An end-to-end quantum algorithm for nonlinear fluid dynamics with bounded quantum advantage

David Jennings[1], Kamil Korzekwa*[1], Matteo Lostaglio[1], Richard Ashworth[2], Emanuele Marsili[2], and Stephen Rolston[2]

[1]PsiQuantum, 700 Hansen Way, Palo Alto, CA 94304, USA
[2]Airbus Operations Ltd, Pegasus House Aerospace Avenue, Filton, Bristol, UK

Computational fluid dynamics (CFD) is a cornerstone of classical scientific computing, and there is growing interest in whether quantum computers can accelerate such simulations. To date, the existing proposals for fault-tolerant quantum algorithms for CFD have almost exclusively been based on the Carleman embedding method, used to encode nonlinearities on a quantum computer. In this work, we begin by showing that these proposals suffer from a range of severe bottlenecks that negate conjectured quantum advantages: lack of convergence of the Carleman method, prohibitive time-stepping requirements, unfavorable condition number scaling, and inefficient data extraction. With these roadblocks clearly identified, we develop a novel algorithm for the incompressible lattice Boltzmann equation that circumvents these obstacles, and then provide a detailed analysis of our algorithm, including all potential sources of algorithmic complexity, as well as gate count estimates. We find that for an end-to-end problem, a modest quantum advantage may be preserved for selected observables in the high-error-tolerance regime. We lower bound the Reynolds number scaling of our quantum algorithm in dimension D at Kolmogorov microscale resolution with $O(\mathrm{Re}^{\frac{3}{4}(1+\frac{D}{2})}\times q_M)$, where q_M is a multiplicative overhead for data extraction with $q_M = O(\mathrm{Re}^{\frac{3}{8}})$ for the drag force. This upper bounds the scaling improvement over classical algorithms by $O(\mathrm{Re}^{\frac{3D}{8}})$. However, our numerical investigations suggest a lower speedup, with a scaling estimate of $O(\mathrm{Re}^{1.936}\times q_M)$ for $D = 2$. Our results give robust evidence that small, but nontrivial, quantum advantages can be achieved in the context of CFD, and motivate the need for additional rigorous end-to-end quantum algorithm development.

Contents

*Lead author email: kkorzekwa@psiquantum.com (authors are listed alphabetically within each affiliation).

1 Overview

1.1 Introduction

The development of classical computers, and associated numerical methods, revolutionized engineering by making it possible to numerically solve formerly intractable differential equations, particularly in fluid dynamics. While early methods focused on special cases such as two-dimensional flows, modern high-performance computers allow one to solve highly-detailed, three-dimensional Navier-Stokes equations [1, 2]. Nevertheless, accurately simulating highly turbulent flows with direct numerical simulation (DNS) – relevant for many industrial applications, but of particular relevance in aircraft design – is often computationally infeasible [3]. Leading practitioners hence resort to less accurate turbulence models such as Reynolds-averaged Navier-Stokes or large eddy simulations [4]. More recently, parallelization approaches using GPUs, as well as machine learning models, have been developed [5]. Nonetheless, access to large-scale DNS solutions with the fine resolution required across vast lengths to model all scales in their complexity is not expected to be possible anytime soon.

A new paradigm of quantum computing potentially offers a way to break such classical computational barriers. On a mathematical level, quantum computing enables efficient implementation of unitary operations on exponentially large Hilbert spaces. While the restriction to unitary operations might seem limiting at first sight, block-encoding techniques can be used to encode general matrices inside blocks of the unitary ones [6]. When these block-encodings are combined with quantum algorithmic techniques, such as linear combination of unitaries or quantum signal processing [7], they allow one to perform linear algebra operations on quantum computers. It then becomes possible to solve large linear systems exponentially faster than using classical computers, subject to important caveats on the condition number, as well as data loading and readout [8]. Correspondingly, discretized linear differential equations can be solved via quantum linear solver algorithms [9–13] as a single linear system [14–21], where the solution is stored coherently in a quantum state and needs to be post-processed in a read-out step to extract classical information about the solution. As the Schrödinger equation is a linear differential equation, it is expected that quantum computers will offer exponential advantage for the simulation of quantum systems [23–27]. Moreover, it has been shown that there exist classical linear dynamical systems that can be solved with a provable exponential quantum speedup compared to classical computers [28].

However, industrially relevant computational fluid dynamics (CFD) problems – which account for a significant portion of modern high-performance computing workloads – are inherently nonlinear. As computations on quantum computers are fundamentally linear, leveraging them for CFD simulations requires reformulating the governing equations using a linear representation. One prominent method is Carleman linearization, where the finite-dimensional nonlinear dynamics is transformed to infinite-dimensional linear dynamics, and then truncated to a very high, but finite-dimensional, linear system [29, 30]. This method has been recently proposed and studied as a means of developing quantum algorithms for solving nonlinear differential equations [31–36]. In the CFD context, this path has been explored by the authors of Refs. [37–40], who investigated quantum algorithms based on Carleman linearization to simulate the lattice Boltzmann equation (LBE) [41].[1]

In this work, we first critically assess these endeavors, pointing to several issues that prevent the current proposals from outperforming the existing classical algorithms. In particular, we pinpoint critical complexity bottlenecks that negate prior claims of quantum advantage in the literature. We then introduce a new approach that aims at overcoming these bottlenecks. Our approach is based on a shifted version of the incompressible lattice Boltzmann equation [45, 46] that is linearized using a discrete Carleman embedding method [47–49] and solved using a quantum linear solver [12]. Through a careful analysis of query and gate complexities, supported by extensive numerical investigations, we show that our algorithm can achieve a modest quantum advantage. However, we also highlight the challenges that algorithms based on Carleman embedding face when applied to industrially relevant problems, as the convergence of the method may be limited to low–Reynolds-number CFD simulations.

[1] For other quantum approaches to CFD problems see Refs. [22, 42, 43] or Sec. I of Ref. [44].

The paper is organized as follows. In the remainder of this section, we first present the necessary background material on the lattice Boltzmann equation, and describe recent quantum algorithms for solving it. We then explain a number of difficulties that these algorithms face, and summarize the results we obtained while trying to overcome them. Next, in Sec. 2, we give the details of our approach to the LBE problem that avoids the difficulties mentioned, formulating a linear system of equations whose solution describes the LBE dynamics. With the LBE problem specified, in Sec. 3 we present the details of our quantum algorithm for solving it and extracting the information about the drag force. Then, in Sec. 4, we analyze the performance of the algorithm, estimating its query and gate complexity costs. In Sec. 5, we discuss potential for quantum advantage and present conclusions. Finally, some of the technical details of our work can be found in appendices, in particular Appendix A contains the summary of the notation used throughout the paper.

1.2 Lattice Boltzmann equation

Lattice Boltzmann equation (LBE) is a kinetic model of fluid dynamics that describes the fluid mesoscopically [41]. It conserves mass and momentum by construction and its low-order moments recover the Navier-Stokes equations in the regime where the characteristic length scales of the problem are much larger than the molecular mean path. Within LBE, the fluid is modeled at time t and position $\boldsymbol{r}$ in D-dimensional space by a vector-valued function $\boldsymbol{f}(\boldsymbol{r},t)$, whose components $f_m(\boldsymbol{r},t)$, with $m \in \{1,\ldots,Q\}$, are proportional to the probability density of finding a fluid particle with a discrete velocity $\boldsymbol{e}_m$. The local mass density ρ and momentum density $\rho\boldsymbol{u}$ of the fluid are then given by:

$$\rho(\boldsymbol{r},t) := \sum_{m=1}^{Q} f_m(\boldsymbol{r},t), \qquad \rho\boldsymbol{u}(\boldsymbol{r},t) := \sum_{m=1}^{Q} f_m(\boldsymbol{r},t)\boldsymbol{e}_m. \tag{1.1}$$

LBE is derived from the *discrete Boltzmann equation* (DBE) describing continuous dynamics of $\boldsymbol{f}$ via [41, 50]

$$\frac{\partial f_m(\boldsymbol{r},t)}{\partial t} + \boldsymbol{e}_m \cdot \nabla f_m(\boldsymbol{r},t) = \Omega_m(\boldsymbol{r},t), \tag{1.2}$$

where $\boldsymbol{e}_m \cdot \nabla f_m$ is an advection term that describes *streaming*, i.e., the flow of the fluid in space, while Ω describes *collisions* that are local in space. The simplest choice of Ω that we will adopt is the Bhatnagar-Gross-Krook (BGK) collision operator given by [51]

$$\Omega_m(\boldsymbol{r},t) = -\frac{1}{\tau}\left[f_m(\boldsymbol{r},t) - f_m^{\mathrm{eq}}(\boldsymbol{r},t)\right], \tag{1.3}$$

where τ is the relaxation time determining the speed of equilibration, and $\boldsymbol{f}^{\mathrm{eq}}$ is the Taylor expansion of the local Maxwell equilibrium up to second order terms:

$$f_m^{\mathrm{eq}}(\boldsymbol{r},t) = \rho(\boldsymbol{r},t)\left(1 + \frac{\boldsymbol{e}_m \cdot \boldsymbol{u}(\boldsymbol{r},t)}{c_s^2} + \frac{(\boldsymbol{e}_m \cdot \boldsymbol{u}(\boldsymbol{r},t))^2}{2c_s^4} - \frac{|\boldsymbol{u}(\boldsymbol{r},t)|^2}{2c_s^2}\right) w_m + O(\mathrm{Ma}^3). \tag{1.4}$$

In the above, c_s denotes the speed of sound, $\mathrm{Ma} := |\boldsymbol{u}|/c_s$ is the Mach number, and $\boldsymbol{w}$ is a vector of Maxwell weightings for the different discrete velocities satisfying $w_m \geq 0$ and $\sum_m w_m = 1$. These weightings are chosen to ensure the lowest-order velocity moments are reproduced and conservation laws are respected. Since the error in the above second-order expansion of the equilibrium density is $O(\mathrm{Ma}^3)$, one assumes a small Mach number, typically of order $\mathrm{Ma} \sim 10^{-2}$.

In order to address the DBE with numerical methods, one needs to discretize space and time. In *lattice Boltzmann methods*, spatial discretization is achieved by choosing a uniform and regular lattice with spacing Δx and a discrete velocity set $\{\boldsymbol{e}_m\}$ whose components are integer multiples of $\Delta x/\Delta t$, where Δt is the duration of one discrete time step. Now, using the method of characteristics, one can integrate the DBE over one time step to obtain

$$f_m(\boldsymbol{r} + \boldsymbol{e}_m\Delta t, t + \Delta t) = f_m(\boldsymbol{r},t) + \int_0^{\Delta t} \Omega_m(\boldsymbol{r} + \boldsymbol{e}_m s, t + s)ds. \tag{1.5}$$

D1Q3	$m = 1$:	$\boldsymbol{e}_m^\star = 0$,	$w_m = 2/3$,
	$m \in \{2, 3\}$:	$\boldsymbol{e}_m^\star \in \{\pm 1\}$,	$w_m = 1/6$.
D2Q9	$m = 1$:	$\boldsymbol{e}_m^\star = (0, 0)$,	$w_m = 4/9$,
	$m \in \{2, \ldots, 5\}$:	$\boldsymbol{e}_m^\star \in \{(\pm 1, 0), (0, \pm 1)\}$,	$w_m = 1/9$,
	$m \in \{6, \ldots, 9\}$:	$\boldsymbol{e}_m^\star \in \{(\pm 1, \pm 1)\}$,	$w_m = 1/36$.
D3Q27	$m = 1$:	$\boldsymbol{e}_m^\star = (0, 0, 0)$,	$w_m = 8/27$,
	$m \in \{2, \ldots, 7\}$:	$\boldsymbol{e}_m^\star \in \{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$,	$w_m = 2/27$,
	$m \in \{8, \ldots, 19\}$:	$\boldsymbol{e}_m^\star \in \{(\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1)\}$,	$w_m = 1/54$,
	$m \in \{20, \ldots, 27\}$:	$\boldsymbol{e}_m^\star \in \{(\pm 1, \pm 1, \pm 1)\}$,	$w_m = 1/216$.

Table 1: Discrete velocities in lattice units for standard LBE models [41]. The discrete weightings w_m are chosen so as to reproduce low-order velocity moments of the Maxwell-Boltzmann distribution.

Using the first-order (rectangular) discretization to approximate the integral on the right hand side,

$$\int_0^{\Delta t} \Omega_m(\boldsymbol{r} + \boldsymbol{e}_m s, t + s) ds = \Delta t \Omega_m(\boldsymbol{r}, t) + O(\Delta t^2), \tag{1.6}$$

one arrives at the original *lattice Boltzmann equation* (LBE):

$$f_m(\boldsymbol{r} + \boldsymbol{e}_m \Delta t, t + \Delta t) = f_m(\boldsymbol{r}, t) - \frac{\Delta t}{\tau} \left[f_m(\boldsymbol{r}, t) - f_m^{\mathrm{eq}}(\boldsymbol{r}, t) \right]. \tag{1.7}$$

The above, however, is only first-order accurate in the step size Δt, and so requires very small $\Delta t \ll \tau$ to accurately simulate the fluid dynamics. This in turn translates into a large number of time steps and usually an inefficient simulation (we will discuss this in more detail in Sec. 1.4.3).

To overcome this first-order accuracy issue, one uses the second-order (trapezoidal) discretization to approximate the considered integral with

$$\int_0^{\Delta t} \Omega_m(\boldsymbol{r} + \boldsymbol{e}_m s, t + s) ds = \frac{\Delta t}{2} \left[\Omega_m(\boldsymbol{r} + \boldsymbol{e}_m \Delta t, t + \Delta t) + \Omega_m(\boldsymbol{r}, t) \right] + O(\Delta t^3). \tag{1.8}$$

One then follows the original idea of Refs. [52, 53], and introduces a new set of variables

$$\bar{f}_m(\boldsymbol{r}, t) = f_m(\boldsymbol{r}, t) - \frac{\Delta t}{2} \Omega_m(\boldsymbol{r}, t), \tag{1.9}$$

as well as the shifted relaxation time

$$\bar{\tau} = \tau + \frac{\Delta t}{2}. \tag{1.10}$$

Using these together with Eq. (1.8) in Eq. (1.5), one arrives at the commonly used lattice Boltzmann equation

$$\bar{f}_m(\boldsymbol{r} + \boldsymbol{e}_m \Delta t, t + \Delta t) = \bar{f}_m(\boldsymbol{r}, t) - \frac{\Delta t}{\bar{\tau}} [\bar{f}_m(\boldsymbol{r}, t) - f_m^{\mathrm{eq}}(\boldsymbol{r}, t)], \tag{1.11}$$

which is second-order accurate and does not restrict one to $\Delta t \ll \tau$. Although in this approach one follows the transformed variables $\bar{f}_m$ rather than the original ones f_m, the macroscopic variables ρ and $\boldsymbol{u}$ stay unchanged, as

$$\sum_{m=1}^{Q} \bar{f}_m = \sum_{m=1}^{Q} f_m, \qquad \sum_{m=1}^{Q} \bar{f}_m \boldsymbol{e}_m = \sum_{m=1}^{Q} f_m \boldsymbol{e}_m. \tag{1.12}$$

Next, "lattice units" are used to represent all physical parameters by dimensionless numbers. Denoting physical quantities in lattice units by a $\star$ superscript, the time steps are chosen to be of unit size, $\Delta t^\star = 1$. Moreover, for the spatial discretization to also be of unit size, $\Delta x^\star = 1$, the velocities $\boldsymbol{e}_m^\star$ are chosen to have unit entries. The typical choice of discrete velocities for one-, two- and three-dimensional problems is given by the D1Q3, D2Q9 and D3Q27 models, where the velocities $\boldsymbol{e}_m^\star$ and their corresponding weights w_m are given in Table 1.

Using lattice units, the first-order and second-order accurate lattice Boltzmann equations, Eqs. (1.7) and (1.11), read

$$f_m(\boldsymbol{r}^\star + \boldsymbol{e}_m^\star, t^\star + 1) = f_m(\boldsymbol{r}^\star, t^\star) - \frac{1}{\tau^\star}[f_m(\boldsymbol{r}^\star, t^\star) - f_m^{\mathrm{eq}}(\boldsymbol{r}^\star, t^\star)], \tag{1.13a}$$

$$\bar{f}_m(\boldsymbol{r}^\star + \boldsymbol{e}_m^\star, t^\star + 1) = \bar{f}_m(\boldsymbol{r}^\star, t^\star) - \frac{1}{\bar{\tau}^\star}[\bar{f}_m(\boldsymbol{r}^\star, t^\star) - f_m^{\mathrm{eq}}(\boldsymbol{r}^\star, t^\star)], \tag{1.13b}$$

where $t^\star$ is an integer between 0 (initial time) and $T^\star$ (final time), whereas each $r_i^\star$ is an integer between 1 (one edge of the simulation region) and N_i (the other edge). The conversion to physical units is given by:

$$t = t^\star \Delta t, \quad T = T^\star \Delta t, \quad \tau = \tau^\star \Delta t = \left(\bar{\tau}^\star - \frac{1}{2}\right)\Delta t, \tag{1.14a}$$

$$\boldsymbol{r} = \boldsymbol{r}^\star \Delta x, \quad L_i = N_i \Delta x, \tag{1.14b}$$

$$\boldsymbol{e}_m = \boldsymbol{e}_m^\star \frac{\Delta x}{\Delta t}, \quad \boldsymbol{u} = \boldsymbol{u}^\star \frac{\Delta x}{\Delta t}, \quad c_s = c_s^\star \frac{\Delta x}{\Delta t} \tag{1.14c}$$

where T is the total evolution time, L_i is the spatial dimension in direction i, and Δx is the physical distance between neighboring lattice points.

The LBE is then usually divided into separate collision and streaming steps in the following way:

$$\bar{\boldsymbol{f}}^C(\boldsymbol{r}^\star, t^\star) = \bar{\boldsymbol{f}}(\boldsymbol{r}^\star, t^\star) - \frac{1}{\bar{\tau}^\star}[\bar{\boldsymbol{f}}(\boldsymbol{r}^\star, t^\star) - \boldsymbol{f}^{\mathrm{eq}}(\boldsymbol{r}^\star, t^\star)], \tag{1.15}$$

$$\bar{f}_m(\boldsymbol{r}^\star + \boldsymbol{e}_m^\star, t^\star + 1) = \bar{f}_m^C(\boldsymbol{r}^\star, t^\star), \tag{1.16}$$

with analogous equations for the unbarred variables corresponding to the first-order accurate method.

Finally, the connection between the LBE and the macroscopic equations of fluid mechanics is provided by the Chapman-Enskog analysis [41]. It is based on a perturbation expansion of $\bar{f}_m = \bar{f}_m^{\mathrm{eq}} + \epsilon \bar{f}_m^{(1)} + \epsilon^2 \bar{f}_m^{(2)} + O(\epsilon^3)$ around the equilibrium distribution with the expansion parameter $\epsilon \propto \mathrm{Kn}$, the *Knudsen number* (the ratio of the mean free path to the characteristic length-scale of the problem). Then, by taking moments of this expansion and keeping only the two lowest order terms, one recovers the Navier-Stokes equations in the $\mathrm{Kn} \ll 1$ regime, with the kinematic shear viscosity ν given by the LBE relaxation time τ as

$$\nu = c_s^2 \tau = c_s^2 \left(\bar{\tau} - \frac{\Delta t}{2}\right). \tag{1.17}$$

We can rewrite the right hand side of the above using the lattice units and physical discretizations,

$$\nu = c_s^{\star 2} \left(\bar{\tau}^\star - \frac{1}{2}\right) \frac{\Delta x^2}{\Delta t}. \tag{1.18}$$

Employing the fact that the lattice speed of sound for D1Q3, D2Q9 and D3Q27 models is given by $1/\sqrt{3}$, one obtains a relation between a physical value of kinematic viscosity ν and the simulation parameters:

$$\Delta t = \frac{\Delta x^2}{3\nu} \left(\bar{\tau}^\star - \frac{1}{2}\right). \tag{1.19}$$

To model the same physical scenario, characterized by viscosity ν, one then has the freedom to choose two out of three model parameters $\bar{\tau}^\star$, Δx, and Δt, so that they satisfy the above relation. Although different choices do not affect the modeled physics, they influence the accuracy, stability and efficiency of both the classical and quantum LBE algorithms.

1.3 Quantum algorithms for fluid dynamics

Let us now briefly summarize the most recent quantum approaches to simulating the lattice Boltzmann equation presented in Refs. [37, 38]. These works focus on the LBE indirectly, by developing

a quantum algorithm for solving the underlying discrete Boltzmann equation. They start from the partial differential equation given by the DBE, Eq. (1.2), and perform spatial discretization to transform it into an ordinary differential equation (ODE). Since the obtained ODE is nonlinear, Carleman linear embedding is then employed, and a linear ODE describing the fluid dynamics in a large linear space is derived. Finally, the resulting linear ODE is solved using known quantum algorithms for solving linear ODEs [16, 21, 33], and the output quantum state, coherently encoding the solution, can be then measured to extract information about the fluid system.

In order to transform the DBE into an ordinary differential equation, the authors of Refs. [37, 38] consider a spatial discretization of the simulation region, assumed to be a 3-dimensional rectangular cuboid of size $L_x \times L_y \times L_z$, into a grid of $N = N_x N_y N_z$ points with nearest neighbors separated by Δx. As a result, instead of Q-dimensional vectors $\boldsymbol{f}(\boldsymbol{r}, t)$ at each point in space and time, they are dealing with d-dimensional vectors $\boldsymbol{f}(t)$ at each point in time with $d = NQ$, where

$$f_{m,\boldsymbol{r}}(t) := f_m(\boldsymbol{r}, t), \qquad \frac{r_i}{\Delta x} \in \{1, \ldots, N_i\}, \qquad \Delta x := \frac{L_i}{N_i}, \qquad i \in \{x, y, z\}. \tag{1.20}$$

With such a discretization, and assuming weak incompressibility, the DBE in Eq. (1.2) can be represented by an ordinary nonlinear differential equation for a vector $\boldsymbol{f}(t)$ as

$$\frac{d\boldsymbol{f}(t)}{dt} = (G + F_1)\boldsymbol{f}(t) + F_2 \boldsymbol{f}^{\otimes 2}(t) + F_3 \boldsymbol{f}^{\otimes 3}(t). \tag{1.21}$$

The details of this derivation can be found in Appendix B, together with the explicit forms of matrices G and F_k.

Next, the authors of Refs. [37, 38] replace the obtained nonlinear ODE from Eq. (1.21) with a linear ODE in a larger space using Carleman embedding. To explain how this can be achieved, let us introduce an infinite vector $\boldsymbol{y}^\infty(t)$ of size $d + d^2 + d^3 + \ldots$,

$$\boldsymbol{y}^\infty(t) := [\boldsymbol{y}_1(t), \boldsymbol{y}_2(t), \ldots]. \tag{1.22}$$

Then, formally the non-linear DBE evolution for $\boldsymbol{f}(t)$ from Eq. (1.21) is equivalent to the following infinite linear ODE for $\boldsymbol{y}^\infty$:

$$\frac{d\boldsymbol{y}^\infty(t)}{dt} = \mathcal{C}^\infty \boldsymbol{y}^\infty(t), \qquad \boldsymbol{y}_k(0) = \boldsymbol{f}^{\otimes k}(0), \tag{1.23}$$

where

$$\mathcal{C}^\infty = \begin{pmatrix} C_1^1 & C_2^1 & C_3^1 & 0 & \ldots \\ 0 & C_2^2 & C_3^2 & C_4^2 & \ldots \\ 0 & 0 & C_3^3 & C_4^3 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad C_j^k := \sum_{i=0}^{k-1} I^{\otimes i} \otimes (F_{j-k+1} + \delta_{j,k} G) \otimes I^{\otimes k-1-i}, \tag{1.24}$$

is the infinite Carleman matrix with 3 non-zero blocks in each row, and $\delta_{j,k}$ is the Kronecker delta. Truncating this infinite set of equations at level N_C, so that one deals with the finite Carleman vector $\boldsymbol{y} := [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{N_C}]$, gives

$$\frac{d\boldsymbol{y}(t)}{dt} = \mathcal{C}\boldsymbol{y}(t), \qquad \boldsymbol{y}_k(0) = \boldsymbol{f}^{\otimes k}(0), \tag{1.25}$$

with $\mathcal{C}$ being the Carleman matrix given by $\mathcal{C}^\infty$ restricted to the first N_C blocks. Such truncation introduces an error ϵ_C, so that

$$\boldsymbol{y}_1(t) \approx_{\epsilon_C} \boldsymbol{f}(t) \tag{1.26}$$

only under certain conditions and for large enough N_C. Identifying conditions under which this occurs is a crucial, and often involved, question one needs to answer towards an efficient quantum algorithm.

Under the assumption that the problem from Eq. (1.25) approximates well enough the discrete Boltzmann system in the sense of Eq. (1.26), one can attempt to run a quantum linear solver

(QLS) algorithm to output its solution. More precisely, modulo time-discretization errors, the solver outputs a normalized Carleman state at the final time T:

$$|\boldsymbol{y}(T)\rangle = \frac{1}{\|\boldsymbol{y}(T)\|}[\boldsymbol{y}_1(T), \ldots \boldsymbol{y}_{N_C}(T)], \tag{1.27}$$

or a *history state* encoding a superposition of the solution at different times:

$$|\boldsymbol{y}_H\rangle = \sum_{m=1}^{\lceil T/\Delta t \rceil} \frac{\|\boldsymbol{y}(m\Delta t)\|}{\sqrt{\mathcal{N}}} |\boldsymbol{y}((m\Delta t))\rangle \otimes |m\Delta t\rangle\,, \qquad \mathcal{N} = \sum_{m=1}^{\lceil T/\Delta t \rceil} \|\boldsymbol{y}(m\Delta t)\|^2. \tag{1.28}$$

The cost of obtaining these solution states depends crucially on the condition number of the linear system of equations obtained after time-discretization, see Refs. [16, 21, 33, 54] and references therein, but also on norm properties. This cost is typically defined in terms of number of times we need to apply two core unitaries, one that encodes $\mathcal{C}$ in one of its blocks, and another that encodes the preparation of a quantum state based on $\boldsymbol{f}(0)$. Establishing constructions for these unitaries, presenting evidence for a favorable scaling of the condition number and favorable norm properties, are necessary conditions for any claim of quantum advantage. Finally, non-trivial post-processing is required to extract data about the fluid system from the output of the QLS, i.e., from the quantum state $|\boldsymbol{y}(T)\rangle$ or $|\boldsymbol{y}_H\rangle$.

The works in Refs. [37, 38] are important, as they push our understanding forward by analyzing various components of this pipeline. However, as we shall see in the next section by identifying bottlenecks preventing quantum advantage, they do not present a convincing argument for the suitability of this approach.

1.4 Current limitations

1.4.1 Problem 1: Applicability of the Carleman approach

The Carleman linear embedding approach to fluid simulation involves extending the nonlinear fluid system into a large, but linear system of equations. The core parameter here is N_C, the Carleman truncation order. The resulting ODE system has dimension $O(d^{N_C})$, and the two core questions are:

1. Does the Carleman truncation error ϵ_C converge? This means that ϵ_C (in the sense of Eq. (1.26)) should decrease with increasing N_C. Certain sufficient conditions are known for special classes of systems (dissipative [30, 31, 33], nonresonant [35], and conservative [36]), but these are not directly applicable to our setting, which is why we resort to numerical evidence.

2. Assuming the Carleman truncation error ϵ_C converges, what is a suitable choice of N_C in practice?

Limited numerical analysis is currently available for the Carleman truncation error ϵ_C for the discrete Boltzmann equation:

- The authors of Ref. [37] presented a preliminary numerical study of ϵ_C for a collisions-only D1Q3 model, with $\tau^\star = 1$, a few hundred time steps, $N = 1$ (since no streaming is included, a single node is taken), suggesting that $N_C = 3$ suffices to get errors below 10^{-14}. The issue with this approach is that, in such a purely collisional setting, one can actually prove analytically that the Carleman error for truncation order $N_C \geq 3$ is exactly zero, see Appendix C for details. Nontrivial ϵ_C for $N_C \geq 3$ can only be generated through the interplay of both collisions and streaming, and hence cannot be assessed via numerics on collisions only. The reported figure $\epsilon_C \approx 10^{-14}$ figure seems then to be due to numerical scheme precision rather than the actual Carleman error.

- An $N_C = 3$ truncation was chosen in the study in Ref. [38], but the question of truncation errors is left to further analysis.

- The authors of Ref. [39] instead analyzed a Kolmogorov flow in the D2Q9 model with $N_x = N_y = 32$ and around 100 time steps, setting a truncation $N_C = 2$, which entirely foregoes the cubic term in the nonlinearity. Root mean square errors of $10^{-4} - 10^{-3}$ were observed in the low Reynolds number regime of $\mathrm{Re} \in [10, 100]$.

- A similar setup was analyzed in Ref. [40] with $N_C \in \{1, 2\}$, $N_x = N_y \in \{16, \ldots, 64\}$, $\mathrm{Re} \in [25, 50]$, where errors of around 10^{-3} were found. Importantly, in the settings considered, it was found that pure linearization, $N_C = 1$, outperforms the $N_C = 2$ Carleman truncation.

It is therefore problematic to draw from these results any conclusions concerning the ability of the Carleman procedure to capture genuinely nonlinear effects for two reasons. First, the pure linearization errors are already fairly small, and so the analysed flows must exhibit very weak nonlinearities. Second, the increase of the Carleman error when going from $N_C = 1$ to $N_C = 2$ suggests that one may already be operating beyond the convergence radius of the Carleman procedure. Thirdly, there is no obvious trend to be gathered from these few scattered studies.

We thus conclude that the question of applicability of the Carleman embedding method to the lattice Boltzmann equation is still mostly unanswered. Since a linear embedding of the nonlinear dynamics is pivotal to any quantum algorithm, in this work we shall consider this problem closely.

1.4.2 Problem 2: Scaling of the condition number

Let us now assume that we are in a scenario where Problem 1 is solved, i.e., for the use-case of interest a high enough Carleman truncation order N_C leads to a good approximation of the first Carleman block: $\boldsymbol{y}_1(t) \approx \boldsymbol{f}(t)$. The next step is to design a quantum algorithm that prepares the Carleman state $|\boldsymbol{y}(T)\rangle$ (or the Carleman history state $|\boldsymbol{y}_H\rangle$) describing the fluid system at the final time T (or during the whole evolution between $t = 0$ and $t = T$). Recent quantum approaches to the discrete Boltzmann equation rely on quantum linear solvers applied to a linear system arising from time-discretization of Eq (1.25). A crucial parameter to be evaluated here is the scaling of the condition number κ of that linear system, since quantum linear solvers have complexity scaling at best linearly with it [10]. Checking this scaling is crucial for any claim of potential quantum advantage, since under an unfavorable scaling the quantum algorithm may end up performing substantially worse than the best known classical algorithms.

A standard approach that has been followed in recent works [37, 38] is based on discretizing time via a truncated Taylor series method applied to Eq. (1.25). Several works presented condition number upper bounds for this setting under slight variations [21, 33, 55]. For the class of problems at hand, it was found that the condition number scales as

$$\kappa = O\left(T, \|\mathcal{C}\|, \max_{t\in[0,T]} \|e^{\mathcal{C}t}\|\right), \tag{1.29}$$

where we recall that T is the total simulation time, $\mathcal{C}$ is the truncated Carleman matrix, and $\|\cdot\|$ denotes the operator norm (i.e., the largest singular value).

Concerning the norm of $\mathcal{C}$, in Ref. [37] [Eq. (93)], it is claimed that $\|\mathcal{C}\| = O(\tau^{-1})$, where τ is a constant. However, the top-left block of the matrix $\mathcal{C}$ includes the finite-difference discretization of the streaming operator $\boldsymbol{e}_m \cdot \nabla f_m$. In turn, this means that the largest element in $\mathcal{C}$ scales as $\Delta x^{-1} = O(N_x, N_y, N_z)$, and hence $\|\mathcal{C}\|$ must also scale as $O(N_x, N_y, N_z)$ or worse. This negates the headline exponential advantage in N, independently of the question of the initial state preparation and data extraction. Also, note that in the context of the lattice Boltzmann equation, one has $\|\mathcal{C}\| = O((\bar{\tau}^\star)^{-1})$ (which does not scale with N), but this is simply because we moved to lattice units, where the advection time is N-dependent, so we just shifted the N-scaling to the number of required time steps. A more detailed analysis of $\|\mathcal{C}\|$ for this case can be found in Ref. [38]. Overall, this tells us that the best we can hope for the quantum algorithm under current approaches is a polynomial (in fact cubic) speedup, not an exponential one.[2]

Now, consider $\max_{t\in[0,T]} \|e^{\mathcal{C}t}\|$. In Ref. [37] [Sec. 5e], it is claimed that this quantity is $O(1)$ under stability assumptions. Formally, the claim is that the norm of $e^{\mathcal{C}t}$ is $O(1)$ when $\mathcal{C}$ is marginally

[2] This limitation could potentially be handled via a different algorithm implementing an 'interaction-picture' in the frame of streaming, but this would require further analysis.

stable, i.e., when the largest real part of the eigenvalues of $\mathcal{C}$ is nonpositive. While it is true that $\|e^{\mathcal{C}t}\|$ can be upper bounded by a quantity independent of t, $\max_{t\in[0,T]}\|e^{\mathcal{C}t}\|$ generally can depend on N.[3] The work in Ref. [38] instead sets $\max_{t\in[0,T]}\|e^{\mathcal{C}t}\|$ equal to 1, as a placeholder until further analysis becomes available.

In summary, pinning down the N dependence of the condition number is crucial for determining the size of a quantum advantage, if any exists. In this work, we shall look at this problem in detail.

1.4.3 Problem 3: Prohibitively small time steps

Previous quantum algorithmic approaches to the lattice Boltzmann equation focused on the discrete Boltzmann equation (DBE) from Eq. (1.2). This is a natural starting point to apply the Carleman linear embedding formalism, as well as quantum algorithms whose analysis normally assumes a time-continuous model. However, accuracy of a direct time-discretization of the DBE requires $\Delta t \ll \tau$. For air at atmospheric pressure and room temperature, one has $\tau \sim 10^{-10}s$, so a typical target evolution timescale of $10^{-2} - 10^{-1}s$ leads to a prohibitively large number of time steps. That is the reason why classical algorithms normally target the lattice Boltzmann equation (LBE) in Eq. (1.15)-(1.16), which allows one to take $\Delta t \gg \tau$, as discussed in Sec. 1.2.

The authors of Ref. [37] (arXiv version) take the approach of direct time-discretization of the DBE. The aforementioned cost appears via the scaling with the inverse Knudsen number Kn^{-1}, which is a prohibitive overhead (typical values in the atmosphere are $\mathrm{Kn} > 10^8$). The approach followed in Ref. [38], and the published version of Ref. [37], is instead to argue that one can consider a time-continuous evolution (as in DBE), but with the physical relaxation time τ replaced by the shifted relaxation time in lattice units $\bar{\tau}^\star$.[4] Since $\bar{\tau}^\star$ is always above $1/2$ due to stability requirements, the time step problem seemingly disappears. The issue here is that, if one wants to retain a time-continuous discrete Boltzmann formulation as in Eq. (1.2), but wishes to replace τ with $\bar{\tau}^\star$ to avert the prohibitive number of time steps, then an extra second-order memory term has to be introduced, leading to a generalized Boltzmann equation (see, e.g., Ref. [56], Eq. (24)). The approach of Refs. [37, 38] is then seen to be equivalent to dropping the memory term from this generalized equation, which has the drawback of losing second-order accuracy and reintroducing the need for a prohibitively large number of time steps.

Therefore, the currently proposed quantum approaches may not be competitive in practice. In this work, we shall address this time-step problem.

1.4.4 Problem 4: Inefficient data extraction

Finally, assuming that one somehow deals with the Carleman convergence problem and the complexity of obtaining the final time Carleman state $|\boldsymbol{y}(T)\rangle$ from Eq. (1.27), there is still the problem of extracting data from the quantum state encoding the dynamics. We will focus on the problem of extracting classical information about the fluid at time T from $|\boldsymbol{y}(T)\rangle$. Note that analogous issues as discussed below appear when one wants to extract information from the history state $|\boldsymbol{y}_H\rangle$.

First, note that it is not difficult to estimate that in the weakly compressible (or incompressible) limit, at all times t one has

$$\sqrt{\frac{N}{Q}} \lesssim \|\boldsymbol{f}(t)\| \lesssim \sqrt{N}. \tag{1.30}$$

As a result, the norm of the k-th Carleman block of $|\boldsymbol{y}(T)\rangle$ scales as

$$\frac{\|\boldsymbol{y}_k(T)\|}{\|\boldsymbol{y}(T)\|} = O(\|\boldsymbol{f}(T)\|^{k-N_C}) = O\left(\sqrt{N}^{k-N_C}\right). \tag{1.31}$$

This, in turn, means that in order to extract information from the first Carleman block (which approximates $\boldsymbol{f}(T)$ well enough), a quantum algorithm incurs a cost $O(\sqrt{N}^{N_C-1})$, simply due to

[3] $\|e^{\mathcal{C}t}\| \leq \kappa_P$ for any $P > 0$ satisfying the Lyapunov condition $P\mathcal{C} + \mathcal{C}^\dagger P < 0$, where κ_P is the condition number of P. The parameter κ_P can bring in an N-dependence.

