Quantum Lattice-Based Modelling & Future Developments

René Steijl
James Watt School of Engineering
University of Glasgow
James Watt South Building
G12 8QQ Glasgow
United Kingdom

rene.steijl@glasgow.ac.uk

ABSTRACT

Quantum Fluid Dynamics approaches based on the concept of lattice-based modelling are briefly reviewed. Then, the widely-used streaming operation in lattice-based algorithms is discussed in detail and efficient quantum-circuit implementations are presented. Since multiple-control Toffoli gates arise that are far removed from native gates on quantum hardware, circuit transformation steps to reduce the number of control qubits are also shown. The discrete-velocity method (DVM) as used in Boltzmann kinetic modelling is introduced, before a detailed description of a quantum-circuit implementation of this method for the collisionless Boltzmann equations. For the lattice-based quantum algorithm for the collionless Boltzmann detailed in the present notes, the quantum-circuit implementation of specular-reflection boundary conditions is presented using a number of worked examples and exercises. A section detailing future developments related to quantum lattice-based modelling completes these notes.

Contents

1.0 INTRODUCTION

Quantum lattice-based approaches to deriving quantum algorithms for Computational Fluid Dynamics applications represent a promising and active area of current research[1–3]. Interestingly, early work in quantum computing relevant to the field of Computational Fluid Dynamics (CFD) mainly involved the work on quantum lattice-gas models by Yepez and co-workers[4, 5]. This work typically used type-II quantum computers, consisting of a large lattice of small quantum computers interconnected in nearest neighbour fashion by classical communication channels. It remained largely unclear how a significant quantum speed-up could be achieved and this line of research is no longer pursued. In more recent work, the Lattice Boltzmann Method (LBM) has been an important focal point and current research aims to create quantum algorithm implementations of the LBM for (nearly) incompressible flows, based on D2Q9 and D3Q27 lattice models for the (nearly) incompressible-flow Navier-Stokes equations. In addition, the kinetic Boltzmann model equations have also been considered as the basis for quantum algorithms[2, 6], where the discrete-velocity method (DVM) was used with a much larger number of discrete velocities than in the D2Q9/D3Q27 models, with 9 and 27 directions or discrete velocities for two- and three-dimensional flows, respectively. Although quite different in context and application, the DVM for kinetic modelling of flows and the LBM for continuum flows, both approaches share the important aspect of how the **nonlinear** terms occur. Specifically, in contrast to widely-used Finite-Volume and Finite-Element methods that discretize the nonlinear convection term directly, the DVM kinetic models and LBM have a linear term representing convection. However, the source term on the right-hand of the (system) of equations contains ’equilibrium’ distribution functions that typically depend on local density, velocity component and, in case of kinetic Boltzmann DVM, also temperature. Since the local continuum properties are defined as numerical moments of the non-equilibrium distribution functions governed by LBM or DVM, it is clear that the non-linearity is represented in this source/collision term.

A common feature of lattice-based modelling is the occurrence of streaming operators that move a component of the non-equilibrium distribution function from one lattice site to a neighbouring site. The quantum-circuit implementation of streaming operators is discussed in Section 2.0. The main part of the present notes describes the quantum-circuit implementation of a lattice-based algorithm for the collisionless Boltzman equation in Section 3.0 and Section 4.0. For the quantum lattice-based algorithm for the collisionless Boltzmann equation introduced in Todorova and Steijl[2], the ’quantum’ benefit and remaining challenges can be summarized as follows:

- The novel algorithm presented here is among the very first algorithms related to fluid dynamics that can be performed fully on a quantum computer, i.e. with information transfer between hybrid and classical hardware taking place only at start (initialization) and end of simulation. It therefore does not involve a frequent exchange of information typical of hybrid quantum/classical approaches;

- A key benefit of the introduced algorithm relative to a classical equivalent is the exponential reduction in memory when expressed in terms of qubits;
- For N_v discrete velocity per coordinate direction and spatial dimension d, the time complexity of a classical DVM is $O(TdN_v^d \times (D/h)^d)$, where h represents the cell-spacing of the uniformly-space mesh assumed for domain size D, and T represents the simulation time;
- For the quantum algorithm, the equivalent time complexity expressed in terms of multi-qubit gate operation was shown by Todorova and Steijl[2] to be $O(TdN_v log_2(D/h))$. Therefore, there is a potential exponential speed up for the quantum algorithm. It should be noted that the complexity of the quantum algorithm quoted here represents the complexity of performing the required operations on the state vector for the time evolution phase of the simulation.

From the review of quantum lattice-based modelling presented so far, as well as, the work detailed in these notes, it is clear that further research work is needed. In particular, to address the questions related to efficiently initializing the algorithm as well as extracting classical information at the end of the simulation.

The final section of these lectures notes, i.e. Section 5.0, details future developments related to quantum lattice-based modelling.

2.0 STREAMING OPERATIONS IN LATTICE-BASED MODELLING

2.1 Streaming of scalar quantity in 1D

As a first example, the 'streaming' of a scalar quantity on a one-dimensional domain with 64 lattice sites and periodic boundary condition is considered. Using 6 qubits, here named $|x5|x4|x3|x2|x1|x0\rangle$, creates the 64 degrees-of-freedom required. The indices of the sites are in the range $[0, 63]$, created as base-2 binary representations of the 6-qubit states. In the following 'left' streaming indicates that a scalar value moves to its nearest neighbour on the left-hand side, with an index 1 lower than the starting site. Similarly, 'right' streaming indicates a move to a lattice site with an index incremented by 1 relative to starting point. Figure 1 shows the quantum-circuit implementation of both operations.

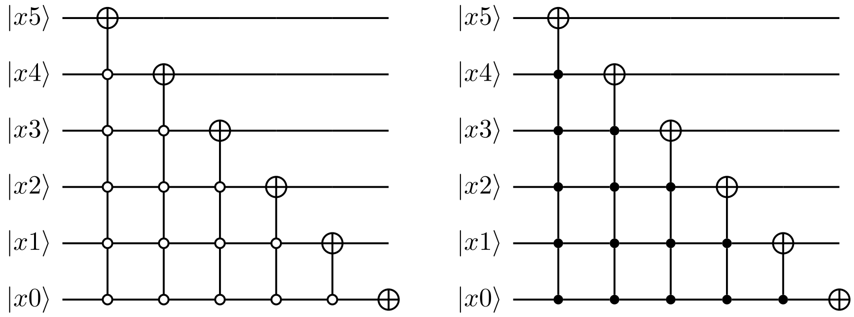


Figure 1: Streaming of scalar data on uniformly-space 1D mesh (64 cells encoded with 6 qubits) with periodic boundary conditions. Circuits for 'Left' and 'Right' streaming.

Clearly, the quantum-circuit implementations shown in Figure 1 involve multiply-controlled Toffoli gates with up to 5 control qubits. For a one-dimensional domain defined by n_x qubits, it follows that up to $n_x - 1$ control qubits occur. Using additional ancillae qubits, a quantum-circuit transformation can be introduced that reduces the number of control qubits. As an example, for the previously considered domain with 64 lattice points, the 'right' streaming operation is transformed to have ≤ 3 control qubits in Figure 2.

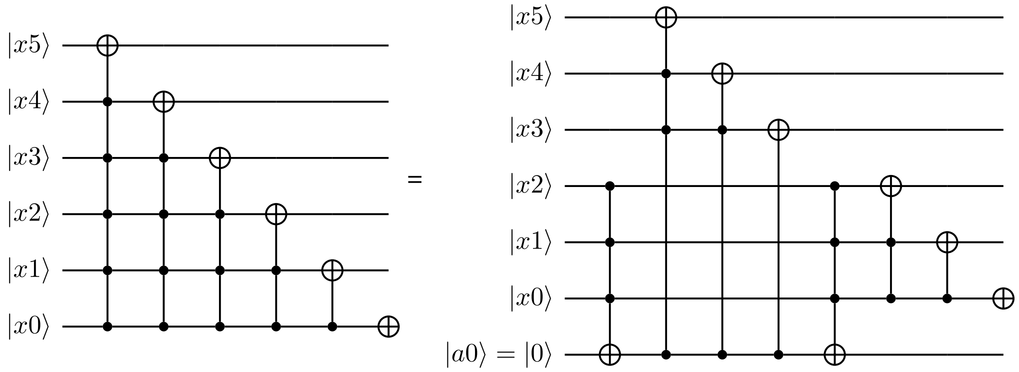


Figure 2: Streaming of scalar data on uniformly-space 1D mesh (64 cells encoded with 6 qubits) with periodic boundary conditions. Circuit for 'Right' streaming was transformed to reduce number of control qubits to ≤ 3.

