Quantum Information Processing (2021) 20:57
https://doi.org/10.1007/s11128-021-02996-3

Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method

Ljubomir Budinski[1]

Received: 23 July 2020 / Accepted: 12 January 2021 / Published online: 5 February 2021

Abstract
A novel quantum algorithm for solving advection–diffusion equation by the lattice Boltzmann method is proposed. The presented quantum algorithm is composed of two major segments. In the first segment, equilibrium distribution function, presented in the form of a non-unitary diagonal matrix, is quantum circuit implemented by using a standard-form encoding approach. For the second segment, the quantum walk procedure as a method for implementing the propagation step is applied. The constructed algorithm is presented as a series of single- and two-qubit gates, as well as multiple-input controlled-NOT gates. In order to demonstrate the validity of the proposed quantum algorithm, the unsteady one-dimensional (1D) and two-dimensional (2D) advection–diffusion equations are solved by using the IBM's quantum computing software development framework Qiskit, while the analytic solution and the classic code are used for verification. Finally, the complexity analysis and directions for future work are discussed.

Keywords Quantum computing · Lattice Boltzmann method · Advection–diffusion equation

1 Introduction

Quantum computing has drawn a lot of attention in the last few years. Twenty years following the foundations of quantum computing have been established by discovery of Grover search algorithm [1], phase (eigenvalue) estimation algorithm [2], quantum Fourier transformation [3] and Shor's algorithm for integer factorization [4], entirely new researcher areas that can benefit from quantum computation emerged thanks to the quantum computing simulators and processors introduced by companies like IBM,

✉ Ljubomir Budinski
ljubab@uns.ac.rs

[1] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, Novi Sad 21000, Serbia

Rigetti and Microsoft. Available through the cloud services, these quantum processors stimulated researchers and computational enthusiast of various fields of interest to work on developing new quantum procedures and techniques which can be applied on quantum devices. New quantum algorithms were discovered, yielding potential quantum speed-up and applications in various fields of science such as linear algebra [5–9], quantum chemistry [10], optimization [11,12] or machine learning [13–15]. However, the currently available quantum devices are error-prone mostly due to the noise produced by the fact that the physical and natural systems do not exist in isolation (Noisy Intermediate-Scale Quantum devices—NISQ). As a consequence, the concept of variational quantum computing (VQC), where the evaluation of the cost function is delegated to a quantum computer while the optimization of variational parameters is performed on a conventional classical computer, attracted considerable interest. While the VQC is proven to be most suitable for machine learning [16–19] and chemistry [10,20], significant research has been done in the area of linear algebra [21,22] and optimization [23–25] as well.

From engineering perspective, complex processes involving time-space dynamics are described mainly by the differential equations (DE). Generally, solving differential equations by classical computers is a hard problem, in particular when the size of the configuration space is large (fluid dynamics). A possible way to overcome the above difficulty is to utilize quantum computing. In case of inhomogeneous linear differential equations, Berry [26] and Childs et al. [27] first formed a system of linear equations by discretizing the differential equation using the finite difference method (FDM), which then are solved by applying the quantum algorithm to the linear systems of equations. Examples of this approach for the Vlasov equation are presented by Costa et al. [28], while in case of Poisson equation Cao et al. [29] and Wang et al. [30] applied HHL [5] algorithm for solving system of linear equations. Another approach, where the Taylor expansion of the analytical solution of the linear differential equations (LDE) is described by the quantum states and the corresponding operators, is firstly proposed by the Berry et al. [31], while application on 4-dimensional LDE with a 4×4 non-unitary matrix is done by Xin et al. [32]. On the other hand, since most of the differential equations used for describing the physical phenomena are essentially nonlinear (Navier–Stokes equations), first attempt to solve systems of nonlinear differential equations, whose nonlinear terms are polynomials, was investigated by Leyton and Osborne [33]. However, since quantum mechanics is represented by linear operators where the time evolution of the quantum state vector is described by the linear differential equation, the presented algorithm turns out to be too ambitious, i.e., it is possible to obtain an algorithm that is far more efficient than the proposed one. This highlights very important fact where the linearity of the quantum state propagation is a perquisite for developing the quantumly efficient algorithm.