[4] In Ref. [38], the model considered can be glanced from Eqs. (16)-(17), where comparing our Eq. (1.18) with Eq. (55) of Ref. [38] we see that their τ is our $\bar{\tau}^\star$. Similarly, the model considered in Ref. [37] can be seen from Eq. (7), and again their τ is our $\bar{\tau}^\star$, as can be seen from the comment just after Eq. (92) that $\tau > 1/2$.

the fact that amplitude is exponentially concentrated in the higher order k-blocks of the Carleman vector [34]. Hence, already for $N_C = 3$, which is the smallest order capturing the cubic nonlinearities of DBE, the cost scales linearly in the number of discretization points, so the same as the classical algorithm. Increasing N_C further to decrease the Carleman truncation error renders the quantum algorithm less efficient than its classical counterpart.

Even if we tried to extract information from some higher block $\boldsymbol{y}_k$ (for the price of getting a higher truncation error), we would still incur an extra scaling $O(\sqrt{N}^{N_C-k})$. This extra cost kills quantum advantage over classical algorithms unless $k = N_C$ (when the dependence on N could disappear), or $k = N_C - 1$ (when one could hope for a quadratic improvement from classical $O(N)$ to $O(\sqrt{N})$). However, $\boldsymbol{y}_{N_C}(t) = \boldsymbol{f}_{\text{lin}}^{\otimes N_C}(t)$ with $\boldsymbol{f}_{\text{lin}}$ describing the solution to the *linear* ODE

$$\frac{d\boldsymbol{f}_{\text{lin}}(t)}{dt} = (G + F_1)\boldsymbol{f}_{\text{lin}}(t), \qquad \boldsymbol{f}_{\text{lin}}(0) = \boldsymbol{f}(0), \tag{1.32}$$

which corresponds to applying naive linearization procedure (i.e., dropping the nonlinear terms) applied to the original DBE given by Eq. (1.2), which we could have obtained in a simpler and more direct manner without invoking any Carleman procedure. Similarly, $\boldsymbol{y}_{N_C-1}$ only describes the solution of the *quadratic* ODE, completely ignoring cubic nonlinearities. Moreover, it is not clear whether choosing a high N_C and extracting data from $\boldsymbol{y}_{N_C-1}$ is any better, in terms of truncation error ϵ_C, than just choosing $N_C = 2$ and focusing on $\boldsymbol{y}_1$. Hence, it is quite likely that even if one is in the regime of converging error, one cannot control it without making the algorithm inefficient.

Most existing literature on quantum algorithms for the lattice Boltzmann equation only discusses how to output the Carleman states $|\boldsymbol{y}(T)\rangle$ or $|\boldsymbol{y}_H\rangle$, without discussing the critical problem of how to extract classical information from a coherent encoding of the dynamics [39, 40, 44]. The two works where extraction of data is discussed are Ref [37] and, in more detail, Ref. [38]. However, they appear to miss this problem:

- In Ref. [37], the crucial ratio from Eq. (1.31) is claimed to be $O(1/\sqrt{N_C})$ (Eq. (102)). This can be traced back to the bound of Eq. (85) in Ref. [37], which implicitly assumes $\|\boldsymbol{f}(t)\| \leq 1$, which does not hold. Technically, one can perform a rescaling $\boldsymbol{f} \mapsto \gamma\boldsymbol{f}$ to ensure the norm is smaller than 1 as required, but this adds an extra scaling with N to the condition number, removing any quantum advantage, as we shall see below.

- The authors of Ref. [38], on the other hand, try to avoid the problem by extracting information from all Carleman blocks. The issue with the argument is the assumption that the Carleman procedure outputs a state of the form $\boldsymbol{y}_k \approx \boldsymbol{f}^{\otimes k}$. In reality, due to the reasons discussed above, the amplitude is concentrated in the block $k = N_C$, which simply contains information about the linear dynamics, $\boldsymbol{y}_{N_C}(t) = \boldsymbol{f}_{\text{lin}}^{\otimes N_C}(t)$. Since tackling relevant applications requires high values of N (Ref. [38] reports use-cases with N between 10^{19} and 10^{25}), this basically means that the output of the quantum algorithm will just reflect the results one would obtain from a naive linearization. As such, it cannot capture the nonlinearity as required.

As anticipated, one potential way around the problem described above is to rescale the DBE according to:

$$\boldsymbol{f} \mapsto \gamma\boldsymbol{f}, \quad G \mapsto G, \quad F_1 \mapsto F_1, \quad F_2 \mapsto F_2/\gamma, \quad F_3 \mapsto F_3/\gamma^2. \tag{1.33}$$

By taking $\gamma < 1/\|\boldsymbol{f}\|$, one can make sure the norm of $\boldsymbol{f}$ is rescaled to be smaller than 1, and so one can avoid the extra complexity factor mentioned. However, as we already mentioned in Sec. 1.4.2, the condition number κ of the linear system describing time-discretized DBE evolution for time T scales as $O(\|\mathcal{C}\|T)$. With the rescaling above, we also have

$$\|\mathcal{C}\| = O(1/\gamma^2) = O(\|\boldsymbol{f}\|^2) = O(N), \tag{1.34}$$

and so $\kappa = O(NT)$. Given that the complexity of the best QLS scales linearly with κ, we end up with a quantum algorithm whose complexity scales the same way as the classical algorithms with the number of spatial discretization points N and simulation time T. We then conclude that if the inefficient data extraction problem cannot be mitigated, no quantum advantage for the simulation of nonlinear dynamics is possible.

1.5 Summary of results

In this work, we propose a novel quantum approach to the lattice Boltzmann equation, addressing the problems described in the previous section.

First, in Sec. 2, we reformulate the standard LBE and the existing Carleman embedding approach to it. In Sec. 2.1, we introduce a *shifted incompressible lattice Boltzmann equation* which, for the price of restricting to incompressible flows, allows us to represent the fluid using a state vector $\boldsymbol{g}$, whose norm scales with the magnitude of the lattice velocity $|\boldsymbol{u}^\star|$. Next, in Sec. 2.2, we explain the choice of simulation parameters that forces $|\boldsymbol{u}^\star|$ to be small enough, so that the norm $\|\boldsymbol{g}\|$ is of $O(1)$. This way our approach mitigates the problem of inefficient data extraction described in Sec. 1.4.4. Then, in Sec. 2.3, we describe a *discrete Carleman embedding* [47–49], which allows us to approximate a nonlinear discrete-time evolution of the d-dimensional system using a discrete-time linear evolution of an enlarged system. By leaving time discretization on the classical side and directly embedding the second-order accurate LBE (instead of embedding the time-continuous DBE), we avoid the problem of prohibitively small time steps described in Sec. 1.4.3. Finally, in Sec. 2.4, we construct linear systems A, whose solutions carry information about fluid systems during LBE evolutions with $T^\star$ time steps.

Section 3 contains the details on implementing the quantum algorithm solving the shifted incompressible LBE problem. First, in Sec. 3.1, we explain how to encode the Carleman embedded shifted LBE data in quantum registers. Then, in Sec. 3.2, we construct the unitary block-encoding of A and unitary state preparations required by a quantum linear solver to produce $|\boldsymbol{y}(T)\rangle$ or $|\boldsymbol{y}_H\rangle$. In Sec. 3.3, we present the details concerning the query complexity q_Q of running the quantum linear solver algorithm for our problem, which scales roughly as $q_Q = O(\alpha_A \kappa_A / \|A\|)$, where α_A is the block-encoding prefactor of A. Finally, in Sec. 3.4, we explain how to perform measurements to extract classical information about the drag force. This extraction procedure brings additional multiplicative factor overhead q_M, scaling as $O(\mathrm{Re}^{3/8})$, to the complexity of our quantum algorithm.

The central Sec. 4 discusses the performance of the proposed quantum algorithm: how well it reproduces the lattice Boltzmann dynamics and at what complexity cost. First, in Sec. 4.1, we numerically investigate the applicability of the Carleman approach for the lattice Boltzmann equation, thus addressing the problem discussed in Sec. 1.4.1. We perform extensive numerical simulations for D1Q3 model with Carleman truncation orders $N_C \in \{1, \ldots, 5\}$ and Reynolds numbers $\mathrm{Re} \in [10, 1000]$; and for D2Q9 model with $N_C \in \{1, \ldots, 3\}$ and $\mathrm{Re} \in [10, 250]$. We assume periodic boundary conditions, no walls, and initial states given by one-dimensional sinusoidal and colliding states, as well as two-dimensional Taylor-Green vortices and Gaussian vortex dipoles. We find that for low enough Reynolds numbers, the Carleman truncation error ϵ_C indeed converges, in the sense that increasing N_C decreases ϵ_C. However, we also observe the existence of a threshold Reynolds number Re_T, such that the Carleman procedure does not converge for $\mathrm{Re} > \mathrm{Re}_T$, i.e., increasing truncation order N_C increases the error ϵ_C instead of decreasing it. For one-dimensional systems, this threshold is relatively low, $\mathrm{Re}_T \sim 10^2$, for two-dimensional systems it is higher by an order of magnitude or more, and we expect it to be even higher for three-dimensional systems.

Note, that this threshold behavior is consistent with physical interpretation of the Reynolds number Re, which can be seen as a measure of the relative strength of nonlinear (inertial) effects compared to linear (viscous) effects in a fluid flow. One can then expect that, as Re grows, the strength of the nonlinearity grows as well, eventually bringing the system outside of the Carleman convergence radius. This seems to be contrary to what was suggested in Ref. [37], where the authors claimed that solving the lattice Boltzmann form of the Navier-Stokes equations, one reduces the nonlinearity from Re-determined to Ma-determined, where Ma is the Mach number characterizing the fluid flow. Our numerical investigations suggest that the picture is more intricate, with both Re and Ma playing their role in convergence, especially when the lattice velocity is chosen to scale with Re, as we do in our simulations. We conclude that the applicability of the Carleman approach to the simulation of highly turbulent flows with high Reynolds number may be limited due to the Carleman error not converging for these cases.

Next, in Sec. 4.2 we estimate the first complexity component $\alpha_A / \|A\|$. We find that it has a negligible contribution compared to κ_A. Nevertheless, we show that it can be well approximated by an exponential in N_C, and we find the best fitting parameters. Section 4.3 is devoted to the analysis

of the condition number κ_A. We first derive lower bounds, proving that at best one can hope for $\kappa_A = O(\sqrt{N}T^\star)$. Then, we present the results of our extensive numerical simulations to estimate the condition number for $D = 1$ and $D = 2$, and $N_C \in \{1, \ldots, 4\}$, using the Lanczos method for Carleman matrices of dimension up to $\sim 10^8$. While sizes corresponding to Carleman systems for high-impact uses-cases would be much larger, and beyond the reach of classical hardware for the foreseeable future, these are – to our knowledge – the largest numerical simulations ever done to establish the scaling of Carleman-based algorithms for fluids (and in fact, for Carleman-based algorithms more generally). We find that the condition number κ_A scales with the power χ of the Reynolds number Re, with χ growing with both the problem dimension D and the Carleman truncation order N_C. As we discuss in the further section, this negatively affects the potential for a quantum advantage. In Sec. 4.4, we then put together both query complexity cost components and derive both lower and upper bounds for the query cost q_Q in terms of the Reynolds number Re and Carleman truncation order N_C. This is followed by Sec. 4.5, where we estimate the T gate cost of a single query, proving that for a fixed N_C this just brings a constant overhead over the query cost q_Q. For low $N_C \leq 5$, this overhead is roughly 10^6 gates per query for $D = 1$, and 10^8 gates for $D = 2$.

Finally, in Sec. 5, we analyze the obtained complexity results and compare them with the complexity of classical LBE algorithms. We conclude that while modest quantum advantage could be expected for $D = 2$ (and, by extrapolation, in $D = 3$), it will be limited to low N_C, and so to low Reynolds numbers or high error outputs.

2 Problem formulation

Throughout the paper, we set the base of log to 2, and when we use the natural logarithm, we denote it by ln. We will also use the Dirac ket notation, e.g., $|\boldsymbol{Y}\rangle$ will denote the normalized quantum state encoding the corresponding vector $\boldsymbol{Y}$.

2.1 Shifted incompressible lattice Boltzmann equation

We now discuss a formulation of the problem that strives to overcome the obstacles that we have described in the previous section. Consider a new vector-valued function $\boldsymbol{g}$ that is a shifted version of LBE state vector $\bar{\boldsymbol{f}}$ from Eq. (1.9):

$$\boldsymbol{g}(\boldsymbol{r}^\star, t^\star) = \bar{\boldsymbol{f}}(\boldsymbol{r}^\star, t^\star) - \boldsymbol{w}, \tag{2.1}$$

and note that $\boldsymbol{g}$ carries information about the displacement of the fluid state from the zero velocity equilibrium state. The mass and momentum densities can be expressed using these new variables via:

$$\delta\rho(\boldsymbol{r}^\star, t^\star) = \sum_{m=1}^{Q} g_m(\boldsymbol{r}^\star, t^\star), \qquad \rho\boldsymbol{u}^\star(\boldsymbol{r}^\star, t^\star) = \sum_{m=1}^{Q} g_m(\boldsymbol{r}^\star, t^\star)\boldsymbol{e}_m, \tag{2.2}$$

where we expand ρ at every lattice site around the reference density as $\rho = 1 + \delta\rho$.

From Eqs. (1.15)-(1.16), we see that this shifted LBE state vector evolves according to the following equations:

$$\boldsymbol{g}^C(\boldsymbol{r}^\star, t^\star) = \boldsymbol{g}(\boldsymbol{r}^\star, t^\star) - \frac{1}{\bar{\tau}^\star}[\boldsymbol{g}(\boldsymbol{r}^\star, t^\star) - \boldsymbol{g}^{\mathrm{eq}}(\boldsymbol{r}^\star, t^\star)], \tag{2.3a}$$

$$g_m(\boldsymbol{r}^\star + \boldsymbol{e}_m^\star, t^\star + 1) = g_m^C(\boldsymbol{r}^\star, t^\star), \tag{2.3b}$$

with

$$g_m^{\mathrm{eq}} = \delta\rho\, w_m + \rho\left(3\boldsymbol{e}_m^\star \cdot \boldsymbol{u}^\star + \frac{9}{2}(\boldsymbol{e}_m^\star \cdot \boldsymbol{u}^\star)^2 - \frac{3}{2}|\boldsymbol{u}^\star|^2\right) w_m, \tag{2.4}$$

where we used the fact that for the considered discrete velocity models from Table 1 one has $c_s^\star = 1/\sqrt{3}$, and for clarity we omitted the dependence on $\boldsymbol{r}^\star$ and $t^\star$. Note that here we restrict our attention to a simple streaming step without walls (just periodic boundary conditions), and

without external driving. We discuss and analyze the extension to non-trivial geometries and driven flows elsewhere [57].

Next, instead of following the recent proposals of Refs. [37, 38] that exploit the weak incompressibility assumption and approximate $1/\rho \approx 2 - \rho$ to express $\boldsymbol{g}^{\mathrm{eq}}$ as a cubic function of $\boldsymbol{g}$, we focus on the incompressible formulation of LBE developed in Ref. [45]. As we explain in Appendix D, this results in replacing ρ in Eq. (2.4) by the reference density (i.e., setting ρ to 1), while retaining the density fluctuation term $\delta\rho$, and allows one to express macroscopic velocity as

$$\boldsymbol{u}^\star(\boldsymbol{r}^\star, t^\star) = \sum_{m=1}^{Q} g_m(\boldsymbol{r}^\star, t^\star)\boldsymbol{e}^\star_m. \tag{2.5}$$

As a result, the equilibrium density takes the following form:

$$g^{\mathrm{eq}}_m = \left(\delta\rho + 3\boldsymbol{e}^\star_m \cdot \boldsymbol{u}^\star + \frac{9}{2}(\boldsymbol{e}^\star_m \cdot \boldsymbol{u}^\star)^2 - \frac{3}{2}|\boldsymbol{u}^\star|^2\right) w_m, \tag{2.6}$$

which can be rewritten explicitly as a quadratic function of $\boldsymbol{g}$:

$$g^{\mathrm{eq}}_m = w_m \left(\sum_{m_1=1}^{Q} g_{m_1} + 3\sum_{m_1=1}^{Q} E_{m,m_1} g_{m_1} + \frac{9}{2}\left(\sum_{m_1=1}^{Q} E_{m,m_1} g_{m_1}\right)^2 - \frac{3}{2}\left|\sum_{m_1=1}^{Q} g_{m_1}\boldsymbol{e}^\star_{m_1}\right|^2\right), \tag{2.7}$$

$$= w_m \left(\sum_{m_1=1}^{Q} g_{m_1} + \sum_{m_1=1}^{Q} 3E_{m,m_1} g_{m_1} + \sum_{m_1=1}^{Q}\sum_{m_2=1}^{Q}\left(\frac{9}{2}E_{m,m_1}E_{m,m_2} - \frac{3}{2}E_{m_1;m_2}\right) g_{m_1} g_{m_2}\right), \tag{2.8}$$

where we introduced a Gram matrix E with matrix elements

$$E_{m,m_1} = \boldsymbol{e}^\star_m \cdot \boldsymbol{e}^\star_{m_1}. \tag{2.9}$$

Given that the D-dimensional position vector $\boldsymbol{r}^\star$ in lattice units takes only integer values, the state of the system at each time step $t^\star$ can be represented by the following vector

$$\boldsymbol{g}(t^\star) = \sum_{\boldsymbol{r}^\star}\sum_{m=1}^{Q} g_m(\boldsymbol{r}^\star, t^\star)\,|\boldsymbol{r}^\star\rangle \otimes |m\rangle\,, \tag{2.10}$$

where we explicitly introduced position and velocity registers, and the dimension of $\boldsymbol{g}$ is

$$d := NQ, \tag{2.11}$$

where

$$N := \prod_{i=1}^{D} N_i \tag{2.12}$$

is the total number of spatial lattice points. We can then rewrite the two steps of the shifted LBE, Eqs. (2.3a)-(2.3b), as

$$\boldsymbol{g}^C(t^\star) = (I + F_1)\,\boldsymbol{g}(t^\star) + F_2\boldsymbol{g}(t^\star)^{\otimes 2}, \tag{2.13a}$$

$$\boldsymbol{g}(t^\star + 1) = S\boldsymbol{g}^C(t^\star), \tag{2.13b}$$

where S is a $d \times d$ permutation matrix (so also a unitary matrix) encoding streaming:

$$S = \sum_{\boldsymbol{r}^\star}\sum_{m=1}^{Q} |\boldsymbol{r}^\star + \boldsymbol{e}^\star_m\rangle\langle\boldsymbol{r}^\star| \otimes |m\rangle\langle m|\,, \tag{2.14}$$

whereas F_k are $d \times d^k$ matrices describing linear and quadratic contributions from collisions, the form of which can be read off from Eq. (2.8) to be:

$$F_1 = \sum_{\boldsymbol{r}^\star} |\boldsymbol{r}^\star\rangle\langle\boldsymbol{r}^\star| \otimes \sum_{m,m_1=1}^{Q} \frac{1}{\bar{\tau}^\star}(w_m + 3w_m E_{m,m_1} - \delta_{m,m_1})\,|m\rangle\langle m_1|\,, \tag{2.15a}$$

$$F_2 = \sum_{\boldsymbol{r}^\star} |\boldsymbol{r}^\star\rangle\langle \boldsymbol{r}^\star, \boldsymbol{r}^\star| \otimes \sum_{m,m_1,m_2=1}^{Q} \frac{w_m}{\bar{\tau}^\star} \left(\frac{9}{2} E_{m,m_1} E_{m,m_2} - \frac{3}{2} E_{m_1,m_2} \right) |m\rangle\langle m_1, m_2| \,. \tag{2.15b}$$

Notice that the above can be rewritten as

$$F_1 = I_{\boldsymbol{r}^\star} \otimes \tilde{F}_1, \tag{2.16a}$$

$$F_2 = I_{\boldsymbol{r}^\star, \boldsymbol{r}^\star} \otimes \tilde{F}_2, \tag{2.16b}$$

where $I_{\boldsymbol{r}^\star}$ and $I_{\boldsymbol{r}^\star,\boldsymbol{r}^\star}$ are $N \times N$ and $N \times N^2$ identity matrices on the position registers, whereas $\tilde{F}_1$ and $\tilde{F}_2$ are $Q \times Q$ and $Q \times Q^2$ collision matrices acting on the velocity register.

Now, as we explain in Appendix D, the term $\delta\rho$ in Eq. (2.6) scales as $O(|\boldsymbol{u}^\star|^2)$, while the remaining terms in that equation have explicit dependence on $\boldsymbol{u}^\star$, so that we can conclude that

$$g_m^{\mathrm{eq}} = O(|\boldsymbol{u}^\star|). \tag{2.17}$$

Thus, making a simplifying assumption that during the evolution the state of the fluid at different lattice sites is close to the corresponding local equilibrium state, we have

$$g_m(\boldsymbol{r}^\star, t^\star) \approx g_m^{\mathrm{eq}}(\boldsymbol{r}^\star, t^\star) = O(|\boldsymbol{u}^\star|). \tag{2.18}$$

Then, denoting the upper bound on the magnitude of lattice velocity $|\boldsymbol{u}^\star|$ during the evolution by $u^\star_{\max}$, we get

$$\|\boldsymbol{g}\| = \sqrt{\sum_{\boldsymbol{r}^\star} \sum_{m=1}^{Q} |g_m(\boldsymbol{r}^\star, t^\star)|^2} = O\left(\sqrt{N} u^\star_{\max}\right). \tag{2.19}$$

Therefore, in order to keep the norm of the solution bounded, $\|\boldsymbol{g}\| = O(1)$, one can simply enforce the following value of lattice velocity

$$u^\star_{\max} = \frac{u^\star_0}{\sqrt{N}} \tag{2.20}$$

for some constant $u^\star_0$. This way one can control the norm of $\boldsymbol{g}$, without a direct negative impact on the condition number as in the rescaling approach (as we shall see in Sec. 4.3), thus overcoming the main obstacle for data extraction that we discussed in Sec. 1.4.4.

Finally, although we analyze the inclusion of non-trivial geometries elsewhere [57], let us briefly explain here how to include walls in our model. One can introduce an indicator function $\mathcal{W}$ encoding the wall boundary conditions in the following way:

$$\mathcal{W}_{\boldsymbol{r}^\star} = \begin{cases} 1 & : \boldsymbol{r}^\star \text{ is a wall node,} \\ 0 & : \boldsymbol{r}^\star \text{ is a fluid node,} \end{cases} \tag{2.21}$$

Then, the bounce-back effect of the walls can be accounted for by modifying the streaming matrix in the following way:

$$S = \sum_{\boldsymbol{r}^\star} \sum_{m=1}^{Q} (1 - \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}^\star_m})(1 - \mathcal{W}_{\boldsymbol{r}^\star}) \, |\boldsymbol{r}^\star + \boldsymbol{e}^\star_m\rangle\langle \boldsymbol{r}^\star| \otimes |m\rangle\langle m| \tag{2.22}$$

$$+ \sum_{\boldsymbol{r}^\star} \sum_{m=1}^{Q} \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}^\star_m} (1 - \mathcal{W}_{\boldsymbol{r}^\star}) \, |\boldsymbol{r}^\star\rangle\langle \boldsymbol{r}^\star| \otimes |-m\rangle\langle m| \tag{2.23}$$

$$+ \sum_{\boldsymbol{r}^\star} \sum_{m=1}^{Q} \mathcal{W}_{\boldsymbol{r}^\star} \, |\boldsymbol{r}^\star\rangle\langle \boldsymbol{r}^\star| \otimes |m\rangle\langle m| \,, \tag{2.24}$$

where, with a slight abuse of notation, $-m$ is a velocity index corresponding to discrete velocity $-\boldsymbol{e}^\star_m$. In the above, the first term corresponds to unmodified streaming when both nodes considered are fluid; the second term encodes bouncing, when a particle in a fluid node hits a wall node; and the third one simply means no evolution (identity matrix) inside the wall nodes. Crucially, the streaming matrix modified according to the above recipe is still a permutation matrix, so S stays unitary.

2.2 Parameter selection

We will now explain how to select simulation parameters N_x (spatial extent in lattice units), $T^\star$ (temporal duration in lattice units), and $\bar{\tau}^\star$ (relaxation time in lattice units) in order to guarantee the required scaling of the lattice velocity from Eq. (2.20) and, at the same time, to be able to simulate general dynamics of a fluid described by kinematic viscosity ν. As we shall see, our main assumptions here are as follows:

- The number of discretization points N should scale with a power of the Reynolds number, typically $\mathrm{Re}^{3D/4}$, in order to resolve the *Kolmogorov microscale* and capture the smallest dynamical features of turbulent flows.
- The simulation has to be run till a steady flow develops, which happens after some constant multiple of the advection time.

First, let us denote the upper bound on the maximal physical velocity during the evolution by $u_{\max}$ and introduce the ratio η_u of the flow velocity u to $u_{\max}$,

$$\eta_u = \frac{u}{u_{\max}}. \tag{2.25}$$

Next, assume for simplicity a cubic simulation region, so that all L_i and N_i are equal, and denote by η_L the ratio of the characteristic length L to the simulation region size L_x,

$$\eta_L := \frac{L}{L_x}. \tag{2.26}$$

Thus, the Reynolds number of the considered fluid problem can be expressed as

$$\mathrm{Re} := \frac{Lu}{\nu} = \frac{L_x u_{\max}}{\nu}\eta_L\eta_u. \tag{2.27}$$

Next, using Chapman-Enskog relation, Eq. (1.19), in the expression for the conversion between physical and lattice velocity, we get

$$u^\star_{\max} = \frac{\Delta t}{\Delta x}u_{\max} = \frac{\Delta x u_{\max}}{\nu}\frac{(\bar{\tau}^\star - \frac{1}{2})}{3} = \frac{L_x u_{\max}}{\nu}\frac{(\bar{\tau}^\star - \frac{1}{2})}{3N_x} = \frac{(\bar{\tau}^\star - \frac{1}{2})}{3N_x}\frac{\mathrm{Re}}{\eta_L\eta_u}. \tag{2.28}$$

Thus, enforcing Eq. (2.20) means

$$\bar{\tau}^\star = \frac{1}{2} + \frac{3u_0^\star}{\mathrm{Re}}\frac{\eta_L\eta_u}{N_x^{D/2-1}}. \tag{2.29}$$

Next, we recall that in order to resolve the Kolmogorov microscale, one requires $\Delta x \simeq L/\mathrm{Re}^{3/4}$. We thus choose the number of discretization points per direction to be the following function of the Reynolds number:

$$N_x = \frac{\mathrm{Re}^\beta}{\eta_L}. \tag{2.30}$$

In what follows, we will keep β general to allow the choice of a better or worse resolution, but when we perform simulations or want to compare with the performance classical algorithms, we will always choose $\beta = 3/4$, which guarantees resolving the Kolmogorov microscale.

With that, the expression for the relaxation constant becomes

$$\bar{\tau}^\star = \frac{1}{2} + \frac{3u_0^\star\eta_L^{D/2}\eta_u}{\mathrm{Re}^{\beta(D/2-1)+1}}. \tag{2.31}$$

Since Δx is fixed by the condition to resolve the Kolmogorov microscale and $\bar{\tau}^\star$ is fixed by the norm condition from Eq. (2.20), Δt is also fixed by the Chapman-Enskog relation, Eq. (1.19), to be

$$\Delta t = \frac{L}{u}\frac{u_0^\star\eta_L^{D/2}\eta_u}{\mathrm{Re}^{\beta(D/2+1)}}. \tag{2.32}$$

Finally, assuming that the steady flow will develop after $1/\eta_T$ advection times,

$$T = \frac{1}{\eta_T}\frac{L}{u}, \tag{2.33}$$

the total number of time steps to simulate the fluid under these conditions is given by

$$T^\star = \frac{T}{\Delta t} = \frac{1}{\eta_T \eta_u \eta_L^{D/2}} \frac{\mathrm{Re}^{\beta(D/2+1)}}{u_0^\star}. \tag{2.34}$$

In this work we will investigate the simplified scenario, where we put

$$\eta_T = \eta_u = \eta_L = 1, \qquad u_0^\star = 1, \tag{2.35}$$

i.e., we simulate for one advection time, the characteristic length of the problem is equal to its spatial extent, the maximal velocity is well approximated by the flow velocity, and the constant $u_0^\star$ controlling the norm $\|\boldsymbol{g}\|$ is 1. In such a case, we get the following simulation parameters:

$$N_x = \left\lceil \mathrm{Re}^{\beta} \right\rceil, \qquad T^\star = \left\lceil \mathrm{Re}^{\beta(D/2+1)} \right\rceil, \quad \bar{\tau}^\star = \frac{1}{2} + \frac{3}{\mathrm{Re}^{\beta(D/2-1)+1}}, \quad u_{\max}^\star = \frac{1}{\mathrm{Re}^{\beta D/2}}, \tag{2.36}$$

where we used the ceiling function to guarantee integer values of N_x and $T^\star$. Thus, for a fixed problem dimension D and spatial resolution parameter β (with $\beta = 3/4$ resolving the Kolmogorov microscale), the above simulation parameters become a simple function of the Reynolds number Re.

2.3 Discrete Carleman embedding

We now have a discrete-time nonlinear problem describing the fluid dynamics via Eqs. (2.13a)-(2.13b) with the parameter choice given by Eq. (2.36). This provides a well-defined problem, and we now wish to consider how it can be simulated on a quantum computer. As discussed previously, this cannot be done directly, but we must construct a linear representation of the problem that is amenable for the quantum computer. Therefore, the next step is to embed the nonlinear equations in a larger dimensional space, so that it can be well approximated by a discrete-time linear problem. To achieve this, we start by introducing infinite vectors $\boldsymbol{y}^{C,\infty}(t^\star)$ and $\boldsymbol{y}^{\infty}(t^\star)$ of size $d+d^2+d^3+\ldots$,

$$\boldsymbol{y}^{C,\infty}(t^\star) := [\boldsymbol{y}_1^C(t^\star), \boldsymbol{y}_2^C(t^\star), \ldots], \qquad \boldsymbol{y}_k^C(t^\star) = [\boldsymbol{g}^C(t^\star)]^{\otimes k}, \tag{2.37a}$$

$$\boldsymbol{y}^{\infty}(t^\star) := [\boldsymbol{y}_1(t^\star), \boldsymbol{y}_2(t^\star), \ldots], \qquad \boldsymbol{y}_k(t^\star) = [\boldsymbol{g}(t^\star)]^{\otimes k}, \tag{2.37b}$$

which carry information about all powers of the shifted LBE evolution, respectively ending with a streaming or a collision step. As this replacing of higher powers with new variables is inspired by the Carleman linear embedding procedure [47–49], we will refer to $\boldsymbol{y}^{C,\infty}(t^\star)$ and $\boldsymbol{y}^{\infty}(t^\star)$ as the *infinite Carleman vectors*. From Eq. (2.13b), we straightforwardly obtain the streaming update rule for $\boldsymbol{y}^{\infty}(t^\star)$:

$$\boldsymbol{y}^{\infty}(t^\star + 1) = \mathcal{S}^{\infty} \boldsymbol{y}^{C,\infty}(t^\star), \tag{2.38}$$

where

$$\mathcal{S}^{\infty} = \begin{pmatrix} S & 0 & 0 & \ldots \\ 0 & S^{\otimes 2} & 0 & \ldots \\ 0 & 0 & S^{\otimes 3} & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{2.39}$$

is an *infinite Carleman streaming matrix*, which is unitary.

To get the collision update rule for $\boldsymbol{y}^{C,\infty}(t^\star)$, let us start from the LBE collision equation, Eq. (2.13a), for k copies of the state vector:

$$[\boldsymbol{g}^C(t^\star)]^{\otimes k} = \left[(I + F_1)\,\boldsymbol{g}(t^\star) + F_2[\boldsymbol{g}(t^\star)]^{\otimes 2}\right]^{\otimes k}$$

$$= \sum_{k_1=0}^{k} \left[(I+F_1)^{\otimes k_1} \otimes F_2^{\otimes k-k_1} + \text{perms.} \right] [\boldsymbol{g}(t^\star)]^{\otimes 2k-k_1}$$
$$= \sum_{l=k}^{2k} \left[(I+F_1)^{\otimes 2k-l} \otimes F_2^{\otimes l-k} + \text{perms.} \right] [\boldsymbol{g}(t^\star)]^{\otimes l}$$
$$= \sum_{l=k}^{2k} C_l^k [\boldsymbol{g}(t^\star)]^{\otimes l}, \tag{2.40}$$

where we have introduced $d^k \times d^l$ matrices

$$C_l^k := (I+F_1)^{\otimes 2k-l} \otimes F_2^{\otimes l-k} + \text{perms.}, \tag{2.41}$$

and where + perms. denotes a sum over distinct permutations over k subsystems, with $2k-l$ subsystems of type 1, and $l-k$ subsystems of type 2. This means that

$$\boldsymbol{y}_k^C(t^\star) = \sum_{l=k}^{2k} C_l^k \boldsymbol{y}_l(t^\star), \tag{2.42}$$

which can be written compactly as

$$\boldsymbol{y}^{C,\infty}(t^\star) = \mathcal{C}^\infty \boldsymbol{y}^\infty(t^\star), \tag{2.43}$$

with the *infinite Carleman collision matrix*

$$\mathcal{C}^\infty = \begin{pmatrix} C_1^1 & C_2^1 & 0 & 0 & \dots & 0 \\ 0 & C_2^2 & C_3^2 & C_4^2 & \dots & 0 \\ 0 & 0 & C_3^3 & C_4^3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix}, \tag{2.44}$$

where in each row k there are $k+1$ non-zero blocks.

The nonlinear update rule for a d-dimensional vector $\boldsymbol{g}(t^\star)$ from Eqs. (2.13a)-(2.13b),

$$\boldsymbol{g}(t^\star+1) = S\left[(I+F_1)\,\boldsymbol{g}(t^\star) + F_2 \boldsymbol{g}(t^\star)^{\otimes 2} \right], \tag{2.45}$$

can thus be formally replaced by a linear update rule for an infinite-dimensional vector $\boldsymbol{y}^\infty(t^\star)$,

$$\boldsymbol{y}^\infty(t^\star+1) = \mathcal{S}^\infty \mathcal{C}^\infty \boldsymbol{y}^\infty(t^\star). \tag{2.46}$$

This infinite system can be then truncated at level N_C, so that we are dealing with finite *Carleman vectors*

$$\boldsymbol{y}^C(t^\star) := [\boldsymbol{y}_1^C(t^\star), \boldsymbol{y}_2^C(t^\star), \dots, \boldsymbol{y}_{N_C}^C(t^\star)], \tag{2.47a}$$
$$\boldsymbol{y}(t^\star) := [\boldsymbol{y}_1(t^\star), \boldsymbol{y}_2(t^\star), \dots, \boldsymbol{y}_{N_C}(t^\star)], \tag{2.47b}$$

of dimension

$$d_C := d + d^2 + \cdots + d^{N_C} = \frac{d(d^{N_C}-1)}{d-1}, \tag{2.48}$$

which carry information about all powers of the shifted incompressible LBE evolution up to the N_C-th power. The collision and streaming steps are now captured by the *Carleman collision matrix* $\mathcal{C}$ and *Carleman streaming matrix* $\mathcal{S}$, which are given by $\mathcal{C}^\infty$ and $\mathcal{S}^\infty$ restricted to the first N_C blocks. Due to the block-diagonal form of $\mathcal{S}$, coming from linearity of the original transformation, it can be implemented exactly for a finite truncation order N_C, i.e., no truncation error is introduced. The collision step, on the other hand, has $\min\{k+1, N_C-k+1\}$ non-zero blocks in each row k and, since truncation removes some of the terms as compared to the original matrix $\mathcal{C}^\infty$, a truncation error is typically expected, i.e., the evolution obtained from a truncated linear system will generally differ from the original shifted LBE dynamics. We will discuss the Carleman truncation error ϵ_C,

and in particular its dependence on the truncation order N_C, in Sec. 4.1. The update rule for the Carleman vector is thus given by

$$\boldsymbol{y}(t^\star + 1) = \mathcal{SC}\boldsymbol{y}(t^\star), \tag{2.49}$$

and the above recurrence relation has a clear solution given by

$$\boldsymbol{y}(t^\star) = (\mathcal{SC})^{t^\star}\boldsymbol{y}_{\text{ini}}, \tag{2.50}$$

where $\boldsymbol{y}_{\text{ini}}$ is the initial Carleman vector obtained from the initial condition $\boldsymbol{g}(0)$:

$$\boldsymbol{y}_{\text{ini}} = [\boldsymbol{g}(0), \boldsymbol{g}(0)^{\otimes 2}, \ldots, \boldsymbol{g}(0)^{\otimes N_C}]. \tag{2.51}$$

Thus, given the initial state $\boldsymbol{g}(0)$ of the original fluid system, its state after $T^\star$ collision and streaming steps is given by

$$\boldsymbol{g}(T^\star) \approx_{\epsilon_C} \left[(\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}\right]_1 \tag{2.52}$$

where $\approx_{\epsilon_C}$ captures the introduced truncation error.