2.2 Streaming of scalar quantity in 2D

To illustrate the extension to 2D domains, the 'streaming' of a scalar quantity on a two-dimensional domain with 64×64 lattice sites and periodic boundary condition is considered. Using 6 qubits for the x-coordinate direction, here named $|x5|x4|x3|x2|x1|x0\rangle$, plus 6 qubits for y-coordinate direction (termed $|y5|y4|y3|y2|y1|y0\rangle$), creates the 64×64 degrees-of-freedom required. Figure 3 and Figure 4 show the quantum-circuit operations for 'left' streaming in $x-$ and $y-$directions, and 'right' streaming in $x-$ and $y-$directions, respectively.

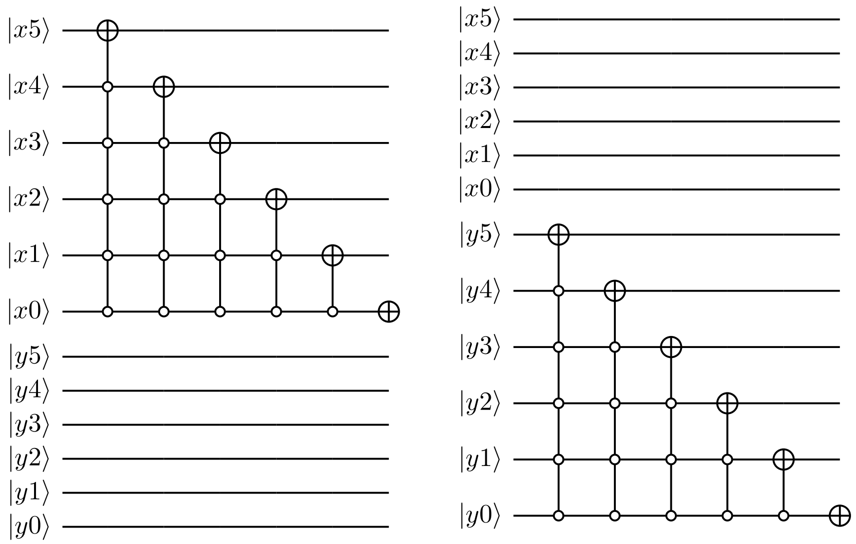


Figure 3: Streaming of scalar data on uniformly-space 2D mesh (64×64 cells encoded with $6 + 6$ qubits) with periodic boundary conditions. Circuits for 'Left' streaming in $x-$ and $y-$direction.

From Figure 3 and Figure 4, the similarity with the streaming operations in 1D is clear. **Crucially, for a 2D domain, the gate complexity for a single streaming operation (either 'left' or 'right') and a single coordinate direction ($x-$ or y-coordinate direction) is the same as a single 'left' or 'right' streaming operation in 1D. This feature can be considered an example of 'quantum parallelism' creating a beneficial effect.**

2.3 Extension to Discrete-Velocity Method

The examples discussed so far involved 'scalar' data. In the LBM and discrete-velocity method (DVM) for kinetic models, the data representing a multi-component distribution function needs to be streamed. As a first illustration of the extension to multi-component values, Figure 5 illustrates the streaming of a 2-component function on a one-dimensional with 64 lattice sites and periodic conditions. Here, an additional qubit $|du\rangle$ is used to double the number of degrees-of-freedom. In the example shown, for $|du\rangle = |0\rangle$ the component of the function is streamed 'left', while 'right' streaming for the component of the function defined by $|du\rangle = |1\rangle$ is used. Clearly, by using additional qubits, e.g. $|u3|u2|u1|u0\rangle$ for 16 component function instead of the single $|du\rangle$, a one-dimensional DVM for kinetic models can be created. This can be extended to two-dimensional DVMs by using multiple qubits defining discrete u-velocities and $v-$discrete velocities. As illustration of this concept, Figure 6 and Figure 7 shows the quantum-circuit implementation of 'left' and 'right' streaming in x- and $y-$ coordinate directions for the 'smallest' discrete-velocity (in magnitude) a 16×16 discrete-velocity

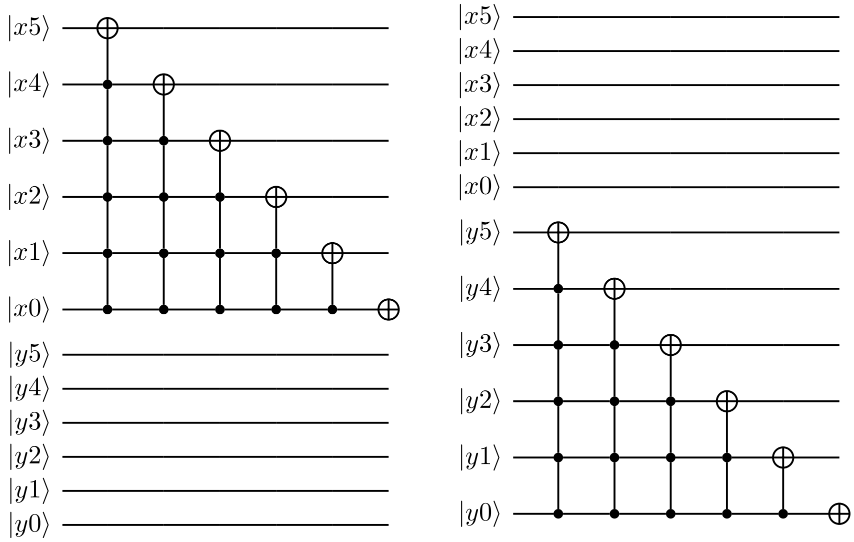


Figure 4: Streaming of scalar data on uniformly-space 2D mesh (64×64 cells encoded with $6 + 6$ qubits) with periodic boundary conditions. Circuits for 'Right' streaming in $x-$ and $y-$direction.

model on a computational domain with 64×64 lattice sites. For the discrete-velocity indexing used, the smallest (in magnitude) negative u-discrete velocity is defined by $|u3|u2|u1|u0\rangle = |0111\rangle$, while the smallest positive u-discrete velocity is defined by $|u3|u2|u1|u0\rangle = |1000\rangle$.

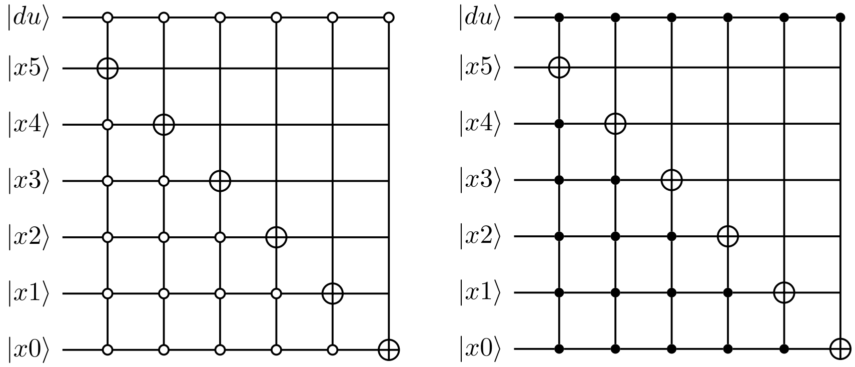


Figure 5: Streaming of 2-valued function data on uniformly-space 1D mesh (64 cells encoded with 6 qubits) with periodic boundary conditions. Circuits for 'Left' (for $|du\rangle = |0\rangle$) and 'Right' (for $|du\rangle = |1\rangle$) streaming.

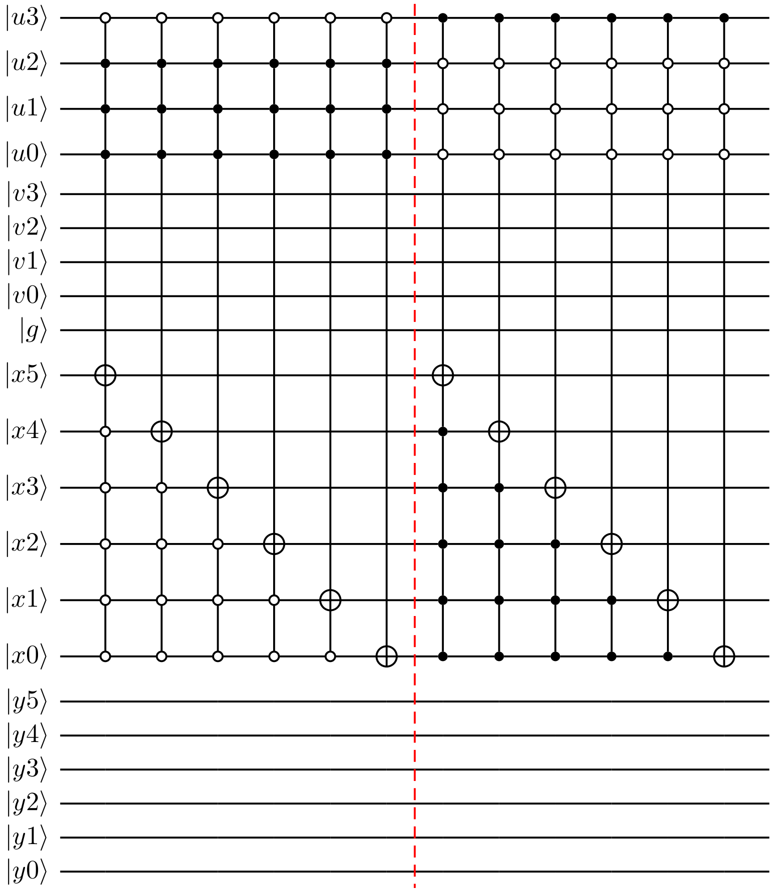


Figure 6: Streaming in $x-$direction. DVM method with 16×16 discrete velocities on 64×64 uniformly-spaced mesh. Smallest (in absolute value) discrete-velocity data is streamed by circuit shown, as defined by conditionals on qubits defining discrete-velocity indices.

