 - \frac{1}{\sqrt{\lambda_i}} |\langle 0 |^{\otimes r} \langle y_i | U |a\rangle |x_i\rangle| \\ &= \min_{U: U U^\dagger = I} \sum_{i=1}^S 1 - \frac{1}{\sqrt{\lambda_i}} + \frac{1}{2\sqrt{\lambda_i}} \|U |a\rangle |x_i\rangle - |0\rangle^{\otimes r} |y_i\rangle\|^2 \end{aligned} \quad (4.41)$$

with probabilities $\lambda_i = \|(\langle 0 |^{\otimes r} \otimes I) U |a\rangle |x_i\rangle\|^2$. This is an optimization problem on the orthogonal manifold $\mathcal{O}_p = \{X \in \mathbb{R}^{p \times p} | XX^T = I_p\}$ with $p = N_v \cdot 2^r$ and, unlike the previous case, does not have an analytical solution. We obtain an optimal solution that we believe to be the global optimum using a gradient-based method called “landing algorithm”, a novel technique presented in [111]. The main difference of the landing algorithm with other optimization algorithms on manifolds is its retraction-free nature.

A retraction is a (not necessarily orthogonal) projection onto the manifold $\mathcal{O}_p$ and usually requires expensive computations. The landing algorithm still converges to a matrix U such that $\|UU^T - I\| = 0$ to numerical precision. The linear operator that acts as the collision operator can be obtained by restricting the unitary to a subspace, $(|0\rangle^{\otimes r} \otimes I)U(|a\rangle \otimes I)$.

We defer the details of numerical simulations to compute the performance of this method to [chapter 5](#). For $r = 2$ and ancillary state $|a\rangle = |+\rangle^{\otimes r}$, the mean error is $1 - |\langle y_i | \hat{y}_i \rangle| \approx 0.002$ with mean probability $\lambda_i \approx 0.74$. This numerical evidence demonstrates the possibility of using a block-encoded operator to perform the ideal collision. The resulting unitary can be implemented on the quantum circuit by Givens rotations by the identification [\(4.39\)](#) and technique in quantum optics called Clements's decomposition of universal interferometers [\[112, 113\]](#). Briefly speaking, a real unitary of size p can be decomposed into $p(p-1)/2$ single-excitation rotations whose common form is shown in [\(4.47\)](#).

To extract the desired output states, we perform a post-selection onto the subspace $|0\rangle^{\otimes r} |\hat{y}_i\rangle$ by measuring the ancillas in the computational basis. The probability of obtaining the desired outcome in round i is $\lambda_i < 1$. For a multi-round LBM simulation to succeed, each intermediate post-selection must succeed; otherwise, the resulting state deviates from the target and must be discarded, requiring the simulation to start from scratch. To mitigate this failure probability, we apply a few rounds of amplitude amplification to boost the likelihood of the “good” state $|0\rangle^{\otimes r} |\hat{y}_i\rangle$. Specifically, we employ the oblivious version [\[79\]](#) of the fixed-point amplitude amplification method from [\[77\]](#) to increase the success probability to at least $1 - \varepsilon$ for any small $\varepsilon > 0$. This involves L iterations of a modified Grover-like operator, each comprising one application of U , $U^\dagger$, and phase rotations on the ancillary qubits. In our context, it suffices to take

$$L \geq \frac{\log(2/\sqrt{\varepsilon})}{\sqrt{\lambda}} \quad (4.42)$$

where $\lambda \leq \min_i \lambda_i$ denotes a uniform lower bound on the individual post-selection probabilities, ensuring the total failure probability is below ε .

4.3.3. Symmetry-preserving collision operator

This section presents an alternative method to design a collision quantum operator based on conservations and symmetries of LBM configurations. The collision operator Ω describe the behavior of particle distributions at a collision event. Due to the symmetric pattern of the velocities and their weights, Ω carries certain physical properties that can be phrased as *invariance* and *equivariance* relations [\[114\]](#).

The invariances of density and momentum are discussed in the paragraph below [\(2.14\)](#). Denote f_i^* the post-collision state of velocity $\mathbf{c}_i$. The invariances are given by conservation equations

$$\sum_i (f_i - f_i^*) = 0, \quad \sum_i (f_i - f_i^*) \mathbf{c}_i = \mathbf{0} \quad (4.43)$$

The equivariance of scale, which is the first-order homogeneity with respect to $\mathbf{f} = (f_1, \dots, f_{N_v})$, is given by $\Omega(\lambda \mathbf{f}) = \lambda \Omega(\mathbf{f})$ for $\lambda \geq 0$. Lastly, the collision operator is equivariant to rotations and reflections of the configurations. For example, the collision operator in the D2Q9 model is equivariant with respect to the action of the dihedral group $D_8 = \langle r, s | r^8 = s^2 = e, srs = r^{-1} \rangle$, which constitute the symmetries of a regular octagon. The equivariance reads

$$\Omega(\sigma \cdot \mathbf{f}) = \sigma \cdot \Omega(\mathbf{f}), \quad \forall \sigma \in D_8, \quad (4.44)$$

where the group actions in the left-hand side and right-hand side apply on the pre-collision and post-collision distributions, respectively. An immediate application of the equivariance is the generation of new data from a few through symmetry-defined permutations $(\mathbf{f}, \mathbf{f}^*) \mapsto \{(\sigma \cdot \mathbf{f}, \sigma \cdot \mathbf{f}^*) : \sigma \in D_8\}$. In the 3-dimensional model D3Q27, the symmetry group of collision contains 48 elements that form symmetries of an octahedral.

Taking D2Q9 as a working example, we consider how to translate those invariances and equivariances to the language of quantum operators. First, the equivariance of scale always holds as the amplitude

encoding assumes normalized distribution. Furthermore, since the collision is realized by a unitary operator, which is norm-preserving and therefore secures the invariance of density. The equivariance of $\sigma \in D_8$ with respect to its unitary representation $W(\sigma)$ is defined by the commutativity

$$U_{\text{col}}W(\sigma) = W(\sigma)U_{\text{col}}, \quad \forall \sigma \in D_8, \quad (4.45)$$

where U_{col} represents the unitary collision operator acting on the Hilbert space of the encoded quantum state (4.33). If a Hermitian operator H satisfies the commutation $[H, W(\sigma)] = 0$, the time evolution of it also has the same commutation, i.e. $[e^{itH}, W(\sigma)] = 0$, $\theta \in \mathbb{R}$. Then U_{col} might be realized as time evolutions of symmetry-preserving Hermitian operators H such that $[H, W(\sigma)] = 0$ for all $\sigma \in D_8$. Such an operator can be obtained by projecting any Hermitian operator H_0 onto the D_8 -invariant subspace. This is accomplished by twirling it with respect to the representation W of D_8 . The concept of twirling and its properties are discussed in subsection 2.3.3. In our context,

$$H = \mathcal{T}_{D_8}(H_0) = \frac{1}{|D_8|} \sum_{\sigma \in D_8} W(\sigma)H_0W(\sigma)^\dagger \quad (4.46)$$

Lastly, the invariance of momentum is not assured by our encoding scheme and quantum collision operator. However, we might let the collision operator to “learn” the invariance by optimizing the evolution time t . It is natural to require the quantum collision operator to preserve the realness of the amplitudes. Our encoding resembles a single-particle state in N_v fermionic orbital modes in the Hartree-Fock basis. Research in quantum chemistry often employs a class of particle-preserving unitaries made of single-excitation and double-excitation rotations[115]. Also referred to as Givens rotations, those unitaries are rotations in 2-dimensional subspaces with a fixed number of particles. As suggested by their names, single- and double-excitation rotations preserve the presence of one and two particles in the subspaces they act on, respectively.

$$\begin{aligned} G^{(1)}(\theta) &= \left(\cos \frac{\theta}{2} |01\rangle + \sin \frac{\theta}{2} |10\rangle \right) \langle 01| + \left(-\sin \frac{\theta}{2} |01\rangle + \cos \frac{\theta}{2} |10\rangle \right) \langle 10| \\ G^{(2)}(\theta) &= \left(\cos \frac{\theta}{2} |0011\rangle + \sin \frac{\theta}{2} |1100\rangle \right) \langle 0011| + \left(-\sin \frac{\theta}{2} |0011\rangle + \cos \frac{\theta}{2} |1100\rangle \right) \langle 1100| \end{aligned} \quad (4.47)$$

The rotations are time evolutions $G^{(i)}(\theta) = \exp(i\theta H^{(i)})$, $i = 1, 2$ of their generating Hermitian operators

$$\begin{aligned} H^{(1)} &= \frac{1}{4}(XY - YX) \\ H^{(2)} &= \frac{1}{16}(XXXXY + XXYX - XYXX + XYYY - YXXX + YXYX - YYXY - YYYX) \end{aligned} \quad (4.48)$$

Let subscripts k, l indicate the qubits (from 1 to N_v) the operators act on. As the encoded state $|\psi_{\mathbf{x}}\rangle = \sum_i \sqrt{f_{\mathbf{x},i}} |e_i\rangle$ has Hamming weight 1, double-excitation rotations act trivially. Meanwhile, single-excitation rotations can be seen as interactions between two qubits, which average to zero upon twirling unless an interaction is between the stationary mode (assume $c_1 = 0$) and any non-stationary mode ($c_k \neq 0$ for $k \neq 1$) in the D2Q9 model.

$$\begin{aligned} \mathcal{T}_{D_8}(H_{k,l}^{(1)}) &= \frac{1}{4|D_8|} \sum_{\sigma \in D_8} W(\sigma)(X_k Y_l - Y_k X_l)W(\sigma)^\dagger \\ &= \frac{1}{64} \sum_{\sigma \in D_8} X_{\sigma(k)} Y_{\sigma(l)} - Y_{\sigma(k)} X_{\sigma(l)} \\ &= \begin{cases} 0, & k, l \neq 1 \\ \pm \frac{1}{8} \sum_{i \neq 1} H_{1,i}^{(1)}, & k = 1 \text{ or } l = 1 \end{cases} \end{aligned} \quad (4.49)$$

That is, twirling single-excitation rotations gives rise to only one operator which captures the interaction between the stationary mode and every other non-stationary mode. The collision operator $U_{\text{col}} \equiv U_{\text{col}}(\theta)$ is now parametrized with a single parameter θ in $\exp\left(i\theta \sum_{i \neq 1} H_{1,i}^{(1)}\right)$, which is implemented by the trotterization

$$\exp\left(i\theta \sum_{i \neq 1} H_{1,i}^{(1)}\right) = \lim_{M \rightarrow \infty} \left[\prod_{i \neq 1} \exp\left(i \frac{\theta}{M} H_{1,i}^{(1)}\right) \right]^M = \lim_{M \rightarrow \infty} \left[\prod_{i \neq 1} G_{1,i}^{(1)}\left(\frac{\theta}{M}\right) \right]^M \quad (4.50)$$

It is evident that a single parameter cannot capture the collision phenomenon in which the equilibrium state alone is determined by at least two physical quantities, namely the density and the velocity field. Furthermore, the mathematical expressions of equilibrium distributions are second-order polynomials in the velocity field $\mathbf{u}$. Applying $U_{\text{col}}(\theta)$ on $|\psi_{\mathbf{x}}\rangle$ can only create new amplitudes that are linear combinations of $\sqrt{f_{\mathbf{x},i}}$, which are incapable to capture the second-order component of the collision. As a result, one might introduce higher-order terms via the encoding

$$\sum_{\mathbf{x}} |\mathbf{x}\rangle |\psi_{\mathbf{x}}\rangle |\psi_{\mathbf{x}}\rangle = \sum_{\mathbf{x}} |\mathbf{x}\rangle \left(\sum_i \sqrt{f_{\mathbf{x},i}} |e_i\rangle \right)^{\otimes 2} \quad (4.51)$$

The new encoding now has Hamming weight 2 that enables meaningful applications of double-excitation rotations within $|\psi_{\mathbf{x}}\rangle^{\otimes 2}$ in addition to single-excitation rotations.