2.4 Linear system formulation

Since the heart of our quantum algorithm for the lattice Boltzmann equation is based on a quantum linear solver, in this section we explain how to cast the Carleman state evolved under discrete-time dynamics from Eq. (2.50) as a solution to a linear system of equations. More precisely, we first construct a linear system A_H, whose solution $\boldsymbol{Y}_H$ carries information about the whole history of the evolution described by the shifted incompressible LBE (i.e., at all discrete time steps between 0 and $T^\star$). We then construct another system A_F, whose solution $\boldsymbol{Y}_F$ can be used to recover the final state of the evolution (i.e., after $T^\star$ time steps). In what follows, we will use a shorthand notation of A and $\boldsymbol{Y}$ to denote either A_H and $\boldsymbol{Y}_H$, or A_F and $\boldsymbol{Y}_F$, when the discussed results for both systems are identical.

History state. The shifted incompressible LBE problem with $T^\star$ streaming and collision steps can be written in the form of the following linear system of dimension $d_C(T^\star + 1)$:

$$\begin{pmatrix} I & 0 & 0 & \ldots & 0 & 0 \\ -\mathcal{SC} & I & 0 & \ldots & 0 & 0 \\ 0 & -\mathcal{SC} & I & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & I & 0 \\ 0 & 0 & 0 & \ldots & -\mathcal{SC} & I \end{pmatrix} \begin{pmatrix} \boldsymbol{y}(0) \\ \boldsymbol{y}(1) \\ \boldsymbol{y}(2) \\ \vdots \\ \boldsymbol{y}(T^\star - 1) \\ \boldsymbol{y}(T^\star) \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_{\text{ini}} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \tag{2.53}$$

or in a more compact form as

$$A_H\boldsymbol{Y}_H = \boldsymbol{b}. \tag{2.54}$$

The vector $\boldsymbol{Y}_H$ is the *Carleman history state*, carrying information about Carleman vector at all times during the simulated evolution. It can be verified by direct calculation that the inverse of A_H is given by

$$A_H^{-1} = \begin{pmatrix} I & 0 & 0 & \ldots & 0 & 0 \\ \mathcal{SC} & I & 0 & \ldots & 0 & 0 \\ (\mathcal{SC})^2 & \mathcal{SC} & I & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\mathcal{SC})^{T^\star-1} & (\mathcal{SC})^{T^\star-2} & (\mathcal{SC})^{T^\star-3} & \ldots & I & 0 \\ (\mathcal{SC})^{T^\star} & (\mathcal{SC})^{T^\star-1} & (\mathcal{SC})^{T^\star-2} & \ldots & \mathcal{SC} & I \end{pmatrix}, \tag{2.55}$$

which is indeed in accordance with Eq. (2.50). We thus see the solution to the set of linear equations above,

$$\boldsymbol{Y}_H = A_H^{-1}\boldsymbol{b} = \left(\boldsymbol{y}_{\text{ini}}, \mathcal{SC}\boldsymbol{y}_{\text{ini}}, (\mathcal{SC})^2\boldsymbol{y}_{\text{ini}}, \ldots, (\mathcal{SC})^{T^\star-1}\boldsymbol{y}_{\text{ini}}, (\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}, \right)^\top, \tag{2.56}$$

carries information about the whole evolution described by the shifted incompressible LBE.

Final state. For the recovery of just the final state, instead of the whole history state, we will consider an alternative linear system, inspired by Ref. [14], with additional $(2^W-1)(T^\star+1)$ idling steps after the original $T^\star+1$ evolution steps:

$$\begin{pmatrix} I & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\mathcal{SC} & I & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -\mathcal{SC} & I & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & I & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & -\mathcal{SC} & I & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & -I & I & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & -I & I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & -I & I \end{pmatrix} \begin{pmatrix} \boldsymbol{y}(0) \\ \boldsymbol{y}(1) \\ \boldsymbol{y}(2) \\ \vdots \\ \boldsymbol{y}(T^\star-1) \\ \boldsymbol{y}(T^\star) \\ \boldsymbol{y}(T^\star) \\ \boldsymbol{y}(T^\star) \\ \vdots \\ \boldsymbol{y}(T^\star) \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_{\text{ini}} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{2.57}$$

where W is an integer to be determined. Again, the above can be rewritten in a more compact form as

$$A_F \boldsymbol{Y}_F = \boldsymbol{b}, \tag{2.58}$$

whose solution is

$$\boldsymbol{Y}_F = A_F^{-1}\boldsymbol{b} = \left(\boldsymbol{y}_{\text{ini}}, \mathcal{SC}\boldsymbol{y}_{\text{ini}}, (\mathcal{SC})^2\boldsymbol{y}_{\text{ini}}, \dots, (\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}, \underbrace{(\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}, \dots, (\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}}_{(2^W-1)(T^\star+1)\text{times}} \right)^\top. \tag{2.59}$$

Clearly, the first $T^\star+1$ blocks of $\boldsymbol{Y}_F$ coincide with the history state $\boldsymbol{Y}_H$, but the remaining $(2^W-1)(T^\star+1)$ blocks contain the Carleman vector corresponding to the state of the system at the final time $T^\star$. As we will explain in Sec. 3.4.1, this will help our quantum algorithm to recover the final state more efficiently than from the history state. Also, the inverse A_F^{-1} has its first $T^\star+1$ rows the same as for A_H^{-1} (see Eq. (2.55)), after which there follow rows, where each row k is given by the last row of A_H^{-1} appended by k identities. We note that the choice of $(2^W-1)(T^\star+1)$ idling steps was made for practical implementation reasons (it can be achieved using W extra qubits) and, as we shall see, the value of W can be chosen to control the probability of recovering the final state.

3 Quantum algorithm implementation

In the previous section, we explained how to rephrase the solution of the shifted incompressible lattice Boltzmann equation as a solution $\boldsymbol{Y}$ of a set of linear equations described by either Eq. (2.54) or Eq. (2.58). In this section, we will construct a quantum algorithm that first employs a quantum linear solver to prepare the solutions $\boldsymbol{Y}$ coherently encoded in the amplitudes of a quantum state; and then post-processes them to extract classical information about the fluid.

First, in Sec. 3.1, we will describe how we encode the data in quantum registers, and then we will present the following three steps of our algorithm:

1. **Input.** In order for a quantum linear solver to output the solution to $A\boldsymbol{Y} = \boldsymbol{b}$, the matrix A an vector $\boldsymbol{b}$ must be passed to it in a way that a quantum computer can deal with. We achieve this in Sec. 3.2 by constructing a unitary block-encoding U_A of A and a unitary state preparation $U_{\boldsymbol{b}}$ of $\boldsymbol{b}$.

2. **Processing.** In Sec. 3.3, we explain how to use the quantum linear solver, with the best theoretical guarantees currently available, to bound the query complexity to the unitary block-encoding U_A and unitary state preparation $U_{\boldsymbol{b}}$ to obtain a state ϵ_Q away from the normalized solution $|\boldsymbol{Y}\rangle$.

3. **Output.** In Sec. 3.4, we describe necessary post-processing steps in the form of measurements performed on $|\boldsymbol{Y}\rangle$ that allow one to extract classical information about the fluid system.

3.1 Data encoding

We will describe our data encoding process in a few steps, starting from the smallest data structure of a single state vector $\boldsymbol{g}$ encoded as a quantum state $|\boldsymbol{g}\rangle$; then we will explain how to encode a Carleman vector $\boldsymbol{y}$ from Eq. (2.47b) into a quantum state $|\boldsymbol{y}\rangle$; and finally how to encode the full data $\boldsymbol{Y}$ including the time register (meaning $\boldsymbol{Y}_H$ in Eq. (2.54) or $\boldsymbol{Y}_F$ in Eq. (2.58)) into a quantum state $|\boldsymbol{Y}\rangle$. When writing the encodings, we will assume $D = 3$, and the smaller dimensions can be straightforwardly obtained by dropping the unnecessary registers. The total number of spatial points is $N = N_x N_y N_z$, with N_i corresponding to the number in the i'th direction. To simplify expressions, when computing qubit counts we give expressions like $\log N_x$, which should be taken to mean $\lceil \log N_x \rceil$, namely we round up to the nearest integer.

To encode the shifted LBE state vector into a quantum state $|\boldsymbol{g}\rangle$, we use three spatial registers composed of $\log N_x$, $\log N_y$, and $\log N_z$ qubits, and three velocity registers, each composed of two qubits:

$$|\boldsymbol{g}(t^\star)\rangle = \frac{1}{\sqrt{\mathcal{N}}} \sum_{r_x^\star=1}^{N_x} \sum_{r_y^\star=1}^{N_y} \sum_{r_z^\star=1}^{N_z} \sum_{v_x,v_y,v_z \in \{00,01,10\}} g_{m(v_x,v_y,v_z)}(\boldsymbol{r}^\star, t^\star) \, |r_x^\star, r_y^\star, r_z^\star\rangle \otimes |v_x, v_y, v_z\rangle \,, \tag{3.1}$$

where $\mathcal{N}$ is the normalization, $r_i^\star$ are discrete spatial positions along axis i in the binary representation, whereas (v_x, v_y, v_z) encode velocity indices according to:

$$v_i = \begin{cases} 00 & : \quad (\boldsymbol{e}_m^\star)_i = 0, \\ 10 & : \quad (\boldsymbol{e}_m^\star)_i = -1, \\ 01 & : \quad (\boldsymbol{e}_m^\star)_i = 1. \end{cases} \tag{3.2}$$

Note that this velocity encoding is almost optimal, as it requires $2D$ qubits, while the optimal one uses 2, 4, and 5 for $D \in \{1, 2, 3\}$. The total number of qubits required to encode an LBE state vector is thus $\log N + 2D$.

Next, we need to encode the Carleman state $|\boldsymbol{y}\rangle$. While the original Carleman vector $\boldsymbol{y}$ is composed of a direct sum of vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_C}$, for practical reasons, it is better to use a tensor product encoding [31]. More precisely, we add an extra Carleman block register $|n_C\rangle$ composed of $\log N_C$ qubits and pad the lower blocks with zeros:

$$|\boldsymbol{y}(t^\star)\rangle = \frac{1}{\sqrt{N_C}} \sum_{n_C=1}^{N_C} |n_C\rangle \otimes |\boldsymbol{y}_{n_C}(t^\star)\rangle \otimes |0\rangle^{\otimes N_C - n_C} \,. \tag{3.3}$$

Above, $|\boldsymbol{y}_{n_C}\rangle$ is composed of n_C single LBE state vector registers (i.e., $n_C(\log N + 2D)$ qubits), and a single $|0\rangle$ is composed of a single LBE state vector register, i.e., of $\log N + 2D$ qubits. Since $\boldsymbol{y}_{n_C}$ lives in a vector space consisting of n_C copies of the space for the LBE state vector, we will denote its single state components by $\boldsymbol{y}^{[l]}$ for $l \in \{1, \ldots, n_C\}$.

Using the proposed tensor product encoding, the structure of the Carleman matrices changes, so that all their Carleman blocks become square matrices with a fixed dimension. More precisely, the action of the Carleman collision matrix can be extended to

$$\mathcal{C} = \sum_{k=1}^{N_C} \sum_{l=k}^{\min\{2k, N_C\}} |k\rangle\langle l| \otimes C_l^k \otimes |0\rangle^{\otimes l-k} \otimes I^{\otimes N_C - l} =: \sum_{k=1}^{N_C} \sum_{l=k}^{\min\{2k, N_C\}} |k\rangle\langle l| \otimes \bar{C}_l^k \otimes I^{\otimes N_C - l}, \tag{3.4}$$

so that every rectangular matrix C_l^k becomes a square matrix $\bar{C}_l^k$ by adding in the output $l-k$ zero states, and then we extend it to a square matrix with a fixed dimension for all blocks by padding it with additional $N_C - l$ identity matrices. Similarly, the Carleman streaming matrix is extended to

$$\mathcal{S} = \sum_{k=1}^{N_C} |k\rangle\langle k| \otimes S^{\otimes k} \otimes I^{\otimes N_C - k}. \tag{3.5}$$

Finally, we need to encode states $|\boldsymbol{Y}_H\rangle$ and $|\boldsymbol{Y}_F\rangle$. We achieve this by adding an extra time register $t^\star$ composed of $\log(T^\star + 1)$ qubits and, in the case of $|\boldsymbol{Y}_F\rangle$, an extra waiting register w

composed of W qubits:

$$|\boldsymbol{Y}_H\rangle = \frac{1}{\sqrt{T^\star+1}} \sum_{t^\star=0}^{T^\star} |t^\star\rangle \otimes |\boldsymbol{y}(t^\star)\rangle\,, \tag{3.6a}$$

$$|\boldsymbol{Y}_F\rangle = \frac{1}{\sqrt{2^W(T^\star+1)}} \left(\sum_{t^\star=0}^{T^\star} |0\rangle \otimes |t^\star\rangle \otimes |\boldsymbol{y}(t^\star)\rangle + \sum_{w=1}^{2^W-1} \sum_{t^\star=0}^{T^\star} |w\rangle \otimes |t^\star\rangle \otimes |\boldsymbol{y}(T^\star)\rangle \right). \tag{3.6b}$$

Summarizing, the full data register is composed of the following:

- A single time register $t^\star$ consisting of $\log(T^\star+1)$ qubits.
- A single waiting register w consisting of W qubits (absent in the case of history state).
- A single Carleman block register n_C consisting of $\log N_C$ qubits.
- N_C single state vector registers $\boldsymbol{y}^{[k]}$, each composed of D spatial registers $r_i^\star$ (consisting of $\log N_i$ qubits each) and D velocity registers v_i (consisting of 2 qubits each).

Adding this all together, we get that the number n_D of qubits in the data register is given by

$$n_D = \log(T^\star+1) + W + \log N_C + N_C(\log N + 2D). \tag{3.7}$$

Using the simulation parameter choice from Eq. (2.36), with $N_x = N_y = N_z$, we can represent it as a function of the Reynolds number Re:

$$n_D = \beta \left[D\left(N_C + \frac{1}{2}\right) + 1 \right] \log \mathrm{Re} + 2DN_C + \log N_C + W, \tag{3.8}$$

and so the size of the data register scales efficiently with the Reynolds number Re and the Carleman truncation order N_C. For example, for a $D = 3$ system with Reynolds number $\mathrm{Re} = 10^8$, parameter $\beta = 3/4$ to resolve the Kolmogorov microscale, $W = 10$ to extract the final state with high probability, and a relatively high truncation order $N_C = 10$, one needs 722 qubits.

3.2 Unitary block-encodings and state preparations

The quantum algorithm involves a range of non-unitary matrices that define the problem instance, as well as state preparation data. Quantum computers are primarily concerned with quantum circuits that generate unitary operations. We must therefore encode the matrices defining the problem within a unitary context that the quantum computer can process. This is achieved through the concept of a *unitary block-encoding* of a matrix A. In what follows, we give the details of these encodings, as well as key results that are central to the algorithm construction.

3.2.1 Preliminaries

Recall that a unitary state preparation of a vector $\boldsymbol{b}$ is any unitary $U_{\boldsymbol{b}}$ such that

$$U_{\boldsymbol{b}}\,|0\rangle = |\boldsymbol{b}\rangle := \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|}, \tag{3.9}$$

while a block-encoding of a rectangular matrix A, mapping m_{in} into $m_{\mathrm{out}} \leq m_{\mathrm{in}}$ qubits, is defined as any unitary U_A of the form

$$U_A = \begin{bmatrix} A/\alpha_A & * \\ * & * \end{bmatrix}. \tag{3.10}$$

In the above, $*$ indicate other non-essential components of the unitary matrix U_A, and α_A is implicitly defined by the equation and is the block-encoding rescaling prefactor. More formally, an $(\alpha_A, n_A, \epsilon)$-block-encoding of A is an $(m_{\mathrm{in}} + n_A)$-qubit unitary U_A such that

$$\left\| A - \alpha_A (I_{m_{\mathrm{out}}} \otimes \langle 0|^{\otimes n_A + m_{\mathrm{in}} - m_{\mathrm{out}}}) U_A (I_{m_{\mathrm{in}}} \otimes |0\rangle^{\otimes n_A}) \right\| \leq \epsilon. \tag{3.11}$$

The parameter α_A typically enters the quantum algorithmic cost, so we ideally devise block-encodings with α_A as small as possible. Note that $\alpha_A \geq \|A\|$, since U_A is unitary, so an *optimal* block-encoding has $\alpha_A = \|A\|$.

We now note the following, where we assume $\epsilon = 0$ for simplicity. Suppose we have a rectangular matrix A and a block-encoding U_A as defined above. This implies that for any $|\psi\rangle$ we have

$$U_A |\psi\rangle \otimes |0\rangle^{\otimes n_A} = \frac{1}{\alpha_A} A |\psi\rangle \otimes |0\rangle^{\otimes (n_A + m_{\text{in}} - m_{\text{out}})} + |\perp\rangle , \tag{3.12}$$

where $|\perp\rangle$ is a vector orthogonal to the first term on the right hand side. Given this, for any rectangular matrix A we can define the *square* matrix $\bar{A}$ in the embedding tensor product space as

$$\bar{A} |\psi\rangle := A |\psi\rangle \otimes |0\rangle^{\otimes (m_{\text{in}} - m_{\text{out}})} . \tag{3.13}$$

It follows that if U_A is a unitary block-encoding of A with n_A incoming ancillary zero states, $n_A + m_{\text{in}} - m_{\text{out}}$ outgoing ancillary zero states, and a prefactor α_A, then U_A is also a unitary block-encoding of $\bar{A}$ with n_A ancillary qubits beginning and ending in the zero state, and with a rescaling prefactor $\alpha_{\bar{A}} = \alpha_A$. In what follows, we will use this fact in the context of the tensor-product embedding of the Carleman matrix.

In the following sections, we construct a unitary block-encoding U_{A_H} of A_H from Eq. (2.54) and a block-encoding U_{A_F} of A_F from Eq. (2.58), which both consist of a series of nested block-encodings:

1. First, we will explain how to construct U_{A_H} and U_{A_F} given the block-encoding $U_{\mathcal{C}}$ of $\mathcal{C}$ (Sec. 3.2.2).
2. Next, we show how to construct $U_{\mathcal{C}}$ given the block-encodings $U_{\bar{C}^k_{k+l}}$ of $\bar{C}^k_{k+l}$ (Sec. 3.2.3).
3. Then, we construct the block-encoding $U_{\bar{C}^k_{k+l}}$ of $\bar{C}^k_{k+l}$ given block-encodings of collision matrices $I + F_1$ and $\bar{F}_2$ (Sec. 3.2.3).
4. Finally, we present a way to block-encode collision matrices $I + F_1$ and $\bar{F}_2$ (Sec. 3.2.4).

This way we close the construction and provide the final expressions for the block-encoding prefactors α_{A_H} and α_{A_F}, and the corresponding number of ancillary qubits needed, n_{A_H} and n_{A_F}. Finally, we explain the construction of a unitary state preparation of $\boldsymbol{b}$.

Before we proceed, we present two crucial results concerning unitary block-encodings that will be useful in block-encoding the investigated linear systems describing the shifted LBE. Although these are well known results, we provide their proofs, as we will use the construction presented there in the coming sections.

Lemma 1 (LCU Block-encodings)**.** *Given an N-qubit matrix*

$$A = \sum_{i=1}^{k} A_i, \tag{3.14}$$

and $(\alpha_i, n_i, 0)$-block-encodings U_{A_i} of A_i, one can construct $(\alpha, n, 0)$-block-encoding U_A of A with

$$\alpha = \sum_{i=1}^{k} \alpha_i, \qquad n = \log k + \max_i n_i. \tag{3.15}$$

Proof. Introduce a state preparation unitary V on $\log k$ qubits, such that

$$V |0\rangle = \frac{1}{\sqrt{\alpha}} \sum_{i=1}^{k} \sqrt{\alpha_i} |i\rangle , \tag{3.16}$$

and denote $n^* = \max_i n_i$. Then, extend $(\alpha_i, n_i, 0)$-block-encodings of A_i to $(\alpha_i, n^*, 0)$-block-encodings $U^*_{A_i}$ of A_i simply by adding extra qubits on which $U^*_{A_i}$ acts trivially. Now, construct the following unitary

$$U_A = (I_N \otimes I_{n^*} \otimes V^\dagger) \left(\sum_{i=1}^{k} U^*_{A_i} \otimes |i\rangle\langle i| \right) (I_N \otimes I_{n^*} \otimes V), \tag{3.17}$$

where I_m is the identity matrix acting on m qubits. We can directly verify that

$$(I_N \otimes \langle 0|^{\otimes n^* + \log k}) U_A (I_N \otimes |0\rangle^{\otimes n^* + \log k}) \tag{3.18}$$

$$= \frac{1}{\alpha} \sum_{j,j'=1}^{k} \sqrt{\alpha_j \alpha_{j'}} (I_N \otimes \langle 0|^{\otimes n^*} \otimes \langle j'|) \left(\sum_{i=1}^{k} U^*_{A_i} \otimes |i\rangle\langle i| \right) (I_N \otimes |0\rangle^{\otimes n^*} \otimes |j\rangle) \tag{3.19}$$

$$= \frac{1}{\alpha} \sum_{i=1}^{k} \alpha_i (I_N \otimes \langle 0|^{\otimes n^*}) U^*_{A_i} (I_N \otimes |0\rangle^{\otimes n^*}) = \frac{1}{\alpha} \sum_{i=1}^{k} A_i = \frac{A}{\alpha}, \tag{3.20}$$

so that U_A is an $(\alpha, n, 0)$-block-encoding of A. □

Lemma 2 (Block-diagonal LCU)**.** *Given a $k \cdot 2^N \times k \cdot 2^N$ matrix*

$$A = \sum_{i=k_1}^{k_2} |i\rangle\langle i| \otimes A_i, \tag{3.21}$$

with $1 \leq k_1 \leq k_2 \leq k$, and $(\alpha_i, n_i, 0)$-block-encodings U_{A_i} of N-qubit matrices A_i, one can construct $(\alpha, n, 0)$-block-encoding U_A of A with

$$\alpha = \max_{i \in \{k_1, \ldots, k_2\}} \alpha_i, \qquad n = \max_{i \in \{k_1, \ldots, k_2\}} n_i + 1. \tag{3.22}$$

Proof. For $i < k_1$ and $i > k_2$ set $\alpha_i = 0$. Then, for all $i \in \{1, \ldots, k\}$, introduce a set of single-qubit state preparation unitaries V_i, such that

$$V_i |0\rangle = \frac{\alpha_i}{\alpha} |0\rangle + \sqrt{1 - \frac{\alpha_i^2}{\alpha^2}} |1\rangle, \tag{3.23}$$

and denote $n^* = \max_i n_i$. Next, for $i \in \{k_1, \ldots, k_2\}$, extend $(\alpha_i, n_i, 0)$-block-encodings U_{A_i} of A_i to $(\alpha_i, n^*, 0)$-block-encodings $U^*_{A_i}$ of A_i simply by adding extra qubits on which $U^*_{A_i}$ acts trivially. Moreover, for $i < k_1$ and $i > k_2$, define $U^*_{A_i} = I_N \otimes I_{n^*}$. Now, construct the following unitary

$$U_A = (I_{\log k} \otimes I_N \otimes I_{n^*} \otimes I_1) \left(\sum_{i=1}^{k} |i\rangle\langle i| \otimes U^*_{A_i} \otimes I_1 \right) \left(\sum_{j=1}^{k} |j\rangle\langle j| \otimes I_N \otimes I_{n^*} \otimes V_j \right), \tag{3.24}$$

where I_m is the identity matrix acting on m qubits. We can directly verify that

$$(I_{\log k} \otimes I_N \otimes \langle 0|^{\otimes n^* + 1}) U_A (I_{\log k} \otimes I_N \otimes |0\rangle^{\otimes n^* + 1}) \tag{3.25}$$

$$= (I_{\log k} \otimes I_N \otimes \langle 0|^{\otimes n^*}) \left(\sum_{i=1}^{k} |i\rangle\langle i| \otimes U^*_{A_i} \langle 0| V_i |0\rangle \right) (I_{\log k} \otimes I_N \otimes |0\rangle^{\otimes n^*}) \tag{3.26}$$

$$= (I_{\log k} \otimes I_N \otimes \langle 0|^{\otimes n^*}) \left(\sum_{i=k_1}^{k_2} |i\rangle\langle i| \otimes U^*_{A_i} \frac{\alpha_i}{\alpha} \right) (I_{\log k} \otimes I_N \otimes |0\rangle^{\otimes n^*}) \tag{3.27}$$

$$= \sum_{i=k_1}^{k_2} |i\rangle\langle i| \otimes \frac{A_i}{\alpha_i} \frac{\alpha_i}{\alpha} = \frac{1}{\alpha} \sum_{i=1}^{k} |i\rangle\langle i| \otimes A_i = \frac{A}{\alpha}, \tag{3.28}$$

so that U_A is an $(\alpha, n, 0)$-block-encoding of A. □

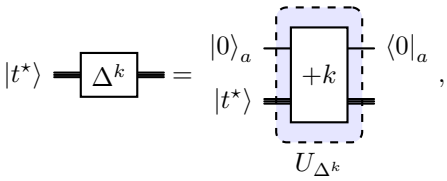
3.2.2 Block-encoding the linear system

History state. First of all, note that we can decompose A_H into a sum over the diagonal and off-diagonal blocks:

$$A_H = \sum_{t^\star=0}^{T^\star} |t^\star\rangle\langle t^\star| \otimes I - \sum_{t^\star=0}^{T^\star-1} |t^\star+1\rangle\langle t^\star| \otimes \mathcal{SC} = I_{t^\star} \otimes I - \Delta_{t^\star} \otimes \mathcal{SC}, \tag{3.29}$$

where $\Delta_t^\star$ is just a $(T^\star+1)$-dimensional shift operator on the first (time) register. Recall that in this section we want to show how to block-encode A_H given a block-encoding of $\mathcal{C}$. So, we need to show how to block-encode $\Delta_{t^\star}$ and how to realize $\mathcal{S}$.

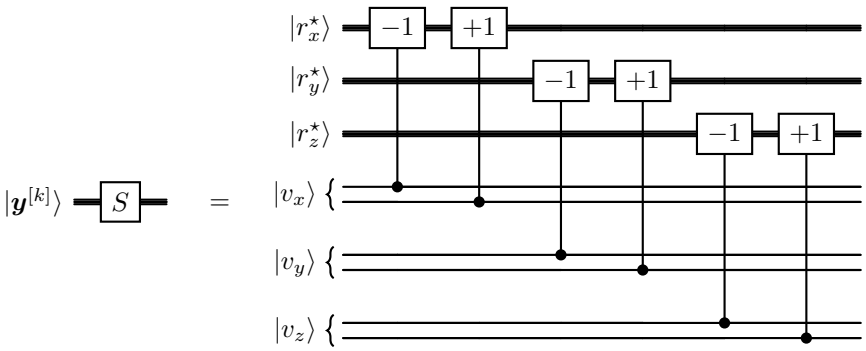
It is easy to construct a $(1,1,0)$-block-encoding U_{Δ^k} of Δ^k (here we just need $k=1$) in the following way:



(3.30)

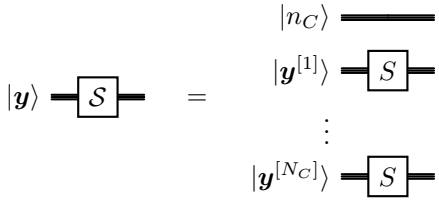
where the $+k$ gate is a quantum adder adding k to a register consisting of $\log(T^\star+1)+1$ qubits. Note that the extra qubit, initialized in $|0\rangle_a$, plays the role of the most significant bit, and the notation $\langle 0|_a$ in Eq. 3.30 indicates that Δ^k is encoded in the $|0\rangle_a$ subspace, in the sense of Eq. (3.11) with $n_A = 1$ and $m_{\text{in}} = m_{\text{out}}$. The block-encoding of $(\Delta^\dagger)^k$ is analogous, just using a $-k$ quantum adder.

Next, since $\mathcal{S}$ is unitary, it has a trivial block-encoding and can be implemented as a quantum circuit in the following way. Note that the action of S on a single state vector register $\boldsymbol{y}^{[k]}$ can be constructed using quantum adders on the corresponding spatial registers controlled by the velocity registers:



(3.31)

Then, the direct way to construct $\mathcal{S}$ is to consecutively apply S to the k-th single state vector register, controlled on the Carleman block register n_C being smaller or equal to k. However, given that we encode the data in a way that the single state vector registers $\boldsymbol{y}^{[k]}$ are all set to zero states when $k > n_C$, and that S acts on these simply as the identity matrix, we can as well remove the controls and encode $\mathcal{S}$ by just applying S to each single state vector register,



(3.32)

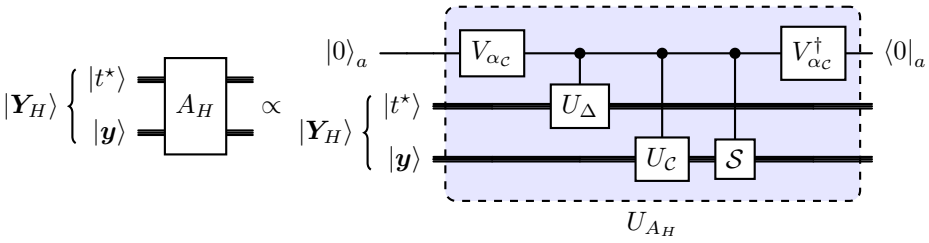
We shall follow this cheaper option.

Finally, let us assume that we have access to $(\alpha_{\mathcal{C}}, n_{\mathcal{C}}, 0)$-block-encoding $U_{\mathcal{C}}$ of $\mathcal{C}$, so that we can construct an $(\alpha_{\mathcal{C}}, n_{\mathcal{C}}+1, 0)$-block-encoding of the second term in Eq. (3.29) using the constructions

presented above. Then, using Lemma 1, we can combine this with the first identity term to construct an $(\alpha_{A_H}, n_{A_H}, 0)$-block-encoding U_{A_H} of A_H with

$$\alpha_{A_H} = \alpha_{\mathcal{C}} + 1, \quad n_{A_H} = n_{\mathcal{C}} + 2. \tag{3.33}$$

The explicit circuit for this is given by:



(3.34)

where V_α is a single qubit gate defined via:

$$V_\alpha |0\rangle \propto |0\rangle + \sqrt{\alpha}\,|1\rangle\,, \tag{3.35}$$

and, with a slight abuse of notation that we will use throughout this section, the block-encoded operators U_Δ and $U_{\mathcal{C}}$ also act on ancillary qubits that are not shown in the circuit.

Final state. Similarly to the case of A_H, we can decompose A_F as follows:

$$A_F = I_w \otimes I_{t^\star} \otimes I - [|0\rangle\langle 0|_w \otimes I_{t^\star} \otimes \mathcal{SC} + (I_w - |0\rangle\langle 0|_w) \otimes I_{t^\star} \otimes I]\,[\Delta_{w,t^\star} \otimes I], \tag{3.36}$$

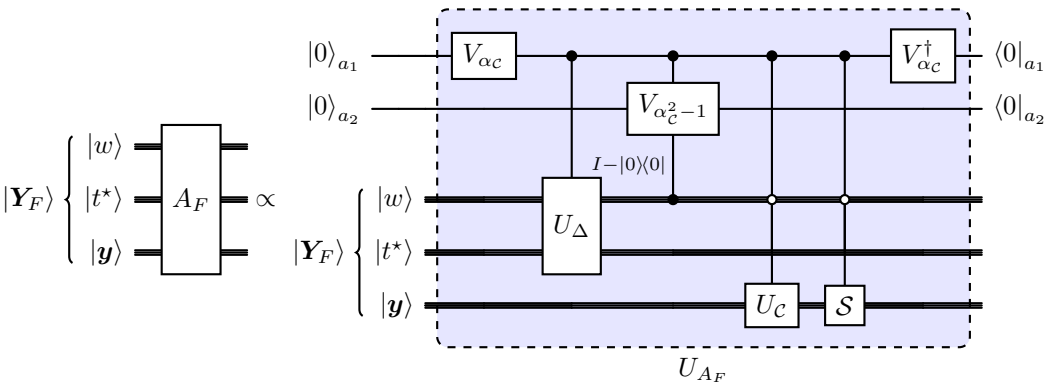
where $\Delta_{w,t^\star}$ is the shift operator on the waiting and time registers acting as:

$$|0\rangle_w \otimes |0\rangle_{t^\star} \to |0\rangle_w \otimes |1\rangle_{t^\star} \to \cdots \to |0\rangle_w \otimes |T^\star\rangle_{t^\star} \to |1\rangle_w \otimes |0\rangle_{t^\star} \to \cdots \to |2^W - 1\rangle_w \otimes |T^\star\rangle_{t^\star}\,. \tag{3.37}$$

Assuming we have access to $(\alpha_{\mathcal{C}}, n_{\mathcal{C}}, 0)$-block-encoding of $\mathcal{C}$ and to $(1, 1, 0)$-block-encoding of Δ, we can use Lemma 2 to construct an $(\alpha_{\mathcal{C}}, n_{\mathcal{C}} + 2, 0)$-block-encoding of the second term in Eq. (3.36). Thus, using Lemma 1 as before, we end up with an $(\alpha_{A_F}, n_{A_F}, 0)$-block-encoding of A_F with

$$\alpha_{A_F} = \alpha_{\mathcal{C}} + 1, \quad n_{A_F} = n_{\mathcal{C}} + 3, \tag{3.38}$$

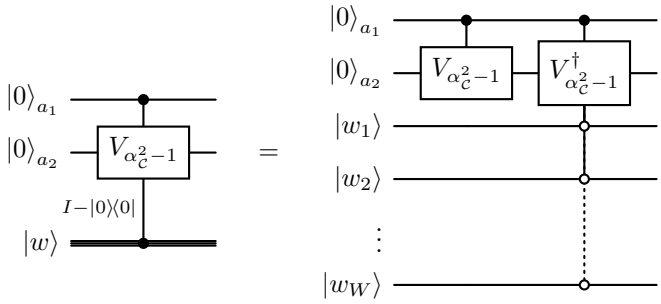
i.e., with the same block-encoding prefactor as in the case of A_H, but with one more ancillary qubit needed. The explicit circuit for U_{A_F} is given by:



(3.39)

where both single qubit V_α unitaries are defined by Eq. (3.35), whereas $I - |0\rangle\langle 0|$ denotes a control that works unless all qubits are in a zero state. Note that this controlled gate can be implemented

via



(3.40)

Summary. In summary, block-encodings of A_H and A_F can be realized with a single call to a block-encoding of $\mathcal{C}$ and with block-encoding prefactors and auxiliary qubits equal to

$$\alpha_{A_F} = \alpha_{A_H} := \alpha_{\mathcal{C}} + 1, \quad n_{A_H} \leq n_{A_F} = n_{\mathcal{C}} + 3, \tag{3.41}$$

where $n_{\mathcal{C}}$ is the number of qubits required to block-encode $\mathcal{C}$. Thus, we will simply denote α_{A_H} and α_{A_F} as α_A, and we will use $n_A = n_{A_F}$ as the ancillary qubit cost for both systems.