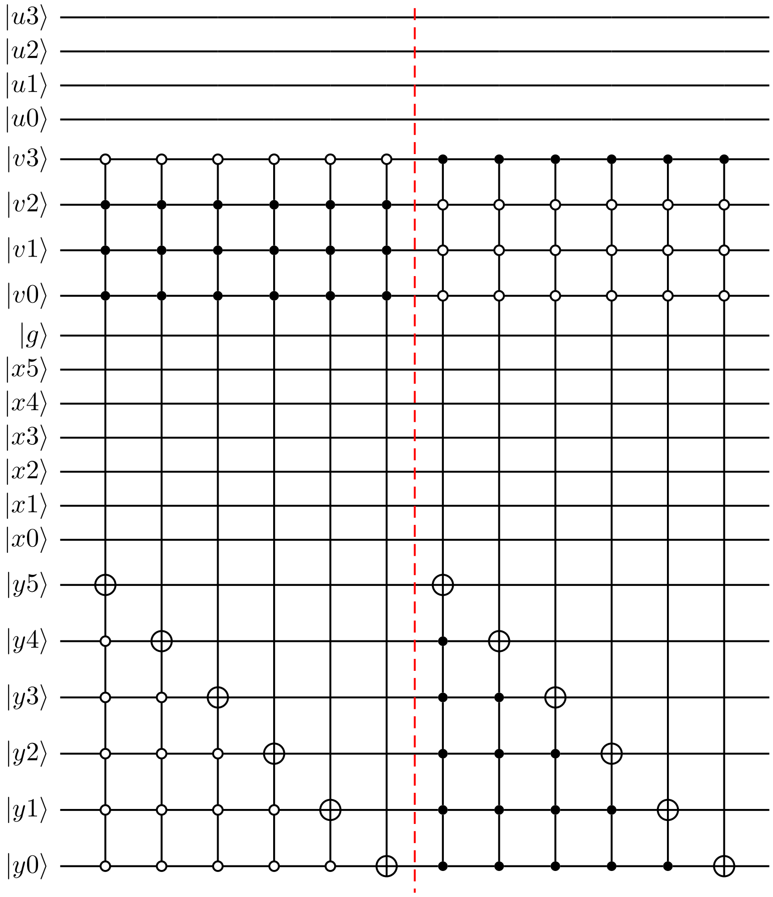


Figure 7: Streaming in $y-$direction. DVM method with 16×16 discrete velocities on 64×64 uniformly-spaced mesh. Smallest (in absolute value) discrete-velocity data is streamed by circuit shown, as defined by conditionals on qubits defining discrete-velocity indices.

3.0 QUANTUM ALGORITHM FOR COLLISIONLESS BOLTZMANN EQUATION

As an example of Quantum Lattice-Based modelling, we will now focus on a quantum lattice-based algorithm for the collisionless Boltzmann equation. The quantum-circuit implementation will be discussed has number of key features:

- The 'full' algorithm is implemented in terms of quantum circuits, i.e. this is not a hybrid quantum/classical algorithm typically used in the existing work in quantum fluid dynamics;
- The algorithm performs multiple time steps of the discrete-velocity method (DVM) used to discretize the collisionless Boltzmann equation in space, velocity-space and time;
- A modified reservoir-technique based time-integration is used to facilitate quantum-circuit implementation. Specifically, for each of the discrete velocities used in the DVM, a 'streaming' operation to a nearest neighbour is used for the considered component of the non-equilibrium distribution function;
- The specular-reflection boundary conditions are also introduced in the quantum-circuit implementation. This type of boundary condition models a molecule elastically colliding with a solid wall, so that the wall-normal velocity component gets reversed while the wall-tangential velocity component remains unchanged

The quantum circuits used for the streaming operations are similar to previous work for the dirac equation[7–9], while the quantum-circuits for the specular-reflection boundary conditions were introduced by the author[2].

3.1 Collisionless Boltzmann equation

The Boltzmann equation defines the single-particle distribution in a three-dimensional phase (velocity-space) for each point in three-dimensional space and therefore involves a seven-dimensional solution space (including time) for a gas consisting of a single monatomic species[10, 11]. In this section, applications involving highly rarefied gas flows (large Knudsen numbers) are considered and the assumption is that that these flows can be modeled as free-molecular gas flows, i.e. neglecting the collisions between the gas molecules. The collisions between gas molecules and domain boundaries is included, since this represents an essential feature of the considered flows. When neglecting inter-particle collisions and further body forces acting on the gas, the Boltzmann equation reduces to the collisionless Boltzmann equation, written here for a single-species flow as,

$$\frac{\partial F(\vec{x},\vec{c};t)}{\partial t} + \vec{c}\cdot\frac{\partial F(\vec{x},\vec{c};t)}{\partial \vec{x}} = 0 \tag{1}$$

$$F_{initial} = \frac{\rho}{\left(2\pi RT\right)^{3/2}}\exp\left[-\frac{\left(\vec{c}-\vec{u}_0\right)^2}{2RT}\right] \tag{2}$$

where $F(\vec{x},\vec{c};t)$ is the single-particle distribution function, and $\vec{x}=(x,y,z)^T$ and $\vec{c}=(c_x,c_y,c_z)^T$ represent three-dimensional space and three-dimensional phase (velocity) space, respectively. Also, $F_{initial}$ defines the Maxwell-Boltzmann equilibrium distribution, for a local gas mass density ρ, temperature T and mean gas velocity $\vec{u}_0$, used in the present work as initial conditions of the simulations. The particle number density n and gas mass density ρ are related as $\rho = nm$, for molecular mass m. For the Maxwell-Boltzmann equilibrium distribution it follows that the most probable (thermal) speed of a particle depends on temperature $V_{mp} = \sqrt{2RT}$ with R the specific gas constant for the gas considered. For a molecular mass m this gas constant

$R = k_b/m$, with k_b the Boltzmann constant. Since monatomic gases are considered, $\gamma = 5/3$. For the flow considered here, periodic boundary conditions are imposed on the edges of the computational domain. For the solid walls considered, specular-reflection boundary conditions are applied, which can be described as follows. For a molecule traveling with a pre-collision velocity $\vec{c} = (u, v, w)^T$, the reflective collision with the wall creates a post-collision velocity $\vec{c}' = \vec{c} - 2\vec{n}(\vec{n} \cdot \vec{c})$, where $\vec{n}$ represents the unit wall normal vector. For the 2D test cases considered here, stationary solid walls will be used aligned with either the $x-$ or $y-$direction, so that $v-$ or $u-$ velocity component, respectively, will be reversed. In the quantum algorithm presented here, this effect is achieved by reversing this velocity for 'halo' lattice sites within the solid object next to the imposed wall, as detailed later.

3.2 Reduction of the collisionless Boltzmann equation to 2D

For quasi-two dimensional flows, a dimensional reduction approach is commonly used to reduce significantly the required overhead. Starting from the full three-dimensional collisionless Boltzmann equation with single-particle distribution function $F(\vec{x}, \vec{c}; t)$, the dimensional reduction to two-dimensional problems replaces F with two reduced distribution functions as follows:

$$\begin{aligned} &f(x, y, c_x, c_y; t) = \int_{-\infty}^{\infty} F(\vec{x}, \vec{c}; t) dc_z \ ; \ g(x, y, c_x, c_y; t) = \int_{-\infty}^{\infty} c_z^2 F(\vec{x}, \vec{c}; t) dc_z \\ &\frac{\partial f}{\partial t} + \vec{c}\frac{df}{d\vec{x}} = 0 \ ; \ \frac{\partial g}{\partial t} + \vec{c}\frac{dg}{d\vec{x}} = 0 \end{aligned} \tag{3}$$

the corresponding gas density, mean velocities and temperature can be obtained from f and g using the following moments,

$$\begin{aligned} &\rho = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f dc_x dc_y \ ; \ \rho \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \begin{pmatrix} c_x \\ c_y \end{pmatrix} f dc_x dc_y \\ &\frac{3}{2}\rho T + \frac{u_0^2 + v_0^2}{2} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\frac{c_x^2 + c_y^2}{2} f + g\right] dc_x dc_y \end{aligned}$$

This shows that in the quantum-circuit implementation for 2D problems, the streaming operations as well as the boundary conditions involve two reduced distribution functions. Since the operations are identical for both these reduced distribution functions, the following discussion omits this detail for clarity ad brevity, i.e. the quantum circuits involve a single distribution function discretized by the DVM.