When a sequence of twirled rotations forms a parametrized unitary $U(\theta)$, the unitary apparently satisfies the equivariance condition

$$U(\theta)W(\sigma) = W(\sigma)U(\theta), \quad \forall \sigma \in D_8, \quad (4.52)$$

This makes it a valid symmetry-preserving ansatz for the collision operator $U_{\text{col}}(\theta)$. The parameters are selected by maximizing the overlap of pre-collision state $|\psi\rangle = \sum_i \sqrt{f_i} |e_i\rangle$ and the post-collision state $|\psi^*\rangle = \sum_i \sqrt{f_i^*} |e_i\rangle$ over a generated dataset $\mathcal{S} = \{(\mathbf{f}, \mathbf{f}^*)\}$ where $\mathbf{f}^*$ are post-collision distributions from incoming distributions $\mathbf{f}$. The optimal parameters implement the collision operator by $U_{\text{col}} = U_{\text{col}}(\theta^{\text{opt}})$ with $\theta^{\text{opt}} = \arg \min \mathcal{L}(\theta)$ for the loss function

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{f}, \mathbf{f}^*) \in \mathcal{S}} 1 - |\langle \psi^* |^{\otimes 2} U_{\text{col}}(\theta) | \psi \rangle^{\otimes 2}|^2 \quad (4.53)$$

4.3.4. Collision ansatz design

Even after we project every generator onto the symmetry-invariant subspace by twirling, the operator pool that survives is still enormous. All symmetry-adapted single-excitation rotations $\tilde{G}^{(1)}(\theta)$ and double-excitation rotations $\tilde{G}^{(2)}(\theta)$ remain available, where the tildes indicate that each rotation has been twirled. Because specifying the collision ansatz architecture by an a priori fixed sequence of such gates is impractical, we grow the circuit adaptively in the spirit of ADAPT-VQE [116, 117].

At every step we consider the operator pool $\mathcal{P}$ of all twirled single- and double-excitation rotations. Starting from the current operator $U_{\text{col}}(\theta^*)$, we evaluate the magnitude of the loss-function derivative with respect to the parameter of each candidate operator $\tilde{G} \in \mathcal{P}$.

$$g_{\tilde{G}} := \left| \frac{\partial \mathcal{L}((\theta^*, \theta))}{\partial \theta} \right|_{\theta=0} \quad (4.54)$$

where loss function now takes $\tilde{G}(\theta) U_{\text{col}}(\theta^*)$ as the collision operation. New rotations to be inserted to the ansatz are sampled from the normalized distribution of $(g_{\tilde{G}})_{\tilde{G} \in \mathcal{P}}$. As new operators are being added to the ansatz, the entire set of parameters should be reoptimized.

In the end of the optimization, we prune redundant gates to control circuit depth. We look for rotations whose current angles fall below a pruning tolerance $\varepsilon_{\text{prune}}$. The following pruning strategy is adopted

from [118]. As newly added rotations have a tendency to have small angles, we bias the pruning towards rotations near the start of the ansatz. To balance between the magnitude and the relative position of a rotation, we introduce a decision factor for each operator.

$$f_i = F_1(\theta_i)F_2(x_i) \quad (4.55)$$

where $x_i := i/N$ is the relative position of the rotation in an ansatz with N operators and θ_i the rotation angle. We emphasize operators with small rotations and low tendency to change

$$F_1(\theta_i) = \frac{1}{\theta_i^2} \left| \frac{\partial \mathcal{L}}{\partial \theta_i} \right|^{-1} \quad (4.56)$$

We also prioritize the removal of operators that appear early in the ansatz, which accounts for the small magnitude of newly added rotations near the convergence regime, by a decaying function $F_2 = e^{-\alpha x_i}$. As the angles and the respective gradients are available after optimizing the loss function, the evaluation of all decision factors f_i can be done in $O(N)$ time. We identify candidates for removal by operators with the largest factors f_i . We only remove candidate operators with magnitude below the threshold $\varepsilon_{\text{prune}}$, which features the average magnitude of the N_L most recently added operators

$$\varepsilon_{\text{prune}} = \frac{0.1}{N_L} \sum_{i=0}^{N_L-1} |\theta_{N-i}| \quad (4.57)$$

The removal is performed after every grow-optimize iteration, as outlined in Algorithm 1. We might proceed the next iteration without reoptimizing parameters as eliminating these small-magnitude operators has a negligible effect on the ansatz.

Algorithm 1 Adaptive collision ansatz

Input: number of new operators each step N_L .

Initialize operator pool $\mathcal{P}$ and initial ansatz $U_{\text{col}}(\theta)$.

repeat

 Compute gradients $g_{\tilde{G}} = |\partial \mathcal{L} / \partial \theta|_{\theta=0}$ for every pool operator $\tilde{G} \in \mathcal{P}$

 Sample N_L operators from $\mathcal{P}$ with probability $\sim g_{\tilde{G}}$.

 Add new operators to the ansatz and extend parameters $\theta \leftarrow (\theta, 0, \dots, 0)$.

 Optimize ansatz parameters $\theta \leftarrow \arg \min \mathcal{L}(\theta)$ with the loss function (4.53).

 Compute the decision factors f_i (4.55) for all ansatz operators.

 Select N_L operators $\tilde{G}_j$ with the largest f_j values.

 Compute the pruning threshold $\varepsilon_{\text{prune}}$ (4.57).

if $|\theta_j| < \varepsilon_{\text{prune}}$ **then.**

 Remove operator $\tilde{G}_j$ from the ansatz and remove θ_j from θ .

end if

 Evaluate the loss function value $\mathcal{L}(\theta)$.

until convergence

Output: optimized collision operator $U_{\text{col}}(\theta)$ and mean fidelity $1 - \mathcal{L}(\theta)$.

4.3.5. Unitary streaming

A quantum implementation of the streaming step that is compatible to the amplitude encoding performs the transformation

$$\sum_{\mathbf{x}} \sum_{i=1}^{N_v} |\mathbf{x}\rangle \sqrt{f_{\mathbf{x},i}} |e_i\rangle \mapsto \sum_{\mathbf{x}} \sum_{i=1}^{N_v} |\mathbf{x}\rangle \sqrt{f_{\mathbf{x}+\mathbf{c}_i,i}} |e_i\rangle = \sum_{\mathbf{x}} \sum_{i=1}^{N_v} |\mathbf{x} - \mathbf{c}_i\rangle \sqrt{f_{\mathbf{x},i}} |e_i\rangle \quad (4.58)$$

Since $|e_i\rangle$ has exactly one 1-bit at the position i , the described operator can be realized by unitaries that perform addition of $-\mathbf{c}_i$ in the spatial-coordinate register controlled on qubit i of the velocity register.

We further separate the coordinate into spatial axes $|\mathbf{x}\rangle = |x\rangle |y\rangle |z\rangle$. Then the streaming operator can be expressed by the unitary

$$U_{\text{str}} = \prod_{i=1}^{N_v} [\text{ADD}(-c_{i,x}) \otimes \text{ADD}(-c_{i,y}) \otimes \text{ADD}(-c_{i,z}) \otimes |1\rangle\langle 1|_i + I^{\otimes 3} \otimes |0\rangle\langle 0|_i] \quad (4.59)$$

Since every velocity $\mathbf{c}_i$ has components $c_{i,x}, c_{i,y}, c_{i,z} \in \{0, 1, -1\}$ in common $DnQm$ models, the operator can be performed by the controlled version of arithmetic circuits INCREMENT_1 and INCREMENT_{-1} illustrated in Figure 4.6. Note that in the presence of two copies of $|\psi_{\mathbf{x}}\rangle$, which is the case for the encoding (4.51), the streaming operator is equivalent to applying two U_{str} operators sequentially, each on a copy of $|\psi_{\mathbf{x}}\rangle$ and the shared coordinated register $|\mathbf{x}\rangle$.

4.3.6. Multi-round LBM simulations

From the theoretical complexity and numerical performance that will be presented in chapter 5, we find that implementing the collision operator as a block-encoded operator is the most feasible option. We propose a pipeline for multi-round LBM simulations in Figure 4.11, which combines amplitude encoding, block-encoded collision (optimization and application), and unitary streaming.

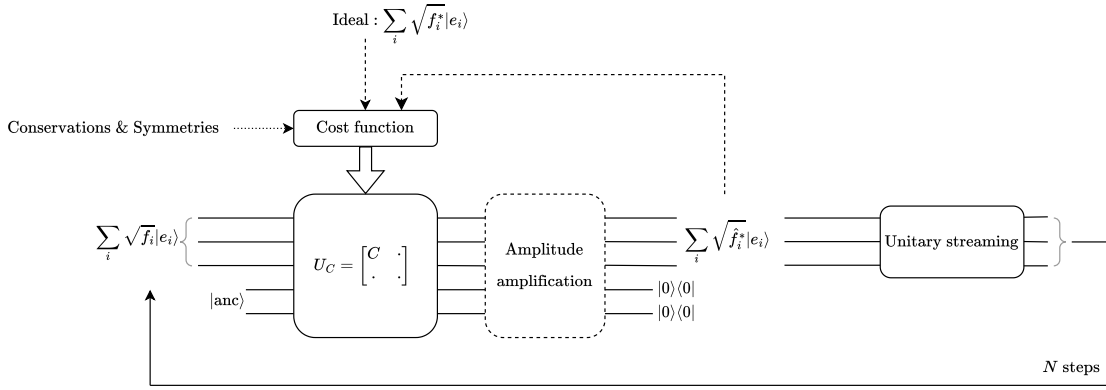


Figure 4.11: Quantum algorithm for the LBM using amplitude encoding. The process begins with the construction of a block-encoded collision unitary, obtained by optimizing a cost function over the orthogonal manifold with respect to physical constraints such as conservation laws and lattice symmetries. In the main simulation loop, each step applies the collision unitary, followed by optional oblivious amplitude amplification to boost post-selection success probability. The ancillas are then measured, and upon successful post-selection, a unitary streaming operation shifts the encoded distributions across lattice sites.

4.4. Error analysis of quantum Lattice Boltzmann simulations

4.4.1. Simulation using tensor-product encoding

The performance of our quantum Lattice Boltzmann method is characterized by three layers of approximation error, the estimation error from phase estimation protocols, the arithmetic error in computing equilibrium distributions, and the error that occurs in streaming.

We first consider the phase estimation procedure used to extract the particle density ρ and momenta p_α for $\alpha \in \{x, y, z\}$. Let $\hat{\rho}$ and $\hat{p}_\alpha$ denote the estimated quantities. If the protocol uses r principal precision qubits, the accuracy of estimation satisfies:

$$\Pr(|\hat{\rho} - \rho| < \delta_{\text{est}}) \geq 1 - \varepsilon_{\text{col}}, \quad \Pr(|\hat{p}_i - p_i| < \delta_{\text{est}}) \geq 1 - \varepsilon_{\text{col}} \quad (4.60)$$

with resolution $\delta_{\text{est}} = 2^{-(r+1)}$. When the tapered protocol introduces an additional r_{taper} ancillary qubits to improve fault tolerance, the failure probability ε_{col} can be bounded by

$$\varepsilon_{\text{col}} \leq \min \left\{ 8 \exp \left[-\frac{2R_0 - 1}{\frac{2}{\pi^2} \log(4R)} \right], 10 \exp \left[-\frac{2R_0 - 6}{\frac{2}{\pi^2} \log(100R_0 + 75)} \right] \right\} \quad (4.61)$$

for $R = 2^{r+r_{\text{taper}}}$ and $R_0 = 2^{r_{\text{taper}}-1} - 1$ [27].

Next, arithmetic subroutines discussed in subsection 4.2.3 are used to compute the equilibrium distribution $f_i = f_i(\rho, \mathbf{p})$ via the expression (4.16). These circuits assume the availability of sufficient ancilla qubits to implement fixed-point arithmetics with QFT-based operations. The output estimates $\hat{f}_i^{\text{eq}}$ satisfy

$$\Pr(|\hat{f}_i^{\text{eq}} - f_i^{\text{eq}}| < C\delta_{\text{est}} =: \delta_{\text{col}}) \geq 1 - \varepsilon_{\text{col}} \quad (4.62)$$

for a constant C that depends on the bit-width of intermediate arithmetic operations and the truncation of binary division (4.17). This bound ensures that the total error incurred in the quantum LBM pipeline remains within tolerable limits for practical fluid simulation.

Finally, the streaming step implements a shift of the distribution components along discrete velocity directions. We have discussed two alternative approaches, space-time encoding and approximate streaming, to address the non-unitary nature of streaming in subsection 4.2.5. When implemented with finite resources such as using registers with bounded precision, the streaming operation incurs its own implementation error. Let $\hat{f}_i^{\text{stream}}$ denote the distribution component after the streaming operation. Assume we can bound the streaming error by

$$\Pr(|\hat{f}_i^{\text{stream}} - f_i^{\text{stream}}| < \delta_{\text{str}}) \geq 1 - \varepsilon_{\text{str}} \quad (4.63)$$

where δ_{str} captures the maximal deviation due to imperfect shift operations, and ε_{str} is the corresponding failure probability. The values of δ_{str} and ε_{str} depend on the choice of propagation method and internal settings. For example, space-time encoding enables a perfect streaming, i.e. $\delta_{\text{str}} = 0$ and $\varepsilon_{\text{str}} = 0$ at the cost of large qubit use. The approximate streaming, on the other hand, incurs errors from the Chapman–Enskog expansion and the numerical precision within a quantum solver and a probability of failure from the algorithm of the quantum solver.