Quantum algorithm for solving the advection–diffusion equation is presented in this work. Since the ADE is actually partial differential equation used to describe transport phenomena in time-space domain, the solution of this kind of equation is, in general obtained by using some of the traditional methods from the numerical analysis (FDM). However, in this paper, the lattice Boltzmann method (LBM) as a numerical technique for an indirect solution of flow equations through a microscopic approach to a macroscopic phenomenon [34–37] is utilized. The main reason for using this implicit

numerical procedure over previously mentioned FDM lies primarily in its simple mathematical structure, where it consists of simple arithmetic calculations of only one single variable, the microscopic distribution function. It is suitable for flows in complex geometry such as flows through porous media for the straightforward implementation of the boundary conditions providing an opportunity for easy simulation of complex flows. The first attempt to solve the processes of fluid dynamics by quantum computing was made by Yepez [38], Berman et al. [39] and Micci and Yepez [40], and mainly involves lattice-gas models and type II quantum computers. Type II machines consist of a number of small type I quantum computers ('pure' quantum computers called nodes) with as few as two qubits in each, connected by classical communications channels carrying bits instead of qubits [41]. However, it is known that lattice-gas model suffers from various problems, like non-isotropic advection, violation of Galilei invariance, and noise due to the usage of Boolean variables. Hence, most recent work on solving the collisionless Boltzmann equation is performed by N.Todorova and Steijl [42], where the quantum circuit implementations for the advection step as a quantum walk process are presented. The most recent attempt to derive quantum algorithm for one-dimensional Navier–Stokes equations by applying quantum amplitude estimation algorithm (QAEA) for solving ordinary differential equations (ODEs) is done by Gaitan [43]. However, it is not clear from the paper how this approach can be further extended to the two- and three-dimensional space and at what cost.

In this work, a quantum algorithm for transport phenomena (ADE) utilizing the complete lattice Boltzmann equation is presented. The non-unitary collision operator is solved by using the standard-form encoding approach [44], while the quantum walk is used for the propagation step. The quantum circuit for the collision and propagation operator is constructed from multiple controlled-NOT gates, as well as single qubit X and R_z gates, and two-qubit $SWAP$ gate. The constructed algorithm is implemented on IBM's open-source quantum computing software development framework Qiskit [45] and it is tested for the case of unsteady one-dimensional and two-dimensional heat transport. The main contribution of the present work is a quantum algorithm simulating the transport phenomena, solving the full lattice Boltzmann equation, i.e., both the collision and propagation step. Besides the re-normalization of the post-selected state vector preformed at the end of each time step, the entire computation within a particular time step is performed solely on the quantum processor. Furthermore, to be suitable for different mesh sizes, the scalability of the presented algorithm is also elaborated. To the best of the author's knowledge, the presented quantum algorithm is the very first algorithm related to fluid dynamics that solves the full lattice Boltzmann equation, which is also tested on the actual platform for quantum computation.

2 Mathematical formulation

2.1 The advection and diffusion equation

The advection–diffusion process is a process where both advection and diffusion take place simultaneously, and both phenomena are governed by the advection–diffusion

equation,

$$\frac{\partial \phi}{\partial t} + \frac{\partial \left(u_i \phi\right)}{\partial x_i} = \frac{\partial}{\partial x_i}\left(D \frac{\partial \phi}{\partial x_i}\right), \tag{1}$$

where ϕ is the depended variable (mass, momentum, energy, species, etc.), t is time, x_i is a Cartesian coordinate, u_i is the fluid velocity in the i-direction and D is the diffusion coefficient. The above equation is valid for general advection and diffusion phenomena, including both steady and unsteady situations.

2.2 The lattice Boltzmann method

The single relaxation time lattice Boltzmann equation [46] is formulated as

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = (1 - \omega)\, f_\alpha(\mathbf{x}, t) + \omega f_\alpha^{eq}, \tag{2}$$

where f_α is the particle distribution function along the α link, f_α^{eq} is the local equilibrium distribution function, $\mathbf{e}_\alpha$ is the particle velocity vector, $\mathbf{x}$ is the space vector defined by Cartesian coordinates, t is time, Δt is the time step and ω=$\Delta t/\tau$, where τ is the single relaxation time. Depending on the spatial dimensions being used, $\mathbf{x}$ is the space vector defined by Cartesian coordinates, i.e., $\mathbf{x} = (x)$ for one-dimensional space and $\mathbf{x} = (x, y)$ in two-dimensional space.