3.3 Discrete-velocity method

The present method implements a discrete-velocity method with a Cartesian, uniform mesh in state (velocity) space. A key feature of applying the DVM method to the collisionless Boltzmann equation is that in contrast to model kinetic equations including a collision term, in the time-integration method the evaluation of the continuum quantities of the flow is not needed at each time step since there is no need to construct a local equilibrium function typically required in a BGK-type relaxation model. In deriving quantum algorithm, the following assumptions related to the DVM were made:

- N_u and N_v define the number of discrete-velocity mesh points in $x-$ and $y-$direction and in this work we assume $N_u = N_v$, although the algorithm is not restricted to this;

- For each direction, the discrete velocities are defined as $c_k \in [c_{min}, \ldots, c_{max}]$, for $k = 0, \ldots, n_{DV} - 1$ (with $n_{DV} = N_u$ or $n_{DV} = N_v$) and a uniform step size in velocity space $\Delta c = (c_{max} - c_{min})/n_{DV}$. Furthermore, $c_{min} = -c_{max}$. The implementation of the specular-reflection boundary condition in the current quantum algorithm requires this assumption;
- Discrete velocities are indexed from 0, i.e. the most negative value, to $n_{DV} - 1$ representing the discrete-velocity with the largest positive value

In all cases, the simulations will be initialized with an equilibrium Maxwellian distribution defined for an initial flow state.

3.4 Multi-step time integration method for 'full' quantum-circuit based simulation

Table 1: Modified reservoir method for velocity-space boundaries ± 8 reference velocity units ($\sqrt{2RT_r}$, most-probable molecular speed at reference temperature T_r) and $\Delta x = 1$.

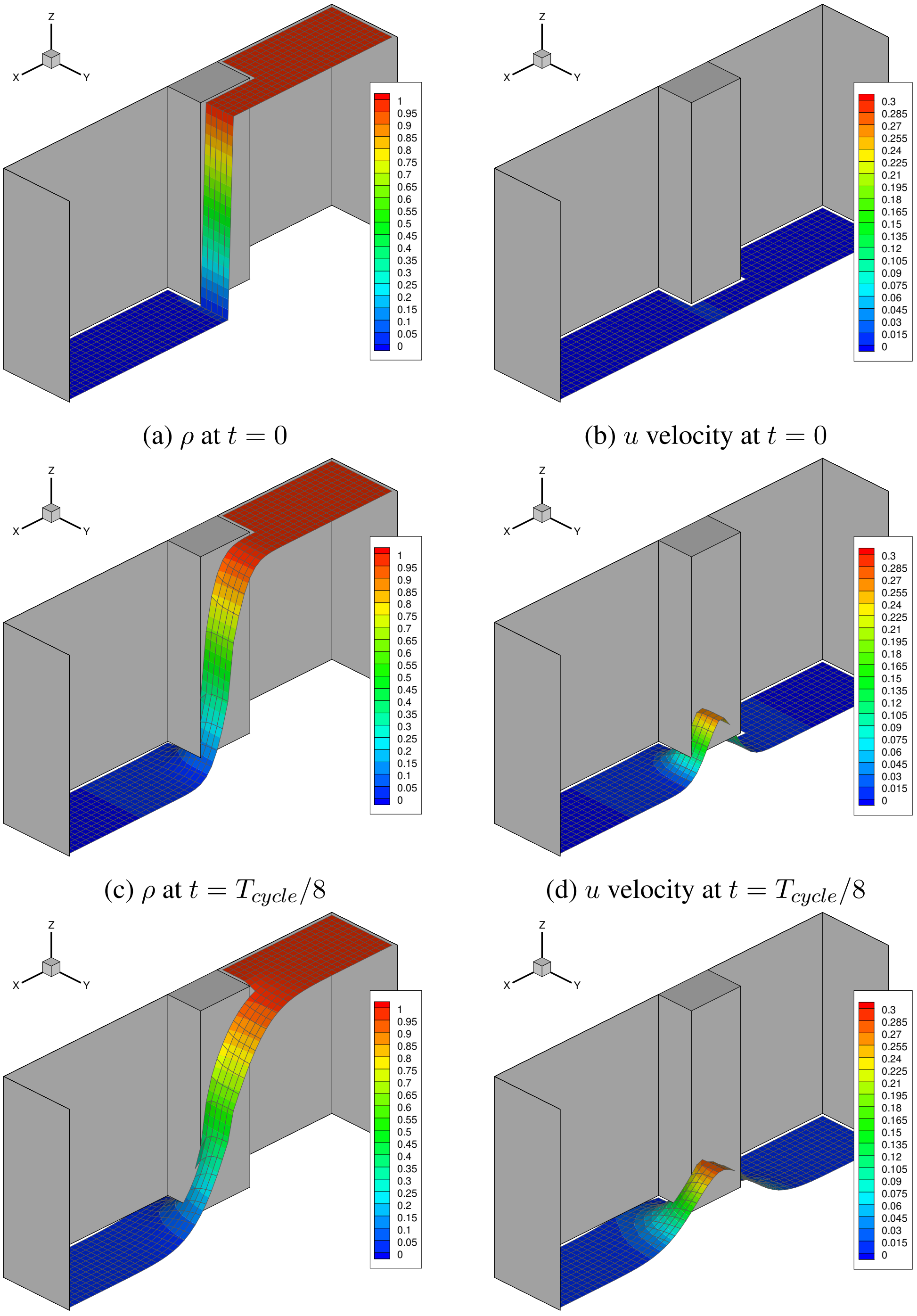
n_{DV}	n_{cycle}	Δc	$c_{k,min}$	$c_{k,max}$	T_{cycle}	ave. Δt
16	49	1.000	0.5000	7.5000	2.0	0.04167
32	213	0.500	0.2500	7.7500	4.0	0.01887
64	825	0.250	0.1250	7.8750	8.0	0.00971
128	3327	0.125	0.0625	7.9375	16.0	0.00481

In the time integration method employed, the underlying idea is to advance each of the components of the distribution functions corresponding to the different discrete velocities in the DVM at its 'CFL=1' condition. This way, data 'streams' from one lattice site to a neighbouring site. In the original DVM method for kinetic Boltzmann equations this is not the case. Specifically, for a typical time step used, the CFL condition on the largest eigenvalue (here, directly related to largest discrete-velocity used) automatically leads to a situation where distribution components corresponding to the smaller discrete-velocity values (in magnitude) only advance by a small fraction of a lattice spacing each time step. The current time-integration method can be regarded as a modified reservoir method with significant alterations to the original method by Alouges et al.[12]. Specifically, the **reservoir values** and **counters** are omitted, such that the time-integration accuracy is not affected. However, time-accurate output can only be output at certain instances in the time-integration process. In the **reservoir method** forming the basis on the current time-integration method, **reservoir values** are used in the interpolation of data between cells when output is required at a time step that does not correspond to the last step in a cycle. To facilitate the implementation as quantum algorithm, this type of reservoirs are actually not stored and when output at a time step which does not coincide with the end of a cycle is required, the interpolation of data from a location in between two neighboring grid points is omitted. This will have a small impact on the accuracy of the output, i.e. for certain steps within a cycle the output will not benefit from the smoothing effect of the data interpolation and will therefore be more likely to involve small oscillations. In the modified reservoir method used here, we can identify a cycle during which data for each discrete velocity gets updated at least once, i.e. this cycle involves a time $T_{cycle} = 1/c_{k,min}$, with $c_{k,min}$ representing the smallest discrete velocity in absolute value. The number of time-steps depends on the eigenvalue spectrum considered. Table 1 shows the details of one cycle in the reservoir scheme for a one-dimensional problem with velocity space bounds $\pm 8\sqrt{2RT_r}$. For the smallest number of discrete velocities, 49 time steps constitute one cycle. It can be seen that for each doubling of the number of discrete velocities, the number of steps per cycle grows approximately by a factor 4. In all cases, the average non-dimensional time step Δt is significantly smaller than the CFL limit imposed by the largest discrete velocity, i.e. $\Delta x / c_{k,max}$ with $c_{k,min}$ representing the largest discrete velocity in absolute value.

Note: because of the 'CFL=1' condition on the propagation of each distribution function component, the 'streaming' operations described previously can be employed.