Assuming the error events are independent and each round consists of these two steps, we define the total pointwise error per round as

$$\delta_{\text{round}} := \delta_{\text{col}} + \delta_{\text{str}}, \quad \varepsilon_{\text{round}} := \varepsilon_{\text{col}} + \varepsilon_{\text{str}} \quad (4.64)$$

Then, over N rounds of simulation, the total pointwise error accumulates linearly in the worst case, and the failure probability accumulates additively by the union bound. Therefore, the final distribution $\hat{f}_i^{(N)}$ after N rounds satisfies

$$\Pr(|\hat{f}_i^{(N)} - f_i^{(N)}| < N\delta_{\text{round}}) \geq 1 - N\varepsilon_{\text{round}} \quad (4.65)$$

This worst-case bound ensures robustness under sequential updates and provides a guideline for choosing initial roundwise precision such that the final error stays within desired tolerance Δ , i.e., by setting

$$\delta_{\text{round}} \leq \frac{\Delta}{N}, \quad \varepsilon_{\text{round}} \leq \frac{\varepsilon_{\text{total}}}{N} \quad (4.66)$$

for target accuracy Δ and total failure probability $\varepsilon_{\text{total}}$.

4.4.2. Simulation using amplitude encoding

In the amplitude encoding approach, the sole source of error arises from the non-perfect nature of the employed collision operator. For a block-encoded collision operator, since the output state must be post-selected on an ancilla subspace, the procedure is inherently probabilistic. To mitigate this, we apply oblivious amplitude amplification, specifically the fixed-point method, to amplify the success probability to at least $1 - \varepsilon_{\text{col}}$ for a small failure probability bound $\varepsilon_{\text{col}} > 0$, while ensuring that the desired output

state is preserved within acceptable error bounds. (See the last paragraph of [subsection 4.3.2](#)). Recall that both the variational method in [Algorithm 1](#) and the unitary optimization in [\(4.41\)](#) find the collision operator by maximizing the fidelity of the estimate $\sum_i (\hat{f}_i^{\text{eq}})^{1/2} |e_i\rangle$ and the target state $\sum_i (f_i^{\text{eq}})^{1/2} |e_i\rangle$. A classical distance between distributions $\mathbf{f}^{\text{eq}} = (f_1^{\text{eq}}, \dots, f_{N_v}^{\text{eq}})$ and $\hat{\mathbf{f}}^{\text{eq}} = (\hat{f}_1^{\text{eq}}, \dots, \hat{f}_{N_v}^{\text{eq}})$ is called *total variation* and is defined by

$$d_{\text{TV}}(\hat{\mathbf{f}}^{\text{eq}}, \mathbf{f}^{\text{eq}}) = \frac{1}{2} \sum_{i=1}^{N_v} |f_i^{\text{eq}} - \hat{f}_i^{\text{eq}}| \quad (4.67)$$

We can relate the total variation distance to quantum fidelity $F = \sum_i |f_i^{\text{eq}} \hat{f}_i^{\text{eq}}|^{1/2}$ by the inequalities

$$1 - F \leq d_{\text{TV}}(\hat{\mathbf{f}}^{\text{eq}}, \mathbf{f}^{\text{eq}}) \leq \sqrt{1 - F^2} \quad (4.68)$$

Since we have optimized the collision operator with randomly sampled input distributions, the error bounds do not hold for individual distributions but for the expectation value over the space of distributions

$$\Pr(\mathbb{E}[d_{\text{TV}}(\hat{\mathbf{f}}^{\text{eq}}, \mathbf{f}^{\text{eq}})] < \delta_{\text{col}}) \geq 1 - \varepsilon_{\text{col}} \quad (4.69)$$

for a value $\delta_{\text{col}} > 0$ incurred by the fidelity error, e.g., the results in [Figure 5.1](#). Although the error bound can also be generalized to N rounds of simulation in a similar manner to [\(4.65\)](#), the actual error can become saturated as the system converges to its steady state. [Figure 5.3](#) illustrates two scenarios: one where the error saturates, and another where it diverges.

The proposed quantum algorithm offers a potential exponential memory advantage over classical lattice Boltzmann solvers by leveraging amplitude encoding. In a classical setting, representing the particle distribution functions $f_i(\mathbf{x})$ across M lattice sites and N_v discrete velocities requires $\mathcal{O}(MN_v)$ memory. In contrast, our amplitude encoding strategy stores this data in the amplitudes of a quantum state over $\lceil \log_2 M \rceil + N_v$ qubits. Note that we might need further r (usually $r \leq 2$) ancillary qubits as they are reusable after each simulation step.

The quantum collision step via block-encoded unitary $U_C \in \mathbb{R}^{p \times p}$ for $p = 2^r N_v$ can be implemented by a sequence of $p(p-1)/2$ two-qubit gates [\[112\]](#). Suppose the amplitude amplification subroutine applies L layers of a Grover-like operator which involve two queries to U_C and $U_C^\dagger$. The fixed-point strategy in [\[77\]](#) is capable of increasing the post-selection success probability to $\geq 1 - \varepsilon$ for $\varepsilon > 0$ for any $L \geq \frac{\log(2/\sqrt{\varepsilon})}{\sqrt{\lambda}}$, where λ is the initial success probability as demonstrated in [Figure 5.1](#). The measurement of ancillas only induces a small complexity, assuming r is small. Lastly, implementing the unitary streaming operator on our encoding requires $\mathcal{O}(N_v^2)$ CNOT gates. The total number of two-qubit gates for each step of the simulation is

$$\mathcal{O}\left(\frac{p(p-1) \log(2/\sqrt{\varepsilon})}{\sqrt{\lambda}}\right) + \mathcal{O}(N_v^2) = \mathcal{O}\left(\frac{N_v^2 4^r \log(2/\sqrt{\varepsilon})}{\sqrt{\lambda}}\right). \quad (4.70)$$

Numerical Simulations and Analysis

5.1. Performance of collision operators by optimization on the orthogonal manifold

We simulate the optimization (4.41) with different number of ancillas to determine whether the collision operator can be implemented by a block-encoding unitary. Figure 5.1 presents the fidelity error of the resulting operator, along with the mean success probability. The figure also includes experiments where two copies of $|\psi\rangle = \sum_i \sqrt{f_i} |e_i\rangle$ are concatenated together to make the data-carrying initial state.

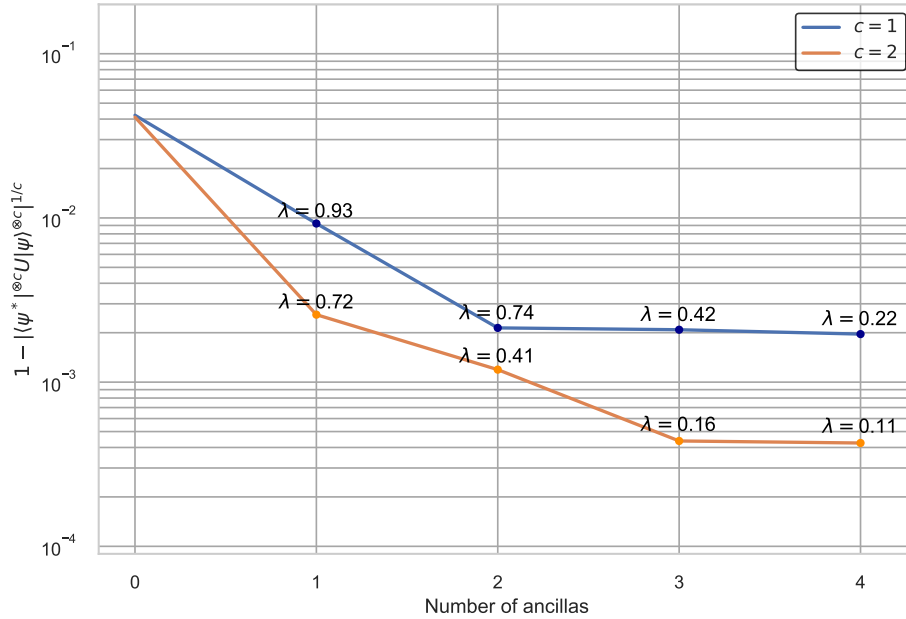


Figure 5.1: Mean fidelity error per copy of the output state of the optimal unitary collision. The optimization becomes finding the best (block-encoded) unitary to map $|+\rangle^{\otimes r} |x\rangle^{\otimes c}$ to $|0\rangle^{\otimes r} |y\rangle^{\otimes c}$ with r being the number of ancillas. Shown values $\lambda = \|(\langle 0|^{\otimes r} \otimes I) U |a\rangle |x\rangle\|^2$ are the probability of measuring the ancillas in the state $|0\rangle^{\otimes r}$ for post-selection. The case $r = 0$ correspond to a fully unitary collision operator, hence $\lambda = 1$.

5.2. Performance of symmetry-preserving collision ansatzes

We examine the training process and convergence of the unitary collision created by Algorithm 1 when the operator pool is created by twirling excitation rotations (4.47) with respect to the symmetry group of the D2Q9 model. The corresponding encodings on 9 qubits of mesotropic distributions are $|\psi\rangle =$

$\sum_{i=1}^9 \sqrt{f_i} |e_i\rangle$ and $|\psi^*\rangle = \sum_{i=1}^9 \sqrt{f_i^*} |e_i\rangle$ for the initial state and the post-collision state, respectively. (See the description of (4.33) for related assumptions and notations.). The parametrized unitary might act on either 9 or 18 qubits; in the latter case, two copies of the states, i.e., $|\psi\rangle^{\otimes 2}$ and $|\psi^*\rangle^{\otimes 2}$, are used as the initial state and target state. Note that the operator pool has size $|\mathcal{P}| = 105$ in the latter case, while the former case has only $|\mathcal{P}| = 1$ twirled operator, which makes the algorithm converge after one iteration. The quality of the collision unitary, measured by the mean fidelity error per copy, is reported in Figure 5.2.

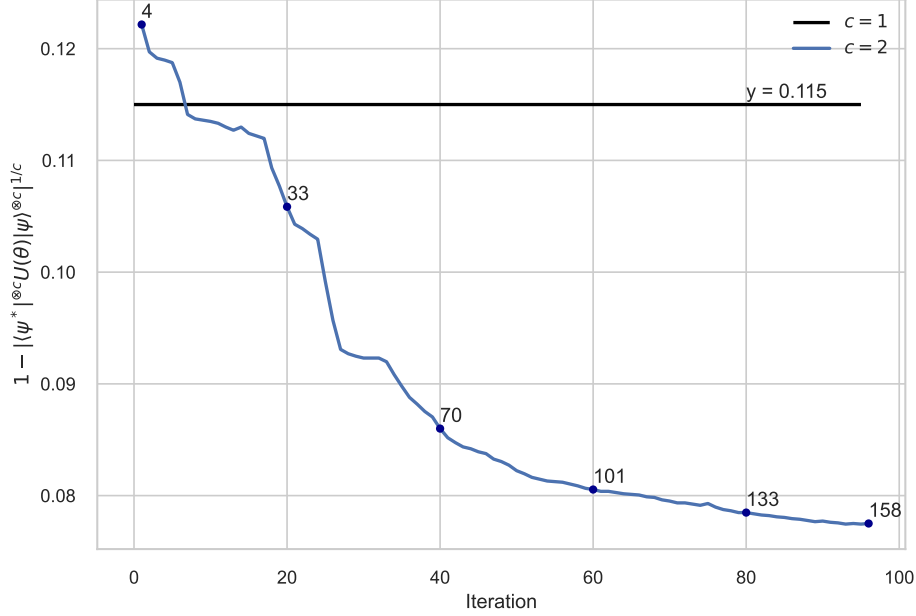


Figure 5.2: Mean fidelity error per copy of the output states of the parametrized collision operator. With $c = 1$ copy of the given states, the ansatz contains exactly one operator. With $c = 2$, the growing and pruning strategies described in Algorithm 1 apply to balance the ansatz performance and its size. At every iteration, the algorithm selects $N_L = 2$ operators to insert and also removes at most N_L existing operators from the circuit. The numbers above small dots indicate the number of operators in the ansatz at an iteration.