In this paper, two spatial lattice models for advection–diffusion equation have been considered. First, a 1D model with the D1Q2 configuration (in DnQm classification Dn stands for n dimensions, while Qm stands for m speeds) is used (Fig. 1). This arrangement uses just two velocity vectors, $e_1 = -e_2 = \Delta x/\Delta t$ for distribution functions f_1 and f_2, respectively, streaming along the links in opposite directions. Equality between two velocity vectors is imposed by the symmetry of the LBM. Furthermore, the crucial part of the lattice Boltzmann method is the local equilibrium distribution function [37], which is for advection–diffusion equation defined as

$$f_\alpha^{eq}(\mathbf{x}, t) = w_\alpha \phi(\mathbf{x}, t)\left(1 + \frac{e_\alpha \cdot \overrightarrow{u}}{c_s^2}\right), \tag{3}$$

where $\overrightarrow{u}$ is the advection velocity vector, c_s is the speed of sound and w_α is the weighting factor in the direction α. In case of a one-dimensional D1Q2 scheme, the weighting factors are $w_{1,2} = 0.5$, while for the speed of sound $c_s = 1$ is used.

For the two-dimensional problem, the D2Q5 scheme with rest particle is utilized (Fig. 2). This model consists of four velocity vectors and a rest particle, defined as

$$e_\alpha = \begin{cases} (0, 0), & \alpha = 0 \\ (\pm e_x, 0), & \alpha = 1, 2 \\ (0, \pm e_y), & \alpha = 3, 4 \end{cases} \tag{4}$$

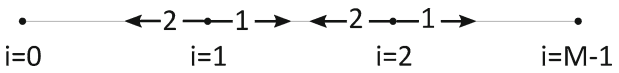

Fig. 1 D1Q2 lattice configuration

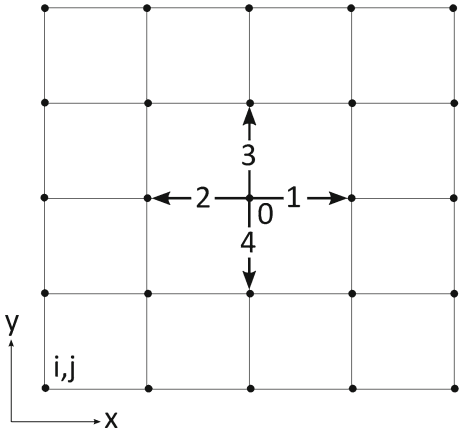

Fig. 2 D2Q5 lattice configuration

where $e_x = e_y = \Delta x(y)/\Delta t$. Furthermore, the equilibrium distribution function is solved by using Eq. (3), while the corresponding weight coefficients and the speed of sound are set to $w_0 = 2/6$, $w_{1,2,3,4} = 1/6$ and $c_s = 1/\sqrt{3}$, respectively. The advection velocity vector for 2D case is $\overrightarrow{u} = ui + vj$, where i and j are unit vectors along x and y direction. It should be noted that following the work of Zhou [47] parameter ω is set to 1.0 in both cases.

The solution procedure for the LBM, in general, consists of two major steps, collision and streaming. In the collision step, relaxation to local equilibrium is performed only

$$\hat{f}_\alpha(\mathbf{x}, t) = (1 - \omega)\, f_\alpha(\mathbf{x}, t) + \omega f_\alpha^{eq}, \tag{5}$$

while in the streaming step propagation of the relaxed distribution functions f_α along the links α is conducted as

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = \hat{f}_\alpha(\mathbf{x}, t). \tag{6}$$

The macroscopic variable $\phi\,(\mathbf{x}, t)$ at the end of each time step is calculated as the zero-order discrete moment of the particle distribution function f_α as

$$\phi\,(\mathbf{x}, t) = \sum_\alpha f_\alpha\,(\mathbf{x}, t)\,. \tag{7}$$

The recovery of the advection–diffusion equation Eq. (1) by applying the Chapman–Enskog expansion procedure can be found in [37].