3.2.3 Block-encoding the Carleman collision matrix

First of all, note that we can decompose $\mathcal{C}$ from Eq. (3.4) into a sum over the diagonal and off-diagonal blocks:

$$\mathcal{C} = \sum_{k=1}^{N_C} \sum_{l=k}^{\min\{2k,N_C\}} |k\rangle\langle l| \otimes \bar{C}_l^k \otimes I^{\otimes N_C - l} \tag{3.42}$$

$$= \sum_{k=1}^{N_C} |k\rangle\langle k| \otimes \bar{C}_k^k \otimes I^{\otimes N_C - k} + \sum_{k=1}^{N_C} \sum_{l=1}^{\min\{k,N_C-k\}} |k\rangle\langle k+l| \otimes \bar{C}_{k+l}^k \otimes I^{\otimes N_C - k - l} \tag{3.43}$$

$$= \sum_{k=1}^{N_C} |k\rangle\langle k| \otimes \bar{C}_k^k \otimes I^{\otimes N_C - k} + \sum_{l=1}^{\lfloor \frac{N_C}{2} \rfloor} \sum_{k=l}^{N_C - l} |k\rangle\langle k| \, (\Delta^\dagger)^l \otimes \bar{C}_{k+l}^k \otimes I^{\otimes N_C - k - l}. \tag{3.44}$$

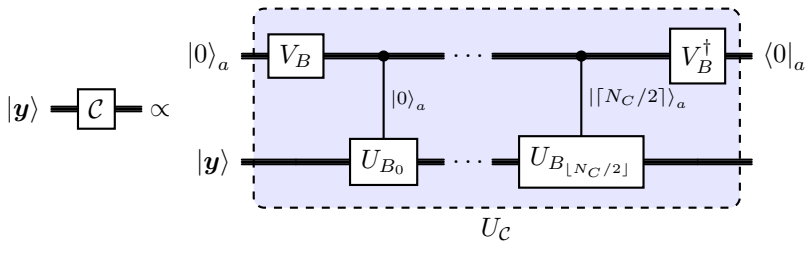
Introducing $l' := \max\{1, l\}$, we rewrite $\mathcal{C}$ as

$$\mathcal{C} = \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \sum_{k=l'}^{N_C - l} |k\rangle\langle k| \, (\Delta^\dagger)^l \otimes \bar{C}_{k+l}^k \otimes I^{\otimes N_C - k - l} =: \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} B_l. \tag{3.45}$$

Thus, assuming we found $(\alpha_{B_l}, n_{B_l}, 0)$-block-encodings U_{B_l} for each B_l, we can use Lemma 1 to construct $(\alpha_{\mathcal{C}}, n_{\mathcal{C}}, 0)$-block-encoding $U_{\mathcal{C}}$ of $\mathcal{C}$ with

$$\alpha_{\mathcal{C}} = \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \alpha_{B_l}, \qquad n_{\mathcal{C}} = \log(\lfloor N_C/2 \rfloor + 1) + \max_{l \in \{0, \ldots, \lfloor N_C/2 \rfloor\}} n_{B_l}. \tag{3.46}$$

One can then explicitly construct $U_{\mathcal{C}}$ via a standard LCU circuit:



, (3.47)

where each U_{B_l} is controlled on the ancillary register, consisting of $\log(\lceil N_C/2 \rceil + 1)$ qubits, being in the state $|l\rangle$, and V_B is defined via

$$V_B |0\rangle \propto \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \sqrt{\alpha_{B_l}} |l\rangle . \tag{3.48}$$

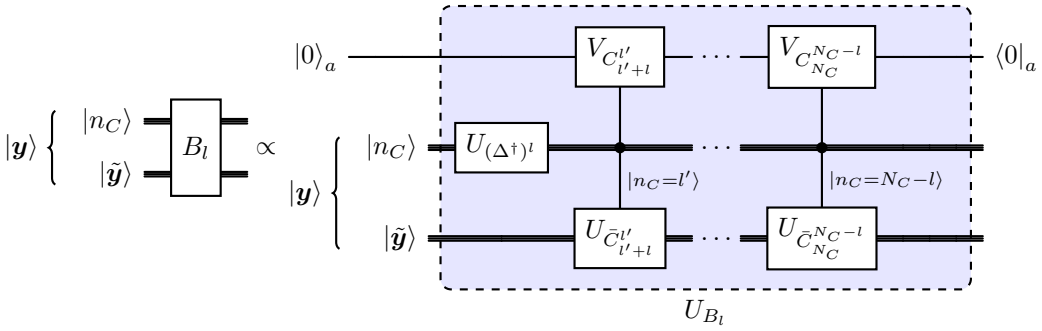
In order to block-encode each B_l,

$$B_l = \left(\sum_{k=l'}^{N_C - l} |k\rangle\langle k| \otimes \bar{C}^k_{k+l} \otimes I^{\otimes N_C - k - l} \right) [(\Delta^\dagger)^l \otimes I^{\otimes N_C}], \tag{3.49}$$

we can block-encode the two terms separately. We have already shown in Sec. 3.2.2 how to construct a $(1, 1, 0)$-block-encodings of the second term. From Lemma 2, we know that we can construct a $(\max_k \alpha_{\bar{C}^k_{k+l}}, \max_k n_{\bar{C}^k_{k+l}} + 1, 0)$-block-encoding of the first term, given a $(\alpha_{\bar{C}^k_{k+l}}, n_{C^k_{k+l}}, 0)$-block-encodings of $\bar{C}^k_{k+l}$. We thus conclude that we can construct $(\alpha_{B_l}, n_{B_l}, 0)$-block-encodings of each B_l with

$$\alpha_{B_l} = \max_{k \in \{l', \ldots, N_C - l\}} \alpha_{\bar{C}^k_{k+l}}, \qquad n_{B_l} = \max_{k \in \{l', \ldots, N_C - l\}} n_{\bar{C}^k_{k+l}} + 1. \tag{3.50}$$

The explicit circuit for each U_{B_l} is given by



(3.51)

where $\tilde{\boldsymbol{y}}$ denotes all the registers of $\boldsymbol{y}$ except for the Carleman block register (i.e., single state vector registers $\boldsymbol{y}^{[1]}$ through $\boldsymbol{y}^{[N_C]}$), and $V_{C^k_{k+l}}$ are single qubit unitaries defined by

$$V_{C^k_{k+l}} |0\rangle \propto |0\rangle + \sqrt{\frac{\alpha^2_{B_l}}{\alpha^2_{\bar{C}^k_{k+l}}} - 1} |1\rangle . \tag{3.52}$$

Next, we introduce

$$F(k, l) := (I + F_1)^{\otimes k - l} \otimes \bar{F}_2^{\otimes l}, \tag{3.53}$$

where $\bar{F}_2$ acts on two single state vector registers as $F_2 \otimes |0\rangle$, analogously to how $\bar{C}^k_l$ depends on C^k_l. We can then rewrite $\bar{C}^k_{k+l}$ as

$$\bar{C}^k_{k+l} = \left(\sum_\pi \Pi_\pi F(k, l) \Pi^\dagger_\pi \right), \tag{3.54}$$

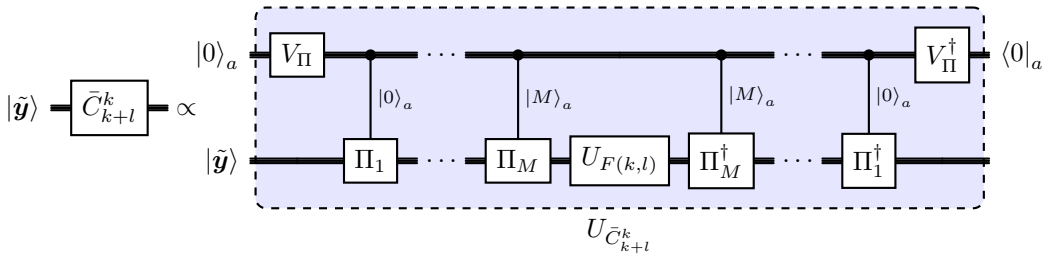
where the summation is over all permutations π of a total of k systems with $k - l$ systems of type 1 (each consisting of 1 subsystem) and l systems of type 2 (each consisting of 2 subsystems). Moreover, Π_π is a unitary permutation over all $k + l$ subsystems that:

1. Acts on the $k - l$ systems of type 1, and on all the first subsystems of the l systems of type 2, as π;
2. Then, each second subsystem of the m-th system of type 2 is shifted forward by to a position $k + m$.

To block-encode $\bar{C}^k_{k+l}$ given block-encodings $U_{F(k,l)}$ of $F(k,l)$ we will use the LCU construction from Lemma 1 again. Importantly, note that given a $(\alpha_{F(k,l)}, n_{F(k,l)}, 0)$-block-encoding of $F(k,l)$, all its permuted versions appearing in the expression for $\bar{C}^k_{k+l}$ have block-encodings with the same prefactors and numbers of ancillary qubits. Thus, assuming we can construct $(\alpha_{F(k,l)}, n_{F(k,l)}, 0)$-block-encodings $U_{F(k,l)}$ of each $F(k,l)$, we can use Lemma 1 to construct $(\alpha_{\bar{C}^k_{k+l}}, n_{\bar{C}^k_{k+l}}, 0)$-block-encodings $U_{\bar{C}^k_{k+l}}$ of $\bar{C}^k_{k+l}$ with

$$\alpha_{\bar{C}^k_{k+l}} = \binom{k}{l}\alpha_{F(k,l)}, \qquad n_{\bar{C}^k_{k+l}} = \log\binom{k}{l} + n_{F(k,l)}. \tag{3.55}$$

The straightforward circuit decomposition of $U_{\bar{C}^k_{k+l}}$ is given by



(3.56)

where we introduced a shorthand notation $M = \binom{k}{l}$, Π_k means Π_{π_k} with π_k being the k-th permutation using some standard ordering, and V_Π prepares a uniform superposition over M states. Note, however, that in this construction the number of $(\log M)$-controlled gates is M, which can grow almost exponentially with N_C, rendering it an inefficient encoding for large N_C. However, in the sub-threshold regime where the Carleman method converges, we expect that N_C will grow only logarithmically with the error, and so will result in an efficient encoding (as we shall see, the cost of these gates is subleading). However, we point out that one could encode the action of all permutations in the circuit above efficiently with growing N_C using the quantum Fisher-Yates shuffle [58].

Finally, assume we have a $(\alpha_{I+F_1}, n_{I+F_1}, 0)$-block-encoding of $(I+F_1)$ and a $(\alpha_{\bar{F}_2}, n_{\bar{F}_2}, 0)$-block-encoding of $\bar{F}_2$. Then, by simply applying these to subsequent single state vector registers, we can construct $(\alpha_{F(k,l)}, n_{F(k,l)}, 0)$-block-encoding of $F(k,l)$ with

$$\alpha_{F(k,l)} = \alpha^{k-l}_{I+F_1}\alpha^l_{\bar{F}_2}, \tag{3.57a}$$

$$n_{F(k,l)} = (k-l)n_{I+F_1} + l n_{\bar{F}_2}. \tag{3.57b}$$

Putting it all together, we get $(\alpha_{\mathcal{C}}, n_{\mathcal{C}}, 0)$-block-encoding of $\mathcal{C}$ with

$$\alpha_{\mathcal{C}} = \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \max_{k\in\{l',\ldots,N_C-l\}} \binom{k}{l}\alpha^{k-l}_{I+F_1}\alpha^l_{\bar{F}_2} = \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \binom{N_C-l}{l}\alpha^{N_C-2l}_{I+F_1}\alpha^l_{\bar{F}_2}, \tag{3.58a}$$

$$\begin{aligned} n_{\mathcal{C}} &= 1 + \log\left(\left\lfloor \frac{N_C}{2} \right\rfloor + 1\right) + \max_{l\in\{0,\ldots,\lfloor N_C/2\rfloor\}} \max_{k\in\{l',\ldots,N_C-l\}} \left(\log\binom{k}{l} + (k-l)n_{I+F_1} + l n_{\bar{F}_2}\right) \\ &= 1 + \log\left(\left\lfloor \frac{N_C}{2} \right\rfloor + 1\right) + \max_{l\in\{0,\ldots,\lfloor N_C/2\rfloor\}} \left(\log\binom{N_C-l}{l} + (N_C-2l)n_{I+F_1} + l n_{\bar{F}_2}\right). \end{aligned} \tag{3.58b}$$

3.2.4 Block-encoding collision matrices

We now proceed to the final step of block-encoding the collision matrices $I+F_1$ and $\bar{F}_2$. Starting from $I+F_1$, we recall Eq. (2.16a) and note that this matrix is local in space. Therefore, it acts non-trivially only on the velocity register,

$$I + F_1 = I_{\boldsymbol{r}^\star} \otimes (I + \tilde{F}_1), \tag{3.59}$$

where $\tilde{F}_1$ is of size $2^{2D} \times 2^{2D}$. In fact, since out of 4 basis states in each velocity register we only use 3, for $I+\tilde{F}_1$ it is enough to block-encode a $Q \times Q$ matrix (with $Q = 3^D$ denoting the number of

discrete velocities in the considered LBE model), as the rest of the unitary can be set to identity. Our problem then reduces to block-encoding a $Q \times Q$ matrix $I + \tilde{F}_1$.

To achieve this, we will use block-encoding based on singular value decomposition (SVD). More precisely, we can always write

$$(I + \tilde{F}_1) = L_1 \Sigma_1 R_1^\dagger, \tag{3.60}$$

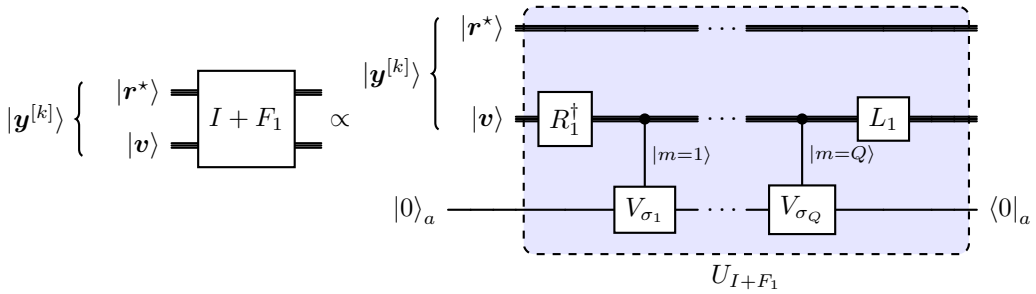
where L_1 and R_1 are $Q \times Q$ unitary matrices, while Σ_1 is a $Q \times Q$ diagonal matrix (in fact, all these matrices have size $2^{2D} \times 2^{2D}$, but only the $Q \times Q$ block is nontrivial). Then, assuming access to the SVD unitaries, which act only on $2D$ qubits and whose explicit construction we postpone to Sec. 4.5.4, we only need to block-encode the diagonal matrix Σ_1. Thus, using Lemma 2, we get $(\alpha_{I+F_1}, 1, 0)$-block-encoding of $(I + F_1)$ with the block-encoding prefactor given by largest element σ_* of Σ_1. This can be found using the explicit formula for $\tilde{F}_1$ and, in the relevant regime of $\bar{\tau}^\star > 1/2$, we get

$$\alpha_{I+F_1} = \sigma_* = \frac{\sqrt{3^D + 2^{D+1}\left(\bar{\tau}^{\star 2} - \bar{\tau}^\star\right) + \sqrt{9^D + 4 \cdot 6^D \left(\bar{\tau}^{\star 2} - \bar{\tau}^\star\right)}}}{\sqrt{2^{D+1}}\,\bar{\tau}^\star}, \tag{3.61}$$

which is a non-increasing function of $\bar{\tau}^\star$ with values:

$$0 \leq \alpha_{I+F_1} \leq \begin{cases} \sqrt{2+\sqrt{3}} & \text{for} \quad D = 1, \\ \sqrt{\frac{1}{2}(7+3\sqrt{5})} & \text{for} \quad D = 2, \\ \frac{1}{2}\sqrt{23+3\sqrt{57}} & \text{for} \quad D = 3. \end{cases} \tag{3.62}$$

The explicit circuit block-encoding $I + F_1$ is given by:



(3.63)

where $\boldsymbol{v}$ is the velocity register consisting of $2D$ qubits, controls depend on states encoding consecutive discrete velocities m, and V_{σ_m} are single qubit unitaries defined via

$$V_{\sigma_m} |0\rangle \propto |0\rangle + \sqrt{\frac{\sigma_*^2}{\sigma_m^2} - 1}, \tag{3.64}$$

with σ_m denoting the m-th singular value of $I + \tilde{F}_1$, i.e., the m-th diagonal element of Σ_1.

We now proceed to block-encoding $\bar{F}_2$. From Eq. (2.16b), we have

$$\bar{F}_2 = I_{\boldsymbol{r}^\star, \boldsymbol{r}^\star} \otimes |0\rangle_{\boldsymbol{r}^\star} \otimes \tilde{F}_2 \otimes |0\rangle\,, \tag{3.65}$$

where $I_{\boldsymbol{r}^\star, \boldsymbol{r}^\star}$ is of size $N \times N^2$ and $\tilde{F}_2$ is of size $2^{2D} \times 2^{4D}$. As before, for the latter matrix we can in fact consider a restriction to a $Q \times Q^2$ matrix. We then need to construct block-encodings of two square matrices: $N^2 \times N^2$ matrix $\bar{I}_{\boldsymbol{r}^\star, \boldsymbol{r}^\star} := I_{\boldsymbol{r}^\star, \boldsymbol{r}^\star} \otimes |0\rangle_{\boldsymbol{r}^\star}$, and $Q^2 \times Q^2$ matrix $\tilde{F}_2 \otimes |0\rangle$. The first of these can be easily block-encoded with a unit prefactor and a single qubit ancilla using the following permutation matrix:

$$U_{\bar{I}_{\boldsymbol{r}^\star, \boldsymbol{r}^\star}} = \bar{I}_{\boldsymbol{r}^\star, \boldsymbol{r}^\star} \otimes I_a + (\bar{I}_{\boldsymbol{r}^\star, \boldsymbol{r}^\star} - I) \otimes X_a, \tag{3.66}$$

where X_a is a Pauli X matrix acting on the ancilla qubit. Concerning the second matrix, as we explained in Sec. 3.2.1, its block-encoding coincides with a block-encoding of a rectangular matrix

$\tilde{F}_2$, and the prefactors are the same, $\alpha_{\tilde{F}_2\otimes|0\rangle} = \alpha_{\tilde{F}_2}$. Therefore, assuming access to $(\alpha_{\tilde{F}_2}, n_{\tilde{F}_2}, 0)$-block-encoding of $\tilde{F}_2$, we can construct $(\alpha_{\tilde{F}_2}, n_{\tilde{F}_2}+1, 0)$-block-encoding of $\bar{F}_2$.

As for the block-encoding of $\tilde{F}_2$, we will use a method based on higher-order singular value decomposition (HOSVD). More precisely, we aim at decomposing the $Q \times Q^2$ matrix $\tilde{F}_2$ as

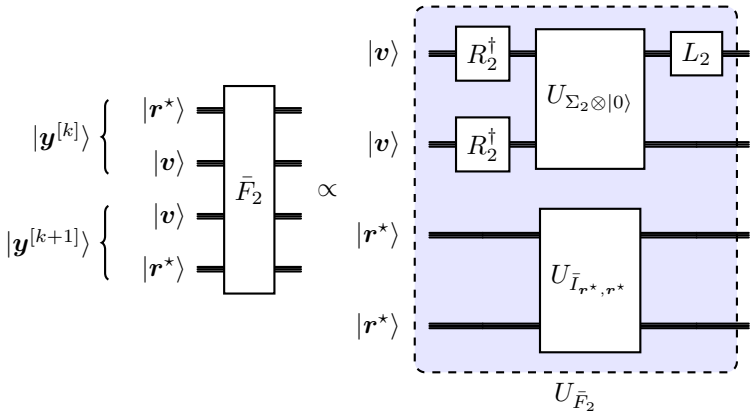
$$\tilde{F}_2 = L_2 \Sigma_2 (R_2 \otimes R_2)^\dagger, \tag{3.67}$$

where L_2 and R_2 are $Q \times Q$ unitary matrices, while Σ_2 is a $Q \times Q^2$ with as small rank as possible. Then, again, we assume access to $2D$-qubit unitaries L_2 and R_2 (that we construct in Sec. 4.5.4), and we only need to block-encode a sparse matrix Σ_2. The exact block-encoding cost of Σ_2 will depend on the particular HOSVD used. However, as we shall see in Sec. 4.5.4, we find constructions for $D \in \{1, 2\}$ where this coincides with the block-encoding prefactors for the diagonal matrix with singular values of $\tilde{F}_2$, while the number of ancillary qubits needed is 1 for $D = 1$ and 8 for $D = 2$. Since we expect that also for $D = 3$ the block-encoding will not differ significantly from this, we will use the largest singular value of $\tilde{F}_2$ for the block-encoding prefactor $\alpha_{\tilde{F}_2}$, and $4D$ for the number of ancillary qubits needed. As the largest singular value can be explictly calculated, we end up with $(\alpha_{\bar{F}_2}, n_{\bar{F}_2}, 0)$-block-encoding of $\bar{F}_2$ with

$$\alpha_{\bar{F}_2} = \frac{2}{3\bar{\tau}^\star}\left(\frac{3}{\sqrt{2}}\right)^D \sqrt{D+2}, \tag{3.68a}$$

$$n_{\bar{F}_2} = 4D + 1. \tag{3.68b}$$

The circuit realizing the block-encoding of $\bar{F}_2$ is given by:



(3.69)

where $U_{\Sigma_2\otimes|0\rangle}$ is the block-encoding of the square embedding of Σ_2 in the full space. We also recall that in the circuit diagram we do not show the auxiliary qubits that begin and end in the zero state, for successful implementation of the block-encoding.

Finally, putting everything we derived in this section together, we end up with the final expression for the block-encoding prefactor α_A and the number of ancillary qubits n_A needed:

$$\alpha_A = 1 + \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \binom{N_C - l}{l} \alpha_{I+F_1}^{N_C-2l} \alpha_{\bar{F}_2}^l, \tag{3.70a}$$

$$n_A = N_C + 1 + \log\left(\left\lfloor \frac{N_C}{2} \right\rfloor + 1\right) + \max_{l\in\{0,\ldots,\lfloor N_C/2\rfloor\}} \left[\log\binom{N_C - l}{l} + l(4D-1)\right], \tag{3.70b}$$

where α_{I+F_1} and $\alpha_{\bar{F}_2}$ are given by Eqs. (3.61) and (3.68a), respectively.

3.2.5 State preparations

The final missing component before we can run a quantum linear solver is a unitary state preparation $U_{\boldsymbol{b}}$ of $|\boldsymbol{b}\rangle$. For that, we assume access to a unitary state preparation $U_{\psi_{\text{ini}}}$ of the initial state

$|\psi_{\text{ini}}\rangle$ for the shifted LBE problem:

$$|\psi_{\text{ini}}\rangle \propto \sum_{\boldsymbol{r}^\star} \sum_{m=1}^{Q} g_m(\boldsymbol{r}^\star, 0)\, |\boldsymbol{r}^\star, m\rangle\,, \tag{3.71}$$

where we use a concise notation for the velocity registers. In this work, we disregard walls, inlets and outlets (which we analyze elsewhere [57]), but consider non-trivial initial states given by $g_{m,\boldsymbol{r}^\star}(0)$ describing e.g., Taylor-Green vortices and vortex dipoles, see Sec. 4.1 for details. The complexity cost of constructing $U_{\psi_{\text{ini}}}$ will clearly depend on the initial state $|\psi_{\text{ini}}\rangle$ considered. Although it may be significant for random or unstructured states, in practice one investigates initial velocity fields with explicit functional dependence on position, and so the corresponding initial states are well-structured. Thus, we will assume that the complexity cost of $U_{\psi_{\text{ini}}}$ is negligible compared to the cost of unitary block-encoding U_A of the linear system.

Moreover, one can also show that the complexity cost of $U_{\boldsymbol{b}}$ does not differ much from that of $U_{\psi_{\text{ini}}}$, and so can be neglected. Namely, in order to prepare $|\boldsymbol{b}\rangle$ using $U_{\psi_{\text{ini}}}$, one can employ a result from Ref. [31], where Lemma 4 shows how to construct $U_{\boldsymbol{b}}$ using $O(N_C)$ queries to $U_{\psi_{\text{ini}}}$ and $O(N_C)$ single qubit gates. Hence, each query to $U_{\boldsymbol{b}}$ translates to $O(N_C)$ queries to $U_{\psi_{\text{ini}}}$ (in fact, exactly N_C, so there are no hidden large constant prefactors).

3.3 Quantum linear solver

Once we have encoded the data about our fluid dynamics problem in unitaries U_A and $U_{\boldsymbol{b}}$, we can use a quantum linear solver to prepare a quantum state encoding the solution to our problem in its amplitudes. We shall use the algorithm from Ref. [12], since at the time of writing, it provides the lowest rigorously guaranteed upper bound to the number of required applications of (controlled) U_A, $U_{\boldsymbol{b}}$ and their inverses, i.e., the best upper bound for the query complexity q_Q. However, in order to get the correct query complexity upper bound, we need to carefully translate our problem to the standardized linear problem studied in Ref. [12]. More precisely, in that work the author considers the linear problem of the form

$$A\boldsymbol{Y} = \boldsymbol{b}, \tag{3.72}$$

where $\boldsymbol{Y}$ is the desired solution vector, $\boldsymbol{b}$ is a normalized vector, and it is assumed that one has access a unitary block-encoding U_A of A with a block-encoding prefactor $\alpha_A = 1$, and to a unitary state preparation $U_{\boldsymbol{b}}$. Under such assumptions, the singular values of A satisfy

$$\sigma(A) \in [1/\kappa_A, 1], \tag{3.73}$$

where κ_A denotes the condition number of A. In such a setting, the work Ref. [12] provides a quantum algorithm for outputting a quantum state that is at most ϵ_Q away from the normalized solution $|\boldsymbol{Y}\rangle$, with the following query upper bound

$$q_Q \leq 56.0\kappa_A + 1.05\kappa_A \ln\left(\frac{\sqrt{1-\epsilon_Q^2}}{\epsilon_Q}\right) + 2.78\ln(\kappa_A)^3 + 3.17. \tag{3.74}$$

More precisely, q_Q is the total number of queries to U_A, $U_A^\dagger$ and their controlled versions, and $2q_Q$ is the total number of queries to $U_{\boldsymbol{b}}$, $U_{\boldsymbol{b}}^\dagger$ and their controlled versions.

The difference then is that our block-encoding prefactor α_A is not 1 and that $\boldsymbol{b}$ is not normalized. To bring our problem to the standardized form described above, note that the linear problems we consider, $A\boldsymbol{Y} = \boldsymbol{b}$, are equivalent to

$$\tilde{A}\tilde{\boldsymbol{Y}} = \boldsymbol{b}, \tag{3.75}$$

where

$$\tilde{A} = \frac{A}{\alpha_A}, \qquad \tilde{\boldsymbol{Y}} = \frac{\alpha_A}{\|\boldsymbol{b}\|}\boldsymbol{Y}. \tag{3.76}$$

Note that having access to $(\alpha_A, n_A, 0)$-block-encoding of A is equivalent to having access to $(1, n_A, 0)$-block-encoding of $\tilde{A}$. Also, considering $\tilde{\boldsymbol{Y}}$ instead of $\boldsymbol{Y}$ makes no difference, as the QLS

prepares a normalized quantum state, so it is insensitive to simple scaling. The only important difference then is that the range of singular values of $\tilde{A}$ differs from that of A:

$$\sigma(\tilde{A}) \in \left[1/\|\tilde{A}^{-1}\|, \|\tilde{A}\|\right] = \left[\frac{1}{\frac{\alpha_A}{\|A\|}\kappa_A}, \frac{\|A\|}{\alpha_A}\right] \subseteq \left[\frac{1}{\frac{\alpha_A}{\|A\|}\kappa_A}, 1\right], \tag{3.77}$$

where we used that $\alpha_A \geq \|A\|$. Comparing with Eq. (3.73), we then end up with the following query complexity upper bound for our problem:

$$q_Q \leq \frac{\alpha_A}{\|A\|}\left(56.0\kappa_A + 1.05\kappa_A \ln\left(\frac{\sqrt{1-\epsilon_Q^2}}{\epsilon_Q}\right) + 2.78\ln(\kappa_A)^3 + 3.17\right). \tag{3.78}$$

Upon successful execution, the quantum linear solver algorithm outputs a coherent encoding of the solution to $A\boldsymbol{Y} = \boldsymbol{b}$ as normalized quantum states. More precisely, for the history state and final state output we get:

$$|\boldsymbol{Y}_H\rangle = \frac{1}{\|\boldsymbol{Y}_H\|}\sum_{t^\star=0}^{T^\star}|t^\star\rangle \otimes \boldsymbol{y}(t^\star) = \frac{1}{\|\boldsymbol{Y}_H\|}\sum_{t^\star=0}^{T^\star}|t^\star\rangle \otimes (\mathcal{SC})^{t^\star}\boldsymbol{y}_{\text{ini}}, \tag{3.79a}$$

$$\begin{aligned}|\boldsymbol{Y}_F\rangle &= \frac{1}{\|\boldsymbol{Y}_F\|}\left(\sum_{t^\star=0}^{T^\star}|0\rangle \otimes |t^\star\rangle \otimes \boldsymbol{y}(t^\star) + \sum_{w=1}^{2^W-1}\sum_{t^\star=0}^{T^\star}|w\rangle \otimes |t^\star\rangle \otimes \boldsymbol{y}(T^\star)\right) \\ &= \frac{1}{\|\boldsymbol{Y}_F\|}\left(\sum_{t^\star=0}^{T^\star}|0\rangle \otimes |t^\star\rangle \otimes (\mathcal{SC})^{t^\star}\boldsymbol{y}_{\text{ini}} + \sum_{w=1}^{2^W-1}\sum_{t^\star=0}^{T^\star}|w\rangle \otimes |t^\star\rangle \otimes (\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}\right).\end{aligned} \tag{3.79b}$$

These two outputs describe the coherent encoding of the dynamics, provided in the form of an entangled quantum state over the registers. We next turn to post-processing and information-extraction.

3.4 Measurements and data extraction

3.4.1 Recovering final and history LBE states

We start by explaining how to recover the Carleman state of the system at the final time $T^\star$ from the state $|\boldsymbol{Y}_F\rangle$. To achieve this, we perform a measurement of both the waiting register w and time register $t^\star$, discarding them afterwards. From Eq. (3.79b), it is clear that if the w outcome is anything but all zero state, then irrespectively of the $t^\star$ outcome, we collapse the system onto the final Carleman state:

$$|\boldsymbol{y}(T^\star)\rangle = \frac{1}{\|\boldsymbol{y}(T^\star)\|}(\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}. \tag{3.80}$$

Introducing

$$\mathcal{N}_H = \sum_{t^\star=0}^{T^\star}\|(\mathcal{SC})^{t^\star}\boldsymbol{y}_{\text{ini}}\|^2, \qquad \mathcal{N}_F = (T^\star+1)\|(\mathcal{SC})^{T^\star}\boldsymbol{y}_{\text{ini}}\|^2, \tag{3.81}$$

we see that this happens with probability

$$p(\boldsymbol{y}(T^\star)) = \frac{1}{1+\frac{1}{2^W-1}\cdot\frac{\mathcal{N}_H}{\mathcal{N}_F}}. \tag{3.82}$$

The above approaches 1 exponentially with increasing the number W of qubits in the waiting register. Moreover, under the assumption that the norm of the Carleman state does not vary significantly during the evolution, one has $\mathcal{N}_H \approx \mathcal{N}_F$. Thus, a single waiting qubit gives a constant probability of success, approximately equal to 1/2. Given the expected high success probability, in practice a repeat-till-success strategy appears to be more convenient than a strategy based on amplitude amplification, although the latter is also possible.

Next, both for the Carleman history state $|\boldsymbol{Y}_H\rangle$ and the final Carleman state $|\boldsymbol{y}(T^\star)\rangle$, we measure the Carleman block register n_C. We keep the outcome only if $n_C = 1$, and discard all the state vector registers except for the first one. Thus, we collapse the system onto

$$|\psi_F\rangle := |\boldsymbol{y}_1(T^\star)\rangle \approx |\boldsymbol{g}(T^\star)\rangle\,, \tag{3.83a}$$

$$|\psi_H\rangle := \sum_{t^\star=0}^{T^\star} |t^\star\rangle \otimes |\boldsymbol{y}_1(t^\star)\rangle \approx \sum_{t^\star=0}^{T^\star} \frac{\|\boldsymbol{g}(t^\star)\|}{\mathcal{N}_{\boldsymbol{g}}} |t^\star\rangle \otimes \boldsymbol{g}(t^\star), \tag{3.83b}$$

where the approximation error comes from the Carleman truncation error ϵ_C and QLS error ϵ_Q, whereas $\mathcal{N}_{\boldsymbol{g}}$ is the normalization factor. Assuming $\|\boldsymbol{y}_k\| \approx \|\boldsymbol{g}\|^k$, one obtains $|\psi_F\rangle$ with with probability

$$p(\psi_F) \approx \frac{\|\boldsymbol{g}(T^\star)\|^2}{\sum_{k=1}^{N_C} \|\boldsymbol{g}(T^\star)\|^{2k}} = \frac{1 - \|\boldsymbol{g}(T^\star)\|^2}{1 - \|\boldsymbol{g}(T^\star)\|^{2N_C}}. \tag{3.84a}$$

Moreover, assuming that the norm of $\boldsymbol{g}(t^\star)$ does not change significantly during the evolution, so that $\|\boldsymbol{g}(t^\star)\| \approx \|\boldsymbol{g}(T^\star)\|$ for all $t^\star$, the probability $p(\psi_H)$ of obtaining $|\psi_H\rangle$ is the same, i.e., $p(\psi_H) \approx p(\psi_F)$. Crucially, note that differently from prior proposals, our introduction of a shifted LBE model and our choice of parameters from Eq. (2.36) guarantees $\|\boldsymbol{g}\| = O(1)$. Hence, we get constant success probabilities: in particular, when $\|\boldsymbol{g}\| \approx 1$, the probabilities become approximately $1/N_C$.[5] To show that this is a genuine solution to Problem 4 of Sec. 1.4.4, rather than just a way to shift Problem 4 into Problem 2, we will need to analyze how our choice of parameters affects the condition number κ_A, which we shall do in Sec. 4.3.

3.4.2 Extracting the drag force

We now proceed to explaining how to perform measurements on $|\psi_F\rangle$ to estimate the drag force at the final time $T^\star$ on some boundary wall. Our analysis will closely follow the proposal presented in Ref. [38], with some small changes due to us working with the shifted LBE vector $\boldsymbol{g}$, and not the LBE vector $\bar{\boldsymbol{f}}$. Using the the standard momentum exchange approach to the drag force [41], the local momentum density change $\Delta\boldsymbol{p}^\star(\boldsymbol{r}^\star, T^\star, m)$, at time $T^\star$ and at the fluid node $\boldsymbol{r}^\star$, with the wall node at $\boldsymbol{r}^\star + \boldsymbol{e}_m^\star$ in lattice units is given by

$$\Delta\boldsymbol{p}^\star(\boldsymbol{r}^\star, T^\star, m) = \left(\bar{f}_m(\boldsymbol{r}^\star, T^\star) + \bar{f}_{-m}(\boldsymbol{r}^\star, T^\star)\right) \boldsymbol{e}_m^\star, \tag{3.85}$$

where, as before, $-m$ is the velocity index corresponding to discrete velocity $-\boldsymbol{e}_m^\star$. Using the definition of the shifted LBE vector from Eq. (2.1), the above can be rewritten as

$$\Delta\boldsymbol{p}^\star(\boldsymbol{r}^\star, T^\star, m) = \Delta\boldsymbol{p}_0^\star(m) + \left(g_m(\boldsymbol{r}^\star, T^\star) + g_{-m}(\boldsymbol{r}^\star, T^\star)\right) \boldsymbol{e}_m^\star, \tag{3.86}$$

with

$$\Delta\boldsymbol{p}_0^\star(m) = 2w_m \boldsymbol{e}_m^\star \tag{3.87}$$

denoting the zero-velocity equilibrium value of momentum density change.

Next, let us encode the wall boundary of interest using the $\mathcal{W}$ function from Eq. (2.21). Since in lattice units the volume of one spatial cell is 1, the total momentum exchange $\Delta\boldsymbol{P}^\star$ at the boundary is just the sum over local momentum density changes:

$$\Delta\boldsymbol{P}^\star(T^\star) = \sum_{\boldsymbol{r}^\star} \sum_m \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}_m^\star} (1 - \mathcal{W}_{\boldsymbol{r}^\star}) \Delta\boldsymbol{p}^\star(\boldsymbol{r}^\star, T^\star, m). \tag{3.88}$$

Now, the drag force $\boldsymbol{F}^\star(T^\star)$ can be calculated by dividing the total momentum change by the time it takes. Since in lattice units one time step has length 1, the drag force is simply equal to $\Delta\boldsymbol{P}^\star(T^\star)$, which can be explicitly written as

$$\boldsymbol{F}^\star(T^\star) = \boldsymbol{F}_0^\star + \sum_{\boldsymbol{r}^\star} \sum_m \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}_m^\star} (1 - \mathcal{W}_{\boldsymbol{r}^\star}) \left(g_m(\boldsymbol{r}^\star, T^\star) + g_{-m}(\boldsymbol{r}^\star, T^\star)\right) \boldsymbol{e}_m^\star, \tag{3.89}$$

[5] Also note that one can improve this constant probability quadratically by using amplitude amplification.

where

$$\boldsymbol{F}_0^\star = \sum_{\boldsymbol{r}^\star} \sum_m \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}_m^\star} (1 - \mathcal{W}_{\boldsymbol{r}^\star}) 2 w_m \boldsymbol{e}_m^\star. \tag{3.90}$$

Hence, if we can estimate $\boldsymbol{F}^\star(T^\star)$, we can get the drag force in physical units by simply multiplying $\boldsymbol{F}^\star(T^\star)$ by $\Delta x^2/\Delta t$ and by the reference mass (i.e., the mass of a single unit cell).

We shall assume that $\boldsymbol{F}_0^\star$ can be efficiently computed classically, and we will use $|\psi_F\rangle$ to estimate the value of the second term on the right hand side of Eq. (3.89), i.e., to get its components:

$$\mathcal{F}_k := \sum_{\boldsymbol{r}^\star} \sum_m \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}_m^\star} (1 - \mathcal{W}_{\boldsymbol{r}^\star}) \left(g_m(\boldsymbol{r}^\star, T^\star) + g_{-m}(\boldsymbol{r}^\star, T^\star) \right) (\boldsymbol{e}_m^\star)_k, \tag{3.91}$$

where $k \in \{x, y, z\}$. To that end, for each component k we prepare the following quantum state encoding the boundary of interest:

$$|\mathcal{W}_k\rangle = \frac{1}{\sqrt{\mathcal{N}_{\mathcal{W}_k}}} \sum_{\boldsymbol{r}^\star} \sum_m \mathcal{W}_{\boldsymbol{r}^\star + \boldsymbol{e}_m^\star} (1 - \mathcal{W}_{\boldsymbol{r}^\star}) (\boldsymbol{e}_m^\star)_k \, |\boldsymbol{r}^\star\rangle \otimes (|m\rangle + |-m\rangle), \tag{3.92}$$

with $\mathcal{N}_{\mathcal{W}_k}$ denoting normalization constants that can be efficiently precomputed, and the velocity registers written in a concise notation. This quantum state can be efficiently prepared when certain simplicity criteria on the boundary function are met, e.g., via rejection sampling techniques [59]. Broadly speaking, when $\mathcal{W}$ is an analytically defined and relatively simple function, this state preparation will have a subleading complexity cost compared to the rest of the algorithm.