5.3. Simulations of the Quantum Lattice Boltzmann Method

From the numerical results presented in the previous sections, we find the collision operators via optimization-on-manifold give excellent performance using a reasonable amount of quantum resources. Therefore, we decide to apply the resulting block-encoding unitary (that corresponds to $c = 1$ and $r = 2$) for follow-up LBM simulations.

Figure 5.3 illustrate the fluid flow and the fidelity error for the evolution of two initial density distributions. Depending on the initial density distribution, the quantum algorithm can maintain high-fidelity outputs or experience a diverging error. A similar numerical simulation (up to 10 steps) for the Gaussian distribution (subfigures (c) and (d)) was presented in a recent work [57] also using amplitude encoding. At step $t = 10$, our method achieves an error that is smaller by a factor of approximately 7 to 8 compared to their method.

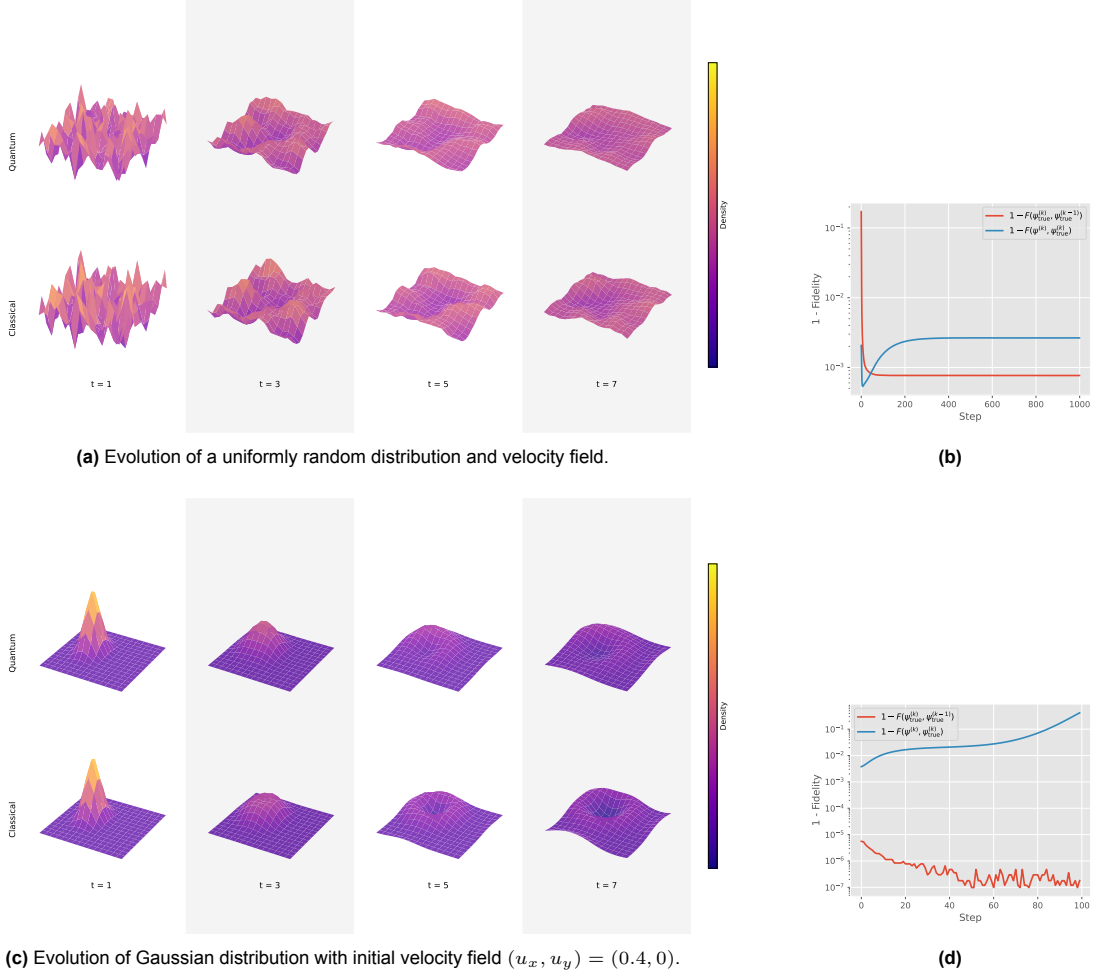


Figure 5.3: Simulation of the quantum algorithm for the lattice Boltzmann method with amplitude encoding strategy for the D2Q9 model on a 16×16 lattice. Every simulation step includes an application of the 2-ancilla block-encoded collision operator (with successful post-selection) followed by exact unitary streaming. **(a,c)** Time evolution of an initial density distribution (top: quantum algorithm, bottom: ideal simulation). **(b,d)** Convergence and error of the quantum algorithm, measured in terms of fidelity, $F(\psi, \phi) = |\langle \psi | \phi \rangle|$. The red lines present the convergence of the ideal flows measure by its change at a step. The blue lines show the fidelity error of the output by the quantum algorithm with respect to the ideal flow.

Discussion and Conclusion

This thesis has explored quantum algorithms for simulating fluid dynamics through the lattice Boltzmann method (LBM), with a particular focus on designing efficient quantum circuits for streaming and collision steps. Central to our quantum algorithms for the LBM are two encoding schemes for representing particle distribution functions as quantum states: amplitude encoding and tensor-product encoding. Each scheme offers distinct advantages and limitations in terms of circuit depth, accuracy, and physical interpretability.

Tensor-product encoding assigns each distribution component to a separate qubit. This preserves locality and allows a more modular representation of distributions. For instance, state preparation and intermediate updates at every time step can be performed simultaneously on the distribution-carry qubits. Collision operations can be decomposed into macroscopic quantity estimations and arithmetics, which are implemented by quantum phase estimation and quantum arithmetics protocols, respectively. This allows a perfect collision operator up to numerical resolution induced by the qubit width r for binary representations, which comes at the cost of resource overheads to perform intermediate computations. The main limitation of this approach is Quantum Fourier Transform operators on $O(r)$ qubits that typically serve as key components in quantum arithmetic circuits performing calculation in the Fourier domain. Additionally, tensor-product encoding inhibits unitary streaming, and alternative methods, such as space-time encoding and approximate streaming, further require a significant amount of resources and entail another source of error. Although tensor-product encoding supports high-precision LBM implementations with a transparent mathematical interpretation, its high gate complexity and the need for precise calibration of intermediate operations render it suitable only for relatively large, fault-tolerant quantum architectures.

In contrast, amplitude encoding encodes the square roots of distribution values directly into the amplitudes of a single quantum state. As this encoding allows a straightforward representation of symmetry groups of LBM models, low-depth symmetry-preserving variational models can be designed to approximate the collision effect. This gate efficiency makes this variational approach well-suited for near-term quantum hardware, where quantum gates are prone to errors. On the other hand, amplitude encoding allows collision operators to be expressed via block-encoded unitaries and manipulated using standard techniques such as oblivious amplitude amplification, altogether implemented by $O(N_v^2)$ two-qubit gates on $O(N_v)$ data-carrying qubits, where N_v is the number of velocity vectors. For $N_v \leq 27$, the necessary system size falls within the projected capacity of early fault-tolerant quantum devices, justifying continued investigation in the near future. This encoding scheme, however, also suffers from several drawbacks. First, post-selection is typically required to extract physically meaningful outputs, which introduces non-determinism into multi-step simulations. Second, the encoding obscures locality and makes it challenging to interpret output states in terms of physical quantities.

To validate the feasibility of amplitude encoding, we implemented preliminary experiments using block-encoded collision operators and a unitary streaming transformation. We studied the relation between probabilities of successful post-selection and the number of ancillas to guide our selection of the block-

encoding model. We also measure errors across multiple rounds along the convergence of the system in a full LBM simulation, verifying the algorithm outputs final quantities to within acceptable error.

Looking ahead, several directions emerge for near-future work. First, theoretical analysis of error accumulation across successive LBM steps, especially under repeated post-selection of quantum superpositions, is essential for understanding long-term stability. In particular, the varying post-selection probabilities associated with intermediate states of the form

$$|\tilde{\psi}_i\rangle = \sqrt{\lambda_i} |0^r\rangle |\psi\rangle + \sqrt{1 - \lambda_i} |0^{r\perp}\rangle |\dots\rangle, \quad i = 0, 1, \dots \quad (6.1)$$

introduce amplitude imbalances when constructing global superpositions like $\sum_i |i\rangle |\tilde{\psi}_i\rangle$. If these variations are not properly accounted for, the block-encoded collision operator may inadvertently distort the amplitudes across lattice sites, effectively driving the quantum state out of the subspace spanned by uniform superpositions. This deviation can lead to a violation of mass conservation, undermining the physical foundation of the simulation.

Second, implementing and comparing different types of boundary conditions, such as bounce-back versus specular reflection, within both encoding frameworks would illuminate how effectively quantum circuits can capture particle interactions at walls. Techniques to impose these constraints unitarily could open up practical routes to model dynamics of confined flows.

In the longer term, one promising direction is to design collision operators for amplitude encoding that incorporate physical parameters, such as viscosity or relaxation time, directly into their construction, rather than relying solely on pre-sampled equilibrium distributions. This would enable a form of physics-informed quantum learning, where the solver is variationally trained to respect macroscopic fluid behavior. Such an approach could allow the quantum circuit to adapt collision dynamics based on flow conditions, improving accuracy and generalizability. Another avenue is to combine symmetry-preserving ansatzes with post-selection techniques, leveraging group-theoretic structure to restrict our search to physically admissible subspaces. This could significantly reduce the training overhead by narrowing the variational search space and improving convergence, especially when learning LBM dynamics constrained by certain conservations and symmetries.

In conclusion, this thesis lays foundational work for quantum fluid simulation using the lattice Boltzmann framework. The two encoding strategies studied here represent two ends of a design spectrum, one optimized for compactness and precision, the other for efficiency and applicability. Their trade-offs, as well as their interplay with circuit depth, error tolerance, and physical interpretation, provide a basis for future development. As new quantum hardware and algorithms continue to advance, the methods developed in this work can serve as building blocks for larger-scale quantum simulations of fluid phenomena.