3 The quantum algorithm

3.1 The D1Q2 model

The entire procedure of establishing the quantum algorithm for D1Q2 lattice Boltzmann model can be divided into four major steps: encoding, collision, propagation and macroscopic variable calculation. In the encoding segment, transformation of the initial variable $\phi(\mathbf{x}, 0)$ into a quantum state vector is performed by employing the normalization procedure and the encoding state protocol, while in the collision and propagation steps corresponding operations based on Eqs. (5) and (6) are utilized. In the last step, addition according to Eq. (7) is conducted, where the output state is used as the input quantum state for the new time step.

In the encoding step, two quantum registers, q and ancilla a are installed. In the first register, q, having the $\log_2(2M)$ qubits, the initial distribution of variable $\phi(\mathbf{x}, 0)$ in form of vector $\boldsymbol{\Phi} = [\phi_{1,0}, \ldots, \phi_{1,M-1}, \phi_{2,0}, \ldots, \phi_{2,M-1}]^T$, where the first index denotes the link α and the second index marks the location number of the lattice site, is encoded into the quantum state as

$$|\psi_0\rangle = |0\rangle_a \sum_{i=0}^{2M-1} \phi_{\alpha,i}/\|\phi\||i\rangle_q. \tag{8}$$

The $|i\rangle$ is the $2M$-dimensional computational basis state, while the $\|\phi\|$ is the Euclidean norm. To achieve this initial state, where the $|000\ldots\rangle$ state is adapted to be, conditionally speaking, a desired arbitrary state, the reverse iterative procedure proposed by Shende et al. [48] is used as part of the built-in functions of the Qiskit [45] quantum framework. It should be noted that this initialization of the desired state is performed only once, at the beginning of the simulation.

The collision step is responsible for the relaxation of the distribution function to the local equilibrium (Eq. (5)). To simplify the collision step without destroying the very core of the lattice Boltzmann method, according to Zhou [47], parameter ω is set to 1.0. As a result, the left side of Eq. (5) now depends only on f_{eq}, hence, the collision step is reduced to derivation of the f_{eq} alone. To perform this step, point-wise multiplication of the vector $\boldsymbol{\Phi}$ and corresponding block-diagonal matrix A with two $M \times M$ main-diagonal blocks is required according to Eq. (3). However, since this block-diagonal matrix is a non-unitary, implementation of the corresponding operator is performed by applying the linear combination of unitaries approach [44]. Hence, controlled operations in form

$$(H^\dagger \otimes I_q)(|0\rangle\langle 0|_a \otimes C_1 + |1\rangle\langle 1|_a \otimes C_2)(H \otimes I_q), \tag{9}$$

is applied, where C_1 and C_2 are the unitary diagonal operators defined as $C_1 = A + i\sqrt{I - A^2}$ and $C_2 = A - i\sqrt{I - A^2}$, while $A = 1/2(C_1 + C_2)$ represents the non-unitary block-diagonal operator, whose entries, $a_{i,i}$, are calculated according to Eq. (3), excluding the depended variable $\phi(\mathbf{x}, t)$. Since there is just one qubit in ancilla register required to achieve control operation, superposition is accomplished

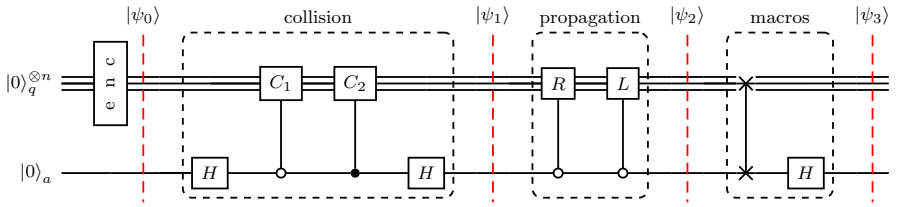

Fig. 3 Quantum circuit for solving the 1D advection–diffusion equation by using the D1Q2 lattice Boltzmann model. For the typesetting quantum circuit diagrams, a Quantikz package [49] is used