3.5 Example: free molecular flow escaping from rectangular container

As representative unsteady test case, the time-dependent flow out of a rectangular container is considered. A 64×16 uniformly spaced mesh was used. Figure 8(a) shows the initial condition, where a uniform gas density is assumed within the container as well as in the rectangular throat, while the rest of the domain is initialized with a vacuum. One half of the overall domain is considered, with a symmetry boundary condition applied on the 'open' right-hand side of shown domain. The initial density and temperature are used as reference values, so that initially gas density $\rho = 1$ and $T = 1$ in the reservoir. The velocity-space discretization employs a uniform 128×128 mesh, with boundaries $[-4, 4]^2$ scaled with most probable speed at reference temperature. The non-dimensional time $T_{cycle} = 32$ for this integration, i.e. within one cycle a particle travelling at the most probable speed will cover $32\Delta x$. Figure 8 shows the flow developing between $T = 0$ and $t = T_{cycle}/4$, while the development up to $t = T_{cycle}/2$ is shown in Figure 9. For all time instances, the non-dimensional density and $u-$velocity are shown. As expected, the flow evacuates the container, forming a jet coming out of the throat. Also, a rarefaction wave runs in the opposite direction into the reservoir. Because of the collisionless nature of the gas, no shock waves are formed. Furthermore, it can be seen that for the 128×128 velocity-space mesh, the output at timesteps not coinciding with the end of a cycle does not lead to significant oscillations in the gas density and velocity field. Therefore, the simplification made to facilitate implementation on a quantum computer is clearly acceptable.



(a) ρ at $t = 0$

(b) u velocity at $t = 0$

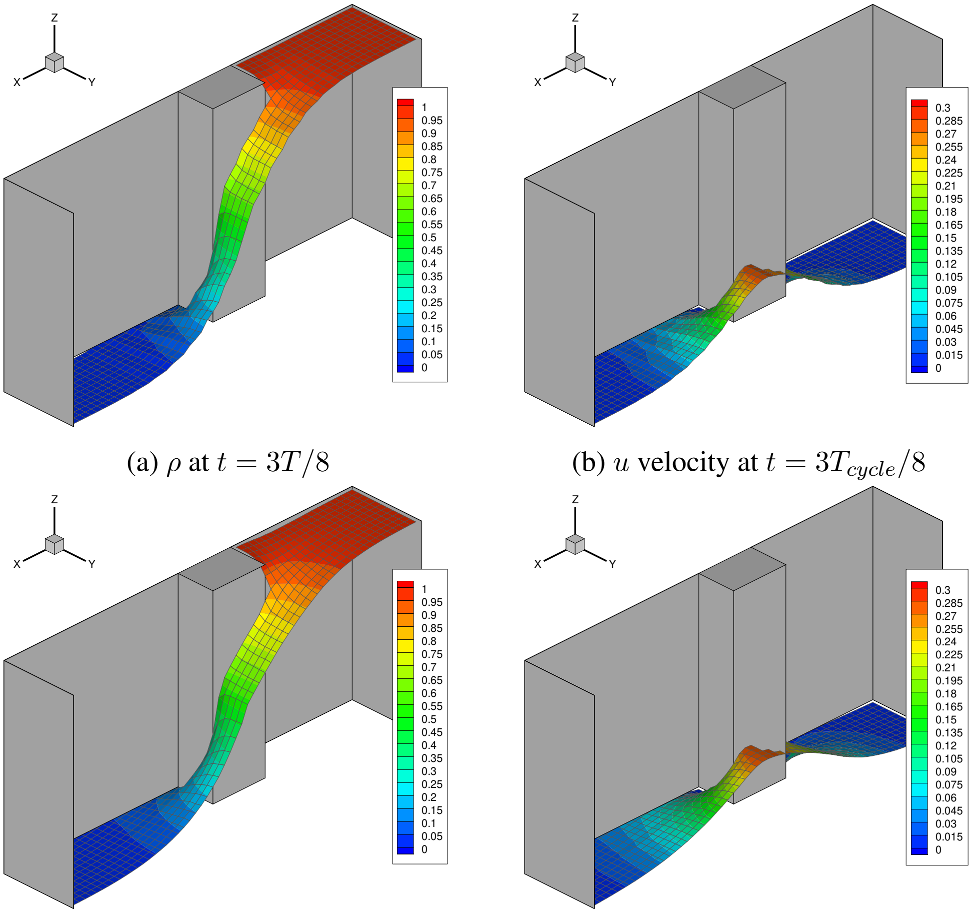
(c) ρ at $t = T_{cycle}/8$

(d) u velocity at $t = T_{cycle}/8$

(e) ρ at $t = T_{cycle}/4$

(f) u velocity at $t = T_{cycle}/4$

Figure 8: Free-molecular flow escaping out of rectangular container in two-dimensional domain simulated with quantum algorithm for collisionless Boltzmann equation. A 64×16 uniformly spaced mesh was used. DVM with 128×128 discrete-velocities. Initial gas region has ρ and T at 1.0. Density is non-dimensionalized with initial density and speed with most probably molecular speed at initial temperature.



(a) ρ at $t = 3T/8$

(b) u velocity at $t = 3T_{cycle}/8$

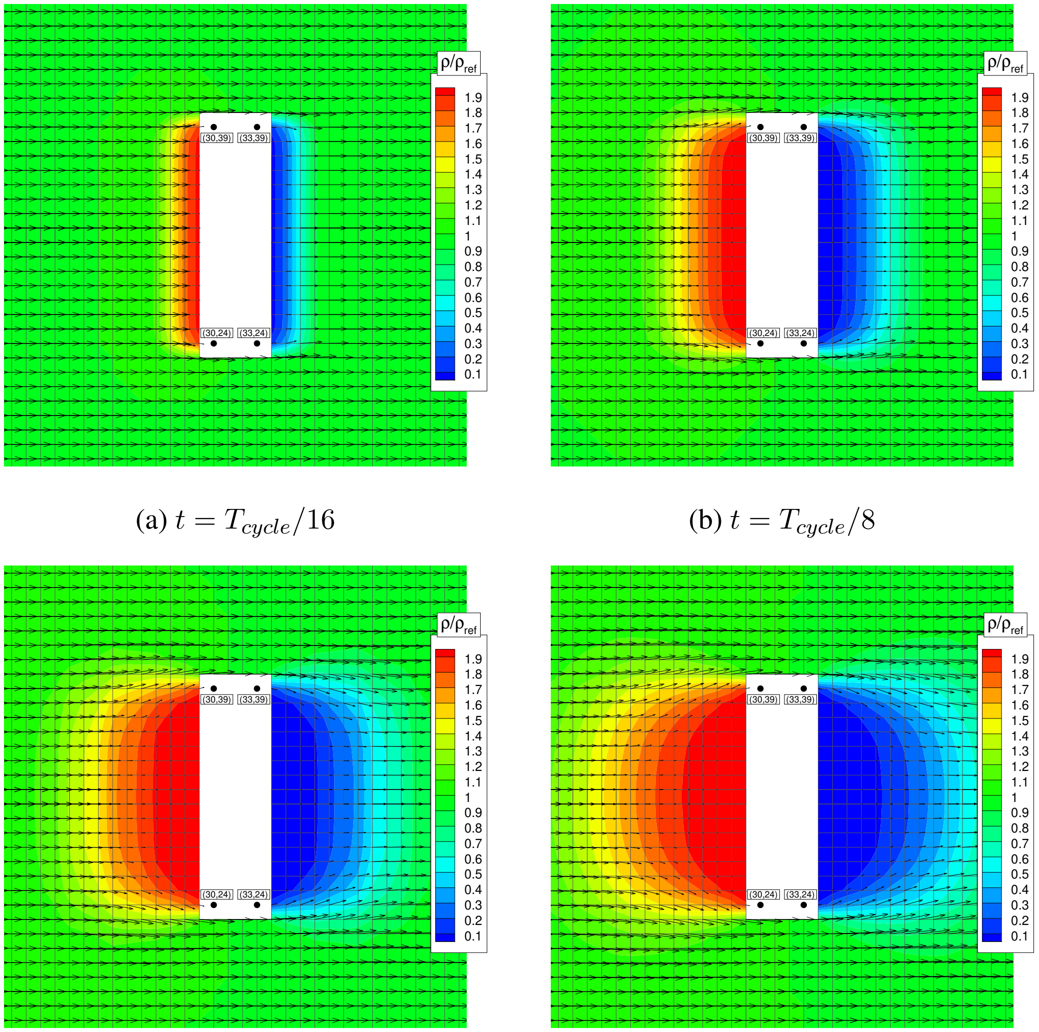
(c) ρ at $t = T/2$

(d) u velocity at $t = T_{cycle}/2$

Figure 9: Free-molecular flow escaping out out of rectangular container in two-dimensional domain simulated with quantum algorithm for collisionless Boltzmann equation. A 64×16 uniformly spaced mesh was used. DVM with 128×128 discrete-velocities. Initial gas region has ρ and T at 1.0. Density is non-dimensionalized with initial density and speed with most probably molecular speed at initial temperature.