References

- [1] Henk Kaarle Versteeg. *An introduction to computational fluid dynamics the finite volume method*, 2/E. Pearson Education India, 2007.
- [2] Mario A. Storti et al. "Dynamic boundary conditions in computational fluid dynamics". In: *Computer Methods in Applied Mechanics and Engineering* 197.13-16 (Feb. 2008), pp. 1219–1232. ISSN: 0045-7825. DOI: [10.1016/j.cma.2007.10.014](https://doi.org/10.1016/j.cma.2007.10.014). URL: <https://doi.org/10.1016/j.cma.2007.10.014>.
- [3] Mehrez Agnaou et al. "On the use of physical boundary conditions for two-phase flow simulations: Integration of control feedback". In: *Computers & Chemical Engineering* 118 (Oct. 2018), pp. 268–282. ISSN: 0098-1354. DOI: [10.1016/j.compchemeng.2018.08.012](https://doi.org/10.1016/j.compchemeng.2018.08.012). URL: <https://doi.org/10.1016/j.compchemeng.2018.08.012>.
- [4] D. Bestion. "The difficult challenge of a two-phase CFD modelling for all flow regimes". In: *Nuclear Engineering and Design* 279 (Nov. 2014), pp. 116–125. ISSN: 0029-5493. DOI: [10.1016/j.nucengdes.2014.04.006](https://doi.org/10.1016/j.nucengdes.2014.04.006). URL: <https://doi.org/10.1016/j.nucengdes.2014.04.006>.
- [5] Takeo Kajishima and Kunihiro Taira. *Computational Fluid Dynamics*. Springer International Publishing, 2017. ISBN: 9783319453026. DOI: [10.1007/978-3-319-45304-0](https://doi.org/10.1007/978-3-319-45304-0). URL: <https://doi.org/10.1007/978-3-319-45304-0>.
- [6] Jian-Xun Wang and Heng Xiao. "Data-driven CFD modeling of turbulent flows through complex structures". In: *International Journal of Heat and Fluid Flow* 62 (Dec. 2016), pp. 138–149. ISSN: 0142-727X. DOI: [10.1016/j.ijheatfluidflow.2016.11.007](https://doi.org/10.1016/j.ijheatfluidflow.2016.11.007). URL: <https://doi.org/10.1016/j.ijheatfluidflow.2016.11.007>.
- [7] Timm Krueger et al. *The Lattice Boltzmann Method: Principles and Practice*. 2016.
- [8] Sauro Succi. *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press, 2001.
- [9] Dieter A. Wolf-Gladrow. *Lattice Gas Cellular Automata and Lattice Boltzmann Models*. Springer Berlin Heidelberg, 2000. ISBN: 9783540669739. DOI: [10.1007/b72010](https://doi.org/10.1007/b72010). URL: <https://doi.org/10.1007/b72010>.
- [10] F. Talati and M. Taghilou. "Lattice Boltzmann application on the PCM solidification within a rectangular finned container". In: *Applied Thermal Engineering* 83 (May 2015), pp. 108–120. ISSN: 1359-4311. DOI: [10.1016/j.applthermaleng.2015.03.017](https://doi.org/10.1016/j.applthermaleng.2015.03.017). URL: <https://doi.org/10.1016/j.applthermaleng.2015.03.017>.
- [11] Ao Xu, Wei Shyy, and Tianshou Zhao. "Lattice Boltzmann modeling of transport phenomena in fuel cells and flow batteries". In: *Acta Mechanica Sinica* 33.3 (Apr. 2017), pp. 555–574. ISSN: 0567-7718. DOI: [10.1007/s10409-017-0667-6](https://doi.org/10.1007/s10409-017-0667-6). URL: <https://doi.org/10.1007/s10409-017-0667-6>.
- [12] Y.Y. Yan, Y.Q. Zu, and Bo Dong. "LBM, a useful tool for mesoscale modelling of single-phase and multiphase flow". In: *Applied Thermal Engineering* 31.5 (Apr. 2011), pp. 649–655. ISSN: 1359-4311. DOI: [10.1016/j.applthermaleng.2010.10.010](https://doi.org/10.1016/j.applthermaleng.2010.10.010). URL: <https://doi.org/10.1016/j.applthermaleng.2010.10.010>.
- [13] B. Dorschner, S. S. Chikatamarla, and I. V. Karlin. "Entropic multirelaxation-time lattice Boltzmann method for moving and deforming geometries in three dimensions". In: *Physical Review E* 95.6 (June 2017). ISSN: 2470-0045. DOI: [10.1103/physreve.95.063306](https://doi.org/10.1103/physreve.95.063306). URL: <https://doi.org/10.1103/physreve.95.063306>.
- [14] Cheng Chang, Chih-Hao Liu, and Chao-An Lin. "Boundary conditions for lattice Boltzmann simulations with complex geometry flows". In: *Computers & Mathematics with Applications* 58.5 (Sept. 2009), pp. 940–949. ISSN: 0898-1221. DOI: [10.1016/j.camwa.2009.02.016](https://doi.org/10.1016/j.camwa.2009.02.016). URL: <https://doi.org/10.1016/j.camwa.2009.02.016>.

- [15] E Calore et al. "Optimization of lattice Boltzmann simulations on heterogeneous computers". In: *The International Journal of High Performance Computing Applications* 33.1 (Apr. 2017), pp. 124–139. ISSN: 1094-3420. DOI: [10.1177/1094342017703771](https://doi.org/10.1177/1094342017703771). URL: <https://doi.org/10.1177/1094342017703771>.
- [16] Andreas Lintermann and Wolfgang Schröder. "Lattice–Boltzmann simulations for complex geometries on high-performance computers". In: *CEAS Aeronautical Journal* 11.3 (May 2020), pp. 745–766. ISSN: 1869-5582. DOI: [10.1007/s13272-020-00450-1](https://doi.org/10.1007/s13272-020-00450-1). URL: <https://doi.org/10.1007/s13272-020-00450-1>.
- [17] Frédéric Kuznik et al. "LBM based flow simulation using GPU computing processor". In: *Computers & Mathematics with Applications* 59.7 (Apr. 2010), pp. 2380–2392. ISSN: 0898-1221. DOI: [10.1016/j.camwa.2009.08.052](https://doi.org/10.1016/j.camwa.2009.08.052). URL: <https://doi.org/10.1016/j.camwa.2009.08.052>.
- [18] Sauro Succi. "Lattice Boltzmann 2038". In: *EPL (Europhysics Letters)* 109.5 (Mar. 2015), p. 50001. ISSN: 0295-5075. DOI: [10.1209/0295-5075/109/50001](https://doi.org/10.1209/0295-5075/109/50001). URL: <https://doi.org/10.1209/0295-5075/109/50001>.
- [19] YaoSong Chen, XiaoWen Shan, and HuDong Chen. "New direction of computational fluid dynamics and its applications in industry". In: *Science in China Series E: Technological Sciences* 50.5 (Oct. 2007), pp. 521–533. ISSN: 1006-9321. DOI: [10.1007/s11431-007-0075-4](https://doi.org/10.1007/s11431-007-0075-4). URL: <https://doi.org/10.1007/s11431-007-0075-4>.
- [20] Anthony Nguyen et al. *3.5-D Blocking Optimization for Stencil Computations on Modern CPUs and GPUs*. Tech. rep. IEEE, Nov. 2010, pp. 1–13. DOI: [10.1109/sc.2010.2](https://doi.org/10.1109/sc.2010.2). URL: <https://doi.org/10.1109/sc.2010.2>.
- [21] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. Tech. rep. ACM Press, 1996, pp. 212–219. DOI: [10.1145/237814.237866](https://doi.org/10.1145/237814.237866). URL: <https://doi.org/10.1145/237814.237866>.
- [22] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 0097-5397. DOI: [10.1137/s0097539795293172](https://doi.org/10.1137/s0097539795293172). URL: <https://doi.org/10.1137/s0097539795293172>.
- [23] Andrew M. Childs et al. "Toward the first quantum simulation with quantum speedup". In: *Proceedings of the National Academy of Sciences* 115.38 (Sept. 2018), pp. 9456–9461. ISSN: 0027-8424. DOI: [10.1073/pnas.1801723115](https://doi.org/10.1073/pnas.1801723115). URL: <https://doi.org/10.1073/pnas.1801723115>.
- [24] Guang Hao Low and Isaac L. Chuang. "Optimal Hamiltonian Simulation by Quantum Signal Processing". In: *Physical Review Letters* 118.1 (Jan. 2017). ISSN: 0031-9007. DOI: [10.1103/physrevlett.118.010501](https://doi.org/10.1103/physrevlett.118.010501). URL: <https://doi.org/10.1103/physrevlett.118.010501>.
- [25] John M. Martyn et al. "Grand Unification of Quantum Algorithms". In: *PRX Quantum* 2.4 (Dec. 2021). ISSN: 2691-3399. DOI: [10.1103/prxquantum.2.040203](https://doi.org/10.1103/prxquantum.2.040203). URL: <https://doi.org/10.1103/prxquantum.2.040203>.
- [26] A Yu Kitaev. "Quantum measurements and the Abelian Stabilizer Problem". In: *arXiv* (1995). URL: <https://doi.org/10.48550/arXiv.quant-ph/9511026>.
- [27] Dhruvil Patel et al. "Optimal Coherent Quantum Phase Estimation via Tapering". In: *arXiv* (2024). URL: <https://doi.org/10.48550/arXiv.2403.18927>.
- [28] Andrew M. Childs et al. *Exponential algorithmic speedup by a quantum walk*. Tech. rep. ACM, June 2003, pp. 59–68. DOI: [10.1145/780542.780552](https://doi.org/10.1145/780542.780552). URL: <https://doi.org/10.1145/780542.780552>.
- [29] Miklos Santha. *Quantum walk based search algorithms*. 2008. URL: <https://doi.org/10.48550/arXiv.0808.0059>.
- [30] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103.15 (Oct. 2009). ISSN: 0031-9007. DOI: [10.1103/physrevlett.103.150502](https://doi.org/10.1103/physrevlett.103.150502). URL: <https://doi.org/10.1103/physrevlett.103.150502>.

- [31] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. “Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision”. In: *SIAM Journal on Computing* 46.6 (Jan. 2017), pp. 1920–1950. ISSN: 0097-5397. DOI: [10.1137/16m1087072](https://doi.org/10.1137/16m1087072). URL: <https://doi.org/10.1137/16m1087072>.
- [32] Daniel S. Abrams and Seth Lloyd. “Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors”. In: *Physical Review Letters* 83.24 (Dec. 1999), pp. 5162–5165. ISSN: 0031-9007. DOI: [10.1103/physrevlett.83.5162](https://doi.org/10.1103/physrevlett.83.5162). URL: <https://doi.org/10.1103/physrevlett.83.5162>.
- [33] David Poulin and Pawel Wocjan. “Preparing Ground States of Quantum Many-Body Systems on a Quantum Computer”. In: *Physical Review Letters* 102.13 (Apr. 2009). ISSN: 0031-9007. DOI: [10.1103/physrevlett.102.130503](https://doi.org/10.1103/physrevlett.102.130503). URL: <https://doi.org/10.1103/physrevlett.102.130503>.
- [34] Yimin Ge, Jordi Tura, and J. Ignacio Cirac. “Faster ground state preparation and high-precision ground energy estimation with fewer qubits”. In: *Journal of Mathematical Physics* 60.2 (Feb. 2019). ISSN: 0022-2488. DOI: [10.1063/1.5027484](https://doi.org/10.1063/1.5027484). URL: <https://doi.org/10.1063/1.5027484>.
- [35] Lin Lin and Yu Tong. “Near-optimal ground state preparation”. In: *Quantum* 4 (Dec. 2020), p. 372. ISSN: 2521-327X. DOI: [10.22331/q-2020-12-14-372](https://doi.org/10.22331/q-2020-12-14-372). URL: <https://doi.org/10.22331/q-2020-12-14-372>.
- [36] Kishor Bharti et al. “Noisy intermediate-scale quantum algorithms”. In: *Reviews of Modern Physics* 94.1 (Feb. 2022). ISSN: 0034-6861. DOI: [10.1103/revmodphys.94.015004](https://doi.org/10.1103/revmodphys.94.015004). URL: <https://doi.org/10.1103/revmodphys.94.015004>.
- [37] M. Cerezo et al. “Variational quantum algorithms”. In: *Nature Reviews Physics* 3.9 (Aug. 2021), pp. 625–644. ISSN: 2522-5820. DOI: [10.1038/s42254-021-00348-9](https://doi.org/10.1038/s42254-021-00348-9). URL: <https://doi.org/10.1038/s42254-021-00348-9>.
- [38] Frank Leymann and Johanna Barzen. “The bitter truth about gate-based quantum algorithms in the NISQ era”. In: *Quantum Science and Technology* 5.4 (Sept. 2020), p. 044007. ISSN: 2058-9565. DOI: [10.1088/2058-9565/abae7d](https://doi.org/10.1088/2058-9565/abae7d). URL: <https://doi.org/10.1088/2058-9565/abae7d>.
- [39] Jonathan Wei Zhong Lau et al. “NISQ computing: where are we and where do we go?” In: *AAPPS Bulletin* 32.1 (Sept. 2022). ISSN: 2309-4710. DOI: [10.1007/s43673-022-00058-z](https://doi.org/10.1007/s43673-022-00058-z). URL: <https://doi.org/10.1007/s43673-022-00058-z>.
- [40] Sitan Chen et al. “The complexity of NISQ”. In: *Nature Communications* 14.1 (Sept. 2023). ISSN: 2041-1723. DOI: [10.1038/s41467-023-41217-6](https://doi.org/10.1038/s41467-023-41217-6). URL: <https://doi.org/10.1038/s41467-023-41217-6>.
- [41] Amara Katabarwa et al. “Early Fault-Tolerant Quantum Computing”. In: *PRX Quantum* 5.2 (June 2024). ISSN: 2691-3399. DOI: [10.1103/prxquantum.5.020101](https://doi.org/10.1103/prxquantum.5.020101). URL: <https://doi.org/10.1103/prxquantum.5.020101>.
- [42] Oriel Kiss et al. “Early fault-tolerant quantum algorithms in practice: Application to ground-state energy estimation”. In: *Quantum* 9 (2025), p. 1682.
- [43] Tom Ginsberg and Vyom Patel. *Quantum Error Detection For Early Term Fault-Tolerant Quantum Algorithms*. Tech. rep. 2025. URL: <https://doi.org/10.48550/arXiv.2503.10790>.
- [44] Qiyao Liang et al. “Modeling the performance of early fault-tolerant quantum algorithms”. In: *Physical Review Research* 6.2 (May 2024). ISSN: 2643-1564. DOI: [10.1103/physrevresearch.6.023118](https://doi.org/10.1103/physrevresearch.6.023118). URL: <https://doi.org/10.1103/physrevresearch.6.023118>.
- [45] Merel A. Schalkers and Matthias Möller. “On the importance of data encoding in quantum Boltzmann methods”. In: *Quantum Information Processing* 23.1 (Jan. 2024). ISSN: 1573-1332. DOI: [10.1007/s11128-023-04216-6](https://doi.org/10.1007/s11128-023-04216-6). URL: <https://doi.org/10.1007/s11128-023-04216-6>.
- [46] Ljubomir Budinski. “Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method”. In: *Quantum Information Processing* 20.2 (Feb. 2021). ISSN: 1570-0755. DOI: [10.1007/s11128-021-02996-3](https://doi.org/10.1007/s11128-021-02996-3). URL: <https://doi.org/10.1007/s11128-021-02996-3>.