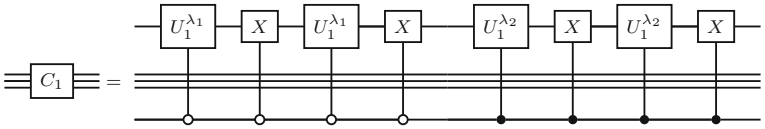

Fig. 4 Quantum circuit for implementing the operator C_1 in case of the D1Q2 LBM

by applying the Hadamard operator $\hat{H}|0\rangle_a$. It should be noted also that by increasing the number of computational points, i.e., the number of qubits in q register, the two block-diagonal matrix structure of the collision operator does not change, leaving unaffected the ancillary register in terms of the used qubits. Following the collision step, the initial quantum state ψ_0 has evolved into:

$$|\psi_1\rangle = |0\rangle_a \sum_{i=0}^{2M-1} a_{i,i}\phi_{\alpha,i}/\|\phi\||i\rangle_q + |1_{\phi}^{\perp}\rangle_{aq}, \tag{10}$$

where $|1_{\phi}^{\perp}\rangle_{aq}$ denotes some orthogonal state of lesser interest. The quantum circuit responsible for achieving the state described by Eq. (10) is given in Fig. 3, while the details of the operator C_1 are shown in Fig. 4. The $U_1 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$ gate in Fig. 4 represents the standard Qiskit phase shift gate, which correspond to R_z rotation up to global phase $e^{-i\lambda/2}$, while the parameters λ_1 and λ_2 corresponds to the first and second block of the diagonal operator C_1, respectively. From the scalability point of view, increasing the number of qubits do not affect operators C_1 and C_2 in terms of required gates, i.e., the qubits are simply added with no additional modifications of the current gate structure and depth whatsoever.

In the propagation step, quantum walk [50,51] as a procedure for the propagation of the distribution functions along the corresponding links is utilized. In general, this procedure implies shifting the distribution function f_1 along the link $\alpha = 1$ with speed $e_1 = 1$, and function f_2 along the link $\alpha = 2$ with speed $e_2 = -1$. To achieve this, corresponding operations in form of matrices for the right and the left shift, P_r and P_l, respectively, are introduced. These operations are implemented into the circuit in form of operators R and L (Fig. 3), where the corresponding sub-circuits are shown in Figs. 5 and 6. It should be noted that the propagation step is implemented as a controlled operation of the quantum walk procedure on the first qubit in register q as $|0\rangle\langle 0| \otimes R_w + |1\rangle\langle 1| \otimes L_w$, where R_w and L_w represents the quantum walk in right

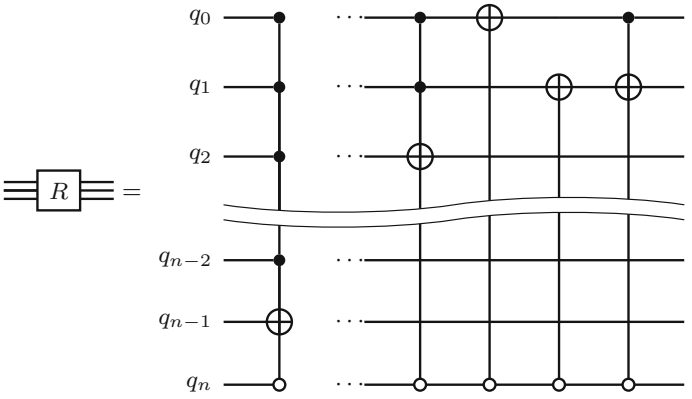

Fig. 5 Quantum circuit for implementing the operator R

and left direction, respectively. Following the propagation step, the quantum state of the entire system is

$$|\psi_2\rangle = |0\rangle_a L R \left(\sum_{i=0}^{2M-1} a_{i,i} \phi_{\alpha,i} / \|\phi\| |i\rangle_q \right) + |1_{\phi}^{\perp}\rangle_{aq}, \tag{11}$$

where again $|1_{\phi}^{\perp}\rangle_{aq}$ represents some state of lesser interest.