Now, recalling that

$$|\psi_F\rangle \approx |\boldsymbol{g}(T^\star)\rangle = \frac{1}{\|\boldsymbol{g}(T^\star)\|} \sum_{\boldsymbol{r}^\star} \sum_m g_m(\boldsymbol{r}^\star, T^\star) \, |\boldsymbol{r}^\star\rangle \otimes |m\rangle \,, \tag{3.93}$$

we see that the overlaps between $|\psi_F\rangle$ and $|\mathcal{W}_k\rangle$ are given by

$$\langle \mathcal{W}_k | \psi_F \rangle \approx \frac{\mathcal{F}_k}{\|\boldsymbol{g}(T^\star)\| \sqrt{\mathcal{N}_{\mathcal{W}_k}}}. \tag{3.94}$$

Note that one can efficiently compute $\mathcal{N}_{\mathcal{W}_k}$ and estimate $\|\boldsymbol{g}(T^\star)\|$ from the QLS algorithm. More precisely, one can estimate $\|\boldsymbol{Y}_F\|$ and then use the knowledge of probabilities $p(\boldsymbol{y}(T^\star))$ and $p(\psi_F)$ to get $\|\boldsymbol{g}(T^\star)\|$.

Thus, the only thing left to estimate $\mathcal{F}_k$ is to find the overlaps $\langle \mathcal{W}_k | \psi_F \rangle$, which can be achieved using quantum amplitude estimation. However, one has to note that the boundary states $|\mathcal{W}_k\rangle$ only have support on $O(N_x^{D-1})$ states, in contrast to $|\psi_F\rangle$ having support on $O(N_x)$ states. Hence, one expects $\langle \mathcal{W}_k | \psi_F \rangle$ to be of the order $O(1/\sqrt{N_x})$, which translates into a multiplicative factor overhead scaling as $O(\sqrt{N_x})$. Recalling the parameter choices in Eq. (2.36), we conclude that, if one wants to extract the drag force as described above, one needs to multiply the complexity of obtaining the state $|\psi_F\rangle$ by

$$q_M := O(\mathrm{Re}^{\beta/2}). \tag{3.95}$$

Moreover, one needs to include the complexity of preparing states $|\mathcal{W}_k\rangle$, which is expected to be negligible for simple boundaries like flat walls, but can become more problematic for complex boundaries.

4 Quantum algorithm performance

4.1 Carleman error convergence

4.1.1 Methodology

To numerically investigate Carleman error convergence, we first solve the shifted incompressible LBE directly, by simulating $T^\star$ streaming and collision steps from Eqs. (2.3a)-(2.3b) with $\boldsymbol{g}^{\mathrm{eq}}$ given by Eq. (2.6) (i.e., assuming incompressibility). This way, we obtain the shifted state vector $\boldsymbol{g}(t^\star)$ for each discrete time $t^\star \in \{0, \ldots, T^\star\}$, with $\boldsymbol{g}(0)$ encoding our selected initial condition

(see below). Next, we numerically solve the shifted incompressible LBE employing the discrete Carleman embedding, by simulating Eq. (2.49) for $T^\star$ time steps. This way, we obtain the Carleman state vector $\boldsymbol{y}(t^\star)$ for each discrete time $t^\star \in \{0, \ldots, T^\star\}$, with $\boldsymbol{y}(0) = \boldsymbol{y}_{\text{ini}}$ encoding the initial condition according to Eq. (2.51). From that, we recover the Carleman approximation $\boldsymbol{y}_1(t^\star)$ of $\boldsymbol{g}(t^\star)$.

In previous literature, a measure of Carleman truncation error was proposed that was based on the root mean square error (RMSE) between the actual LBE vector and its Carleman approximation [39, 40]. Here, we decide to not use this measure for the following reason. Namely, for different Reynolds numbers Re, the values of discretization parameters Δx and Δt vary. Thus, for the same values of RMSE (which are calculated in lattice units), but for two different Reynolds numbers, the actual errors in the physical velocity field will be different. To allow for easy comparison with previous studies, however, we present the results for RMSE-based Carleman error in Appendix E. For example, one can clearly see there that the Carleman error for $N_C = 1$ (so for pure linearization) decreases with increasing Re. This behavior may be misleading if one does not take into account unit conversion, as increasing Re means stronger nonlinear effects and so one expects pure linearization to give higher errors for for higher Re.

Here, we decided to quantify the Carleman truncation error using the average error in the physical velocity field relative to the characteristic velocity u of the problem, e.g., the flow velocity or, in the absence of driving, maximal velocity in the initial state. More precisely, denote the lattice velocities obtained from $\boldsymbol{g}$ and its Carleman approximation $\boldsymbol{y}_1$ by $\boldsymbol{u}^\star(\boldsymbol{r}^\star, t^\star)$ and $\tilde{\boldsymbol{u}}^\star(\boldsymbol{r}^\star, t^\star)$, respectively. Then, for all positions $\boldsymbol{r}^\star$ and time steps $t^\star$, introduce the velocity error vector:

$$\Delta \boldsymbol{u}^\star(\boldsymbol{r}^\star, t^\star) = \tilde{\boldsymbol{u}}^\star(\boldsymbol{r}^\star, t^\star) - \boldsymbol{u}^\star(\boldsymbol{r}^\star, t^\star), \tag{4.1}$$

and its version $\Delta \boldsymbol{u}(\boldsymbol{r}^\star, t^\star)$ in physical units. We now define the Carleman truncation error ϵ_C as follows:

$$\epsilon_C := \max_{t^\star \in \{1,\ldots,T^\star\}} \frac{1}{N} \sum_{\boldsymbol{r}^\star} \frac{\|\Delta \boldsymbol{u}\|}{u}, \tag{4.2}$$

i.e., as the relative physical velocity error with respect to the flow velocity u, averaged over all N lattice sites, and maximized over all times. Note that we have

$$\frac{\|\Delta \boldsymbol{u}\|}{u} = \frac{\|\Delta \boldsymbol{u}^\star\|}{u} \frac{\Delta x}{\Delta t} = \frac{\|\Delta \boldsymbol{u}^\star\|}{u} \frac{3\nu}{\Delta x \left(\tau^\star - \frac{1}{2}\right)} = \text{Re}^{\frac{\beta D}{2}} \|\Delta \boldsymbol{u}^\star\|, \tag{4.3}$$

where we first converted $\Delta \boldsymbol{u}$ to the lattice units, then used Chapman-Enskog relation from Eq. (1.19), and finally used the choice of parameter scaling with Reynolds number Re from Eq. (2.36). Thus, one can easily calculate ϵ_C using the numerical data on $\Delta \boldsymbol{u}^\star$ as follows:

$$\epsilon_C = \frac{\text{Re}^{\frac{\beta D}{2}}}{N} \max_{t^\star \in \{1,\ldots,T^\star\}} \sum_{\boldsymbol{r}^\star} \|\Delta \boldsymbol{u}^\star\| = \text{Re}^{\frac{-\beta D}{2}} \max_{t^\star \in \{1,\ldots,T^\star\}} \sum_{\boldsymbol{r}^\star} \|\Delta \boldsymbol{u}^\star\|. \tag{4.4}$$

Due to immense memory requirements for classically simulating Carleman evolution of three-dimensional LBE systems,[6] we restrict our investigations to one- and two-dimensional settings. We consider the spatial discretization parameter $\beta = 3/4$, which is the value allowing one to resolve the Kolmogorov microscale. The simulation parameters N_x, $T^\star$, $\bar{\tau}^\star$, and $u^\star_{\text{ini}}$ are chosen according to Eq. (2.36), so that

$$N_x = \left\lceil \text{Re}^{3/4} \right\rceil, \qquad T^\star = \left\lceil \text{Re}^{\frac{3}{4}(D/2+1)} \right\rceil, \quad \bar{\tau}^\star = \frac{1}{2} + \frac{3}{\text{Re}^{\frac{3}{4}(D/2-1)+1}}, \quad u^\star_{\text{ini}} = \frac{1}{\text{Re}^{3D/8}}. \tag{4.5}$$

Note that in the original Eq. (2.36) it was $u^\star_{\max}$, and not $u^\star_{\text{ini}}$ scaling as above. Let us remind that the parameter choice leading to such a scaling was motivated by keeping $\|\boldsymbol{g}(t^\star)\| \leq 1$ at all times. In all of our numerical simulations, we confirmed that choosing $u^\star_{\text{ini}}$ (that we have control over)

[6] For a system with Re = 100, assuming spatial resolution parameter $\beta = 3/4$ to resolve the Kolmogorov microscale, the Carleman vector for the smallest nontrivial truncation order $N_C = 2$ has dimension approximately equal to 7.8×10^{11}. Assuming each entry is stored as a double precision real number taking 8 bytes, it means that the Carleman vector takes up roughly 6.2 terabytes.

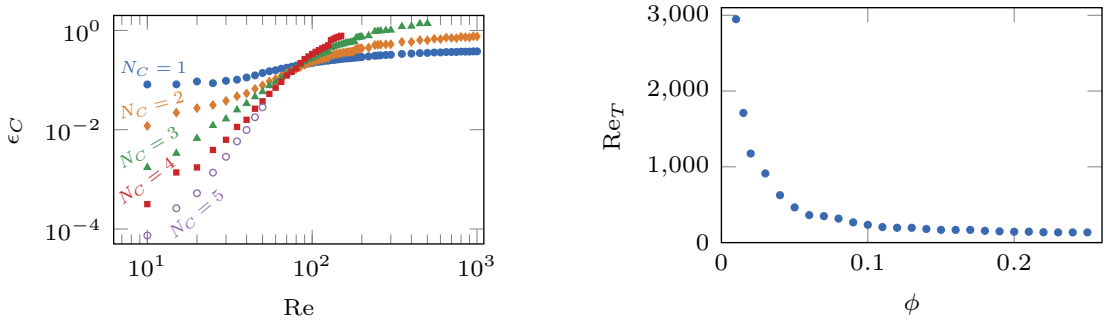


Figure 1: **Carleman truncation error and convergence for $D = 1$.** Left: Truncation error ϵ_C dependence on the Reynolds number Re and truncation order N_C for D1Q3 model with $\beta = 3/4$ (corresponding to spatial discretization resolving the Kolmogorov microscale), $u_0^\star = 1$ (enough to keep the norm of the solution below 1 at all times), and the initial sinusoidal state. Right: The threshold value Re_T above which the Carleman truncation error ϵ_C is larger for truncation order $N_C = 2$ than for $N_C = 1$. Data for D1Q3 model with $\beta = 3/4$, $u_0^\star = 1$, and initial states given by colliding states with fraction parameter ϕ.

to scale as above instead of $u^\star_{\max}$ (that we have no control over) guarantees that $\|\boldsymbol{g}(t^\star)\|$ is always below 1.[7] We perform simulations for a range of Reynolds numbers $\mathrm{Re} \in [10, 1000]$ and, for the convenience of the reader, we list the corresponding numerical values of simulation parameters for selected values of Re in Appendix F.

4.1.2 Convergence for $D = 1$

In the one-dimensional case, the compressible Navier–Stokes equations admit non-trivial solutions in the form of pressure waves, since density variations are allowed. By contrast, the incompressible Navier–Stokes equations admit only a trivial, spatially uniform solution, since the velocity field must be divergence-free. Acknowledging that incompressible LBE, to leading order, can only recover a spatially constant state in one-dimension, we nevertheless assess Carleman error convergence for this case, as larger N_C are numerically accessible allowing us to present the main ideas.

We consider two families of initial states, the *sinusoidal states* defined by

$$\boldsymbol{g}(\boldsymbol{r}^\star, 0) = \boldsymbol{g}^{\mathrm{eq}}\left(u^\star = u^\star_{\mathrm{ini}} \sin \frac{2\pi r^\star_x}{N_x}\right), \tag{4.6}$$

and the *colliding states*, parametrized by the fraction parameter ϕ as

$$\boldsymbol{g}(\boldsymbol{r}^\star, 0) = \begin{cases} \boldsymbol{g}^{\mathrm{eq}}(u^\star = u^\star_{\mathrm{ini}}) & \text{for} \quad 1 \le r^\star_x \le \lceil \phi N_x \rceil, \\ \boldsymbol{g}^{\mathrm{eq}}(u^\star = 0) & \text{for} \quad \lceil \phi N_x \rceil < r^\star_x < N_x - \lceil \phi N_x \rceil + 1, \\ \boldsymbol{g}^{\mathrm{eq}}(u^\star = -u^\star_{\mathrm{ini}}) & \text{for} \quad N_x - \lceil \phi N_x \rceil + 1 \le r^\star_x \le N_x. \end{cases} \tag{4.7}$$

In the above, $\boldsymbol{g}^{\mathrm{eq}}(u^\star)$ is the equilibrium distribution corresponding to zero density fluctuations, $\delta\rho = 0$, and lattice velocity $u^\star$ (see Eq. (2.6)), while $u^\star_{\mathrm{ini}}$ depends on the Reynolds number Re according to the parameter choice from Eq. (4.5).

Our numerical results on the behavior of the Carleman truncation error ϵ_C as a function of the Reynolds number Re and the Carleman truncation order N_C are presented in Fig. 1. In the left panel, for the sinusoidal states, one can clearly observe the existence of a threshold Reynolds number Re_T, such that ϵ_C converges (i.e., decreases as N_C increases) for $\mathrm{Re} \le \mathrm{Re}_T$, and diverges otherwise. For the colliding state, the same behavior is observed, just with the position of the threshold depending on the fraction parameter ϕ. As we illustrate in the right panel of Fig. 1, the

[7] To be more precise, it is the norm $\|\boldsymbol{g}(t^\star)\|$ that stays below 1, not $\|\boldsymbol{y}_1(t^\star)\|$. For most simulations $\|\boldsymbol{y}_1(t^\star)\|$ also stays below 1, but this stops being the case when ϵ_C becomes large, as is the case e.g., for $D = 1$ system with $N_C > 1$ and Reynolds number Re above the threshold value Re_T, see the left panel of Fig. 1. However, in such cases the Carleman output anyway is very erroneous, so even if one could overcome the data extraction bottleneck created by $\|\boldsymbol{y}_1(t^\star)\| > 1$, the extracted data would not approximate the real fluid system being simulated.

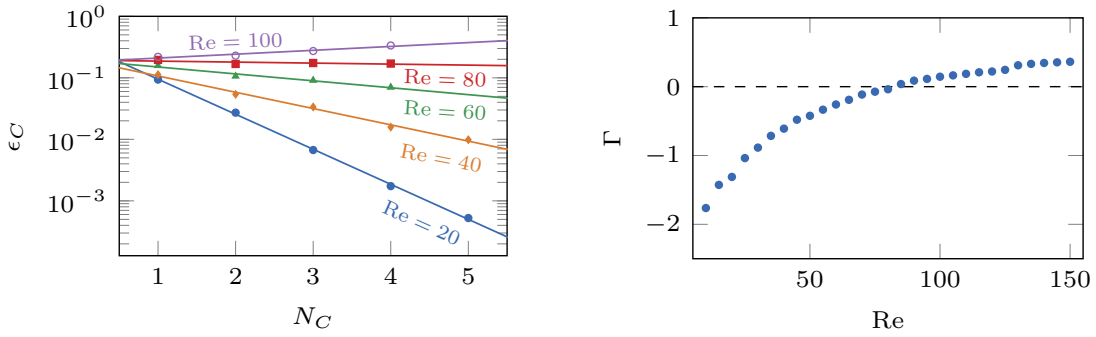


Figure 2: **Carleman error convergence and divergence for $D = 1$.** Left: Dependence on the Carleman error ϵ_C (points) on the truncation order N_C for selected values of the Reynolds number Re below and above the threshold Re_T, together with the best exponential fits (solid lines). Data obtained for the $D = 1$ system with the initial sinusoidal state. Right: Values of the corresponding exponential damping parameters Γ (see Eq. (4.8)) as a function of the Reynolds number Re.

threshold becomes higher for smaller ϕ, i.e., when there are less nodes with non-zero velocity, which directly translates to a smaller average magnitude of velocity. The existence of the threshold can be understood, as Re quantifies the strength of nonlinear fluid behavior, and it is known that the Carleman procedure converges only when nonlinearities are weak enough. Similarly, the behavior in ϕ is not surprising, as lower ϕ means lower average velocity, and so weaker nonlinearities for the same Re.

A more detailed analysis of the behavior of Carleman truncation error ϵ_C with N_C and Re reveals that its convergence (when $\mathrm{Re} \leq \mathrm{Re}_T$) and divergence (when $\mathrm{Re} > \mathrm{Re}_T$) with N_C is, to a good approximation, exponential. As we present in the left panel of Fig. 2 for the sinusoidal states, we have $\epsilon \propto \exp(\Gamma N_C)$, where Γ is negative for $\mathrm{Re} \leq \mathrm{Re}_T$ and positive otherwise. In the right panel of Fig. 2, we show the dependence on the convergence parameter Γ on the Reynolds number Re, where we can again clearly see the position of the threshold Re_T when Γ changes sign. We can then model the dependence of the Carleman truncation error ϵ by

$$\epsilon_C = E \exp(\Gamma N_C), \tag{4.8}$$

where E can be assumed approximately constant, and Γ becomes negative for low enough Re. As we will see in Sec. 4.4, this will allow us to estimate the dependence of the query complexity of our quantum algorithm on the desired level of error.

4.1.3 Convergence for $D = 2$

In contrast to the one-dimensional case, the incompressible Navier-Stokes equations in two dimensions admit spatially non-trivial solutions. To assess Carleman convergence error, we consider the time evolution of the *Taylor-Green vortex* and the *Gaussian vortex dipole* (see Fig. 3):

$$\boldsymbol{u}_V(\boldsymbol{r}) = [\sin r_x \cos r_y, -\cos r_x \sin r_y], \tag{4.9}$$

$$\boldsymbol{u}_{VD}(\boldsymbol{r}) = \left[\frac{\partial \psi}{\partial r_y}, -\frac{\partial \psi}{\partial r_x}\right], \qquad \psi = e^{-\frac{(r_x-s)^2+r_y^2}{2\sigma^2}} - e^{-\frac{(r_x+s)^2+r_y^2}{2\sigma^2}}, \tag{4.10}$$

where $r_x, r_y \in [-\pi, \pi)$ are the spatial variables, $2s$ is the dipole separation, and σ is the width of the vortex. As there is no forcing to balance viscous dissipation, both solutions decay exponentially in time. The time evolution of the Taylor-Green vortex has a known analytical solution and is therefore used to validate numerical methods [41]. While an analytical solution for the time evolution of the Gaussian vortex dipole does not exist, the vortices' counter-rotation causes them to self-advect [60]. Therefore, although the vortex dipole weakens in intensity and ultimately decays, its time evolution is more nonlinear as compared with the Taylor-Green vortex.

We investigate two families of initial states based on the velocity fields defined above. More precisely, we consider

$$\boldsymbol{g}(\boldsymbol{r}^\star, 0) = \boldsymbol{g}^{\mathrm{eq}}\left(\boldsymbol{u}^\star = u^\star_{\mathrm{ini}} \boldsymbol{u}_X([-\pi + 2\pi r^\star_x/N_x, -\pi + 2\pi r^\star_y/N_y])\right), \tag{4.11}$$

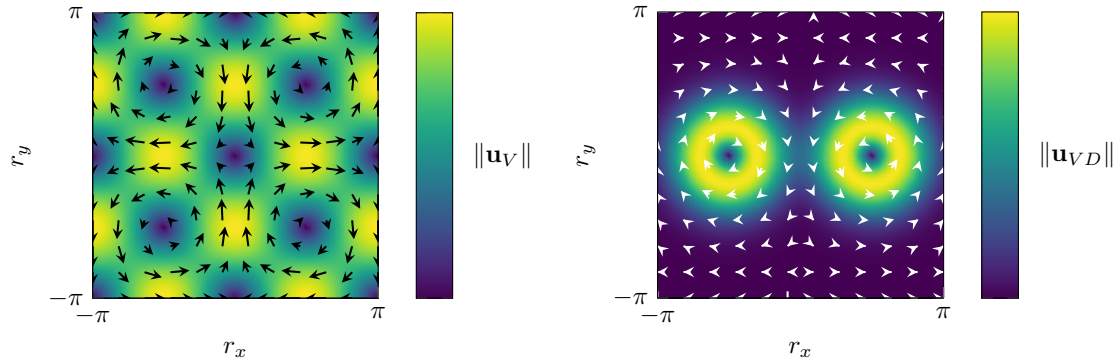


Figure 3: **Velocity field for the investigated initial states.** Velocity field $\boldsymbol{u}^\star(\boldsymbol{r}^\star, 0)$ (arrows) and its magnitude $\|\boldsymbol{u}^\star(\boldsymbol{r}^\star, 0)\|$ (color map) for the Taylor-Green vortex (left) and Gaussian vortex dipole (right) initial states for $D = 2$.

where $\boldsymbol{u}_X$ is either $\boldsymbol{u}_V$ or $\boldsymbol{u}_{VD}$, $\boldsymbol{g}^{\mathrm{eq}}(\boldsymbol{u}^\star)$ is the equilibrium distribution corresponding to zero density fluctuations, $\delta\rho = 0$, and lattice velocity $\boldsymbol{u}^\star$ (see Eq. (2.6)), while $u^\star_{\mathrm{ini}}$ depends on the Reynolds number Re according to Eq. (4.5).

Our numerical results on the dependence of the Carleman truncation error ϵ_C on the Reynolds number Re and the Carleman truncation order N_C are presented in Fig. 4. In contrast to the one-dimensional case, we do not observe the threshold behavior within the numerically accessible regime of Reynolds numbers. For the Taylor-Green vortex, the larger Re gets, the better the improvement in the truncation error ϵ_C when going from $N_C = 1$ to $N_C = 2$, hence not only do we not observe the threshold, but we cannot even extrapolate the data to estimate its position. On the other hand, for the Gaussian vortex dipole, the gap between ϵ_C for $N_C = 1$ and $N_C = 2$ visibly shrinks as Re gets higher, suggesting that Re_T is around a few hundreds. To confirm that, we extended the range of accessible Reynolds number to $\mathrm{Re} = 500$ by reducing the resolution parameter to $\beta = 1/2$. In such a setting, we observe the threshold behavior for the Gaussian vortex dipole at around $\mathrm{Re}_T \approx 400$, while the error behavior for the Taylor-Green vortex stays unchanged (i.e., the error gap widens with growing Re).

Let us now briefly discuss the potential explanation for the observed behavior of ϵ_C. First, the much higher threshold for $D = 2$ as compared to $D = 1$ may be coming from a stronger velocity renormalization, since $u^\star \sim 1/\mathrm{Re}^{\beta D/2}$. Smaller velocity could translate into weaker nonlinearity, so that the same Re (physically describing the same strength of nonlinearity) that lies outside the Carleman convergence radius for $D = 1$ could lie inside it for $D = 2$. This would suggest that for $D = 3$ the threshold could be even higher, as $u^\star$ is even smaller for the same Re. Also, the fact that initial states for $D = 1$ did not satisfy the divergence-free condition could play a role. This is because in that case numerical instabilities might be driving the threshold to lower values. Finally, the difference in the behavior of ϵ_C between the Taylor-Green vortex and the Gaussian vortex dipole may be coming from their fundamentally different dynamics, with the first one simply decaying, and the second one exhibiting non-trivial advection as well. This would suggest that the Carleman linearization method performance may strongly depend on the type of investigated fluid dynamics problem, not only on its Reynolds number.

4.2 Block-encoding prefactor analysis

We now proceed to analyzing the block-encoding prefactor $\alpha_A/\|A\|$ appearing in the upper bound for the query complexity in Eq. (3.78). We have already derived the expression for α_A as a function of the relaxation parameter $\bar{\tau}^\star$ and the Carleman truncation order N_C, see Eq. (3.70a). To estimate $\|A\|$, we note that the norm of A_H (recall Eq. (2.53)) can be bounded as follows:

$$\|A_H\| \leq \|I\| + \|A_H - I\| = 1 + \|A_H - I\|. \tag{4.12}$$

Then, using the definition of operator norm and unitarity of $\mathcal{S}$, we have

$$\|A_H\| \leq 1 + \|\mathcal{C}\|. \tag{4.13}$$

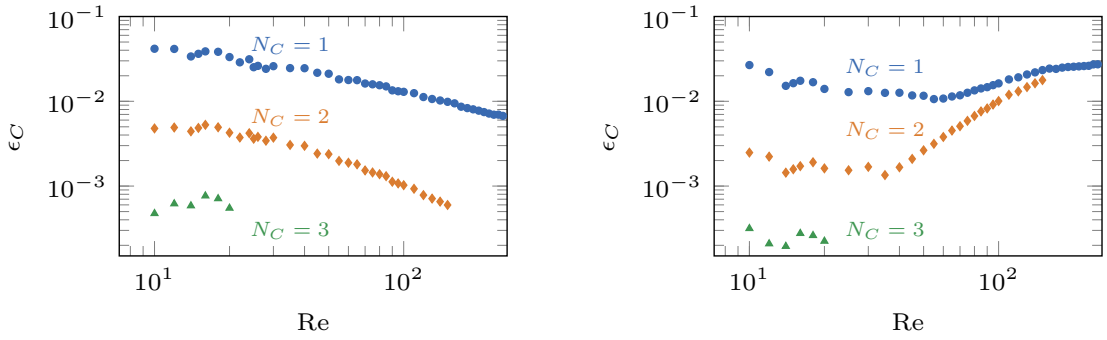


Figure 4: **Carleman truncation error ϵ_C for $D = 2$.** Data for D2Q9 model with $\beta = 3/4$ (corresponding to spatial discretization resolving the Kolmogorov microscale) and $u_0^\star = 1$ (enough to keep the norm of the solution below 1 at all times). Initial states given by the Taylor-Green vortex (left) and Gaussian vortex dipole with $s = \pi/2$ and $\sigma = \pi/5$ (right).

On the other hand, taking a vector $\boldsymbol{v} = [0, \boldsymbol{x}, 0, \dots]^\top$ with $\boldsymbol{x}$ such that $\|\mathcal{C}\boldsymbol{x}\| = \|\mathcal{C}\|\|\boldsymbol{x}\|$, we get

$$\|A_H\| \geq \frac{\|A_H \boldsymbol{v}\|}{\|\boldsymbol{v}\|} = \sqrt{1 + \|\mathcal{C}\|^2}. \tag{4.14}$$

Exactly the same argument applies to the linear system A_F for the final Carleman state, and so we conclude that

$$\sqrt{1 + \|\mathcal{C}\|^2} \leq \|A\| \leq 1 + \|\mathcal{C}\|. \tag{4.15}$$

Note that the above bound is pretty tight, in the sense that the difference between the upper and lower bounds is at most 1.

Now, combining the above with Eq. (3.70a), we get

$$\frac{\alpha_A}{\|A\|} \leq \frac{1 + \sum_{l=0}^{\lfloor \frac{N_C}{2} \rfloor} \binom{N_C - l}{l} \alpha_{I+F_1}^{N_C - 2l} \alpha_{\bar{F}_2}^{l}}{\sqrt{1 + \|\mathcal{C}\|^2}} =: \mathrm{BE}_A^<, \tag{4.16}$$

where the approximation is justified as the relevant terms above grow quickly with N_C and become much larger than 1 in the interesting regime.

Now, the only missing component to get the upper bound $\mathrm{BE}_A^<$ for the block-encoding prefactor $\alpha_A/\|A\|$ is the norm of the Carleman collision matrix $\|\mathcal{C}\|$. To estimate it, note that due to the locality of collision matrices F_k (recall Eqs. (2.15a)-(2.15b)), the norm of $\mathcal{C}$ for a system with N lattice sites is the same as the norm of a matrix $\mathcal{C}$ for a system with a single lattice site. As such, $\|\mathcal{C}\|$ depends only on the relaxation parameter $\bar{\tau}^\star$ and the Carleman truncation order N_C. Moreover, instead of dealing with the norm of d_C-dimensional matrix, one only needs to consider a matrix of dimension

$$\tilde{d}_C := \frac{Q(Q^{N_C} - 1)}{Q - 1}, \tag{4.17}$$

where we recall that $Q = 3^D$ is the number of discrete velocities in the considered LBE model. This way, we can efficiently calculate $\|\mathcal{C}\|$ for small values of N_C and a range of $\bar{\tau}^\star \in [1/2, 1]$. Using these numerical values of $\|\mathcal{C}\|$, together with the expression for α_A, one can calculate upper bounds $\mathrm{BE}_A^<$. We present these in Fig. 5 for the extreme cases of $\bar{\tau}^\star = 1/2$ and $\bar{\tau}^\star = 1$, where one can see that $\mathrm{BE}_A^<$ grows exponentially with N_C, so that we get the following approximate upper bound:

$$\frac{\alpha_A}{\|A\|} \lesssim b \exp(a N_C). \tag{4.18}$$

To get the numerical values of coefficients, we first note that for all D the value of $\mathrm{BE}_A^<$ is largest for $\bar{\tau}^\star = 1/2$. Thus, for each dimension, we take the worst case scenario and perform best linear fits to the following equation:

$$\ln \mathrm{BE}_A^< = a N_C + \ln b. \tag{4.19}$$

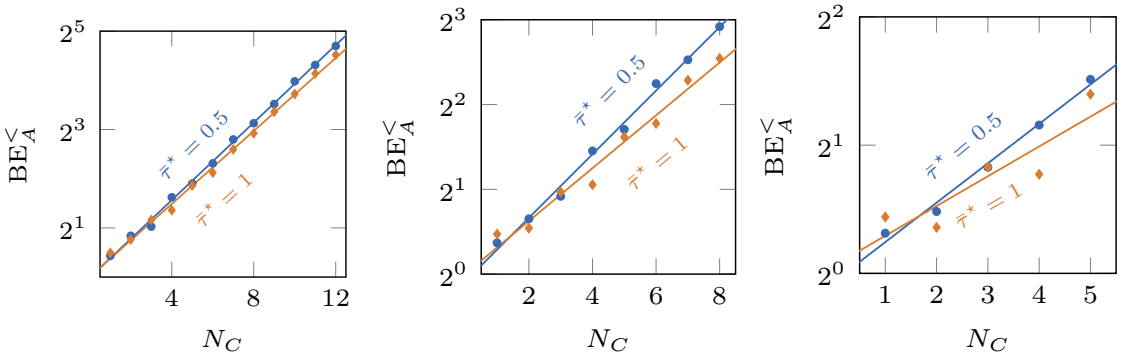


Figure 5: **Upper bounds for block-encoding prefactors.** The numerical values of $BE_A^<$ from Eq. (4.16) for different values of the Carleman truncation order N_C, relaxation parameter $\bar{\tau}^\star$, and dimension D (left to right: $D = 1$ to $D = 3$), together with their best exponential fits as in Eq. (4.19). Note that the fits showcase slightly different slopes for the even and odd N_C, but we took an overall fit.

This way, we obtain the following:

$$a = \begin{cases} 0.273 & \text{for} \quad D = 1, \\ 0.260 & \text{for} \quad D = 2, \\ 0.213 & \text{for} \quad D = 3, \end{cases} \qquad b = \begin{cases} 0.993 & \text{for} \quad D = 1, \\ 0.942 & \text{for} \quad D = 2, \\ 0.957 & \text{for} \quad D = 3, \end{cases} \tag{4.20}$$

so that approximating $a \approx 1/4$ and $b \approx 1$ for all D, we get the following simple approximate bound:

$$\frac{\alpha_A}{\|A\|} \lesssim \exp\left(\frac{N_C}{4}\right). \tag{4.21}$$

4.3 Condition number analysis

4.3.1 Lower bound

To lower bound the condition number κ_A of A, we need bounds on both $\|A\|$ and $\|A^{-1}\|$. In the previous section we have already derived a tight bound for $\|A\|$ in Eq. (4.15), so we just need to bound $\|A^{-1}\|$. We start by noting that, since A_F^{-1} has the same structure as A_H^{-1} (just appended with extra blocks), we have

$$\|A^{-1}\| \geq \|A_H^{-1}\|. \tag{4.22}$$

Now, let us introduce a normalized vector

$$|x_\theta\rangle = \frac{1}{\sqrt{T^\star + 1}} \sum_{t^\star=0}^{T^\star} e^{i\theta t^\star} |t^\star\rangle \otimes |\xi_\theta\rangle, \tag{4.23}$$

where $|\xi_\theta\rangle$ is the eigenvector of $\mathcal{SC}$ with eigenvalue $e^{i\theta}$ for some real θ (we will justify its existence below). We then have

$$A_H^{-1} |x_\theta\rangle = \frac{1}{\sqrt{T^\star + 1}} \sum_{r=0}^{T^\star} \sum_{c=0}^{r} \sum_{t^\star=0}^{T^\star} e^{i\theta t^\star} |r\rangle\langle c| \, t^\star\rangle \otimes (\mathcal{SC})^{r-c} |\xi_\theta\rangle \tag{4.24}$$

$$= \frac{1}{\sqrt{T^\star + 1}} \sum_{r=0}^{T^\star} \sum_{c=0}^{r} e^{i\theta r} |r\rangle \otimes |\xi_\theta\rangle \tag{4.25}$$

$$= \frac{1}{\sqrt{T^\star + 1}} \sum_{r=0}^{T^\star} e^{i\theta r} (r+1) |r\rangle \otimes |\xi_\theta\rangle, \tag{4.26}$$

which can be used to obtain the following lower bound:

$$\|A^{-1}\| \geq \|A_H^{-1}\| \geq \|A_H^{-1}\,|x\rangle\,\| = \frac{\sqrt{\sum_{r=0}^{T^\star}(r+1)^2}}{\sqrt{T^\star+1}} = \sqrt{\frac{(2+T^\star)(3+2T^\star)}{6}} \geq \frac{T^\star}{\sqrt{3}}. \tag{4.27}$$

Putting the above together with Eq. (4.15), we end up with

$$\kappa_A \geq \frac{\|\mathcal{C}\|T^\star}{\sqrt{3}}. \tag{4.28}$$

We now argue for the existence of an eigenvector $|\xi_\theta\rangle$ of $\mathcal{SC}$ with eigenvalue $e^{i\theta}$. Here, we will just consider the simpler case with no walls, when the streaming matrix S is given by Eq. (2.14), and we will show how to construct $|\xi_0\rangle$ (i.e., the eigenvector with eigenvalue 1). In Appendix G, we consider the harder case, when there are non-trivial boundaries with the streaming operator described by Eq. (2.22), and we construct the eigenstate for $\theta = \pi$, i.e., a state $|\xi_\pi\rangle$ that is a -1 eigenstate of $\mathcal{SC}$.

First, we set all Carleman blocks, except for the first one, to zero:

$$|\xi_0\rangle = [|\zeta_0\rangle, 0, \ldots, 0]. \tag{4.29}$$

Thus,

$$\mathcal{SC}\,|\xi_0\rangle = [S(I+F_1)\,|\zeta_0\rangle, 0, \ldots, 0]. \tag{4.30}$$

Next, choosing $|\zeta_0\rangle$ to be translationally invariant in space,

$$|\zeta_0\rangle = \frac{1}{\sqrt{N}}\sum_{\boldsymbol{r}^\star} |\boldsymbol{r}^\star\rangle \otimes |\psi_0\rangle, \tag{4.31}$$

to guarantee that $S\,|\zeta_0\rangle = |\zeta_0\rangle$. Finally, recalling that F_1 acts locally in space (see Eq. (2.16a)), we note that taking $|\psi_0\rangle$ to be the zero eigenstate of $\tilde{F}_1$ (its existence can be checked by direct calculation for $Q\times Q$ matrices), we get $(I+F_1)\,|\zeta_0\rangle = |\zeta_0\rangle$. We thus conclude that $|\xi_0\rangle$ constructed as described above is indeed an eigenstate of $\mathcal{SC}$ with eigenvalue 1.

4.3.2 Numerical evaluation

While a lower bound on the condition number scaling worse than the classical complexity of solving an LBE problem would rule out the possibility for a quantum advantage, an upper bound scaling better than classical would prove its existence. Thus, deriving such an upper bound is central to any claim of quantum advantage. In Appendix H, we explain how one can derive the following bound for κ_{A_H} and κ_{A_F}:

$$\kappa_{A_H} \leq (\|\mathcal{C}\|+1)\sqrt{\sum_{t^\star=0}^{T^\star}(T^\star - t^\star + 1)\|(\mathcal{SC})^{t^\star}\|^2}, \tag{4.32a}$$

$$\kappa_{A_F} \leq (\|\mathcal{C}\|+1)\sqrt{\sum_{t^\star=0}^{T^\star}[(2^W(T^\star+1)-t^\star)\|(\mathcal{SC})^{t^\star}\|^2] + 2^{2W-1}(T^\star+1)^2}. \tag{4.32b}$$

The problem with analyzing the above bounds is that one needs to estimate the values of $\|(\mathcal{SC})^{t^\star}\|$, which turns out to be a non-trivial problem. One can then resort to numerics and evaluate the necessary norms. It turns out, however, that it is not much more computationally demanding to directly calculate the condition number.