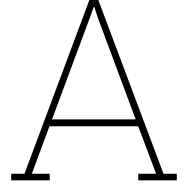
- [47] Budinski Ljubomir. “Quantum algorithm for the Navier–Stokes equations by using the streamfunction-vorticity formulation and the lattice Boltzmann method”. In: *International Journal of Quantum Information* 20.02 (Feb. 2022). ISSN: 0219-7499. DOI: [10.1142/s0219749921500398](https://doi.org/10.1142/s0219749921500398). URL: <https://doi.org/10.1142/s0219749921500398>.
- [48] Melody Lee et al. “A multiple-circuit approach to quantum resource reduction with application to the quantum lattice Boltzmann method”. In: *arXiv* (2024). URL: <https://doi.org/10.48550/arXiv.2401.12248>.
- [49] E. Dinesh Kumar and Steven H. Frankel. “Quantum unitary matrix representation of the lattice Boltzmann model for low Reynolds fluid flow simulation”. In: *AVS Quantum Science* 7.1 (Mar. 2025). ISSN: 2639-0213. DOI: [10.1116/5.0245082](https://doi.org/10.1116/5.0245082). URL: <https://doi.org/10.1116/5.0245082>.
- [50] David Wawrzyniak et al. “A quantum algorithm for the lattice-Boltzmann method advection-diffusion equation”. In: *Computer Physics Communications* 306 (Jan. 2025), p. 109373. ISSN: 0010-4655. DOI: [10.1016/j.cpc.2024.109373](https://doi.org/10.1016/j.cpc.2024.109373). URL: <https://doi.org/10.1016/j.cpc.2024.109373>.
- [51] Claudio Sanavio and Sauro Succi. “Lattice Boltzmann–Carleman quantum algorithm and circuit for fluid flows at moderate Reynolds number”. In: *AVS Quantum Science* 6.2 (Apr. 2024). ISSN: 2639-0213. DOI: [10.1116/5.0195549](https://doi.org/10.1116/5.0195549). URL: <https://doi.org/10.1116/5.0195549>.
- [52] Wael Itani, Katepalli R. Sreenivasan, and Sauro Succi. “Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term”. In: *Physics of Fluids* 36.1 (Jan. 2024). ISSN: 1070-6631. DOI: [10.1063/5.0176569](https://doi.org/10.1063/5.0176569). URL: <https://doi.org/10.1063/5.0176569>.
- [53] David Wawrzyniak et al. “Unitary Quantum Algorithm for the Lattice-Boltzmann Method”. In: *arXiv* (2024). URL: <https://doi.org/10.48550/arXiv.2405.13391>.
- [54] David Wawrzyniak et al. “Dynamic Circuits for the Quantum Lattice-Boltzmann Method”. In: *arXiv* (2025). URL: <https://doi.org/10.48550/arXiv.2502.02131>.
- [55] René Steijl. “Quantum Circuit Implementation of Multi-Dimensional Non-Linear Lattice Models”. In: *Applied Sciences* 13.1 (Dec. 2022), p. 529. ISSN: 2076-3417. DOI: [10.3390/app13010529](https://doi.org/10.3390/app13010529). URL: <https://doi.org/10.3390/app13010529>.
- [56] C. Sanavio et al. “Three Carleman routes to the quantum simulation of classical fluids”. In: *Physics of Fluids* 36.5 (May 2024). ISSN: 1070-6631. DOI: [10.1063/5.0204955](https://doi.org/10.1063/5.0204955). URL: <https://doi.org/10.1063/5.0204955>.
- [57] Apurva Tiwari et al. “Algorithmic Advances Towards a Realizable Quantum Lattice Boltzmann Method”. In: *arXiv preprint arXiv:2504.10870* (2025). URL: <https://doi.org/10.48550/arXiv.2504.10870>.
- [58] Clement Kleinstreuer. *Modern Fluid Dynamics*. Springer Netherlands, 2010. ISBN: 9781402086694. DOI: [10.1007/978-1-4020-8670-0](https://doi.org/10.1007/978-1-4020-8670-0). URL: <https://doi.org/10.1007/978-1-4020-8670-0>.
- [59] Akihiro Sasoh. *Compressible Fluid Dynamics and Shock Waves*. Springer Singapore, 2020. ISBN: 9789811505034. DOI: [10.1007/978-981-15-0504-1](https://doi.org/10.1007/978-981-15-0504-1). URL: <https://doi.org/10.1007/978-981-15-0504-1>.
- [60] P. L. Bhatnagar, E. P. Gross, and M. Krook. “A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems”. In: *Physical Review* 94.3 (May 1954), pp. 511–525. ISSN: 0031-899X. DOI: [10.1103/physrev.94.511](https://doi.org/10.1103/physrev.94.511). URL: <https://doi.org/10.1103/physrev.94.511>.
- [61] Sydney Chapman and Thomas George Cowling. *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 1990.
- [62] J Li. “Appendix: Chapman-Enskog Expansion in the Lattice Boltzmann Method”. In: *arXiv* (2015). URL: <https://doi.org/10.48550/arXiv.1512.02599>.
- [63] Li-Shi Luo. *Lattice-gas automata and lattice Boltzmann equations for two-dimensional hydrodynamics*. Georgia Institute of Technology, 1993.

- [64] Xiaowen Shan and Hudong Chen. “Simulation of nonideal gases and liquid-gas phase transitions by the lattice Boltzmann equation”. In: *Physical Review E* 49.4 (Apr. 1994), pp. 2941–2948. ISSN: 1063-651X. DOI: [10.1103/physreve.49.2941](https://doi.org/10.1103/physreve.49.2941). URL: <https://doi.org/10.1103/physreve.49.2941>.
- [65] J. M. Buick and C. A. Greated. “Gravity in a lattice Boltzmann model”. In: *Physical Review E* 61.5 (May 2000), pp. 5307–5320. ISSN: 1063-651X. DOI: [10.1103/physreve.61.5307](https://doi.org/10.1103/physreve.61.5307). URL: <https://doi.org/10.1103/physreve.61.5307>.
- [66] A.A. Mohamad and A. Kuzmin. “A critical evaluation of force term in lattice Boltzmann method, natural convection problem”. In: *International Journal of Heat and Mass Transfer* 53.5-6 (Feb. 2010), pp. 990–996. ISSN: 0017-9310. DOI: [10.1016/j.ijheatmasstransfer.2009.11.014](https://doi.org/10.1016/j.ijheatmasstransfer.2009.11.014). URL: <https://doi.org/10.1016/j.ijheatmasstransfer.2009.11.014>.
- [67] Laura Clinton, Johannes Bausch, and Toby Cubitt. “Hamiltonian simulation algorithms for near-term quantum hardware”. In: *Nature Communications* 12.1 (Aug. 2021). ISSN: 2041-1723. DOI: [10.1038/s41467-021-25196-0](https://doi.org/10.1038/s41467-021-25196-0). URL: <https://doi.org/10.1038/s41467-021-25196-0>.
- [68] Andrew J. Daley et al. “Practical quantum advantage in quantum simulation”. In: *Nature* 607.7920 (July 2022), pp. 667–676. ISSN: 0028-0836. DOI: [10.1038/s41586-022-04940-6](https://doi.org/10.1038/s41586-022-04940-6). URL: <https://doi.org/10.1038/s41586-022-04940-6>.
- [69] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. “An introduction to quantum machine learning”. In: *Contemporary Physics* 56.2 (Oct. 2014), pp. 172–185. ISSN: 0010-7514. DOI: [10.1080/00107514.2014.964942](https://doi.org/10.1080/00107514.2014.964942). URL: <https://doi.org/10.1080/00107514.2014.964942>.
- [70] M. Cerezo et al. “Challenges and opportunities in quantum machine learning”. In: *Nature Computational Science* 2.9 (Sept. 2022), pp. 567–576. ISSN: 2662-8457. DOI: [10.1038/s43588-022-00311-3](https://doi.org/10.1038/s43588-022-00311-3). URL: <https://doi.org/10.1038/s43588-022-00311-3>.
- [71] Alessandro Luongo. *Quantum algorithms for data analysis*. 2020. URL: <https://quantumalgorithms.org/>.
- [72] Ashley Montanaro and Sam Pallister. “Quantum algorithms and the finite element method”. In: *Physical Review A* 93.3 (Mar. 2016). ISSN: 2469-9926. DOI: [10.1103/physreva.93.032324](https://doi.org/10.1103/physreva.93.032324). URL: <https://doi.org/10.1103/physreva.93.032324>.
- [73] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [74] Thomas G Draper. “Addition on a Quantum Computer”. In: *arXiv* (2000). URL: <https://doi.org/10.48550/arXiv.quant-ph/0008033>.
- [75] R. Cleve et al. “Quantum algorithms revisited”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (Jan. 1998), pp. 339–354. ISSN: 1364-5021. DOI: [10.1098/rspa.1998.0164](https://doi.org/10.1098/rspa.1998.0164). URL: <https://doi.org/10.1098/rspa.1998.0164>.
- [76] Gilles Brassard et al. *Quantum amplitude amplification and estimation*. 2002. DOI: [10.1090/conm/305/05215](https://doi.org/10.1090/conm/305/05215). URL: <https://doi.org/10.1090/conm/305/05215>.
- [77] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. “Fixed-Point Quantum Search with an Optimal Number of Queries”. In: *Physical Review Letters* 113.21 (Nov. 2014). ISSN: 0031-9007. DOI: [10.1103/physrevlett.113.210501](https://doi.org/10.1103/physrevlett.113.210501). URL: <https://doi.org/10.1103/physrevlett.113.210501>.
- [78] Dominic W. Berry et al. *Exponential improvement in precision for simulating sparse Hamiltonians*. Tech. rep. ACM, May 2014, pp. 283–292. DOI: [10.1145/2591796.2591854](https://doi.org/10.1145/2591796.2591854). URL: <https://doi.org/10.1145/2591796.2591854>.
- [79] Bao Yan et al. “Fixed-point oblivious quantum amplitude-amplification algorithm”. In: *Scientific Reports* 12.1 (Aug. 2022). ISSN: 2045-2322. DOI: [10.1038/s41598-022-15093-x](https://doi.org/10.1038/s41598-022-15093-x). URL: <https://doi.org/10.1038/s41598-022-15093-x>.
- [80] András Gilyén et al. *Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics*. Tech. rep. ACM, June 2019, pp. 193–204. DOI: [10.1145/3313276.3316366](https://doi.org/10.1145/3313276.3316366). URL: <https://doi.org/10.1145/3313276.3316366>.