4.0 IMPOSING SPECULAR-REFLECTION BOUNDARY CONDITIONS

The Mach 2 free-molecular flow over a rectangular body is considered here as an example[2]. The velocities are non-dimensionalized by the reference velocity ($\sqrt{2RT_\infty}$, most-probable molecular speed at free-stream temperature T_∞), so that Mach 2 corresponds to a non-dimensional speed of $2 \times \sqrt{\gamma RT_\infty}/\sqrt{2RT_\infty} = \sqrt{2\gamma}$. The flow field is initially (at time $T = 0$) set at free stream conditions: $(u_\infty, v_\infty)^T = (\sqrt{2\gamma}, 0)^T =$. Since we are considering a monatomic gas, $\gamma = 5/3$. Figure 10 shows the time evolution of the flow through the 1st quarter of the update cycle, showing the formation of a stagnation zone ahead of the rectangular and a low-density wake region behind it.



(a) $t = T_{cycle}/16$

(b) $t = T_{cycle}/8$

(c) $t = 3T_{cycle}/16$

(d) $t = T_{cycle}/4$

Figure 10: Free-molecular Mach 2 flow field around blunt body, 64×64 discrete-velocity space. Specular-reflection boundary conditions are applied. Results show evolution of flow field at 4 time instances following initialization with uniform free-stream flow.

4.1 Worked example - rectangular body - height 16 and width 4 lattice spacings

In the example shown in Figure 10, a 64×64 uniformly-space domain with periodic boundary conditions in x- and y-direction on the 4 domain edges is considered. The body has a height of 16 times the lattice spacing and a width of 4 times the lattice spacing. The body is located in the centre of the square computational domain. In the example shown, a 64×64 discrete-velocity mesh was used in the DVM. For reference, the figure shows the (i, j) index range involved in imposing the boundary conditions by defining 'halo' lattice points 'within' the solid body. In these 'halo' lattice points, the velocity component normal to solids walls is reflected (i.e. changes sign) while tangential component remains unchanged.

4.1.1 Front and rear of solid body

For the front and rear parts of the rectangular solid body, the u-velocity in x-direction needs changing sign for the 'halo' lattice points within the body along the front and rear surfaces. For the selected size of rectangular body, the lattice sites with $i = 30$ and $j \in [24, 39]$ represent the halo sites closest to front surface, while the lattice sites with $i = 30$ and $j \in [24, 39]$ represent the halo sites closest to rear surface. In the example quantum-circuit implementation for the solid b.c. in x-direction (for clarity, a reduced-size DVM mesh with 16×16 discrete velocities was used), shown in Figure 11, the required sign changes are introduced in 4 steps:

1 : conditional on $|x5|x4|x3|x2|x1\rangle = |01111\rangle$ ($i \in [30, 31]$) and $|y5|y4|y3\rangle = |011\rangle$ ($j \in [24, 31]$);

2 : conditional on $|x5|x4|x3|x2|x1\rangle = |01111\rangle$ ($i \in [30, 31]$) and $|y5|y4|y3\rangle = |100\rangle$ ($j \in [32, 39]$);

3 : conditional on $|x5|x4|x3|x2|x1\rangle = |10000\rangle$ ($i \in [32, 33]$) and $|y5|y4|y3\rangle = |011\rangle$ ($j \in [24, 31]$);

4 : conditional on $|x5|x4|x3|x2|x1\rangle = |10000\rangle$ ($i \in [32, 33]$) and $|y5|y4|y3\rangle = |100\rangle$ ($j \in [32, 39]$).

For the 16×16 discrete-velocity mesh, $|u3|u2|u1|u0\rangle = |0000\rangle$ corresponds to the largest (in magnitude) negative u-velocity and $|u3|u2|u1|u0\rangle = |1111\rangle$ corresponds to the largest (in magnitude) positive u-velocity, with all other u discrete velocities uniformly-spaced in between. Applying a controlled NOT on each u-qubit then imposes a sign change.

4.1.2 Upper and lower surfaces of solid body

For the upper and lower surfaces of the rectangular solid body, v-velocity in y-direction needs changing sign for the 'halo' lattice points within the body, similar to the approach detailed for x-direction previously. For the selected size of rectangular body, the lattice sites with $j = 24$ and $i \in [30, 33]$ represent the halo sites closest to lower surface, while the lattice sites with $j = 39$ and $i \in [30, 33]$ represent the halo sites closest to upper surface. In the example quantum-circuit implementation for the solid b.c. in y-direction (for clarity, a reduced-size DVM mesh with 16×16 discrete velocities was used), shown in Figure 12, the required sign changes are introduced in 4 steps:

1 : conditional on $|x5|x4|x3|x2|x1\rangle = |01111\rangle$ ($i \in [30, 31]$) and $|y5|y4|y3|y2|y1\rangle = |01100\rangle$ ($j \in [24, 25]$);

2 : conditional on $|x5|x4|x3|x2|x1\rangle = |10000\rangle$ ($i \in [32, 33]$) and $|y5|y4|y3|y2|y1\rangle = |01100\rangle$ ($j \in [24, 25]$);

3 : conditional on $|x5|x4|x3|x2|x1\rangle = |01111\rangle$ ($i \in [30, 31]$) and $|y5|y4|y3|y2|y1\rangle = |10011\rangle$ ($j \in [38, 39]$);

4 : conditional on $|x5|x4|x3|x2|x1\rangle = |10000\rangle$ ($i \in [32, 33]$) and $|y5|y4|y3|y2|y1\rangle = |10011\rangle$ ($j \in [38, 39]$).

For the 16×16 discrete-velocity mesh, $|v3|v2|v1|v0\rangle = |0000\rangle$ corresponds to the largest (in magnitude) negative v-velocity and $|v3|v2|v1|v0\rangle = |1111\rangle$ corresponds to the largest (in magnitude) positive v-velocity, with all other v discrete velocities uniformly-spaced in between. Applying a controlled NOT on each v-qubit then imposes a sign change.

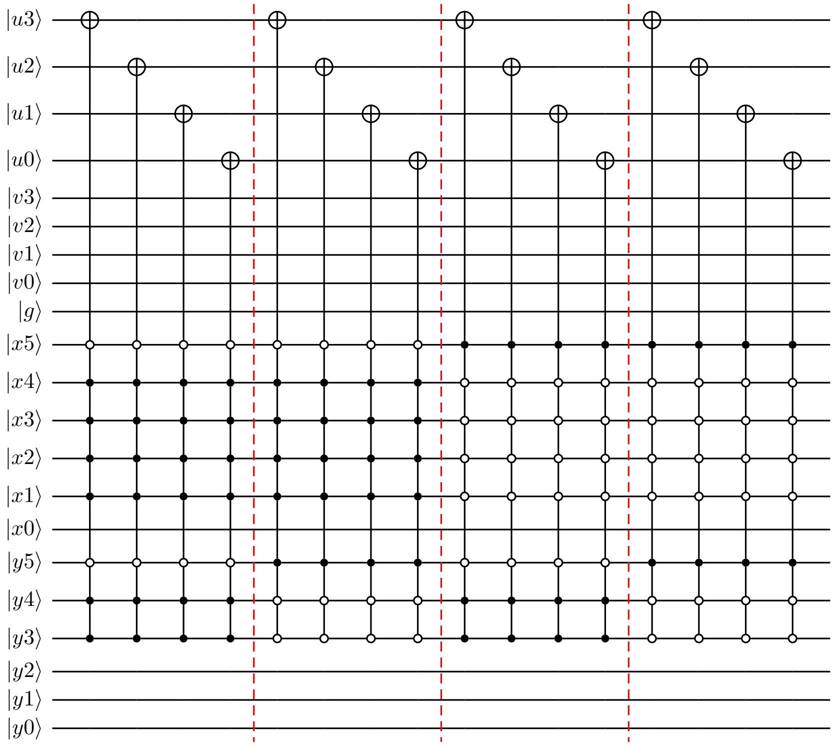


Figure 11: Imposing specular-reflection boundary condition on streaming in $x-$direction. DVM method with 16×16 discrete velocities on 64×64 uniformly-spaced mesh. Conditionals on $|x5|x4|x3|x2|x1\rangle$ and $|y5|y4|y3\rangle$ are used to define rectangular body with 16 lattice spacing height and a width of 4 lattice spacings.