Thus, instead of bounding it, we evaluate the value of κ_{A_H} for $D=1$ and $D=2$ systems with Reynolds number $\mathrm{Re}\in[10,1000]$, and extrapolate to get the scaling behavior of κ_{A_H}. Note that κ_{A_H} is a lower bound for κ_{A_F}, so the query complexity upper bound for the final state system will be at least the one we find for the history state case. At this stage, we will use κ_{A_H} as an estimate for κ_{A_F}, but of course this could be improved upon. More precisely, we first approximate $\|A_H\|$ using the upper bound from Eq. (4.15) (recall that this approximation differs by at most 1 from the

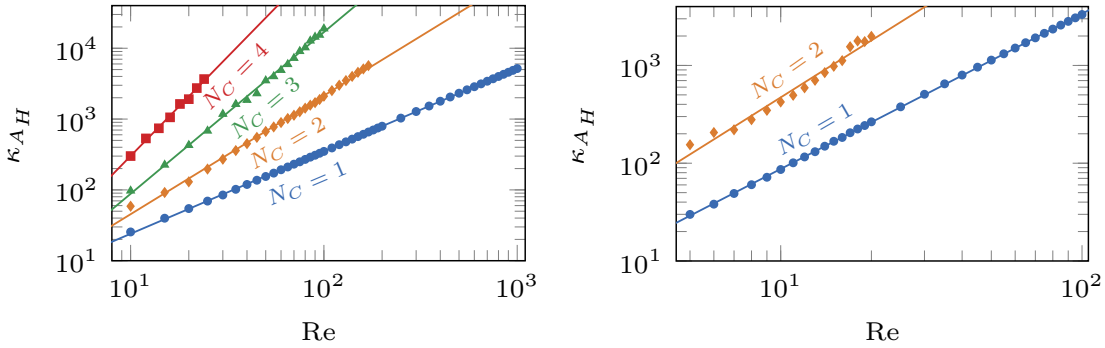


Figure 6: **Condition number.** Scaling of the condition number $\kappa(A_H)$ with the Reynolds number Re for D1Q3 model (left) and D2Q9 model (right), and different Carleman truncation orders N_C. The considered systems have periodic boundary conditions with no walls and $\beta = 3/4$.

real value of $\|A_H\|$). Then, we numerically evaluate $\|A_H^{-1}\|$, by using the Lanczos method to find the largest eigenvalue of $(A_H^{-1})^\dagger A_H^{-1}$, and taking its square root. As before, we consider the spatial discretization parameter $\beta = 3/4$ (resolving the Kolmogorov microscale), so that all the simulation parameters entering A_H^{-1} (i.e., N_x, $T^\star$, and $\bar{\tau}^\star$) become simple functions of the Reynolds number Re according to Eq. (4.5).

We present the dependence of κ_{A_H} on the Reynolds number using doubly logarithmic plots in Fig. 6. Using best linear fits to

$$\ln \kappa_{A_H} = \chi \ln \mathrm{Re} + \ln c, \tag{4.33}$$

one can approximate the condition number well by

$$\kappa_{A_H} \approx c\ \mathrm{Re}^{\chi}, \tag{4.34}$$

where parameters χ and c depend both on problem dimension D and the Carleman truncation order N_C:

$$\chi = \begin{cases} 1.167 & \text{for} \quad D=1,\ N_C=1, \\ 1.691 & \text{for} \quad D=1,\ N_C=2, \\ 2.283 & \text{for} \quad D=1,\ N_C=3, \\ 2.792 & \text{for} \quad D=1,\ N_C=4, \\ 1.588 & \text{for} \quad D=2,\ N_C=1, \\ 1.936 & \text{for} \quad D=2,\ N_C=2, \end{cases} \qquad c = \begin{cases} 1.635 & \text{for} \quad D=1,\ N_C=1, \\ 0.905 & \text{for} \quad D=1,\ N_C=2, \\ 0.459 & \text{for} \quad D=1,\ N_C=3, \\ 0.486 & \text{for} \quad D=1,\ N_C=4, \\ 2.254 & \text{for} \quad D=2,\ N_C=1, \\ 5.460 & \text{for} \quad D=2,\ N_C=2. \end{cases} \tag{4.35}$$

Note that, while the increase of χ with D could be expected and might be handled (after all, classical complexity also increases with D), the strong dependence on N_C is troublesome. That is because obtaining low error output may require a high value of N_C, which would then directly translate into complexity cost via the dependence of the condition number on Re^{χ}. Thus, while we postpone a detailed discussion on a possible quantum advantage to Sec. 5, we note here that it may be restricted to high error outputs.

4.4 Estimating query complexity

We now have all the necessary ingredients to bound the query complexity of running a quantum linear solver for our shifted lattice Boltzmann problem. First, let us briefly comment on the lower bound for the query complexity q_Q. It is known that q_Q can at best scale linearly with the condition number κ_A of the investigated linear system A [10]. As we analytically proved a lower bound on κ_A that scales linearly with the number of time steps (recall Eq. (4.28)), we then have

$$q_Q = O(T^\star). \tag{4.36}$$

Recalling the simulation parameter choice from Eq. (2.36), we conclude that at best (i.e., ignoring all other potential dependence on the Reynolds number not arising from the dependence on $T^\star$) we have:

$$q_Q = O\left(\mathrm{Re}^{\beta(D/2+1)}\right). \tag{4.37}$$

Thus, even ignoring the block-encoding cost, at best the complexity will scale with the Reynolds number as $\mathrm{Re}^{\beta(D/2+1)}$.

Now, let us proceed to the upper bound. We start by simplifying the bound on q_Q from Eq. (3.78), by slightly weakening it through the observation that for all $\epsilon_Q \geq 10^{-10}$ and all $\kappa \geq 1$, one has

$$1.05 \ln\left(\frac{\sqrt{1-\epsilon_Q^2}}{\epsilon_Q}\right) + \frac{2.78\ln(\kappa)^3 + 3.17}{\kappa} \leq 29.0. \tag{4.38}$$

Since the Carleman truncation errors are expected to be much larger than 10^{-10}, this assumption is justified, and so we get

$$q_Q \leq \frac{85.0\ \alpha_A \kappa_A}{\|A\|}. \tag{4.39}$$

Next, we upper bound the block-encoding prefactor using Eq. (4.21), to obtain

$$q_Q \lesssim 85.0 \exp\left(\frac{N_C}{4}\right) \kappa_A. \tag{4.40}$$

Then, focusing as before on $\beta = 3/4$, we approximate the condition number using Eq. (4.34) to finally arrive at

$$q_Q \lesssim 85.0c \exp\left(\frac{N_C}{4}\right) \mathrm{Re}^{\chi}, \tag{4.41}$$

where, as expected, the main factor comes from Re^{χ}, with χ depending N_C.

We also note that by rewriting

$$\exp\left(\frac{N_C}{4}\right) = [\exp(\Gamma N_C)]^{\frac{1}{4\Gamma}} \tag{4.42}$$

and using the error model from Eq. (4.8) under the assumption that we are operating below the threshold Reynolds number ($\mathrm{Re} \leq \mathrm{Re}_T$), so that $\Gamma < 0$, we get

$$\exp\left(\frac{N_C}{4}\right) = \left(\frac{E}{\epsilon_C}\right)^{\frac{1}{4|\Gamma|}}. \tag{4.43}$$

Plugging this back into the query complexity upper bound, we get

$$q_Q \lesssim 85.0c \left(\frac{E}{\epsilon_C}\right)^{\frac{1}{4|\Gamma|}} \mathrm{Re}^{\chi}, \tag{4.44}$$

where we note that χ depends on N_C, which in turn should be chosen as

$$N_C = \left\lceil \frac{1}{|\Gamma|} \ln \frac{E}{\epsilon_C} \right\rceil. \tag{4.45}$$

4.5 Estimating gate complexity

4.5.1 Preliminaries

In this section we aim at estimating the cost of a single query to U_{A_F} or U_{A_H} in terms of the number of T gates needed to construct quantum circuits implementing them. We denote the T-cost of implementing exactly a general unitary circuit U by $G[U]$, and the cost of implementing it up to error ϵ by $G[U^\epsilon]$. Moreover, a gate U controlled on k qubits will be denoted by c^kU. In what follows, we focus on estimating $G[U_{A_F}]$, which also acts as an upper bound for $G[U_{A_H}]$.

In the process, we will use the following estimates of T-costs based on known results on circuit compilation. First, we focus on estimating the cost of a general unitary U controlled on k qubits, as a function of the cost of implementing just U. As we explain in Appendix I, for $k \geq 2$ one can realize c^kU with $2k-2$ Toffoli gates, $k-1$ auxiliary qubits and a single cU gate. Moreover, since

each Toffoli gate can be replaced with 7 T gates [61, 62], we we can realize an arbitrary $c^k U$ with a T-count of

$$G[c^k U] = 14k - 14 + G[cU]. \tag{4.46}$$

Next, given a decomposition of U into Clifford and T gates, one can estimate the T-cost of cU in the following way. First, all the controlled versions of the single-qubit Clifford gates can be realized with CNOTs and Cliffords, so they do not contribute to $G[cU]$. Then, every CNOT appearing in the decomposition of U becomes a Toffoli in cU, and so contributes 7 T gates. Finally, every T gate in U becomes a controlled T in cU, and each such gate can be constructed from 9 T gates [61, 62]. Here, we will make a crude approximation by assuming that the number of CNOTs in the decomposition of U is roughly equal to the number of T gates. We thus approximate the T-cost of cU by

$$G[cU] \approx 16G[U]. \tag{4.47}$$

Consequently, for $k \geq 2$, we get

$$G[c^k U] \approx 14k - 14 + 16G[U]. \tag{4.48}$$

For the special case when $U = X$ is a single qubit NOT gate, we use a slightly better estimate based on Ref. [63]:

$$G[c^k X] = 8k - 12, \tag{4.49}$$

which requires a single extra ancilla. Moreover, we take the T-cost of a controlled Hadamard gate to be given by

$$G[cH] = 2, \tag{4.50}$$

due to a decomposition $cH = (I \otimes R_y(\pi/4))cZ(I \otimes R_y^\dagger(\pi/4))$ and $R_y(\pi/4) = SHTHS^\dagger$.

Now, we proceed to the estimates for the cost of implementing an n-qubit unitary U. For $n = 1$, the cost of implementing a single qubit rotation $R_{\hat{n}}(\theta)$ around axis $\hat{n} \in \{x, y, z\}$ up to error ϵ is given by [64]:

$$G[R_{\hat{n}}^\epsilon(\theta)] \approx 3 \log \frac{1}{\epsilon}. \tag{4.51}$$

We choose equal error ϵ on all approximate single-qubit gates used, and assume that the total error arising from gates implemented up to ϵ_1 and ϵ_2 is $\epsilon_1 + \epsilon_2$. For $n \geq 2$, we explain in Appendix I how to estimate the T-cost for synthesizing U in terms of the T-costs $G[\mathrm{PREP}^{\epsilon_j}(u_j)]$, where $\mathrm{PREP}^{\epsilon_j}(u_j)$ is a state preparation of the j-th column of U to error ϵ_j. For an n-qubit unitary U with d columns u_j specified, this estimate is given by

$$G[U^\epsilon] = 32 \sum_{j=1}^{d} G[\mathrm{PREP}^{\epsilon_j}(u_j)] + d(8n - 12), \tag{4.52}$$

where the resulting error ϵ depends on ϵ_j as

$$\epsilon = \sqrt{2} \sum_{j=1}^{d} \epsilon_j. \tag{4.53}$$

Note that such unitary state preparations can be constructed from single-qubit rotations, and that a uniform superposition over 2^k states can be prepared using only Clifford gates. Moreover, a state preparation unitary $\mathrm{PREP}(\psi)$ of a general n-qubit state $|\psi\rangle$ can be realized using roughly 2^{n+1} single qubit rotations [65], and so the cost of implementing it up to error $2^{n+1}\epsilon$ can be approximated by

$$G[\mathrm{PREP}^{2^{n+1}\epsilon}(\psi)] \approx 3 \cdot 2^{n+1} \log \frac{1}{\epsilon}. \tag{4.54}$$

Finally, we assume that the T-cost of a quantum adder acting on n qubits is given by [66]

$$G[\mathrm{ADD}(n)] \approx 4n. \tag{4.55}$$

4.5.2 Gate cost of the linear system circuit

We start by using the gate decomposition of U_{A_F} presented in Eq. (3.39), and employing Eq. (3.40), to get

$$G[U_{A_F}] = G[cU_\Delta] + G[c^2\mathcal{S}] + G[c^2U_\mathcal{C}] + 2G[V_{\alpha_\mathcal{C}}] + G[cV_{\alpha_\mathcal{C}^2-1}] + G[c^{W+1}V_{\alpha_\mathcal{C}^2-1}]. \tag{4.56}$$

All the state preparation unitaries V are single-qubit y rotations, and so implementing them up to error ϵ has the following cost:

$$G[V^\epsilon_{\alpha_\mathcal{C}}] \approx 3\log\frac{1}{\epsilon}, \tag{4.57a}$$

$$G[cV^\epsilon_{\alpha_\mathcal{C}^2-1}] \approx 16G[V^\epsilon_{\alpha_\mathcal{C}^2-1}] \approx 48\log\frac{1}{\epsilon}, \tag{4.57b}$$

$$G[c^{W+1}V^\epsilon_{\alpha_\mathcal{C}^2-1}] \approx 14W + G[cV^\epsilon_{\alpha_\mathcal{C}^2-1}] \approx 14W + 48\log\frac{1}{\epsilon}. \tag{4.57c}$$

Then, from Eq. (3.30), we have

$$G[cU_\Delta] \approx 16G[\mathrm{ADD}(\log(T^\star+1)+W+1)] \approx 64(\log T^\star + W + 1). \tag{4.58}$$

Next, from Eqs. (3.31) and (3.32), we get

$$G[c^2\mathcal{S}] = N_C G[c^2 S] = 2DN_C G[c^2\mathrm{ADD}(\log N_x)] \approx 2DN_C(14 + 16G[\mathrm{ADD}(\log N_x)]) \tag{4.59}$$

$$\approx 2DN_C(14 + 64\log N_x). \tag{4.60}$$

Putting it together, we get

$$G[U^{4\epsilon+\epsilon_\mathcal{C}}_{A_F}] \approx G[c^2U^{\epsilon_\mathcal{C}}_\mathcal{C}] + 78W + 64 + 102\log\frac{1}{\epsilon} + 64\log T^\star + 2DN_C(14 + 64\log N_x). \tag{4.61}$$

Moreover, recalling the choice of simulation parameters, N_x and $T^\star$, from Eq. (2.36) and restricting to the case $\beta = 3/4$ (allowing one to resolve the Kolmogorov microscale), the above can be expressed as a function of the Reynolds number:

$$G[U^{4\epsilon+\epsilon_\mathcal{C}}_{A_F}] \approx G[c^2U^{\epsilon_\mathcal{C}}_\mathcal{C}] + 78W + 28DN_C + 64 + 102\log\frac{1}{\epsilon} + 24(D(1+4N_C)+2)\log \mathrm{Re}. \tag{4.62}$$

4.5.3 Gate cost of the Carleman collision circuit

We are now facing the central problem of estimating the T-cost of $c^2U_\mathcal{C}$. First, let us introduce some shorthand notation:

$$n_B := \lceil \log\lfloor N_C/2\rfloor\rceil, \qquad n_C = \lceil \log N_C\rceil, \qquad n_\Pi = \left\lceil \log\binom{k}{l}\right\rceil. \tag{4.63}$$

From Eq. (3.47) we have

$$G[c^2U_\mathcal{C}] = 2G[V_B] + \sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor} G[c^{2+n_B}U_{B_l}]. \tag{4.64}$$

Since V_B is an n_B-qubit unitary, its approximation has the following T-cost according to Eq. (4.54):

$$G[V_B^{N_C\epsilon}] \approx 3N_C\log\frac{1}{\epsilon}. \tag{4.65}$$

Then, from Eq. (3.51), we see that

$$G[c^{2+n_B}U_{B_l}] = G[c^{2+n_B}U_{(\Delta^\dagger)^l}] + \sum_{k=l'}^{N_C-l}\left(G\left[c^{2+n_B+n_C}V_{C^k_{k+l}}\right] + G\left[c^{2+n_B+n_C}U_{C^k_{k+l}}\right]\right). \tag{4.66}$$

The first term can be easily estimated to be

$$G[c^{2+n_B}U_{(\Delta^\dagger)^l}] \approx 14+14n_B+G[c(\Delta^\dagger)^l] \approx 14+14n_B+16G[\mathrm{ADD}(n_C)] \approx 14+14n_B+64n_C. \quad (4.67)$$

As $V_{C^k_{k+l}}$ is a single qubit unitary, we get the second term via

$$G\left[c^{2+n_B+n_C}V^\epsilon_{C^k_{k+l}}\right] \approx 14+14(n_B+n_C)+G\left[cV^\epsilon_{C^k_{k+l}}\right] \approx 14+14(n_B+n_C)+48\log\frac{1}{\epsilon}. \quad (4.68)$$

Turning to the last term of Eq. (4.66), we use Eq. (3.56) to get

$$G[c^{2+n_B+n_C}U_{C^k_{k+l}}] = 2G[V_\Pi] + 2\sum_{i=1}^{\binom{k}{l}} G[c^{2+n_B+n_C+n_\Pi}\Pi_i] + G[c^{2+n_B+n_C}U_{F(k,l)}]. \quad (4.69)$$

Since V_Π is a uniform state preparation, we ignore its cost, as it will be negligible. We estimate the cost of each multiply-controlled Π_i as

$$G[c^{2+n_B+n_C+n_\Pi}\Pi_i] \approx 14+14(n_B+n_C+n_\Pi)+G[c\Pi_i] \approx 14+14(n_B+n_C+n_\Pi), \quad (4.70)$$

where we made a crude assumption that controlled permutations have zero T-cost. The cost of $c^{2+n_B+n_C}F(k,l)$ can be approximated by

$$G[c^{2+n_B+n_C}U_{F(k,l)}] \approx 14+14(n_B+n_C)+G[cU_{F(k,l)}] \approx 14+14(n_B+n_C)+16G[U_{F(k,l)}] \quad (4.71)$$

$$\approx 14+14(n_B+n_C)+16(k-l)G[U_{I+F_1}]+16lG[U_{\bar{F}_2}]. \quad (4.72)$$

Putting things together, we get:

$$G[c^2U^{\epsilon_\mathcal{C}}_\mathcal{C}] = 6N_C\log\frac{1}{\epsilon} + \sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}(14+14n_B+64n_C) \quad (4.73)$$

$$+\sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}\sum_{k=l'}^{N_C-l}\left(14+14(n_B+n_C)+48\log\frac{1}{\epsilon}\right) \quad (4.74)$$

$$+2\sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}\sum_{k=l'}^{N_C-l}\sum_{i=1}^{\binom{k}{l}}(14+14(n_B+n_C+n_\Pi)) \quad (4.75)$$

$$+\sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}\sum_{k=l'}^{N_C-l}\left(14+14(n_B+n_C)+16(k-l)G[U^{\epsilon_{F_1}}_{I+F_1}]+16lG[U^{\epsilon_{F_2}}_{\bar{F}_2}]\right), \quad (4.76)$$

where $\epsilon_\mathcal{C}$ is just the sum over errors of all ϵ-approximate single-qubit gates used so far, together with the errors introduced while compiling U_{I+F_1} and $U_{\bar{F}_2}$ (we will discuss these in the next section):

$$\epsilon_\mathcal{C} = 2N_C\epsilon + \sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}\sum_{k=l'}^{N_C-l}[\epsilon+(k-l)\epsilon_{F_1}+l\epsilon_{F_2}] \quad (4.77)$$

$$= \frac{N_C\left[6(12+N_C)\epsilon+(N_C+2)((2N_C+5)\epsilon_{F_1}+(N_C+1)\epsilon_{F_2})\right]}{24}. \quad (4.78)$$

Most of the sums can be explicitly calculated to yield

$$G[c^2U^{\epsilon_\mathcal{C}}_\mathcal{C}] \approx 6N_C\left(n_B+1+\log\frac{1}{\epsilon}\right)+\frac{N_C+2}{2}(14+14n_B+64n_C) \quad (4.79)$$

$$+\frac{N_C^2+4N_C}{4}\left(14+14(n_B+n_C)+48\log\frac{1}{\epsilon}\right) \quad (4.80)$$

$$+\left[\left(2+\frac{4}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{N_C}-4\right](14+14(n_B+n_C))+2h_{N_C} \tag{4.81}$$

$$+\frac{N_C^2+4N_C}{4}(14+14(n_B+n_C)) \tag{4.82}$$

$$+\frac{2N_C(N_C+2)}{3}\left[(2N_C+5)G[U_{I+F_1}^{\epsilon_{F_1}}]+(N_C+1)G[U_{\bar{F}_2}^{\epsilon_{F_2}}]\right], \tag{4.83}$$

where we used

$$\sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}\sum_{k=l'}^{N_C-l}\binom{k}{l}=\left(1+\frac{2}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^{N_C}-2, \tag{4.84}$$

and introduced

$$h_{N_C}:=\sum_{l=0}^{\lfloor\frac{N_C}{2}\rfloor}\sum_{k=l'}^{N_C-l}\binom{k}{l}\log\binom{k}{l}. \tag{4.85}$$

4.5.4 Gate cost of the collision circuits

To estimate the T gate cost of compiling U_{I+F_1} and $U_{\bar{F}_2}$, we will consider separately the cases of dimension $D=1$ and $D=2$, and leave $D=3$ case for future work.

Case $D=1$. Performing the SVD of $I+\tilde{F}_1=L_1\Sigma_1R_1^\dagger$, one finds that not only Σ_1 depends on $\bar{\tau}^\star$, but also the unitaries L_1 and R_1 do. However, independently of the specific value of $\bar{\tau}^\star$, the columns of L_1 and R_1 have a very specific structure. Namely, up to permutations, we have:

$$\text{2 columns with pattern}: \quad (a,a,b) \tag{4.86a}$$

$$\text{1 column with pattern}: \quad (1/\sqrt{2},-1/\sqrt{2},0), \tag{4.86b}$$

where a,b are real numbers. Thus, we can estimate the T-cost of L_1 and R_1 using Eq. (4.52) and the costs of state preparation for the above states. Clearly, the last state can be prepared with Clifford gates only. Moreover, the first two states each require just one general single-qubit rotation and one controlled Hadamard gate. Hence:

$$G[L_1^{2\sqrt{2}\epsilon}]\approx 32\left(2\cdot(G[R_y^\epsilon(\theta)]+2)\right)+12\approx 140+192\log\frac{1}{\epsilon}, \tag{4.87}$$

and the same cost for R_1.

Next, Σ_1 can be found to be

$$\Sigma_1=\operatorname{diag}(1,\sigma_1,\sigma_2). \tag{4.88}$$

From Eq. (3.63), we see that this then reduces to three single-qubit unitaries, V_{σ_1}, V_{σ_2}, and V_{σ_3}, controlled on two qubits. By noticing that $V_{\sigma_1}=I$ and that encoding never uses the basis state $|11\rangle$, it can be reduced to two single-qubit unitaries controlled on a single-qubit. Thus, we get

$$G[U_{\Sigma_1}^{2\epsilon}]=2G[cR_y^\epsilon(\theta)]\approx 96\log\frac{1}{\epsilon}. \tag{4.89}$$

Putting it together with the costs for L_1 and R_1, we get:

$$G[U_{I+F_1}^{\epsilon_{F_1}}]\approx 280+480\log\frac{1}{\epsilon}, \tag{4.90}$$

with

$$\epsilon_{F_1}=(4\sqrt{2}+2)\epsilon. \tag{4.91}$$

Now, we find the following HOSVD of $\tilde{F}_2=L_2\Sigma_2(R_2\otimes R_2)^\dagger$:

$$L_2=\begin{pmatrix}\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}}\\ -\frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}}\\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}}\end{pmatrix},\quad R_2=\begin{pmatrix}\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}}\\ 0 & 1 & 0\\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}}\end{pmatrix}, \tag{4.92}$$

and Σ_2 has a single non-zero element $(\Sigma_2)_{11} = \sqrt{6}/\bar{\tau}^\star$. Thus, it is actually an SVD and $(\Sigma_2)_{11}$ is the largest singular value of $\tilde{F}_2$. The right unitary R_2 can be compiled using only Clifford gates, whereas the T-cost of the left can be estimated using Eq. (4.52) to be

$$G[L_2^{\sqrt{2}(\epsilon_1+\epsilon_2+\epsilon_3)}] \approx 32 \sum_{j=1}^{3} G[L_{2,j}^{\epsilon_j}] + 12, \tag{4.93}$$

where $L_{2,j}$ is a unitary state preparation of the j-th column of L_2. Since the second column can be prepared using Clifford gates, whereas the first and third column require each just one single-qubit gate and one controlled Hadamard gate, we conclude that

$$G[L_2^{2\sqrt{2}\epsilon}] \approx 64(G[R_y^\epsilon(\theta)] + 2) + 12 \approx 192 \log \frac{1}{\epsilon} + 140. \tag{4.94}$$

Moreover, block-encoding of Σ_2 requires one single-qubit unitary gate controlled on two qubits, meaning that

$$G[U_{\Sigma_2}^\epsilon] \approx G[c^2 R_y^\epsilon(\theta)] \approx 14 + 48 \log \frac{1}{\epsilon}. \tag{4.95}$$

We thus conclude that

$$G[U_{\tilde{F}_2}^{\epsilon_{F_2}}] \approx 154 + 240 \log \frac{1}{\epsilon} \tag{4.96}$$

where

$$\epsilon_{F_2} = (1 + 2\sqrt{2})\epsilon. \tag{4.97}$$

Case $D = 2$. Performing the SVD of $I + \tilde{F}_1 = L_1 \Sigma_1 R_1^\dagger$, one finds that, up to permutations, the columns of L_1 and R_1 have the following structure:

$$\text{2 columns with pattern :} \quad (a, a, a, a, b, b, b, b, c) \tag{4.98a}$$
$$\text{4 columns with pattern :} \quad (a, -a, b, -b, c, -c, d, -d, 0), \tag{4.98b}$$
$$\text{3 column with pattern :} \quad (a, a, b, b, c, c, d, d, e), \tag{4.98c}$$

with a, b, c, d, e real. As before, we can estimate the T-cost of L_1 and R_1 using Eq. (4.52) and the costs of state preparation for the above states. The first two states can be prepared with a single qubit rotation, followed by a single qubit rotation controlled on a single qubit, followed by two controlled Hadamard gates. The next four states can be prepared with a single qubit rotation, followed by two single qubit rotations controlled on a single qubit, followed by Clifford gates. Finally, the last three states can be obtained by a single qubit rotation, followed by a single qubit rotation controlled on a single qubit, followed by two single qubit rotations controlled on two qubits, followed by a controlled Hadamard. Hence:

$$G[L_1^{28\sqrt{2}\epsilon}] \approx 32 \left(2(G[R_y^\epsilon(\theta)] + G[cR_y^\epsilon(\theta)] + 4) + 4(G[R_y^\epsilon(\theta)] + 2G[cR_y^\epsilon(\theta)]\right. \tag{4.99}$$
$$\left. + 3(G[R_y^\epsilon(\theta)] + G[cR_y^\epsilon(\theta))] + 2G[c^2 R_y^\epsilon(\theta)] + 2)\right) + 180 \tag{4.100}$$
$$\approx 3316 + 30048 \log \frac{1}{\epsilon}, \tag{4.101}$$

and the same cost for R_1.

Next, Σ_1 can be found to be

$$\Sigma_1 = \text{diag}\left(\sigma_1, \sigma_1, \sigma_1, \sigma_2, \sigma_2, \sigma_3, \sigma_3, \sigma_4, \sigma_5\right), \tag{4.102}$$

From Eq. (3.63), we see that this then can be achieved by nine single-qubit unitaries, $V_{\sigma_1}, \ldots, V_{\sigma_9}$, controlled on four qubits. By noticing that the first three rotations are the same, and there are also two equal pairs, this can be reduced to one uncontrolled single-qubit unitary rotation, two single-qubit unitaries controlled on a three qubits, and two single-qubit unitaries controlled on four qubits. Thus, we get

$$G[U_{\Sigma_1}^{5\epsilon}] = G[R_y^\epsilon(\theta)] + 2G[c^3 R_y^\epsilon(\theta)] + 2G[c^4 R_y^\epsilon(\theta)] \approx 140 + 195 \log \frac{1}{\epsilon}. \tag{4.103}$$

Putting it together with the costs for L_1 and R_1, we get:

$$G[U_{I+F_1}^{\epsilon_{F_1}}] \approx 6412 + 60291 \log \frac{1}{\epsilon}, \tag{4.104}$$

with

$$\epsilon_{F_1} = (28\sqrt{2} + 5)\epsilon. \tag{4.105}$$

Next, we find the following HOSVD of $\tilde{F}_2 = L_2 \Sigma_2 (R_2 \otimes R_2)^\dagger$:

$$L_2 = \begin{pmatrix}
-\frac{1}{3}\left(2\sqrt{2}\right) & 0 & 0 & 0 & \frac{1}{5} & \frac{2\sqrt{\frac{2}{17}}}{5} & \frac{4}{\sqrt{595}} & \frac{4}{\sqrt{1855}} & \frac{2\sqrt{\frac{2}{53}}}{3} \\
\frac{1}{6\sqrt{2}} & -\frac{1}{2} & 0 & 0 & 0 & 0 & \sqrt{\frac{17}{35}} & \frac{17}{\sqrt{1855}} & -\frac{19}{6\sqrt{106}} \\
\frac{1}{6\sqrt{2}} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{\frac{53}{2}}}{6} \\
\frac{1}{6\sqrt{2}} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \sqrt{\frac{35}{53}} & \frac{17}{6\sqrt{106}} \\
\frac{1}{6\sqrt{2}} & \frac{1}{2} & 0 & 0 & 0 & 0 & \sqrt{\frac{17}{35}} & -\frac{18}{\sqrt{1855}} & \frac{17}{6\sqrt{106}} \\
\frac{1}{6\sqrt{2}} & 0 & \frac{1}{2} & -\frac{1}{\sqrt{2}} & \frac{2}{5} & \frac{4\sqrt{\frac{2}{17}}}{5} & -\frac{1}{2\sqrt{595}} & -\frac{1}{2\sqrt{1855}} & -\frac{1}{6\sqrt{106}} \\
\frac{1}{6\sqrt{2}} & 0 & -\frac{1}{2} & 0 & 0 & \frac{5}{\sqrt{34}} & -\frac{1}{2\sqrt{595}} & -\frac{1}{2\sqrt{1855}} & -\frac{1}{6\sqrt{106}} \\
\frac{1}{6\sqrt{2}} & 0 & -\frac{1}{2} & 0 & \frac{4}{5} & -\frac{9}{5\sqrt{34}} & -\frac{1}{2\sqrt{595}} & -\frac{1}{2\sqrt{1855}} & -\frac{1}{6\sqrt{106}} \\
\frac{1}{6\sqrt{2}} & 0 & \frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{2}{5} & \frac{4\sqrt{\frac{2}{17}}}{5} & -\frac{1}{2\sqrt{595}} & -\frac{1}{2\sqrt{1855}} & -\frac{1}{6\sqrt{106}}
\end{pmatrix}, \tag{4.106}$$

$$R_2 = \begin{pmatrix}
0 & -\frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & -\frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{3}} & 0 & 0 & \frac{1}{\sqrt{3}} \\
0 & -\frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & -\frac{1}{2\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\
0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{3}} \\
0 & \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} & 0 & 0 & 0 & \frac{1}{\sqrt{3}} & 0 \\
0 & -\frac{1}{\sqrt{15}} & 0 & \frac{1}{\sqrt{15}} & 0 & 0 & \sqrt{\frac{3}{5}} & \frac{2}{\sqrt{15}} & 0 \\
0 & -\frac{1}{\sqrt{15}} & 0 & -\frac{1}{\sqrt{15}} & 0 & \sqrt{\frac{3}{5}} & 0 & 0 & \frac{2}{\sqrt{15}} \\
0 & 0 & 0 & \frac{1}{\sqrt{30}} & \sqrt{\frac{5}{6}} & \frac{1}{\sqrt{30}} & -\frac{1}{\sqrt{30}} & \frac{1}{\sqrt{30}} & -\frac{1}{\sqrt{30}} \\
0 & \frac{1}{\sqrt{30}} & \sqrt{\frac{5}{6}} & 0 & 0 & \frac{1}{\sqrt{30}} & \frac{1}{\sqrt{30}} & -\frac{1}{\sqrt{30}} & -\frac{1}{\sqrt{30}} \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}, \tag{4.107}$$

and Σ_2 is a rectangular matrix with only nonzero elements being:

$$(\Sigma_2)_{1,1} = (\Sigma_2)_{1,11} = \frac{3\sqrt{3}}{\bar{\tau}^\star}, \quad (\Sigma_2)_{2,2} = (\Sigma_2)_{1,10} = -\frac{3}{\bar{\tau}^\star}, \quad (\Sigma_2)_{3,1} = \frac{3}{2\bar{\tau}^\star} = -(\Sigma_2)_{3,11}. \tag{4.108}$$

We then need to estimate the cost of the state preparations corresponding to each column of L_2. We summarize the results in Table 2, where USP_n denotes a unitary state preparation over n states. Using Eq. (4.52), it follows that the state preparation cost for the $n = 4$ qubit unitary L_2 is

$$G[L_2^{19\sqrt{2}\epsilon}] = 32 \times 57 \log \frac{1}{\epsilon} + 9(8 \times 4 - 12) = 180 + 1824 \log \frac{1}{\epsilon}. \tag{4.109}$$

Now, let us do the same for R_2 (for convenience we analyze $R_2^\dagger$). We summarize the results in Table 3. The overall cost for $R_2 \otimes R_2$ is then

$$G[R_2^{14\sqrt{2}\epsilon} \otimes R_2^{14\sqrt{2}\epsilon}] = 2\left(32 \times 42 \log \frac{1}{\epsilon} + 9(8 \times 4 - 12)\right) = 360 + 2688 \log \frac{1}{\epsilon}. \tag{4.110}$$

We still need to block-encode Σ_2. To achieve this, consider the following 90×90 unitary:

$$U_{\Sigma_2} = \begin{bmatrix} \Sigma_2 & \sqrt{I_9 - \Sigma_2 \Sigma_2^\dagger} \\ \sqrt{I_{81} - \Sigma_2^\dagger \Sigma_2} & -\Sigma_2^\dagger. \end{bmatrix} \tag{4.111}$$

By direct inspection, U_{Σ_2} is an optimal block-encoding of Σ_2 (the block-encoding prefactor is 6, which equals $\|\Sigma_2\|$). For simplicity, let us consider the matrix $U_{\Sigma_2}^\dagger$. Now, we need to estimate the

cost of the state preparations corresponding to each column. We summarize the results in Table 4. The overall cost for U_{Σ_2} is then:

$$G[U_{\Sigma_2}^{8\sqrt{2}\epsilon}] = 32 \times 24 \log \frac{1}{\epsilon} + 90(8 \times 7 - 12) = 3960 + 768 \log \frac{1}{\epsilon}. \tag{4.112}$$

Combining the results, we thus conclude that T-cost of a block-encoding of $\tilde{F}_2$ is given by

$$G[U_{\tilde{F}_2}^{\epsilon_{F_2}}] = 4500 + 5280 \log \frac{1}{\epsilon}, \tag{4.113}$$

with

$$\epsilon_{F_2} = 59\sqrt{2}\epsilon. \tag{4.114}$$

4.5.5 Final gate cost

Combining Eqs. (4.62) and (4.79), we arrive at the final T-gate cost of approximating U_{A_F} up to error ϵ_{tot}:

$$\begin{aligned}
G[U_{A_F}^{\epsilon_{\text{tot}}}] \approx\ & 78W + 28DN_C + 64 + 102 \log \frac{1}{\epsilon} + 24(D(1+4N_C)+2) \log \text{Re} && (4.115)\\
& + 6N_C \left(n_B + 1 + \log \frac{1}{\epsilon} \right) + \frac{N_C + 2}{2}(14 + 14n_B + 64n_C) && (4.116)\\
& + \frac{N_C^2 + 4N_C}{4} \left(28 + 28(n_B + n_C) + 48 \log \frac{1}{\epsilon} \right) && (4.117)\\
& + \left[\left(2 + \frac{4}{\sqrt{5}} \right) \left(\frac{1+\sqrt{5}}{2} \right)^{N_C} - 4 \right] (14 + 14(n_B + n_C)) + 2h_{N_C} && (4.118)\\
& + \frac{2N_C(N_C+2)}{3} [(2N_C + 5)G[U_{I+F_1}^{\epsilon_{F_1}}] + (N_C + 1)G[U_{\tilde{F}_2}^{\epsilon_{F_2}}]], && (4.119)
\end{aligned}$$

where

$$\epsilon_{\text{tot}} = 4\epsilon + \frac{N_C \left[6(12 + N_C)\epsilon + (N_C + 2)((2N_C + 5)\epsilon_{F_1} + (N_C + 1)\epsilon_{F_2}) \right]}{24}, \tag{4.120}$$

whereas ϵ_{F_1} and ϵ_{F_2} depend on the problem dimension in the following way:

$$\epsilon_{F_1} = \begin{cases} (4\sqrt{2} + 2)\epsilon & \text{for} \quad D = 1, \\ (28\sqrt{2} + 5)\epsilon & \text{for} \quad D = 2, \end{cases} \qquad \epsilon_{F_2} = \begin{cases} (1 + 2\sqrt{2})\epsilon & \text{for} \quad D = 1, \\ 59\sqrt{2}\epsilon & \text{for} \quad D = 2. \end{cases} \tag{4.121}$$

Column	Operations	Cost
1	1 single-qubit y-rotation, 1 USP_8	$3 \log \frac{1}{\epsilon}$
2	USP_4, (-1) phases	0
3	USP_4, (-1) phases	0
4	USP_2, (-1) phases	0
5	3 single-qubit y-rotations	$9 \log \frac{1}{\epsilon}$
6	4 single-qubit y-rotations	$12 \log \frac{1}{\epsilon}$
7	3 single-qubit y-rotations, USP_4	$9 \log \frac{1}{\epsilon}$
8	4 single-qubit y-rotations, USP_4	$12 \log \frac{1}{\epsilon}$
9	4 single-qubit y-rotations, USP_2	$12 \log \frac{1}{\epsilon}$
Total	19 single-qubit y-rotations	$57 \log \frac{1}{\epsilon}$

Table 2: State preparation costs for the columns of L_2. Here ϵ is the per-gate error.