- [81] Andrew M. Childs and Nathan Wiebe. “Hamiltonian simulation using linear combinations of unitary operations”. In: *Quantum Information and Computation* 12.11&12 (Nov. 2012), pp. 901–924. ISSN: 1533-7146. DOI: [10.26421/qic12.11-12-1](https://doi.org/10.26421/qic12.11-12-1). URL: <https://doi.org/10.26421/qic12.11-12-1>.
- [82] Shantanav Chakraborty. “Implementing any Linear Combination of Unitaries on Intermediate-term Quantum Computers”. In: *Quantum* 8 (Oct. 2024), p. 1496. ISSN: 2521-327X. DOI: [10.22331/q-2024-10-10-1496](https://doi.org/10.22331/q-2024-10-10-1496). URL: <https://doi.org/10.22331/q-2024-10-10-1496>.
- [83] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum principal component analysis”. In: *Nature Physics* 10.9 (July 2014), pp. 631–633. ISSN: 1745-2473. DOI: [10.1038/nphys3029](https://doi.org/10.1038/nphys3029). URL: <https://doi.org/10.1038/nphys3029>.
- [84] Dominic W. Berry et al. “Simulating Hamiltonian Dynamics with a Truncated Taylor Series”. In: *Physical Review Letters* 114.9 (Mar. 2015). ISSN: 0031-9007. DOI: [10.1103/physrevlett.114.090502](https://doi.org/10.1103/physrevlett.114.090502). URL: <https://doi.org/10.1103/physrevlett.114.090502>.
- [85] Christoph Sünderhauf, Earl Campbell, and Joan Camps. “Block-encoding structured matrices for data input in quantum computing”. In: *Quantum* 8 (Jan. 2024), p. 1226. ISSN: 2521-327X. DOI: [10.22331/q-2024-01-11-1226](https://doi.org/10.22331/q-2024-01-11-1226). URL: <https://doi.org/10.22331/q-2024-01-11-1226>.
- [86] Zhenyu Cai et al. “Quantum error mitigation”. In: *Reviews of Modern Physics* 95.4 (Dec. 2023). ISSN: 0034-6861. DOI: [10.1103/revmodphys.95.045005](https://doi.org/10.1103/revmodphys.95.045005). URL: <https://doi.org/10.1103/revmodphys.95.045005>.
- [87] Christoph Dankert et al. “Exact and approximate unitary 2-designs and their application to fidelity estimation”. In: *Physical Review A* 80.1 (July 2009). ISSN: 1050-2947. DOI: [10.1103/physreva.80.012304](https://doi.org/10.1103/physreva.80.012304). URL: <https://doi.org/10.1103/physreva.80.012304>.
- [88] Benjamin Lévi et al. “Efficient error characterization in quantum information processing”. In: *Physical Review A* 75.2 (Feb. 2007). ISSN: 1050-2947. DOI: [10.1103/physreva.75.022314](https://doi.org/10.1103/physreva.75.022314). URL: <https://doi.org/10.1103/physreva.75.022314>.
- [89] Joseph Emerson, Robert Alicki, and Karol Życzkowski. “Scalable noise estimation with random unitary operators”. In: *Journal of Optics B: Quantum and Semiclassical Optics* 7.10 (Sept. 2005), S347–S352. ISSN: 1464-4266. DOI: [10.1088/1464-4266/7/10/021](https://doi.org/10.1088/1464-4266/7/10/021). URL: <https://doi.org/10.1088/1464-4266/7/10/021>.
- [90] Michael M Bronstein et al. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *arXiv* (2021). URL: <https://doi.org/10.48550/arXiv.2104.13478>.
- [91] Carlos Esteves. “Theoretical Aspects of Group Equivariant Neural Networks”. In: *Arxiv* (2020). URL: <https://arxiv.org/abs/2004.05154v2>.
- [92] J Han et al. “Geometrically Equivariant Graph Neural Networks: A Survey”. In: *arXiv* (2022). URL: <https://doi.org/10.48550/arXiv.2202.07230>.
- [93] Jan E. Gerken et al. “Geometric deep learning and equivariant neural networks”. In: *Artificial Intelligence Review* 56.12 (June 2023), pp. 14605–14662. ISSN: 0269-2821. DOI: [10.1007/s10462-023-10502-7](https://doi.org/10.1007/s10462-023-10502-7). URL: <https://doi.org/10.1007/s10462-023-10502-7>.
- [94] Taco Cohen et al. “Gauge Equivariant Convolutional Networks and the Icosahedral CNN”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 1321–1330. URL: <https://proceedings.mlr.press/v97/cohen19d.html>.
- [95] Taco Cohen and Max Welling. “Group Equivariant Convolutional Networks”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2990–2999. URL: <https://proceedings.mlr.press/v48/cohenc16.html>.
- [96] Quynh T. Nguyen et al. “Theory for Equivariant Quantum Neural Networks”. In: *PRX Quantum* 5.2 (May 2024). ISSN: 2691-3399. DOI: [10.1103/prxquantum.5.020328](https://doi.org/10.1103/prxquantum.5.020328). URL: <https://doi.org/10.1103/prxquantum.5.020328>.

- [97] Su Yeon Chang et al. *Approximately Equivariant Quantum Neural Network for $p4m$ Group Symmetries in Images*. Tech. rep. IEEE, Sept. 2023, pp. 229–235. DOI: [10.1109/qce57702.2023.00033](https://doi.org/10.1109/qce57702.2023.00033). URL: <https://doi.org/10.1109/qce57702.2023.00033>.
- [98] Andrea Skolik et al. “Equivariant quantum circuits for learning on weighted graphs”. In: *npj Quantum Information* 9.1 (May 2023). ISSN: 2056-6387. DOI: [10.1038/s41534-023-00710-y](https://doi.org/10.1038/s41534-023-00710-y). URL: <https://doi.org/10.1038/s41534-023-00710-y>.
- [99] Zachary P. Bradshaw et al. “Learning equivariant maps with variational quantum circuits”. In: *Phys. Rev. Appl.* 23 (4 Apr. 2025), p. 044007. DOI: [10.1103/PhysRevApplied.23.044007](https://link.aps.org/doi/10.1103/PhysRevApplied.23.044007). URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.23.044007>.
- [100] Jeffrey Yepez. “Quantum Computation of Fluid Dynamics”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1999, pp. 34–60. ISBN: 9783540655145. DOI: [10.1007/3-540-49208-9_3](https://doi.org/10.1007/3-540-49208-9_3). URL: https://doi.org/10.1007/3-540-49208-9_3.
- [101] Jeffrey Yepez. “Quantum lattice-gas model for computational fluid dynamics”. In: *Physical Review E* 63.4 (Mar. 2001). ISSN: 1063-651X. DOI: [10.1103/physreve.63.046702](https://doi.org/10.1103/physreve.63.046702). URL: <https://doi.org/10.1103/physreve.63.046702>.
- [102] Jeffrey Yepez. “Quantum lattice-gas model for the Burgers equation”. In: *Journal of Statistical Physics* 107.1/2 (2002), pp. 203–224. ISSN: 0022-4715. DOI: [10.1023/a:1014514805610](https://doi.org/10.1023/a:1014514805610). URL: <https://doi.org/10.1023/a:1014514805610>.
- [103] Kosuke Mitarai, Masahiro Kitagawa, and Keisuke Fujii. “Quantum analog-digital conversion”. In: *Physical Review A* 99.1 (Jan. 2019). ISSN: 2469-9926. DOI: [10.1103/physreva.99.012301](https://doi.org/10.1103/physreva.99.012301). URL: <https://doi.org/10.1103/physreva.99.012301>.
- [104] Akira SaiToh. “Quantum digital-to-analog conversion algorithm using decoherence”. In: *Quantum Information Processing* 14 (2015), pp. 2729–2748.
- [105] S. S. Zhou et al. “Quantum Fourier transform in computational basis”. In: *Quantum Information Processing* 16.3 (Feb. 2017). ISSN: 1570-0755. DOI: [10.1007/s11128-017-1515-0](https://doi.org/10.1007/s11128-017-1515-0). URL: <https://doi.org/10.1007/s11128-017-1515-0>.
- [106] Merel A. Schalkers and Matthias Möller. “Efficient and fail-safe quantum algorithm for the transport equation”. In: *Journal of Computational Physics* 502 (Apr. 2024), p. 112816. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2024.112816](https://doi.org/10.1016/j.jcp.2024.112816). URL: <https://doi.org/10.1016/j.jcp.2024.112816>.
- [107] Blaga N. Todorova and René Steijl. “Quantum algorithm for the collisionless Boltzmann equation”. In: *Journal of Computational Physics* 409 (May 2020), p. 109347. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2020.109347](https://doi.org/10.1016/j.jcp.2020.109347). URL: <https://doi.org/10.1016/j.jcp.2020.109347>.
- [108] Soichiro Yamazaki et al. “Quantum algorithm for collisionless Boltzmann simulation of self-gravitating systems”. In: *Computers & Fluids* 288 (Feb. 2025), p. 106527. ISSN: 0045-7930. DOI: [10.1016/j.compfluid.2024.106527](https://doi.org/10.1016/j.compfluid.2024.106527). URL: <https://doi.org/10.1016/j.compfluid.2024.106527>.
- [109] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. “Quantum arithmetic with the quantum Fourier transform”. In: *Quantum Information Processing* 16.6 (Apr. 2017). ISSN: 1570-0755. DOI: [10.1007/s11128-017-1603-1](https://doi.org/10.1007/s11128-017-1603-1). URL: <https://doi.org/10.1007/s11128-017-1603-1>.
- [110] Himanshu Thapliyal et al. “Quantum Circuit Designs of Integer Division Optimizing T-count and T-depth”. In: *IEEE Transactions on Emerging Topics in Computing* 9.2 (2021), pp. 1045–1056. DOI: [10.1109/TETC.2019.2910870](https://doi.org/10.1109/TETC.2019.2910870).
- [111] Pierre Ablin and Gabriel Peyré. “Fast and accurate optimization on the orthogonal manifold without retraction”. In: *arXiv* (2021). URL: <https://doi.org/10.48550/arXiv.2102.07432>.
- [112] William R. Clements et al. “Optimal design for universal multiport interferometers”. In: *Optica* 3.12 (Dec. 2016), p. 1460. ISSN: 2334-2536. DOI: [10.1364/optica.3.001460](https://doi.org/10.1364/optica.3.001460). URL: <https://doi.org/10.1364/optica.3.001460>.
- [113] Dario Cilluffo. “Commentary on the decomposition of universal multiport interferometers: how it works in practice”. In: *arXiv* (2024). URL: <https://doi.org/10.48550/arXiv.2412.11955>.

- [114] Alessandro Corbetta et al. "Toward learning Lattice Boltzmann collision operators". In: *The European Physical Journal E* 46.3 (Mar. 2023). ISSN: 1292-8941. DOI: [10.1140/epje/s10189-023-00267-w](https://doi.org/10.1140/epje/s10189-023-00267-w). URL: <https://doi.org/10.1140/epje/s10189-023-00267-w>.
- [115] Juan Miguel Arrazola et al. "Universal quantum circuits for quantum chemistry". In: *Quantum* 6 (June 2022), p. 742. ISSN: 2521-327X. DOI: [10.22331/q-2022-06-20-742](https://doi.org/10.22331/q-2022-06-20-742). URL: <https://doi.org/10.22331/q-2022-06-20-742>.
- [116] César Fenou et al. "Overlap-ADAPT-VQE: practical quantum chemistry on quantum computers via overlap-guided compact Ansätze". In: *Communications Physics* 6.1 (July 2023). ISSN: 2399-3650. DOI: [10.1038/s42005-023-01312-y](https://doi.org/10.1038/s42005-023-01312-y). URL: <https://doi.org/10.1038/s42005-023-01312-y>.
- [117] Harper R. Grimsley et al. "An adaptive variational algorithm for exact molecular simulations on a quantum computer". In: *Nature Communications* 10.1 (July 2019). ISSN: 2041-1723. DOI: [10.1038/s41467-019-10988-2](https://doi.org/10.1038/s41467-019-10988-2). URL: <https://doi.org/10.1038/s41467-019-10988-2>.
- [118] Nonia Vaquero-Sabater, Abel Carreras, and David Casanova. "Pruned-ADAPT-VQE: compacting molecular ansatzes by removing irrelevant operators". In: *arXiv* (2025). URL: <https://doi.org/10.48550/arXiv.2504.04652>.



Tapered Quantum Phase Estimation

In the end of [subsection 4.2.2](#), we discuss a technique called tapered Quantum Phase Estimation (tQPE) to, with a high probability, significantly increase the accuracy of the estimation. Unlike the standard QPE protocol that starts with the initial state $|+\rangle^{\otimes r}$, tQPE requires $r' = r + r_{\text{taper}}$ precision qubits to be initialized in the state

$$|\varphi\rangle := \sum_k \frac{2^{r'-1} \sin(\pi k / 2^{r'})}{\sqrt{2^{r'-1}}} |k\rangle \quad (\text{A.1})$$

Then applying $\sum_{k=0}^{2^{r'}-1} |k\rangle \langle k| \otimes U_k$, followed by the inverse quantum Fourier transform on the precision qubits, increases the probability of getting an estimate error below $\delta = 2^{-(r+1)}$ to $1 - \varepsilon$. In particular, the original work [\[27\]](#) shows that $r_{\text{taper}} = 1 + \lceil \log_2(\log[10/\varepsilon] + 4) \rceil$ suffices for $\varepsilon \geq 10^{-81}$. For example, ensuring the failure rate below $e = 10^{-11}$ requires only $r_{\text{taper}} = 6$ extra qubits.