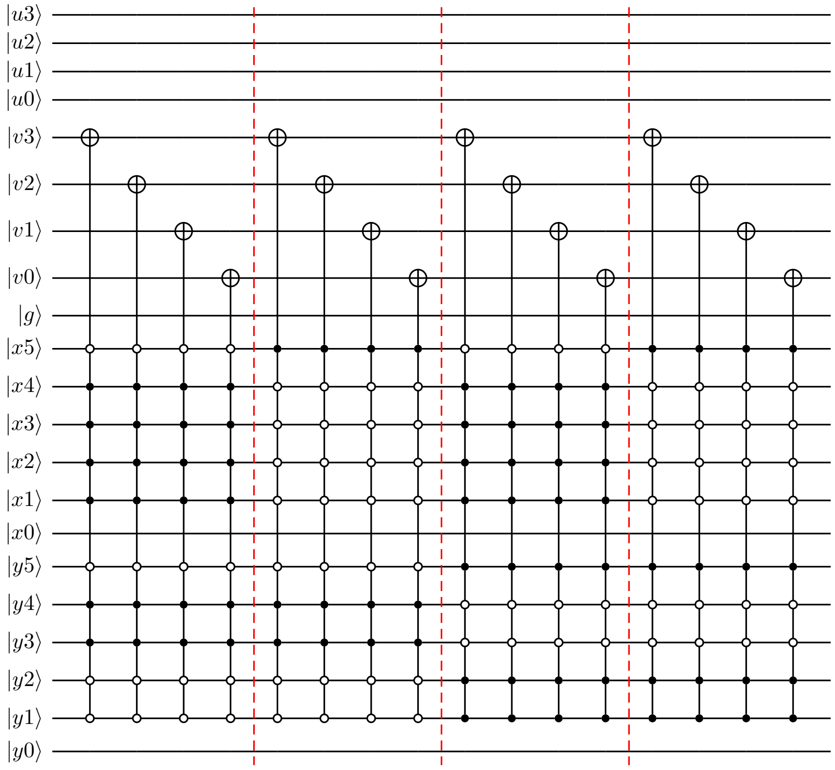


Figure 12: Imposing specular-reflection boundary condition on streaming in $y-$direction. DVM method with 16×16 discrete velocities on 64×64 uniformly-spaced mesh. Conditionals on $|x5|x4|x3|x2|x1\rangle$ and $|y5|y4|y3\rangle$ are used to define rectangular body with 16 lattice spacing height and a width of 4 lattice spacings.

4.2 Worked example with fewer control qubits

For the worked example detailed in Section 4.1, the quantum-circuit implementation of the specular-reflection boundary conditions involved multiple-control Toffoli gates to 'swap' the sign of the molecular velocities. These controls originate from the need to restrict the sign-changing operations to selections of 'halo' lattice sites within the considered solid body. Clearly, such multiple-control Toffoli will not exist as 'native' gates implemented on real quantum processors. As a first step to a quantum-circuit transformed for 'real-world' quantum hardware, this section illustrates how for 2 of the 4 sub-circuits in the quantum-circuit implementation for x-direction boundary conditions (shown in Figure 11), the number of control qubits can be reduced by the introduction of 3 additional ancillae qubits. This transformation is shown in Figure 13. Clearly, similar transformation can be derived for the remaining 2 sub-circuits in Figure 11 as well as for the y-direction boundary condition imposition shown in Figure 12.

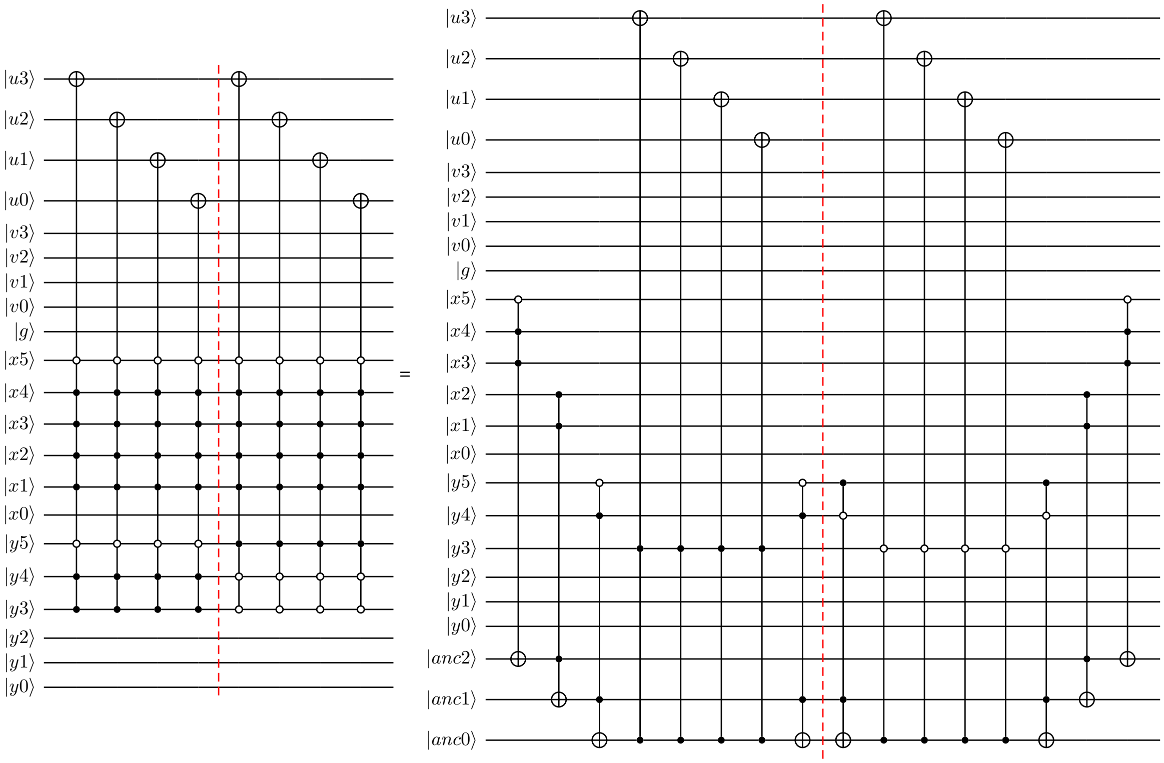


Figure 13: Imposing specular-reflection boundary condition on streaming in $x-$direction. Illustration of how circuit can be transformed to gates with fewer control qubits. DVM method with 16×16 discrete velocities on 64×64 uniformly-spaced mesh. Conditionals on $|x5|x4|x3|x2|x1\rangle$ and $|y5|y4|y3\rangle$ are used to define rectangular body with 16-spacing height and 4-spacing width.

4.3 Exercise 1 - rectangular body - height 16 and width 8 lattice spacings

As a first exercise, consider a rectangular body with height 16 lattice spacings (as in worked example), but now with an increased with of 8 lattice spacings. As before, imposing the specular-reflection boundary conditions in x- and y-directions can be implemented in quantum circuits for a 16×16 discrete-velocity mesh and the previously-used 64×64 periodic computational domain, in 'sub-circuits' designed to swap sign of selected groups of 'halo' lattice sites.

Question: how do quantum circuits previously shown for imposing boundary conditions in the worked example need to be changed to account for this change in body width?

4.4 Exercise 2 - rectangular body - height 4 and width 32 lattice spacings

As second example consider a 'flat-plate' like body with width 32 lattice spacings and height equal to 4 lattice spacing. For the x- and y-direction specular-reflection boundary conditions, the quantum circuits can again be based on the use of sub-circuits targeting selected groups of 'halo' lattice sites 'within' the solid body.

Task: work out quantum-circuit implementations for the x- and y-direction specular-reflection boundary conditions. Hint: first work out a suitable division in groups for the 'halo' lattice sites that need changing sign.

5.0 FUTURE DEVELOPMENTS

The present set of notes detailed quantum lattice-based modellings for the collisionless Boltzmann equation. A key feature of this kinetic model is the **linearity** since the complex, non-linear term of the right-hand side of the ’full’ Boltzmann equation representing molecular collisions was eliminated. Also, for accurate results for free-molecular flows as shown in the present notes, it should be remembered that the discrete-velocity method (DVM) used here involved many more discrete velocities than used in Lattice Boltzmann Method (LBM). This difference stems from the fact that LBM was developed for continuum flows of (nearly) incompressible flows, while Boltzmann kinetic model are typically used for rarefied-flow applications where such a continuum-flow assumption is not valid. Despite this observations, the work detailed in the present set of notes can still be regarded as first steps towards DVM implementation of non-linear kinetic-Boltzmann models (e.g. Bhatnagar-Gross-Krook, BGK[11]) as well as illustration of how streaming steps in the LBM can effectively be implemented.