Column	Operations	Cost
1	2 single-qubit y-rotations, 1 USP_4	$6\log\frac{1}{\epsilon}$
2	2 single-qubit y-rotations, 1 USP_4, (-1) phases	$6\log\frac{1}{\epsilon}$
3	USP_3	$3\log\frac{1}{\epsilon}$
4	USP_3, (-1) phases	$3\log\frac{1}{\epsilon}$
5	2 single-qubit y-rotations, Hadamard	$6\log\frac{1}{\epsilon}$
6	2 single-qubit y-rotations, Hadamard	$6\log\frac{1}{\epsilon}$
7	3 single-qubit y-rotations, USP_4	$9\log\frac{1}{\epsilon}$
8	3 single-qubit y-rotations, USP_4	$9\log\frac{1}{\epsilon}$
9	permutation	0
Total	14 single-qubit y-rotations, USP_3	$45\log\frac{1}{\epsilon}$

Table 3: State preparation costs for the columns of $R_2^\dagger$. We cost USP_3 as $3\log_3(1/\epsilon)$, as it can be realized via a single qubit y rotation followed by a controlled Hadamard. Here ϵ is the per-gate error.

Column	Operations	Cost
1	Hadamard	0
2	2 Hadamards, (-1) phases	0
3	1 single-qubit y-rotation, Hamadard, (-1) phases	$3\log\frac{1}{\epsilon}$
4-9	permutation	0
10	2 single-qubit y-rotations, Hadamard, (-1) phases	$6\log\frac{1}{\epsilon}$
11	2 single-qubit y-rotations, permutation, (-1) phases	$6\log\frac{1}{\epsilon}$
12-18	permutation	0
19	2 single-qubit y-rotations, permutation, (-1) phases	$6\log\frac{1}{\epsilon}$
20	1 single-qubit y-rotation, permutation, (-1) phases	$3\log\frac{1}{\epsilon}$
21-90	permutation	0
Total	8 single-qubit y-rotations	$24\log\frac{1}{\epsilon}$

Table 4: State preparation costs for the columns of U_{Σ_2}. Here ϵ is the per-gate error.

while $G[U_{I+F_1}^{\epsilon_{F_1}}]$ and $G[U_{\bar{F}_2}^{\epsilon_{F_2}}]$ are given by

$$G[U_{I+F_1}^{\epsilon_{F_1}}] = \begin{cases} 280 + 480\log\frac{1}{\epsilon} & \text{for} \quad D = 1, \\ 6412 + 60291\log\frac{1}{\epsilon} & \text{for} \quad D = 2, \end{cases} \tag{4.122a}$$

$$G[U_{\bar{F}_2}^{\epsilon_{F_2}}] = \begin{cases} 154 + 240\log\frac{1}{\epsilon} & \text{for} \quad D = 1, \\ 4500 + 5280\log\frac{1}{\epsilon} & \text{for} \quad D = 2. \end{cases} \tag{4.122b}$$

Although we took care to cost all elements of the circuit, it turns out that the bulk of the cost comes from the last term in Eq. (4.119). For example, for $D = 2$, $N_C = 2$, $\epsilon = 10^{-6}$, $W = 10$ and $\mathrm{Re} \in (0, 10^{10}]$, it constitutes more than 99.9% of the total cost, and this fraction only increases for higher Carleman truncation orders N_C and lower approximation error ϵ. Thus, we can use a much simpler estimate for the total T-gate cost of our circuit given by Eq. (4.119), resulting in the

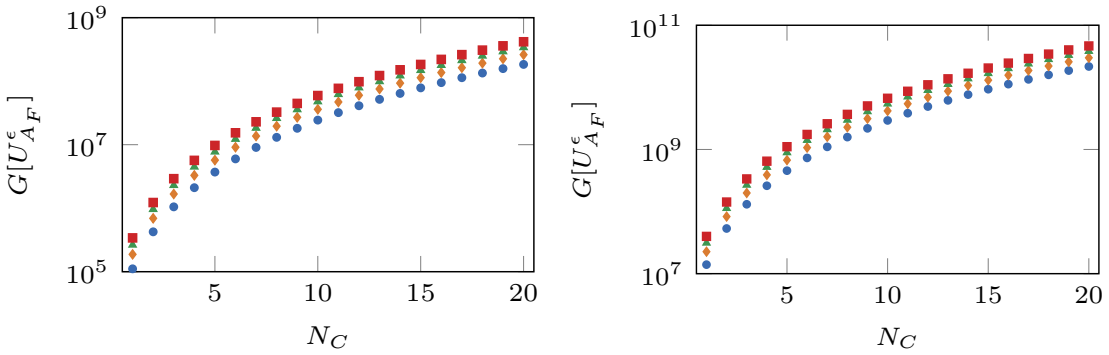


Figure 7: **T-gate cost of $U^\epsilon_{A_F}$.** The estimates for the number $G[U^\epsilon_{A_F}]$ of T gates needed to prepare a circuit ϵ-close to a unitary U_{A_F} encoding the LBE problem with dimension $D = 1$ (left) and $D = 2$ (right) as function of Carleman truncation order N_C. Different symbols, bottom to top, correspond to the allowed error level $\epsilon = 10^{-3k}$ for $k \in \{1, 2, 3, 4\}$. Note that the dependence of $G[U^\epsilon_{A_F}]$ on the Reynolds number Re is only logarithmic, making the presented data applicable with a very good approximation to all $\mathrm{Re} \geq 1$.

following T-gate cost of U_{A_F}:

$$G[U^\epsilon_{A_F}] \approx \begin{cases} N_C(N_C+2)\left[(800N_C+1760)\log\frac{K_1}{\epsilon} + (476N_C+1036)\right] & \text{for} \quad D=1, \\ N_C(N_C+2)\left[(83908N_C+204490)\log\frac{K_2}{\epsilon} + \frac{8}{3}(4331N_C+9140)\right] & \text{for} \quad D=2, \end{cases} \tag{4.123}$$

with

$$K_1 = \frac{1}{24}\left[96 + (94+44\sqrt{2})N_C + (27+42\sqrt{2})N_C^2 + (5+10\sqrt{2})N_C^3\right], \tag{4.124a}$$

$$K_2 = \frac{1}{24}\left[96 + (122+398\sqrt{2})N_C + (51+429\sqrt{2})N_C^2 + (10+115\sqrt{2})N_C^3\right]. \tag{4.124b}$$

We present the T-gate cost dependence on the truncation order N_C and approximation level ϵ for both cases $D = 1$ and $D = 2$ in Fig. 7.

It is thus clear that the T-gate cost $G[U^\epsilon_{A_F}]$ is directly almost independent of the Reynolds number Re or the parameter W controlling the success probability of getting the Carleman state at the final time $T^\star$. Instead, it has a cubic dependence on the Carleman truncation error N_C and logarithmic dependence on the inverse of the gate approximation error ϵ (note that this also holds for the case $D = 3$). Thus, if we decide to run our algorithm for a fixed N_C and ϵ, the difference between the query cost q_Q and the total T-gate cost $q_Q G[U^\epsilon_{A_F}]$ becomes a constant factor overhead. For example, for low $N_C \leq 5$ and error $\epsilon = 10^{-6}$, it is roughly 10^6 for $D = 1$ and 10^8 for $D = 2$, so one can expect it would be around 10^{10} for $D = 3$. Note, however, that these are just rough estimates that do not include any compilation optimization, and so one can expect that the actual numbers can be a few orders of magnitude better.

Finally, assuming we operate below the threshold Reynolds number ($\mathrm{Re} \leq \mathrm{Re}_T$) and modeling the Carleman truncation error ϵ_C via Eq. (4.8), we can replace the $O(N_C^3)$ scaling with $O(\Gamma^{-3}\log^3\epsilon_C)$, where $\Gamma < 0$ is the convergence parameter depending on the Reynolds number. We thus see that the gate cost depends poly-logarithmically on the total solution error, as well as indirectly on the Reynolds number via the cubic dependence on the convergence parameter Γ.

5 Discussion and conclusions

Having analyzed the performance of our quantum algorithm for the LBE problem, we are now in a position to compare it with classical LBE algorithms. We begin with a brief comparison between classical and quantum memory requirements. In a classical simulation, it is necessary to store the local LBE state vectors at each lattice site. Given that there are $N = N_x^D$ lattice sites, and parameterizing the number of discretization points per direction as before to be $N_x = \mathrm{Re}^\beta$, we get

that the number n_c of memory bits needed scales as

$$n_c = O(\mathrm{Re}^{\beta D}). \tag{5.1}$$

On the other hand, the total number of data qubits n_D required by our algorithm, given by Eq. (3.8), and ancilla qubits n_A, specified by Eq. (3.70b), scales as

$$n_D + n_A = O(N_C \log \mathrm{Re}). \tag{5.2}$$

Note that the dependence on N_C translates via the error model from Eq. (4.8) into a dependence on $\log \frac{1}{\epsilon_C}$. Hence, for a fixed acceptable error level, we get an exponential memory improvement compared to classical algorithms.

Next, we will measure the classical computational complexity q_c using the number of updates of the local LBE state vector that have to be performed during the evolution. Given that there are $\mathrm{Re}^{\beta D}$ lattice sites and $T^\star$ time steps, we have

$$q_c = O(\mathrm{Re}^{\beta D} T^\star). \tag{5.3}$$

Moreover, note that $T^\star$ must also scale at least as Re^{β}, as otherwise the lattice velocity would diverge to infinity with growing Re. Thus, the classical complexity q_c grows with the Reynolds number Re as

$$q_c = O(\mathrm{Re}^{\beta(D+1)}), \tag{5.4}$$

which becomes a familiar computational fluid dynamics scaling of $O(\mathrm{Re}^3)$ for a three-dimensional system and a choice of $\beta = 3/4$ guaranteeing resolution of the Kolmogorov microscale [41] .

We will compare the classical complexity q_c with the quantum query complexity q_Q and not with the total number of T-gates. Two main reasons for that are as follows. First, a single classical query is already a complex task, consisting of many elementary logical operations needed to update a local LBE state vector. Thus, it would be unfair to treat on equal footing such a task and the use of an elementary single-qubit T gate. Second, we will be mostly interested in the asymptotic scaling of classical and quantum complexities, and as we have explained in the previous section, the difference between measuring in queries and in T-gates is just a constant factor overhead (i.e., if we assume a fixed N_C).

Let us then compare the classical complexity q_c of outputting the final LBE state with the query complexity q_Q for outputting a coherent encoding of the final Carleman state (with a fixed truncation order N_C) for the shifted LBE problem. If the fraction $q_c/q_Q > 1$, we can thus speak of a quantum advantage. First, recalling Eq. (4.37), we see that this fraction will at most scale as

$$\frac{q_c}{q_Q} = O\left(\mathrm{Re}^{\frac{\beta D}{2}}\right). \tag{5.5}$$

Hence, for the problem of dimension $D = 3$ and with the resolution parameter $\beta = 3/4$ to resolve the Kolmogorov microscale, the best one can hope for is an improvement by a factor of $\mathrm{Re}^{9/8}$, replacing a classical cubic dependence $O(\mathrm{Re}^3)$ with the quantum slightly-lower-than-quadratic dependence $O(\mathrm{Re}^{1.875})$. Instead, if we use our upper bound on q_Q from Eq. (4.41), and the numerically extracted data on the power χ from Eq. (4.35) for the case $\beta = 3/4$, we get that at worst

$$\frac{q_c}{q_Q} = O(\mathrm{Re}^{\lambda}), \tag{5.6}$$

with

$$D = 1: \quad \lambda = \begin{cases} 0.333 & \text{for} \quad N_C = 1, \\ -0.191 & \text{for} \quad N_C = 2, \\ -0.783 & \text{for} \quad N_C = 3, \\ -1.292 & \text{for} \quad N_C = 4, \end{cases} \qquad D = 2: \quad \lambda = \begin{cases} 0.662 & \text{for} \quad N_C = 1, \\ 0.314 & \text{for} \quad N_C = 2, \end{cases} \tag{5.7}$$

where we note that one can achieve quantum advantage for positive λ. Thus, while it is in principle possible for the complexity of a quantum algorithm to scale more favorably with the Reynolds number Re than for classical algorithms, it is limited to low values of the Carleman truncation

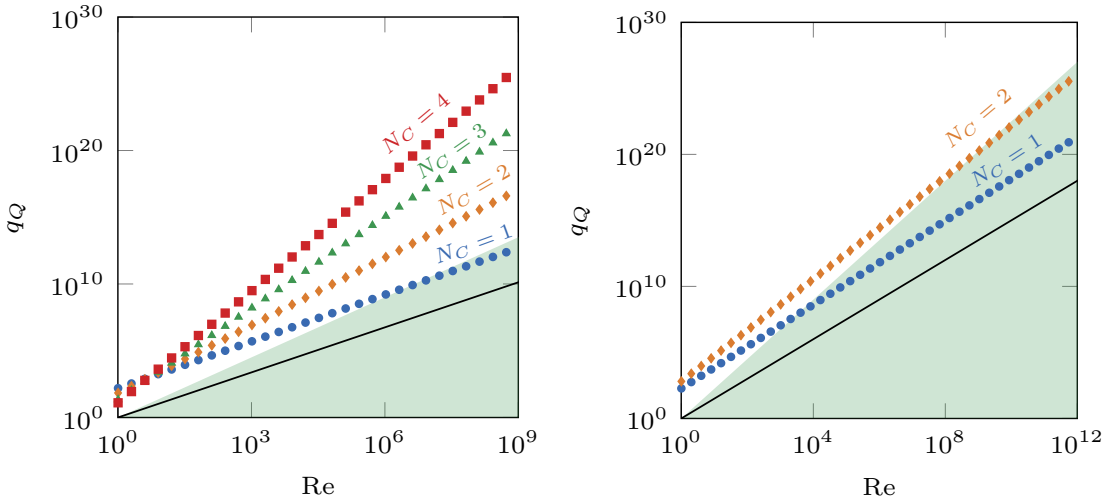


Figure 8: **Query complexity.** The lower bound from Eq. (4.37) (black line) and upper bounds from Eq. (4.41) for different values of Carleman truncation order N_C (different symbols) of the query complexity q_Q of our quantum LBE algorithm with $\beta = 3/4$ and for systems with dimension $D = 1$ (left) and $D = 2$ (right). The green highlighted region indicates potential for quantum advantage, when q_Q is smaller than classical complexity q_c. In the expressions of the form $O(\mathrm{Re}^x)$ for q_c and the lower bound for q_Q, we assumed a constant 1 and plotted Re^x.

order N_C. Via the error model from Eq. (4.8), this then limits applications to high-error outputs or low Reynolds number cases (values of Re far from the threshold Re_T require lower N_C for the same error ϵ_C). In Fig. 8, we present the dependence of q_Q on the Reynolds number Re and truncation order N_C, highlighting the region of potential quantum advantage (under the condition that Re is below a threshold value Re_T).

So far, we have discussed the complexity of outputting the coherently encoded solution. As discussed in Sec. 3.4.2, if we want to extract classical information about the drag force, we need to incur additional complexity cost scaling as $q_M = O(\mathrm{Re}^{\beta/2})$. Then, the best we can hope for is

$$\frac{q_c}{q_M q_Q} = O\left(\mathrm{Re}^{\frac{\beta(D-1)}{2}}\right). \tag{5.8}$$

Hence, there is no chance for a quantum advantage for $D = 1$, whereas for $D = 3$ one can get an improvement by a factor of Re^{β} at best. For the case of $\beta = 3/4$ this replaces the classical scaling $O(\mathrm{Re}^3)$ with the quantum $O(\mathrm{Re}^{2.25})$. Moreover, if we use our numerical estimates on the behavior of the condition number from Eq. (4.35), then all potential for quantum advantage is gone except for the case $D = 2$ and $N_C = 1$, where one could get an improvement by a factor of $\mathrm{Re}^{0.320}$. However, let us emphasize that this extra complexity cost arises from the boundary nature of the drag force observable, and certain bulk observables of interest may avoid such costs, potentially offering greater quantum advantage.

Finally, let us reiterate that for the quantum algorithm to work, the Carleman truncation error ϵ_C needs to converge. Since our analysis in Sec. 4.1 showed that such convergence happens only for Reynolds numbers Re below a threshold value Re_T, the actual value of Re_T for industrially-relevant problems is crucial. This is because one may face two opposing constraints: on the one hand, one needs large Re for the quantum algorithm to overcome constant overheads and become more efficient than the classical algorithm; on the other hand, Re may need to be low enough to guarantee error convergence. Also note that although the inclusion of boundaries, driving, and different initial states may potentially increase Re_T (see Ref. [57] for more details), we do not expect this effect to be significant. Hence, if the rescaling of the lattice velocity does not push Re_T high enough, the applicability of the Carleman approach to simulating highly turbulent flows may be limited.

Authors contributions: Authors are listed alphabetically within each affiliation. DJ contributed to the blocking-encoding and compilation theory of the quantum algorithm. KK identified complexity bottlenecks, developed the new model, designed the quantum algorithm, analyzed its complexity, and performed numerical simulations. ML identified complexity bottlenecks, supported the development of the new model, the design of the quantum algorithm, and its complexity analysis. RA performed numerical studies of truncation errors and operator norms as input to complexity bounds. EM performed numerical comparison between LB and Carleman approximation. SR supported the definition of the end user requirements, and the application use cases. KK wrote the paper with the assistance of DJ and ML.

Acknowledgments: We would like to thank Paul Mannix for his invaluable support and input on the physics of the model and CFD methods. We also want to thank Thomas Astoul, Thomas Bendokat, Giovanni Giustini, and Alessandro De Rosis for helpful discussions concerning fluid dynamics and the lattice Boltzmann method. Finally, we are grateful to Alice Barthe, Sam Morley-Short, and Trevor Vincent for their assistance with numerical simulations.

A Notation

	Symbol	Description
Fluid system	D	Number of spatial dimensions
	$\boldsymbol{r}$	Position in space
	t	Moment in time
	L_x, L_y, L_z	System size along x, y, and z
	T	Total evolution time
	$\boldsymbol{u}$	Fluid velocity
	c_s	Speed of sound
	L	Characteristic length of the system
	ν	Kinematic viscosity of the fluid
	Re	Reynolds number
	Ma	Mach number
Lattice Boltzmann model	Q	Number of discrete velocities
	$\boldsymbol{e}_m, \boldsymbol{e}_m^\star$	Discrete velocities in physical and lattice units
	$\boldsymbol{w}$	Vector of Maxwell weightings for discrete velocities
	$\boldsymbol{f}$, $\boldsymbol{g}$	LBE and shifted LBE state vectors
	$\boldsymbol{f}^{\mathrm{eq}}, \boldsymbol{g}^{\mathrm{eq}}$	Equilibrium states for LBE and shifted LBE
	N_x, N_y, N_z	Number of spatial discretization points along x, y, and z
	N	Total number of spatial discretization points
	$\boldsymbol{r}^\star$	Position in lattice units
	$t^\star$	Time step
	$T^\star$	Total number of time steps
	Δx	Distance between neighboring lattice sites
	Δt	Duration of a time step
	β	Spatial resolution parameter
	$\boldsymbol{u}^\star$	Fluid velocity in lattice units
	$u^\star_{\max}$	Maximal fluid velocity during the evolution (in lattice units)
	$u_0^\star$	Lattice velocity constant controlling the norm of the solution
Quantum algorithm	d	Dimension of the LBE system
	d_C	Dimension of the Carleman-embedded LBE system
	N_C	Carleman truncation order
	ϵ_C	Carleman truncation error
	$\boldsymbol{y}, \boldsymbol{y}_k$	Carleman vector and its k-th block
	F_1, F_2	Linear and quadratic collision matrices
	$\mathcal{C}, \mathcal{S}$	Carleman collision and streaming matrices
	κ_A	Condition number of a matrix A
	α_A	Block-encoding prefactor of a matrix A
	$A_H, \boldsymbol{Y}_H$	Linear system and its solution for the LBE history state
	$A_F, \boldsymbol{Y}_F$	Linear system and its solution for the LBE final state
	W	Number of qubits in the waiting register

Table 5: Symbols appearing in the text and their meaning.

B Transforming DBE into an ODE

With the discretization described by Eq. (1.20), the streaming term in DBE, Eq. (1.2), can be represented using central differencing as follows:

$$-\boldsymbol{e}_m \cdot \nabla f_m(\boldsymbol{r}, t) \qquad \to \qquad -\sum_{i \in \{x,y,z\}} (\boldsymbol{e}_m)_i \frac{f_{m,\boldsymbol{r}+\boldsymbol{a}_i}(t) - f_{m,\boldsymbol{r}-\boldsymbol{a}_i}(t)}{2\Delta x}, \tag{B.1}$$

where we introduced $\boldsymbol{a}_x = (1,0,0)$, $\boldsymbol{a}_y = (0,1,0)$ and $\boldsymbol{a}_z = (0,0,1)$. This can be written concisely as

$$-\boldsymbol{e}_m \cdot \nabla f_m(\boldsymbol{r},t) \qquad \rightarrow \qquad G\boldsymbol{f}(t), \tag{B.2}$$

where G is a $d \times d$ matrix describing the streaming generator with matrix elements given by

$$(G)_{m,\boldsymbol{r};m_1,\boldsymbol{r}_1} = -\delta_{m;m_1} \sum_{i\in\{x,y,z\}} \frac{(\boldsymbol{e}_m)_i(\delta_{\boldsymbol{r}+\boldsymbol{a}_i;\boldsymbol{r}_1} - \delta_{\boldsymbol{r}-\boldsymbol{a}_i;\boldsymbol{r}_1})}{2\Delta x}, \tag{B.3}$$

where δ denotes a Kronecker delta.

Next, one needs to rewrite the collision operator explicitly as a function of $\boldsymbol{f}$. Using the fact that for the considered discrete velocity models the lattice speed of sound is given by $c_s^\star = 1/\sqrt{3}$, the expression for the equilibrium distribution from Eq. (1.4) can be rewritten as:

$$f_m^{\mathrm{eq}}(\boldsymbol{r},t) = \rho(\boldsymbol{r},t) w_m \left(1 + 3\boldsymbol{e}_m^\star \cdot \boldsymbol{u}^\star(\boldsymbol{r},t) + \frac{9}{2}(\boldsymbol{e}_m^\star \cdot \boldsymbol{u}^\star(\boldsymbol{r},t))^2 - \frac{3}{2}|\boldsymbol{u}^\star(\boldsymbol{r},t)|^2\right). \tag{B.4}$$

Then, the authors of Refs. [37, 38] assume a weakly compressible flow, which means that the deviation of ρ from some reference density is small. Normalizing such reference density to 1, the following approximation is then justified:

$$\frac{1}{\rho(\boldsymbol{r},t)} \approx 2 - \rho(\boldsymbol{r},t). \tag{B.5}$$

Using it in Eq. (B.4), the equilibrium distribution can be expressed as a third order polynomial in $\boldsymbol{f}$ as follows (for clarity we omit the dependence on $\boldsymbol{r}$ and t):

$$\begin{aligned} f_m^{\mathrm{eq}} =& w_m \left(\sum_{m_1}^{Q} f_{m_1} + 3\boldsymbol{e}_m^\star \cdot \sum_{m_1=1}^{Q} \boldsymbol{e}_{m_1}^\star f_{m_1}\right) \\ &+ 2w_m \left(\frac{9}{2}\left(\boldsymbol{e}_m^\star \cdot \sum_{m_1=1}^{Q} \boldsymbol{e}_{m_1}^\star f_{m_1}\right)^2 - \frac{3}{2}\left(\sum_{m_1=1}^{Q} \boldsymbol{e}_{m_1}^\star f_{m_1}\right)^2\right) \\ &- w_m \left(\sum_{m_1=1}^{Q} f_{m_1}\right)\left(\frac{9}{2}\left(\boldsymbol{e}_m^\star \cdot \sum_{m_1=1}^{Q} \boldsymbol{e}_{m_1}^\star f_{m_1}\right)^2 - \frac{3}{2}\left(\sum_{m_1=1}^{Q} \boldsymbol{e}_{m_1}^\star f_{m_1}\right)^2\right), \end{aligned} \tag{B.6}$$

where we explicitly separated the expression into linear, quadratic and cubic terms.

Using the above, the BGK collision operator from Eq. (1.3) can be replaced by

$$\Omega(\boldsymbol{r},t) \qquad \rightarrow \qquad F_1\boldsymbol{f}(t) + F_2\boldsymbol{f}^{\otimes 2}(t) + F_3\boldsymbol{f}^{\otimes 3}(t), \tag{B.7}$$

where F_k are $d \times d^k$ matrices describing linear, quadratic and cubic contributions with matrix elements given by

$$(F_1)_{m,\boldsymbol{r};m_1,\boldsymbol{r}_1} = \frac{1}{\tau}\delta_{\boldsymbol{r};\boldsymbol{r}_1}(-\delta_{m;m_1} + w_m + 3w_m E_{m;m_1}), \tag{B.8a}$$

$$(F_2)_{m,\boldsymbol{r};m_1,\boldsymbol{r}_1,m_2,\boldsymbol{r}_2} = \frac{w_m}{\tau}\delta_{\boldsymbol{r};\boldsymbol{r}_1}\delta_{\boldsymbol{r};\boldsymbol{r}_2}\left(9E_{m;m_1}E_{m;m_2} - 3E_{m_1;m_2}\right), \tag{B.8b}$$

$$(F_3)_{m,\boldsymbol{r};m_1,\boldsymbol{r}_1,m_2,\boldsymbol{r}_2,m_3,\boldsymbol{r}_3} = -\frac{w_m}{2\tau}\delta_{\boldsymbol{r};\boldsymbol{r}_1}\delta_{\boldsymbol{r};\boldsymbol{r}_2}\delta_{\boldsymbol{r};\boldsymbol{r}_3}\left(9E_{m;m_1}E_{m;m_2} - 3E_{m_1;m_2}\right), \tag{B.8c}$$

and where we have introduced the following $Q \times Q$ matrix:

$$E_{m;m_1} := \boldsymbol{e}_m^\star \cdot \boldsymbol{e}_{m_1}^\star. \tag{B.9}$$

C Vanishing Carleman error for purely collisional DBE

Start with Eq. (1.21) with a single lattice site and no streaming:

$$\frac{d\boldsymbol{f}(t)}{dt} = F_1\boldsymbol{f}(t) + F_2\boldsymbol{f}^{\otimes 2}(t) + F_3\boldsymbol{f}^{\otimes 3}(t), \tag{C.1}$$

where F_k are $Q \times Q^k$ from Eqs. (B.8a)-(B.8c). Given these small sizes, it can be verified by direct calculations that the linear collision matrix F_1 has only two degenerate eigenvalues: 0 and $-1/\tau$. Denoting by V the matrix that diagonalizes F_1, we thus have

$$V^{-1}F_1V = -\frac{1}{\tau}\Pi_-, \tag{C.2}$$

where Π_- is the projector onto the negative eigenspace, and we have $\Pi_0 + \Pi_- = I$ with Π_0 denoting the projector onto the zero eigenspace of $V^{-1}F_1V$. Moreover, as can also be verified by direct calculations, the action of matrices $V^{-1}F_kV^{\otimes k}$ for k being 2 or 3 is to send vectors from $\Pi_0^{\otimes k}$ to Π_-. In other words, their kernel is the complement of $\Pi_0^{\otimes k}$, and their image is Π_-. This then leads to a set of algebraic relations between matrices F_k. In particular, we have:

$$F_1F_2 = -F_2, \tag{C.3a}$$
$$F_1F_3 = -F_3, \tag{C.3b}$$
$$F_2(F_1 \otimes I) = F_2(I \otimes F_1) = 0, \tag{C.3c}$$
$$F_3(F_1 \otimes I^{\otimes 2}) = F_3(I \otimes F_1 \otimes I) = F_3(I^{\otimes 2} \otimes F_1) = 0, \tag{C.3d}$$
$$\exp(F_1x)F_2 = e^{-x}F_2, \tag{C.3e}$$
$$\exp(F_1x)F_3 = e^{-x}F_3, \tag{C.3f}$$
$$F_2(\exp(F_1x) \otimes I) = F_2(I \otimes \exp(F_1x)) = F_2, \tag{C.3g}$$
$$F_3(\exp(F_1x) \otimes I^{\otimes 2}) = F_3(I \otimes \exp(F_1x) \otimes I) = F_3(I^{\otimes 2} \otimes \exp(F_1x)) = F_3, \tag{C.3h}$$
$$F_2(F_2 \otimes I) = F_2(I \otimes F_2) = 0, \tag{C.3i}$$
$$F_2(F_3 \otimes I) = F_2(I \otimes F_3) = 0, \tag{C.3j}$$
$$F_3(F_2 \otimes I^{\otimes 2}) = F_3(I^{\otimes 2} \otimes F_2) = 0, \tag{C.3k}$$
$$F_3(F_3 \otimes I^{\otimes 2}) = F_3(I \otimes F_3 \otimes I) = F_3(I^{\otimes 2} \otimes F_3) = 0. \tag{C.3l}$$

We now claim that the following is a solution to Eq. (C.1):

$$\boldsymbol{f}(t) = e^{F_1t}\boldsymbol{f}(0) + (1 - e^{-t})\left[F_2\boldsymbol{f}^{\otimes 2}(0) + F_3\boldsymbol{f}^{\otimes 3}(0)\right], \tag{C.4}$$

whose derivative can be straightforwardly calculated to be:

$$\frac{d\boldsymbol{f}(t)}{dt} = F_1e^{F_1t}\boldsymbol{f}(0) + e^{-t}\left[F_2\boldsymbol{f}^{\otimes 2}(0) + F_3\boldsymbol{f}^{\otimes 3}(0)\right]. \tag{C.5}$$

Indeed, substituting the postulated solution to Eq. (C.1), we get

$$\frac{d\boldsymbol{f}(t)}{dt} = F_1\left[e^{F_1t}\boldsymbol{f}(0) + (1 - e^{-t})F_2\boldsymbol{f}^{\otimes 2}(0) + (1 - e^{-t})F_3\boldsymbol{f}^{\otimes 3}(0)\right] \tag{C.6}$$
$$+ F_2\left[e^{F_1t}\boldsymbol{f}(0) + (1 - e^{-t})F_2\boldsymbol{f}^{\otimes 2}(0) + (1 - e^{-t})F_3\boldsymbol{f}^{\otimes 3}(0)\right]^{\otimes 2} \tag{C.7}$$
$$+ F_3\left[e^{F_1t}\boldsymbol{f}(0) + (1 - e^{-t})F_2\boldsymbol{f}^{\otimes 2}(0) + (1 - e^{-t})F_3\boldsymbol{f}^{\otimes 3}(0)\right]^{\otimes 3}, \tag{C.8}$$

which can be simplified using the algebraic identities introduced above to

$$\frac{d\boldsymbol{f}(t)}{dt} = F_1e^{F_1t}\boldsymbol{f}(0) - (1 - e^{-t})F_2\boldsymbol{f}^{\otimes 2}(0) - (1 - e^{-t})F_3\boldsymbol{f}^{\otimes 3}(0) \tag{C.9}$$
$$+ F_2\boldsymbol{f}^{\otimes 2}(0) \tag{C.10}$$
$$+ F_3\boldsymbol{f}^{\otimes 3}(0), \tag{C.11}$$

which in turn is exactly the derivative from Eq. (C.5).

Next, let us introduce the Carleman variables,

$$\boldsymbol{y}_k(t) = \boldsymbol{f}^{\otimes k}(t), \tag{C.12}$$

and assume we truncate the Carleman linearization at level N_C. Then, for all $k \leq N_C - 2$ we have:

$$\frac{d\boldsymbol{y}_k(t)}{dt} = F_1^{(k)}\boldsymbol{y}_k(t) + F_2^{(k)}\boldsymbol{y}_{k+1}(t) + F_3^{(k)}\boldsymbol{y}_{k+2}(t), \tag{C.13a}$$

$$\frac{d\boldsymbol{y}_{N_C-1}(t)}{dt} = F_1^{(N_C-1)}\boldsymbol{y}_{N_C-1}(t) + F_2^{(N_C-1)}\boldsymbol{y}_{N_C}(t), \tag{C.13b}$$

$$\frac{d\boldsymbol{y}_{N_C}(t)}{dt} = F_1^{(N_C)}\boldsymbol{y}_{N_C}(t), \tag{C.13c}$$

where

$$F_l^{(k)} = \sum_{j=0}^{k-1} I^{\otimes j} \otimes F_l \otimes I^{\otimes k-i-1}. \tag{C.14}$$

We can now write a formal solution of the above:

$$\boldsymbol{y}_k(t) = e^{F_1^{(k)}t}\boldsymbol{y}_k(0) + \int_0^t ds e^{F_1^{(k)}(t-s)}[F_2^{(k)}\boldsymbol{y}_{k+1}(s) + F_3^{(k)}\boldsymbol{y}_{k+2}(s)], \tag{C.15a}$$

$$\boldsymbol{y}_{N_C-1}(t) = e^{F_1^{(N_C-1)}t}\boldsymbol{y}_{N_C-1}(0) + \int_0^t ds e^{F_1^{(N_C-1)}(t-s)}F_2^{(N_C-1)}\boldsymbol{y}_{N_C}(s), \tag{C.15b}$$

$$\boldsymbol{y}_{N_C}(t) = e^{F_1^{(N_C)}t}\boldsymbol{y}_{N_C}(0). \tag{C.15c}$$

Assuming $N_C \geq 3$ and substituting such solutions for $\boldsymbol{y}_2$ and $\boldsymbol{y}_3$ into the solution for $\boldsymbol{y}_1$, while using the algebraic identities listed above, one obtains

$$\boldsymbol{y}_1(t) = e^{F_1 t}\boldsymbol{y}_1(0) + (1 - e^{-t})\left[F_2\boldsymbol{y}_2(0) + F_3\boldsymbol{y}_3(0)\right]. \tag{C.16}$$

Finally, since for the truncated system at time $t = 0$ we have $\boldsymbol{y}_k(0) = \boldsymbol{f}^{\otimes k}(0)$, comparing with Eq. (C.4), we conclude that

$$\boldsymbol{y}_1(t) = \boldsymbol{f}(t). \tag{C.17}$$

Thus, whenever $N_C \geq 3$, the Carleman linearization procedure reproduces exactly the evolution of one copy of the DBE state vector with a single lattice site.