Since the original work does not address the preparation of the initial state [\(A.1\)](#), we describe a circuit with $O(r)$ simple gates to prepare $|\varphi\rangle$ with an ancilla. First, apply $H^{\otimes(r'+1)}$ on $|0\rangle^{\otimes r'} |0\rangle$ to create the state $\frac{1}{\sqrt{2^{r'}}} \sum_{k=0}^{2^{r'}-1} |k\rangle \otimes \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Then, apply controlled phase gradients in the following manner. For every precision qubit $j = 1, 2, \dots, r'$, apply the phase gate $P(\theta_j)$ and the ancilla-controlled phase gate $CP(-2\theta_j)$ for $\theta_j = \pi/2^{r'-j}$. Then, apply H on the ancilla to create the state

$$\frac{1}{\sqrt{2^{r'}}} \sum_{k=0}^{2^{r'}-1} \left(\cos\left(\frac{\pi k}{2^{r'}}\right) |k\rangle |0\rangle + i \sin\left(\frac{\pi k}{2^{r'}}\right) |k\rangle |1\rangle \right) \quad (\text{A.2})$$

We shall amplify the amplitude of the subspace where the ancilla is $|1\rangle$. We can raise the corresponding probability of measuring the ancilla in state $|1\rangle$ from $\frac{1}{2}$ to 1 using a single iteration of modified amplitude amplification. Denote A the quantum circuit that prepares the state [\(A.2\)](#) from $|0\rangle^{\otimes r'} |0\rangle$. After one iteration of $A \cdot C_{0^{r'}} P(\pi/2) \cdot A^{-1} \cdot (I^{\otimes r'} \otimes P(\pi/2))$, the ancilla is deterministically disentangled in the final state $|\varphi\rangle |1\rangle$. In the expression, $P(\pi/2) = \text{diag}(1, i)$ is called the quantum S gate, and $C_{0^{r'}} P(\pi/2)$ denotes the S gate acting on the ancilla when the precision qubits are in the state $|0\rangle^{\otimes r'}$.

B

Approximate streaming

We can approximate the post-streaming distributions $f_i^{\text{eq}}(\mathbf{x} + \Delta\mathbf{x}) \approx f_i^{\text{eq}}(\mathbf{x}) + \Delta\mathbf{x} \cdot \nabla f_i^{\text{eq}}(\mathbf{x})$ with the conservations of density and momentum applied on the non-equilibrium part $f_i^{\text{neq}} = f_i - f_i^{\text{eq}}$, i.e., $\sum_i f_i^{\text{neq}} = 0$ and $\sum_i \mathbf{c}_i f_i^{\text{neq}} = 0$.

We establish the relation between f_i^{neq} and ∇f_i^{eq} with the first-order approximation¹ in the Chapman-Enskog expansion [7],

$$f_i^{\text{neq}} \approx -\tau (\partial_t f_i^{\text{eq}} + \mathbf{c}_i \cdot \nabla f_i^{\text{eq}}) \quad (\text{B.1})$$

Applying the chain rule on the expression (2.16) of equilibrium distributions yields

$$\begin{aligned} \nabla f_i^{\text{eq}} &= \frac{\partial f_i^{\text{eq}}}{\partial \rho} \nabla \rho + \frac{\partial f_i^{\text{eq}}}{\partial \mathbf{u}} \nabla \mathbf{u} \\ &= w_i \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right] \nabla \rho + w_i \nabla \mathbf{u} \cdot \left[\frac{\mathbf{c}_i}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u}) \mathbf{c}_i}{c_s^4} - \frac{\mathbf{u}}{c_s^2} \right] \end{aligned} \quad (\text{B.2})$$

where $\nabla \mathbf{u} \equiv (\nabla \mathbf{u})_{ij} = \nabla_j u_i$ denotes the Jacobian of the velocity field. We substitute this to the approximation (B.1) to evaluate

$$\begin{aligned} \sum_i \mathbf{c}_i f_i^{\text{neq}} &\approx -\tau \sum_i \mathbf{c}_i (\partial_t f_i^{\text{eq}} + \mathbf{c}_i \cdot \nabla f_i^{\text{eq}}) \\ &= -\tau \left[\sum_i \mathbf{c}_i \partial_t f_i^{\text{eq}} \right. \\ &\quad + \sum_i w_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2} \right) (\mathbf{c}_i \cdot \nabla \rho) \mathbf{c}_i \\ &\quad \left. + \rho \sum_i w_i \left(\mathbf{c}_i \cdot \nabla \mathbf{u} \cdot \left[\frac{\mathbf{c}_i}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u}) \mathbf{c}_i}{c_s^4} - \frac{\mathbf{u}}{c_s^2} \right] \right) \mathbf{c}_i \right] \end{aligned} \quad (\text{B.3})$$

The first term vanishes as $\partial_t f_i^{\text{eq}} = O(\text{Kn})^2$, which is the same order as the error of the approximation (B.1).

¹The exact expression obtained from the force-free Lattice Boltzmann equation (2.18) is $-\frac{1}{\tau} f_i^{(1)} = (\partial_{t_1} f_i^{\text{eq}} + \mathbf{c}_i \cdot \nabla_1 f_i^{\text{eq}})$, where $f_i^{(1)}$, ∂_{t_1} , and ∇_1 come from the multiscale expansion in several orders in $\varepsilon = O(\text{Kn})$ that $f_i = f_i^{\text{eq}} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} + \dots$, $\partial_t = \varepsilon \partial_{t_1} + \varepsilon^2 \partial_{t_2} + \dots$, and $\nabla = \varepsilon \nabla_1$. Neglecting second and higher order terms leads to the approximate expression $-\frac{1}{\tau} f_i^{\text{neq}} = (\partial_t f_i^{\text{eq}} + \mathbf{c}_i \cdot \nabla f_i^{\text{eq}})$, which has an error of $O(\varepsilon)$. Provided that the LBM formulation is accurate up to $O(\varepsilon^2)$, this approximation presents a considerable source of error compared to the standard LBM.

²Knudson number (Kn): the ratio of molecular mean free path to a characteristic length

In the second term, only components with an even order of $\mathbf{c}_i$ make a contribution, as odd-order moments vanish by the isotropy conditions (2.15). The remaining parts are $-\tau \sum_i w_i \left(1 - \frac{|\mathbf{u}|^2}{2c_s^2}\right) (\mathbf{c}_i \cdot \nabla \rho) \mathbf{c}_i$ and $-\frac{\tau}{2c_s^4} \sum_i w_i (\mathbf{c}_i \cdot \mathbf{u})^2 (\mathbf{c}_i \cdot \nabla \rho) \mathbf{c}_i$. Those terms can be evaluated upon rearranging $\mathbf{c}_i$ factors

$$\begin{aligned} \sum_i w_i \left(1 - \frac{|\mathbf{u}|^2}{2c_s^2}\right) (\mathbf{c}_i \cdot \nabla \rho) \mathbf{c}_i &= \left(1 - \frac{|\mathbf{u}|^2}{2c_s^2}\right) (\nabla \rho \otimes \mathbf{I}) \cdot \sum_i w_i \mathbf{c}_i \otimes \mathbf{c}_i \\ &= \left(1 - \frac{|\mathbf{u}|^2}{2c_s^2}\right) (\nabla \rho \otimes \mathbf{I}) \cdot c_s^2 \mathbf{I} \\ &= \text{Tr}(\mathbf{I}) \left(c_s^2 - \frac{|\mathbf{u}|^2}{2}\right) \nabla \rho \end{aligned} \quad (\text{B.4})$$

where $\text{Tr}(\mathbf{I}) = d$ is the number of spatial dimensions, and

$$\begin{aligned} \sum_i w_i (\mathbf{c}_i \cdot \mathbf{u})^2 (\mathbf{c}_i \cdot \nabla \rho) \mathbf{c}_i &= (\mathbf{u} \otimes \mathbf{u} \otimes \nabla \rho \otimes \mathbf{I}) \cdot \sum_i w_i \mathbf{c}_i \otimes \mathbf{c}_i \otimes \mathbf{c}_i \otimes \mathbf{c}_i \\ &= (\mathbf{u} \otimes \mathbf{u} \otimes \nabla \rho \otimes \mathbf{I}) \cdot c_s^4 \mathbf{I}^{(4)} \end{aligned} \quad (\text{B.5})$$

The rank-4 tensor $\mathbf{I}^{(4)}$ is defined as $I_{\alpha\beta\gamma\delta}^{(4)} = \delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}$. We can further simplify the last equation as $|\mathbf{u}|^2 \nabla \rho + 2(\mathbf{u} \cdot \nabla \rho) \mathbf{u}$, hence,

$$\sum_i w_i \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2c_s^2}\right) (\mathbf{c}_i \cdot \nabla \rho) = \left(dc_s^2 + \left(1 - \frac{d}{2}\right) |\mathbf{u}|^2\right) \nabla \rho + 2(\mathbf{u} \cdot \nabla \rho) \mathbf{u} \quad (\text{B.6})$$

We can remove odd-order moments in the third term and evaluate it in a similar manner

$$\begin{aligned} \rho \sum_i w_i \left(\mathbf{c}_i \cdot \nabla \mathbf{u} \cdot \left[\frac{\mathbf{c}_i}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u}) \mathbf{c}_i}{c_s^4} - \frac{\mathbf{u}}{c_s^2} \right] \right) \mathbf{c}_i &= \frac{\rho}{2c_s^4} \sum_i w_i (\mathbf{c}_i \cdot \nabla \mathbf{u} \cdot \mathbf{c}_i) (\mathbf{c}_i \cdot \mathbf{u}) \mathbf{c}_i \\ &\quad - \frac{\rho}{c_s^2} \sum_i w_i (\mathbf{c}_i \cdot \nabla \mathbf{u} \cdot \mathbf{u}) \mathbf{c}_i \\ &= \frac{\rho}{2c_s^4} (\nabla \mathbf{u} \otimes \mathbf{u} \otimes \mathbf{I}) \cdot \sum_i w_i \mathbf{c}_i \otimes \mathbf{c}_i \otimes \mathbf{c}_i \otimes \mathbf{c}_i \\ &\quad - \frac{\rho}{c_s^2} ((\nabla \mathbf{u} \cdot \mathbf{u}) \otimes \mathbf{I}) \cdot \sum_i w_i \mathbf{c}_i \otimes \mathbf{c}_i \\ &= \frac{\rho}{2} (\nabla \mathbf{u} \otimes \mathbf{u} \otimes \mathbf{I}) \cdot \mathbf{I}^{(4)} - \rho ((\nabla \mathbf{u} \cdot \mathbf{u}) \otimes \mathbf{I}) \cdot \mathbf{I} \\ &= \frac{\rho}{2} \left[(\nabla \cdot \mathbf{u}) \mathbf{u} + \frac{1}{2} \nabla |\mathbf{u}|^2 + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] - d\rho (\mathbf{u} \cdot \nabla) \mathbf{u} \\ &= \rho \left[\frac{1}{2} (\nabla \cdot \mathbf{u}) \mathbf{u} + \frac{1}{4} \nabla |\mathbf{u}|^2 + \left(\frac{1}{2} - d\right) (\mathbf{u} \cdot \nabla) \mathbf{u} \right] \end{aligned} \quad (\text{B.7})$$

By combining the three computed terms in (B.3), we obtain an approximate equation for $\sum_i \mathbf{c}_i f_i^{\text{neq}} = \mathbf{0}$ that is linear in both $\nabla \rho$ and $\nabla \mathbf{u}$. Another equation is needed to obtain needed values to evaluate the gradient in (B.2). The same calculation for $\sum_i f_i^{\text{eq}} = 0$ yields the equation

$$-\tau (\mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u}) = 0 \quad (\text{B.8})$$

Assume a quantum linear solver outputs a solution $(\nabla \rho, \nabla \mathbf{u})$ to the system of equations, then we can compute ∇f_i^{eq} from it and the post-streaming distributions $f_i^{\text{eq}}(\mathbf{x} + \Delta \mathbf{x}) \approx f_i^{\text{eq}}(\mathbf{x}) + \Delta \mathbf{x} \cdot \nabla f_i^{\text{eq}}(\mathbf{x})$. With $\Delta \mathbf{x} = -\mathbf{c}_i$, we result in an *approximate streaming* scheme consistent with the quantum LBM algorithm

for tensor-product encoding. The error of this streaming method is not only $O(K_n)$ in its formulation but also depends on specific quantum algorithms to solve the linear system of equations and compute subsequent evaluations.