In terms of quantum lattice-based modelling, a number of future developments can be highlighted:

I. Quantum Lattice-Based modelling that uses the link between hydrodynamics and the dynamics of quantum systems[13]. More specifically, the ’streaming operation’ used for the Boltzman equation shown in these notes is an example of a Discrete Time Quantum Walk (DTQW)[14], and a similar approach can be used to simulate the dynamics of a classical Dirac field coupled to electromagnetic fields. As an example, Molfetta and co-workers have been working on generalized Madelung transformations, introduced to map the classical Dirac dynamics into potential flows of nonlinear, relativistic, quantum, charged fluids with spin immersed in electromagnetic fields. Their work shows that Dirac dynamics thus connects quantum walks to hydrodynamics (as discussed elsewhere in this Lecture Series);

II. Extension of quantum lattice-based models derived initially for linear models to (weakly) non-linear LBM models using Carlemann-type linearization. Recently progress in this direction has been made[3]. Application to high-Reynolds number flows using D2Q9 and D3Q27 LBM models remains an outstanding research challenge;

III. Lattice-based quantum computing approaches to modelling fluids by and large employ quantum amplitude-based encoding to benefit from the quantum parallelism as well as the ’storage’ space for degrees-of-freedom for considered fluid problem that grows exponentially with the number of qubits. In the quantum circuit model, non-linear terms then form a key challenge. This challenge can be (partly) overcome by changing to quantum computational basis encoding. However, this introduces a series of new challenges: maintaining quantum speed-up as well as the need for efficient transformations between amplitude-based and computational-based encodings[15].

Clearly this list reflects the author’s experience and insight and inevitably will omit some important work related to fluid dynamics modelling using quantum computers.

ACKNOWLEDGMENTS

The author would like to acknowledge the support of the University of Glasgow to B.N. Todorova during the work on the algorithms for the collisionless Boltzmann equation.

REFERENCES

[1] Budinski, L., “Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method,” *Quantum Information Processing*, Vol. 20, No. 2, 2021, pp. 57.

[2] Todorova, B. and Steijl, R., “Quantum Algorithm for the collisionless Boltzmann equation,” *J. Comp. Phys.*, Vol. 409, 2020, pp. 109347.

[3] Itani, W. and Succi, S., “Analysis of Carleman Linearization of Lattice Boltzmann,” *Fluids*, Vol. 7, No. 1, 2022.

[4] Yepez, J., “Quantum lattice-gas model for computational fluid dynamics,” *Phys. Rev. E*, Vol. 63, 2001, pp. 046702.

[5] Berman, G., Ezhov, A., Kamenev, D., and Yepez, J., “Simulation of the diffusion equation on a type-II quantum computer,” *Phys. Rev. A*, Vol. 66, 2002, pp. 012310.

[6] Steijl, R. and Barakos, G., “Parallel evaluation of quantum algorithms for computational fluid dynamics,” *Computers & Fluids*, Vol. 173, 2018, pp. 22–28.

[7] Douglas, B. and Wang, J., “Efficient quantum circuit implementation of quantum walks,” *Phys. Rev. A*, Vol. 79, 2009, pp. 052335.

[8] Fillion-Gourdeau, F., MacLean, S., and Laflamme, R., “Algorithm for the solution of the Dirac equation on digital quantum computers,” *Phys. Rev. A*, Vol. 95, Apr 2017, pp. 042343.

[9] Fillion-Gourdeau, F. and Lorin, E., “Simple digital quantum algorithm for symmetric first-order linear hyperbolic systems,” *Numerical Algorithms*, Vol. 82, 2019, pp. 1009–1045.

[10] Vincenti, W. and Kruger, C., *Introduction to physical gas dynamics*, John Wiley and Sons, New York, 2nd ed., 1967.

[11] Cercignani, C., *The Boltzmann Equations and its Applications*, Springer-Verlag, New York, 2nd ed., 1987.

[12] Alouges, F., De Vuyst, F., Le Coq, G., and Lorin, E., “The reservoir technique: a way to make Godunov-type schemes zero or very low diffuse. Application to Colella–Glaz solver,” *European Journal of Mechanics - B/Fluids*, Vol. 27, No. 6, 2008, pp. 643 – 664.

[13] Arrighi, P., Nesme, N., and Forets, M., “The Dirac equation as a quantum walk: higher dimensions, observational convergence,” *Journal of Physics A: Mathematical and Theoretical*, Vol. 47, No. 46, nov 2014, pp. 465302.

[14] Venegas-Andraca, S., “Quantum walks: a comprehensive review,” *Quantum Information Processing*, Vol. 11, No. 5, 2012, pp. 1015–1106.

[15] Steijl, R., “Quantum algorithms for nonlinear equations in fluid mechanics. In: Zhao, Y. (ed.) Quantum Computing and Communications. IntechOpen:London. ISBN 9781839681332 (doi: 10.5772/intechopen.95023),” 2022.

APPENDIX - SOLUTION TO EXERCISE 1 AND EXERCISE 2

Exercise 1

The specular-reflection boundary conditions in x- and y-directions can be implemented in quantum circuits for a 16×16 discrete-velocity mesh and the previously-used 64×64 periodic computational domain, by applying multiply-controlled NOTs applied to $|u3|u2|u1|u0\rangle$ (change of sign for u-velocity) and multiply-controlled NOTs applied to $|v3|v2|v1|v0\rangle$ (change of sign for v-velocity), where a combination of (some of) the qubits defining x-coordinate ($|x5|x4|x3|x2|x1|x0\rangle$) and (some of) the qubits defining y-coordinate ($|y5|y4|y3|y2|y1|y0\rangle$) acts as control qubits. As before, the quantum circuit implementation is best constructed in terms of 'sub-circuits' designed to swap sign of velocity components in selected groups of 'halo' lattice sites. For the solid body with height equal to 16 lattice spacings and width 8 lattice spacings, the following grouping can be used for boundary conditions in x-direction:

1 : NOTs on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |01110\rangle$ ($i \in [28, 29]$) and $|y5|y4|y3\rangle = |011\rangle$ ($j \in [24, 31]$);

2 : NOTs on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |01110\rangle$ ($i \in [28, 29]$) and $|y5|y4|y3\rangle = |100\rangle$ ($j \in [32, 39]$);

3 : NOTs on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |10001\rangle$ ($i \in [34, 35]$) and $|y5|y4|y3\rangle = |011\rangle$ ($j \in [24, 31]$);

4 : NOTS on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |10001\rangle$ ($i \in [34, 35]$) and $|y5|y4|y3\rangle = |100\rangle$ ($j \in [32, 39]$).

Similarly for the specular-reflection boundary conditions in y-direction:

1 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |01110\rangle$ ($i \in [28, 29]$) and $|y5|y4|y3|y2|y1\rangle = |01100\rangle$ ($j \in [24, 25]$);

2 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |10001\rangle$ ($i \in [34, 35]$) and $|y5|y4|y3|y2|y1\rangle = |01100\rangle$ ($j \in [24, 25]$);

3 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |01110\rangle$ ($i \in [28, 29]$) and $|y5|y4|y3|y2|y1\rangle = |10011\rangle$ ($j \in [38, 39]$);

4 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |10001\rangle$ ($i \in [34, 35]$) and $|y5|y4|y3|y2|y1\rangle = |10011\rangle$ ($j \in [38, 39]$).

Exercise 2

Although the shape of the solid body is now quite different, the quantum circuit implementation can still be constructed in terms of 'sub-circuits' designed to swap sign of velocity components in selected groups of 'halo' lattice sites. For the solid body with a width equal to 32 lattice spacings and height equal to 4 lattice spacings, the following grouping can be used for boundary conditions in x-direction:

1 : NOTs on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |01000\rangle$ ($i \in [16, 17]$) and $|y5|y4|y3|y2|y1\rangle = |01111\rangle$ ($j \in [30, 31]$);

2 : NOTs on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |01000\rangle$ ($i \in [16, 17]$) and $|y5|y4|y3|y2|y1\rangle = |10000\rangle$ ($j \in [32, 33]$);

3 : NOTs on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |10111\rangle$ ($i \in [46, 47]$) and $|y5|y4|y3|y2|y1\rangle = |01111\rangle$ ($j \in [30, 31]$);

4 : NOTS on $|u3|u2|u1|u0\rangle$ conditional on $|x5|x4|x3|x2|x1\rangle = |10111\rangle$ ($i \in [46, 47]$) and $|y5|y4|y3|y2|y1\rangle = |10000\rangle$ ($j \in [32, 33]$).

And for the specular-reflection boundary conditions in y-direction:

1 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4\rangle = |01\rangle$ ($i \in [16, 31]$) and $|y5|y4|y3|y2|y1\rangle = |01111\rangle$ ($j \in [30, 31]$);

2 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4\rangle = |10\rangle$ ($i \in [32, 47]$) and $|y5|y4|y3|y2|y1\rangle = |10000\rangle$ ($j \in [32, 33]$);

3 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4\rangle = |01\rangle$ ($i \in [16, 31]$) and $|y5|y4|y3|y2|y1\rangle = |01111\rangle$ ($j \in [30, 31]$);

4 : NOTs on $|v3|v2|v1|v0\rangle$ conditional on $|x5|x4\rangle = |10\rangle$ ($i \in [32, 47]$) and $|y5|y4|y3|y2|y1\rangle = |10000\rangle$ ($j \in [32, 33]$).