D Incompressible LBE

When $\mathrm{Ma} \ll 1$, compressibility effects become negligible and it is useful to follow the approach of Ref. [45]. It proposes a modified collision integral that recovers an approximation of the incompressible Navier-Stokes equation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = -\frac{1}{\rho_0}\nabla p + \nu\nabla^2\boldsymbol{u}, \tag{D.1a}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{D.1b}$$

where ρ_0, ν, and p are the constant reference density, kinematic viscosity, and pressure, respectively (recall that in the main text we set $\rho_0 = 1$). In order to recover this approximation, one needs to modify the equilibrium distribution from Eq. (1.4), so that only the density variations that do not strongly violate Eq. (D.1b) are retained. Expanding the density as $\rho = \rho_0 + \delta\rho$ with $|\delta\rho| \ll \rho_0$ and letting the pressure fluctuation be given by $p = c_s^2\delta\rho$, we can substitute these expressions into Eq. (1.4) to arrive at

$$f_m^{\mathrm{eq}}(\boldsymbol{r},t) = w_m\left[\rho_0 + \frac{p}{c_s^2} + \rho_0\left(\frac{\boldsymbol{e}_m\cdot\boldsymbol{u}(\boldsymbol{r},t)}{c_s^2} + \frac{(\boldsymbol{e}_m\cdot\boldsymbol{u}(\boldsymbol{r},t))^2}{2c_s^4} - \frac{|\boldsymbol{u}(\boldsymbol{r},t)|^2}{2c_s^2}\right)\right] + O(\mathrm{Ma}^3). \tag{D.2}$$

In addition to terms proportional to $\boldsymbol{u}$, the term $p \sim O(\mathrm{Ma}^2)$ must retained as in the incompressible Navier-Stokes equation the pressure term corrects for the divergences produced by the convective nonlinearity (see, e.g., Chapter 1 of Ref. [67]). However, as $w_m\rho_0 \sim O(1)$ is constant, we can also shift the density $f_m - w_m\rho_0 \to g_m$, such that the shifted equilibrium density becomes

$$g_m^{\mathrm{eq}}(\boldsymbol{r},t) = w_m\left[\frac{p}{c_s^2} + \rho_0\left(\frac{\boldsymbol{e}_m\cdot\boldsymbol{u}(\boldsymbol{r},t)}{c_s^2} + \frac{(\boldsymbol{e}_m\cdot\boldsymbol{u}(\boldsymbol{r},t))^2}{2c_s^4} - \frac{|\boldsymbol{u}(\boldsymbol{r},t)|^2}{2c_s^2}\right)\right] + O(\mathrm{Ma}^3), \tag{D.3}$$

with the pressure and velocity given by

$$p(\boldsymbol{r},t) := c_s^2\sum_{m=1}^{Q} g_m(\boldsymbol{r},t), \quad \boldsymbol{u}(\boldsymbol{r},t) := \frac{1}{\rho_0}\sum_{m=1}^{Q} g_m(\boldsymbol{r},t)\boldsymbol{e}_m. \tag{D.4}$$

Therefore, at the expense of discarding weakly compressible effects, which in low Mach number flows are already negligible, we can obtain an $O(\mathrm{Ma}^2)$ incompressible LBE formulation whose collision integral is quadratic in g_m. As discussed in Chapter 3 of Ref. [68], this formulation recovers exactly the momentum equation, Eq. (D.1a), but approximates the divergence free condition, Eq. (D.1b), as

$$\nabla\cdot\boldsymbol{u} = \frac{1}{c_s^2}\frac{\partial p}{\partial t}, \tag{D.5}$$

where the right hand side is $O(\mathrm{Ma}^2)$. Although this approach is clearly suited for steady or slowly varying flows, in contrast the collision integral proposed by Ref. [46] (also quadratic in g_m) is exact and therefore preferable for time-dependent flows. Because the collision integral of Ref. [46] involves additional coefficients and is more complex, we have restricted this work to the simpler formulation of Ref. [45].

E Carleman error using RMSE

In order to quantify the Carleman error based on root mean square error, we first convert $\boldsymbol{g}(t^\star)$ and $\boldsymbol{y}_1(t^\star)$ into $\bar{\boldsymbol{f}}(t^\star)$ and its Carleman approximation $\tilde{\bar{\boldsymbol{f}}}(t^\star)$ using Eq. (2.1). Next, following Refs. [39, 40], for every discrete velocity direction m, we first calculate the root mean square error over the lattice sites:

$$\mathrm{RMSE}(m,t^\star) = \sqrt{\frac{1}{N}\sum_{\boldsymbol{r}^\star}\left(1 - \frac{\tilde{\bar{f}}_m(\boldsymbol{r}^\star,t^\star)}{\bar{f}_m(\boldsymbol{r}^\star,t^\star)}\right)^2}. \tag{E.1}$$

We then compute the uniform mean over discrete velocities:

$$\langle\mathrm{RMSE}(t^\star)\rangle = \frac{1}{Q}\sum_{m=1}^{Q}\mathrm{RMSE}(m,t^\star). \tag{E.2}$$

Finally, we define the RMSE-based Carleman truncation error as the maximal value of $\langle\mathrm{RMSE}(t^\star)\rangle$ during the whole evolution:

$$\epsilon_C^{\mathrm{RMSE}} := \max_{t^\star\in\{1,\ldots,T^\star\}}\langle\mathrm{RMSE}(t^\star)\rangle. \tag{E.3}$$

In Fig. 9, we present the results of our numerical simulations described in Sec. 4.1 using this RMSE-based measure of the Carleman error.

F Selected simulation parameters

In Tables 6 and 7, we collect the values of simulation parameters N_x, $T^\star$, $\bar{\tau}^\star$, and $u^\star$ for $D = 1$ and $D = 2$ systems, and a representative subset of Reynolds numbers chosen in our simulations. We also present dimensions of Carleman collision matrices $\mathcal{C}$ (so also of the simulated Carleman systems used to evaluate Carleman truncation errors ϵ_C) and the dimension of matrices A_H describing Carleman linear systems (whose condition numbers we evaluate to estimate the query complexity of our quantum algorithm). Absence of dim A_H in some cases means that we have not estimated the condition number for that choice of parameters (typically because sizes of the corresponding matrices were beyond the reach of our numerical methods in reasonable time).

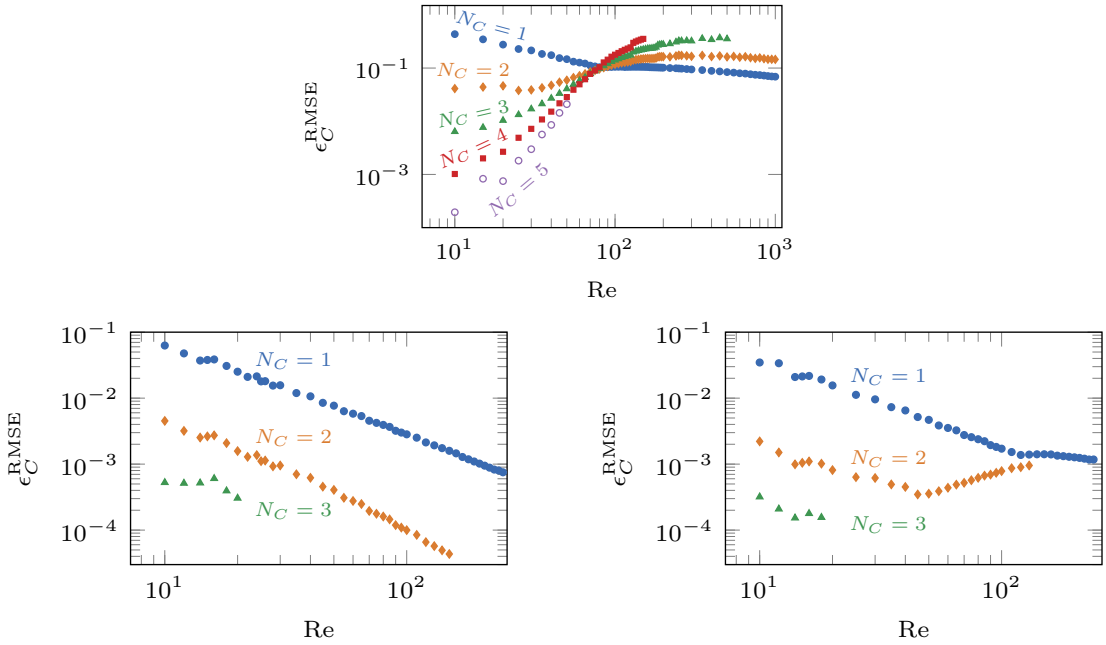


Figure 9: **RMSE-based Carleman error.** Top: Plot corresponding to data presented in the left panel of Fig. 1. Bottom: Plots corresponding to data presented in Fig. 4.

N_C	Re	N_x	$T^\star$	$\bar{\tau}^\star$	$u^\star$	dim $\mathcal{C}$	dim A_H
1	1000	178	2372	0.540	0.0750	534	1267182
2	200	54	388	0.6094	0.1361	26406	10271934
	1000	178	2372	0.540	0.0750	285690	-
3	100	32	178	0.6687	0.1768	894048	160034592
	500	106	1088	0.5617	0.0971	32258874	-
4	30	13	46	0.8580	0.2774	2374320	111593040
	150	43	281	0.6310	0.1525	279086340	-
5	50	19	82	0.7602	0.2294	612436557	-

Table 6: **Simulation parameters for $D = 1$ systems with $\beta = 3/4$.**

N_C	Re	N_x	$T^\star$	$\bar{\tau}^\star$	$u^\star$	dim $\mathcal{C}$	dim A_H
1	100	32	1000	0.5300	0.0313	9216	9225216
	250	63	3953	0.5120	0.0159	35721	-
2	20	10	90	0.6500	0.1000	810900	73791900
	150	43	1838	0.5200	0.0233	276939522	-
3	20	10	90	0.6500	0.1000	729810900	-

Table 7: **Simulation parameters for $D = 2$ systems with $\beta = 3/4$.**

G Eigenstate of collision-streaming operator

In this appendix, we present the construction of the -1 eigenstate $|\xi_\pi\rangle$ of the $\mathcal{SC}$ operator. As in Sec. 4.3.1, we set all Carleman blocks, except for the first one, to zero:

$$|\xi_\pi\rangle = [|\zeta_\pi\rangle, 0, \ldots, 0]. \tag{G.1}$$

Thus,

$$\mathcal{SC}\,|\xi_\pi\rangle = [S(I + F_1) = |\zeta_\pi\rangle, 0, \ldots, 0], \tag{G.2}$$

and so we reduce the problem to finding the -1 eigenstate of $S(I + F_1)$.

We start with $D = 1$ case. It can be easily verified that the following state is an eigenstate of $I + F_1$ corresponding to eigenvalue 1:

$$|\psi\rangle = \overset{-1}{\leftarrow} \overset{0}{\bullet} \overset{1}{\rightarrow}, \tag{G.3}$$

where we used a graphical notation to relate coefficients to discrete velocity components. We then claim that the following state is a -1 eigenstate of $S(I + F_1)$

$$|\zeta_\pi\rangle = \cdots \overset{|\psi\rangle}{\bullet} \quad \overset{-|\psi\rangle}{\bullet} \quad \overset{|\psi\rangle}{\bullet} \quad \overset{-|\psi\rangle}{\bullet} \quad \overset{|\psi\rangle}{\bullet} \cdots, \tag{G.4}$$

where different dots correspond to different lattice sites. One can indeed verify that the application of $I + F_1$ at every site leaves all local states invariant, which is then followed by the streaming operator mapping every $|\psi\rangle$ into $-|\psi\rangle$ and vice versa. Thus, the overall effect is to map $|\zeta_\pi\rangle$ to $-|\zeta_\pi\rangle$. Moreover, note that due to the antisymmetry of the state $|\psi\rangle$ with respect to the reflection around the node point, $|\zeta_\pi\rangle$ is the -1 eigenstate of $S(I + F_1)$ even in the presence of walls.

Next, consider $D = 2$. It can be verified by direct calculation that the following two states are eigenstates of $I + F_1$ corresponding to the eigenvalue 1:

$$|\psi_1\rangle = \begin{array}{ccccc} & & -2 & & \\ & 0 & \uparrow & -1 & \\ & \nwarrow & & \nearrow & \\ 2 & \leftarrow & 0 & \rightarrow & -2 \\ & \swarrow & & \searrow & \\ & 1 & \downarrow & 0 & \\ & & 2 & & \end{array}, \qquad |\psi_2\rangle = \begin{array}{ccccc} & & 2 & & \\ & 1 & \uparrow & 0 & \\ & \nwarrow & & \nearrow & \\ 2 & \leftarrow & 0 & \rightarrow & -2 \\ & \swarrow & & \searrow & \\ & 0 & \downarrow & -1 & \\ & & -2 & & \end{array}. \tag{G.5}$$

We then claim that the following state is a -1 eigenstate of $S(I + F_1)$:

$$|\zeta_\pi\rangle = \cdots \begin{array}{ccccccccccc} -|\psi_1\rangle & & |\psi_2\rangle & & -|\psi_1\rangle & & |\psi_2\rangle & & -|\psi_1\rangle & & \\ \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \\ -|\psi_2\rangle & & |\psi_1\rangle & & -|\psi_2\rangle & & |\psi_1\rangle & & -|\psi_2\rangle & & \\ \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \\ -|\psi_1\rangle & & |\psi_2\rangle & & -|\psi_1\rangle & & |\psi_2\rangle & & -|\psi_1\rangle & & \cdots \\ \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \\ -|\psi_2\rangle & & |\psi_1\rangle & & -|\psi_2\rangle & & |\psi_1\rangle & & -|\psi_2\rangle & & \\ \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \\ -|\psi_1\rangle & & |\psi_2\rangle & & -|\psi_1\rangle & & |\psi_2\rangle & & -|\psi_1\rangle & & \\ \bullet & & \bullet & & \bullet & & \bullet & & \bullet & & \end{array} \tag{G.6}$$

This can be confirmed by verifying that the streaming operator brings a negative phase factor, and that this still holds in the presence of nontrivial walls due to the antisymmetric nature of $|\psi_1\rangle$ and $|\psi_2\rangle$.

Finally, in $D = 3$, there exist three eigenstates of $I + F_1$ corresponding to eigenvalue 1 and being antisymmetric under the reflection with respect to the node point. One can then again find a spatial distribution of these states with phase factors to construct a -1 eigenstate of $S(I + F_1)$.

H Upper bound on the condition number

Consider a block decomposition of a matrix A,

$$A = \sum_{i=1}^{d}\sum_{j=1}^{d} |i\rangle\langle j| \otimes A_{ij}, \tag{H.1}$$

and note that

$$\max_{\boldsymbol{x}:\ \|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\| = \max_{\substack{\{\boldsymbol{x}_k\} \\ \sum_k \|\boldsymbol{x}_k\|^2=1}} \left\| \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d} \left(|i\rangle\langle j| \otimes A_{ij}\right)\left(|k\rangle \otimes \boldsymbol{x}_k\right) \right\| \tag{H.2}$$

$$= \max_{\substack{\{\boldsymbol{x}_k\} \\ \sum_k \|\boldsymbol{x}_k\|^2=1}} \left\| \sum_{i=1}^{d} |i\rangle \otimes \sum_{k=1}^{d} A_{ik}\boldsymbol{x}_k \right\| = \max_{\substack{\{\boldsymbol{x}_k\} \\ \sum_k \|\boldsymbol{x}_j\|^2=1}} \sqrt{\sum_{i=1}^{d} \left\| \sum_{k=1}^{d} A_{ik}\boldsymbol{x}_k \right\|^2} \tag{H.3}$$

$$\leq \max_{\substack{\{\boldsymbol{x}_k\} \\ \sum_k \|\boldsymbol{x}_k\|^2=1}} \sqrt{\sum_{i=1}^{d} \left(\sum_{k=1}^{d} \|A_{ik}\| \|\boldsymbol{x}_k\| \right)^2} \leq \max_{\substack{\{\boldsymbol{x}_k\} \\ \sum_k \|\boldsymbol{x}_k\|^2=1}} \sqrt{\sum_{i=1}^{d} \left(\sum_{j=1}^{d} \|A_{ij}\|^2 \sum_{k=1}^{d} \|\boldsymbol{x}_k\|^2 \right)} \tag{H.4}$$

$$= \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d} \|A_{ij}\|^2}. \tag{H.5}$$

Thus,

$$\|A\| \leq \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d} \|A_{ij}\|^2}. \tag{H.6}$$

Applying the above to bound the norm of A_H^{-1} from Eq. (2.55), noting that one can sum blocks over diagonals, leads to

$$\|A_H^{-1}\| \leq \sqrt{\sum_{t^\star=0}^{T^\star} (T^\star - t^\star + 1) \|(\mathcal{SC})^{t^\star}\|^2}. \tag{H.7}$$

Combining this with an upper bound for $\|A_H\|$ from Eq. (4.15) leads to the claimed bound on the condition number from Eq. (4.32a). Next, recall that A_F^{-1} has the same blocks as A_H^{-1} followed by rows of blocks, where each row $k \in \{1, \ldots, (2^W-1)(T^\star+1)\}$ consists of the blocks from the last row of A_H^{-1} followed by k identity blocks. Thus, we get the following upper bound

$$\|A_F^{-1}\|^2 \leq \sum_{t^\star=0}^{T^\star} (T^\star - t^\star + 1) \|(\mathcal{SC})^{t^\star}\|^2 + (2^W-1)(T^\star+1) \sum_{t^\star=0}^{T^\star} \|(\mathcal{SC})^{t^\star}\|^2 + \sum_{k=1}^{(2^W-1)(T^\star+1)} k \tag{H.8}$$

$$= \sum_{t^\star=0}^{T^\star} (2^W(T^\star+1) - t^\star) \|(\mathcal{SC})^{t^\star}\|^2 + \frac{1}{2}(2^W-1)(T^\star+1)(2^W + (2^W-1)T^\star) \tag{H.9}$$

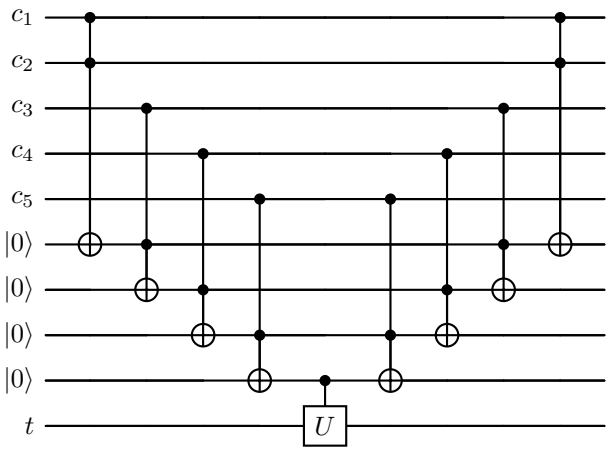
$$= \sum_{t^\star=0}^{T^\star} (2^W(T^\star+1) - t^\star) \|(\mathcal{SC})^{t^\star}\|^2 + 2^{2W-1}(T^\star+1)^2. \tag{H.10}$$

Combining it with the bound for $\|A_F\|$, which is the same as for $\|A_H\|$, we end up with the claimed bound for κ_{A_F} from Eq. (4.32b).

I Details on gate compilation

We start with explaining how to realize c^kU using cU and Toffoli gates. Given control qubits $c_1, \ldots, c_k$, the auxiliary qubits $a_1, \ldots, a_{k-1}$ are prepared in $|0\rangle$ and are then in turn used to iteratively record $\mathrm{AND}(c_1, c_2)$, $\mathrm{AND}(c_3, a_1) \ldots \mathrm{AND}(c_k, a_{k-2})$. A cU gate is then applied to the target

t controlled on a_{k-1}, and then we uncompute the auxiliary qubits. The explicit circuit for $k = 5$ is as follows:

$c^5U =$ 

(I.1)

This uses $2(k-1) = 2k-2$ Toffoli gates for the auxiliary qubits, and a single cU.

We now proceed to deriving a T-cost for general unitary synthesis based on state preparations. In Ref. [69], it is shown that with a single auxiliary qubit we can embed any unitary U as $\tilde{U} := |0\rangle\langle 1| \otimes U + |1\rangle\langle 0| \otimes U^\dagger$, and so we can simulate U by $\tilde{U}\,|1\rangle \otimes |\psi\rangle = |0\rangle \otimes U\,|\psi\rangle$. Then, it can be shown that

$$\tilde{U} = \prod_{j=1}^{2^n} (I - 2\,|w_j\rangle\langle w_j|), \tag{I.2}$$

where

$$|w_j\rangle := \frac{1}{\sqrt{2}} \left(|1\rangle \otimes |j\rangle - |0\rangle \otimes |u_j\rangle\right), \tag{I.3}$$

with $|u_j\rangle$ denoting the j-th column of U. Each reflection term $W_j := I - 2\,|w_j\rangle\langle w_j|$ can be approximately realized using an ϵ_j–approximate state preparation of $|w_j\rangle$ as

$$W_j^{\epsilon_j''} = \mathrm{PREP}^{\epsilon_j'}(w_j)(I - 2\,|0\rangle\langle 0|)\mathrm{PREP}^{\epsilon_j'}(w_j)^\dagger, \tag{I.4}$$

where we assume $\mathrm{PREP}^{\epsilon_j'}(w_j)$ is a unitary that prepares the state $|w_j\rangle$ from $|0^{n+1}\rangle = |0\rangle^{\otimes n+1}$ with error ϵ_j'. The preparation of $|w_j\rangle$ can be realized with a single controlled use of $\mathrm{PREP}^{\epsilon_j}(u_j)$, and so the total cost for generating $W_j^{\epsilon_j''}$ is

$$G[W_j^{\epsilon_j''}] = 2G[\mathrm{cPREP}_{\epsilon_j}(u_j)] + G\left[I - \left|0^{n+1}\right\rangle\!\left\langle 0^{n+1}\right|\right]. \tag{I.5}$$

We now provide error analysis for this step. The unitary $\mathrm{PREP}^{\epsilon_j}(u_j)$ obeys $\|\mathrm{PREP}^{\epsilon_j}(u_j)\,|0\rangle^{\otimes n} - |u_j\rangle\,\| \leq \epsilon_j$. This implies that

$$\left\| \frac{1}{\sqrt{2}}(|1\rangle \otimes |j\rangle - |0\rangle \otimes \mathrm{PREP}^{\epsilon_j}(u_j)\,|0\rangle) - |w_j\rangle \right\| \leq \frac{\epsilon_j}{\sqrt{2}} = \epsilon_j'. \tag{I.6}$$

Moreover, it is readily checked that

$$\left\| \mathrm{PREP}^{\epsilon_j'}(w_j)\left|0^{n+1}\right\rangle\!\left\langle 0^{n+1}\right| \mathrm{PREP}^{\epsilon_j'}(w_j)^\dagger - |w_j\rangle\langle w_j| \right\| \leq \epsilon_j' \sqrt{1 - \epsilon_j'^2/4} \leq \epsilon_j'. \tag{I.7}$$

This implies

$$\|\mathrm{PREP}^{\epsilon_j'}(w_j)(I - 2\,|0\rangle\langle 0|)\mathrm{PREP}^{\epsilon_j'}(w_j)^\dagger - (I - 2\,|w_j\rangle\langle w_j|)\| \leq 2\epsilon_j' = \sqrt{2}\epsilon_j. \tag{I.8}$$

Thus, $\epsilon_j'' = \sqrt{2}\epsilon_j$.

To generate $I - 2\left|0^{n+1}\right\rangle\!\left\langle 0^{n+1}\right|$, we first note that this is a unitary $\mathrm{diag}(-1, 1, 1, \ldots 1)$, which can be viewed as a $c^n(-Z)$ unitary, controlled on n 0's. Up to Clifford gates, this is equivalent to

$c^n X$, and so this provides a T-count of $G[c^n X]$. The total T-count of unitary synthesis from this method is then

$$G[U^\epsilon] = \sum_{j=1}^{2^n} (2G[\text{cPREP}_{\epsilon_j}(u_j)] + G[c^n X]) = 32 \sum_{j=1}^{2^n} G[\text{PREP}^{\epsilon_j}(u_j)] + 2^n(8n - 12), \tag{I.9}$$

where again we use that $G[cU] \approx 16G[U]$, the total error introduced is

$$\epsilon = \sqrt{2} \sum_{j=1}^{2^n} \epsilon_j, \tag{I.10}$$

and a single auxiliary qubit is used in the construction.

References

[1] P. Stefanin Volpiani, J.-B. Chapelier, A. Schwöppe, J. Jägersküpper, and S. Champagneux, "Aircraft simulations using the new CFD software from ONERA, DLR, and Airbus," *J. Aircr.*, vol. 61, no. 3, pp. 857–869, 2024. DOI: 10.2514/1.C037506.

[2] H. Patel, T. Gerhold, and N. Ashton, "Assessing the HPC performance of CODA for the NASA Common Research Model," in *AIAA Aviation Forum and Ascend 2024*, 2024, p. 3792. DOI: 10.2514/6.2024-3792.

[3] P. R. Spalart, "Strategies for turbulence modelling and simulations," *Int. J. Heat Fluid Flow*, vol. 21, no. 3, pp. 252–263, 2000. DOI: 10.1016/S0142-727X(00)00007-2.

[4] S. B. Pope, *Turbulent Flows*. Cambridge University Press, 2000. DOI: 10.1017/CBO9780511840531.

[5] J. Appa, M. Turner, and N. Ashton, "Performance of CPU and GPU HPC Architectures for off-design aircraft simulations," in *AIAA Scitech 2021 Forum*, 2021, p. 0141. DOI: 10.2514/6.2021-0141.

[6] L. Lin, "Lecture notes on quantum algorithms for scientific computation," 2022. arXiv: 2201.08309 [quant-ph].

[7] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 193–204.

[8] S. Aaronson, "Read the fine print," *Nat. Phys.*, vol. 11, no. 4, pp. 291–293, 2015, ISSN: 1745-2473. DOI: 10.1038/nphys3272.

[9] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, p. 150 502, 2009, ISSN: 0031-9007. DOI: 10.1103/physrevlett.103.150502.

[10] P. C. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, "Optimal scaling quantum linear-systems solver via discrete adiabatic theorem," *PRX Quantum*, vol. 3, no. 4, p. 040 303, 2022. DOI: 10.1103/PRXQuantum.3.040303.

[11] D. Jennings, M. Lostaglio, S. Pallister, A. T. Sornborger, and Y. Subaşı, "Efficient quantum linear solver algorithm with detailed running costs," 2023. arXiv: 2305.11352 [quant-ph].

[12] A. M. Dalzell, "A shortcut to an optimal quantum linear system solver," 2024. arXiv: 2406.12086 [quant-ph].

[13] M. E. Morales, L. Pira, P. Schleich, K. Koor, P. Costa, D. An, A. Aspuru-Guzik, L. Lin, P. Rebentrost, and D. W. Berry, "Quantum linear system solvers: A survey of algorithms and applications," 2024. arXiv: 2411.02522 [quant-ph].

[14] D. W. Berry, "High-order quantum algorithm for solving linear differential equations," *J. Phys. A*, vol. 47, no. 10, p. 105 301, 2014. DOI: 10.1088/1751-8113/47/10/105301.

[15] A. Montanaro and S. Pallister, "Quantum algorithms and the finite element method," *Phys. Rev. A*, vol. 93, no. 3, p. 032 324, 2016. DOI: 10.1103/PhysRevA.93.032324.

[16] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, "Quantum algorithm for linear differential equations with exponentially improved dependence on precision," *Commun. Math. Phys.*, vol. 356, pp. 1057–1081, 2017. DOI: 10.1007/s00220-017-3002-y.

[17] P. C. Costa, S. Jordan, and A. Ostrander, "Quantum algorithm for simulating the wave equation," *Phys. Rev. A*, vol. 99, no. 1, p. 012 323, 2019. DOI: 10.1103/PhysRevA.99.012323.

[18] N. Linden, A. Montanaro, and C. Shao, "Quantum vs. classical algorithms for solving the heat equation," *Commun. Math. Phys.*, vol. 395, no. 2, pp. 601–641, 2022. DOI: 10.1007/s00220-022-04442-6.

[19] D. W. Berry and P. C. Costa, "Quantum algorithm for time-dependent differential equations using Dyson series," *Quantum*, vol. 8, p. 1369, 2024. DOI: 10.22331/q-2024-06-13-1369.

[20] A. Ameri, E. Ye, P. Cappellaro, H. Krovi, and N. F. Loureiro, "Quantum algorithm for the linear Vlasov equation with collisions," *Phys. Rev. A*, vol. 107, p. 062 412, 2023. DOI: 10.1103/PhysRevA.107.062412.

[21] D. Jennings, M. Lostaglio, R. B. Lowrie, S. Pallister, and A. T. Sornborger, "The cost of solving linear differential equations on a quantum computer: Fast-forwarding to explicit resource counts," *Quantum*, vol. 8, p. 1553, 2024. DOI: 10.22331/q-2024-12-10-1553.

[22] Z. Meng, L. Chen, J.-P. Liu, and G. He, "Toward end-to-end quantum simulation of rapidly distorted turbulence," 2025. arXiv: 2511.18802 [quant-ph].

[23] R. P. Feynman, "Simulating physics with computers," in *Feynman and computation*, cRc Press, 2018, pp. 133–153.

[24] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996. DOI: 10.1126/science.273.5278.1073.

[25] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," *Commun. Math. Phys.*, vol. 270, no. 2, pp. 359–371, 2007. DOI: 10.1007/s00220-006-0150-x.

[26] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated quantum computation of molecular energies," *Science*, vol. 309, no. 5741, pp. 1704–1707, 2005. DOI: 10.1126/science.1113479.

[27] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," *Rev. Mod. Phys.*, vol. 92, no. 1, p. 015 003, 2020. DOI: 10.1103/RevModPhys.92.015003.

[28] R. Babbush, D. W. Berry, R. Kothari, R. D. Somma, and N. Wiebe, "Exponential quantum speedup in simulating coupled classical oscillators," *Phys. Rev. X*, vol. 13, no. 4, p. 041 041, 2023. DOI: 10.1103/PhysRevX.13.041041.

[29] T. Carleman, "Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires," 1932.

[30] M. Forets and A. Pouly, "Explicit error bounds for Carleman linearization," 2017. arXiv: 1711.02552 [math-NA].

[31] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, "Efficient quantum algorithm for dissipative nonlinear differential equations," *PNAS*, vol. 118, no. 35, e2026805118, 2021. DOI: 10.1073/pnas.2026805118.

[32] J.-P. Liu, D. An, D. Fang, J. Wang, G. H. Low, and S. Jordan, "Efficient quantum algorithm for nonlinear reaction–diffusion equations and energy estimation," *Commun. Math. Phys.*, vol. 404, no. 2, pp. 963–1020, 2023. DOI: 10.1007/s00220-023-04857-9.

[33] H. Krovi, "Improved quantum algorithms for linear and nonlinear differential equations," *Quantum*, vol. 7, p. 913, 2023. DOI: 10.22331/q-2023-02-02-913.

[34] P. C. Costa, P. Schleich, M. E. Morales, and D. W. Berry, "Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling," *npj Quantum Inf.*, vol. 11, no. 1, p. 141, 2025. DOI: 10.1038/s41534-025-01084-z.

[35] H.-C. Wu, J. Wang, and X. Li, "Quantum algorithms for nonlinear dynamics: Revisiting Carleman linearization with no dissipative conditions," *SIAM J. Sci. Comput.*, vol. 47, no. 2, A943–A970, 2025. DOI: 10.1137/24M1665799.

[36] D. Jennings, K. Korzekwa, M. Lostaglio, A. T. Sornborger, Y. Subasi, and G. Wang, "Quantum algorithms for general nonlinear dynamics based on the Carleman embedding," 2025. arXiv: 2509.07155 [quant-ph].

[37] X. Li, X. Yin, N. Wiebe, J. Chun, G. K. Schenter, M. S. Cheung, and J. Mülmenstädt, "Potential quantum advantage for simulation of fluid dynamics," *Phys. Rev. Res.*, vol. 7, no. 1, p. 013 036, 2025. DOI: 10.1103/PhysRevResearch.7.013036. arXiv: 2303.16550 [quant-ph].

[38] J. Penuel, A. Katabarwa, P. D. Johnson, C. Farquhar, Y. Cao, and M. C. Garrett, "Feasibility of accelerating incompressible computational fluid dynamics simulations with fault-tolerant quantum computers," 2024. arXiv: 2406.06323 [quant-ph].

[39] C. Sanavio and S. Succi, "Lattice Boltzmann–Carleman quantum algorithm and circuit for fluid flows at moderate Reynolds number," *AVS Quantum Science*, vol. 6, no. 2, 2024. DOI: 10.1116/5.0195549.

[40] F. Turro, A. Lignarolo, and D. Dragoni, "Practical application of the quantum Carleman lattice Boltzmann method in industrial CFD simulations," 2025. arXiv: 2504.13033 [quant-ph].

[41] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The Lattice Boltzmann Method: Principles and Practice*. Springer, 2016.

[42] S. Succi, W. Itani, K. Sreenivasan, and R. Steijl, "Quantum computing for fluids: Where do we stand?" *EPL*, vol. 144, no. 1, p. 10 001, 2023. DOI: 10.1209/0295-5075/acfdc7.

[43] F. Tennie, S. Laizet, S. Lloyd, and L. Magri, "Quantum computing for nonlinear differential equations and turbulence," *Nat. Rev. Phys.*, pp. 1–11, 2025. DOI: 10.1038/s42254-024-00799-w.

[44] W. Itani, K. R. Sreenivasan, and S. Succi, "Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term," *Phys. Fluids*, vol. 36, no. 1, 2024. DOI: 10.1063/5.0176569.

[45] X. He and L.-S. Luo, "Lattice Boltzmann model for the incompressible Navier–Stokes equation," *J. Stat. Phys.*, vol. 88, no. 3, pp. 927–944, 1997. DOI: 10.1023/B:JOSS.0000015179.12689.e4.

[46] Z. Guo, B. Shi, and N. Wang, "Lattice BGK model for incompressible Navier–Stokes equation," *J. Comput. Phys.*, vol. 165, no. 1, pp. 288–306, 2000. DOI: 10.1006/jcph.2000.6616.

[47] G Berkolaiko, S Rabinovich, and S Havlin, "Analysis of Carleman representation of analytical recursions," *J. Math. Anal. Appl.*, vol. 224, no. 1, pp. 81–90, 1998. DOI: 10.1006/jmaa.1998.5986.

[48] S. Pruekprasert, T. Takisaka, C. Eberhart, A. Cetinkaya, and J. Dubut, "Moment propagation of discrete-time stochastic polynomial systems using truncated Carleman linearization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 462–14 469, 2020. DOI: 10.1016/j.ifacol.2020.12.1447.

[49] S. Pruekprasert, J. Dubut, T. Takisaka, C. Eberhart, and A. Cetinkaya, "Moment propagation of polynomial systems through Carleman linearization for probabilistic safety analysis," *Automatica*, vol. 160, p. 111 441, 2024. DOI: 10.1016/j.automatica.2023.111441.

[50] S. Succi, *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford University Press, 2001.

[51] 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," *Phys. Rev.*, vol. 94, no. 3, p. 511, 1954. DOI: 10.1103/PhysRev.94.511.

[52] X. He, X. Shan, and G. D. Doolen, "Discrete Boltzmann equation model for nonideal gases," *Phys. Rev. E*, vol. 57, no. 1, R13, 1998. DOI: 10.1103/PhysRevE.57.R13.

[53] X. He, S. Chen, and G. D. Doolen, "A novel thermal model for the lattice Boltzmann method in incompressible limit," *J. Comput. Phys.*, vol. 146, no. 1, pp. 282–300, 1998. DOI: 10.1006/jcph.1998.6057.

[54] D. An, A. Onwunta, and G. Yang, "Fast-forwarding quantum algorithms for linear dissipative differential equations," 2024. arXiv: 2410.13189 [quant-ph].

[55] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, "Simulating Hamiltonian dynamics with a truncated Taylor series," *Phys. Rev. Lett.*, vol. 114, no. 9, p. 090 502, 2015. DOI: 10.1103/PhysRevLett.114.090502.

[56] S Ubertini, P. Asinari, and S Succi, "Three ways to lattice Boltzmann: A unified time-marching picture," *Phys. Rev. E*, vol. 81, no. 1, p. 016 311, 2010. DOI: 10.1103/PhysRevE.81.016311.

[57] D. Jennings, K. Korzekwa, M. Lostaglio, P. Mannix, R. Ashworth, E. Marsili, and S. Rolston, "Simulating non-trivial flows with a quantum lattice Boltzmann algorithm," in preparation, 2025.

[58] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, "Improved techniques for preparing eigenstates of fermionic Hamiltonians," *npj Quantum Inf.*, vol. 4, no. 1, p. 22, 2018. DOI: 10.1038/s41534-018-0071-5.

[59] J. Lemieux, M. Lostaglio, S. Pallister, W. Pol, K. Seetharam, S. Sim, and B. Şahinoğlu, "Quantum sampling algorithms for quantum state preparation and matrix block-encoding," 2024. arXiv: 2405.11436 [quant-ph].

[60] I. Delbende and M. Rossi, "The dynamics of a viscous vortex dipole," *Phys. Fluids*, vol. 21, no. 7, 2009. DOI: 10.1063/1.3183966.

[61] P. Selinger, "Quantum circuits of T-depth one," *Phys. Rev. A*, vol. 87, no. 4, p. 042 302, 2013. DOI: 10.1103/PhysRevA.87.042302.

[62] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, 2013. DOI: 10.1109/TCAD.2013.2244643.

[63] T. Khattar and C. Gidney, "Rise of conditionally clean ancillae for efficient quantum circuit constructions," *Quantum*, vol. 9, p. 1752, 2025. DOI: 10.22331/q-2025-05-21-1752.

[64] N. J. Ross and P. Selinger, "Optimal ancilla-free Clifford+T approximation of z-rotations," 2014. arXiv: 1403.2975 [quant-ph].

[65] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," 2004. arXiv: quant-ph/0407010 [quant-ph].

[66] C. Gidney, "Halving the cost of quantum addition," *Quantum*, vol. 2, p. 74, 2018. DOI: 10.22331/q-2018-06-18-74.

[67] C. R. Doering and J. D. Gibbon, *Applied analysis of the Navier-Stokes equations*. Cambridge university press, 1995.

[68] Z. Guo and C. Shu, *Lattice Boltzmann method and its application in engineering*. World Scientific, 2013, vol. 3.

[69] V. Kliuchnikov, "Synthesis of unitaries with Clifford+T circuits," 2013. arXiv: 1306.3200 [quant-ph].