In the last step, the calculation of the macroscopic variables by Eq. (7), i.e., the point-wise addition of the two quantum states, is performed. Since state ψ_2 (Eq. (11)) indicates that both distribution functions are located in the subsystem controlled by the $|0\rangle_a$, for the procedure of addition it is required first to perform the SWAP gate between ancilla register a and the last qubit q_n in the q register (Fig. 3). This operation actually switches between the states of the register q controlled by the $|0\rangle_a$ and $|1\rangle_a$ in the ancilla register, positioning, therefore, the sub-state of the second distribution function, controlled by $|0\rangle_a$, to the state controlled by $|1\rangle_a$. The last step includes application of the Hadamard gate in the ancilla register, followed by the re-normalization of the post-selected $|0\rangle_a$ state by the factor $2\|\phi\|/\sqrt{2}$. As a result, the spatial distribution of the variable ϕ for the next time level $t+1$ is obtained, and the entire procedure is then repeated to achieve desired time level. In order to retrieve this post-selected state on real devices, the state tomography is required.

$$P_r = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & 1 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, P_l = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & 0 & 0 & 1 \\ 1 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

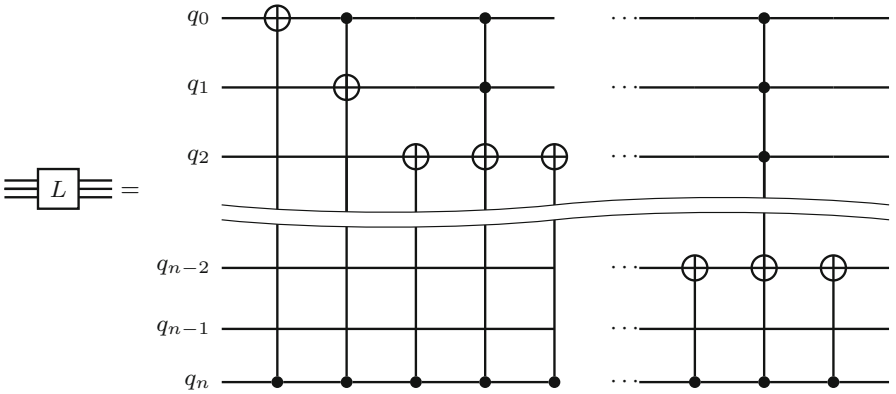

Fig. 6 Quantum circuit for implementing the operator L

3.2 The D2Q5 model

In case of the two-dimensional D2Q5 lattice Boltzmann model, 5 distribution functions have been utilized, where, in comparison with the 1D case, two distribution functions along y axis and a rest particle have been added (Fig. 2). Hence, variable $\phi(\mathbf{x}, \mathbf{0})$ in this case is represented in vector form as $\mathbf{\Phi} = [(\phi_{0,i,j}, \ldots, \phi_{0,M-1,M-1}), \ldots, (\phi_{4,i,j}, \ldots, \phi_{4,M-1,M-1})]^T$, where $i, j = 0, \ldots, M-1$. The corresponding quantum state following the encoding step is

$$|\psi_0\rangle = |0\rangle_a \sum_{k=0}^{8M^2-1} \phi_{\alpha,k}/\|\phi\||k\rangle_q. \tag{12}$$

The number of required qubits in the q register is determined as $\log_2(5(M \times M)) + 1$, where an additional qubit is introduced for the purpose of the collision step only. Again, the encoding step is performed by the previously mentioned procedure [48].

In the collision step, the same procedure is applied as in the case of the D1Q2 model (Fig. 3). In contrast to the one-dimensional case, operators C_1 and C_2 now have eight-block diagonal form, therefore modifying the corresponding quantum subcircuit which is shown in Fig. 7. Following the collision step, the initial quantum state ψ_0 evolves into

$$|\psi_1\rangle = |0\rangle_a \sum_{k=0}^{8M^2-1} a_{k,k}\phi_{\alpha,k}/\|\phi\||k\rangle_q + |1_{\phi}^{\perp}\rangle_{aq}. \tag{13}$$

It should be noted that the $a_{k,k}$ are the entries of the eight-block diagonal operator, where each block corresponds to one of the quantum sub-states representing the distribution functions. Since we have only 5 distribution functions (Fig. 2), the remaining 3

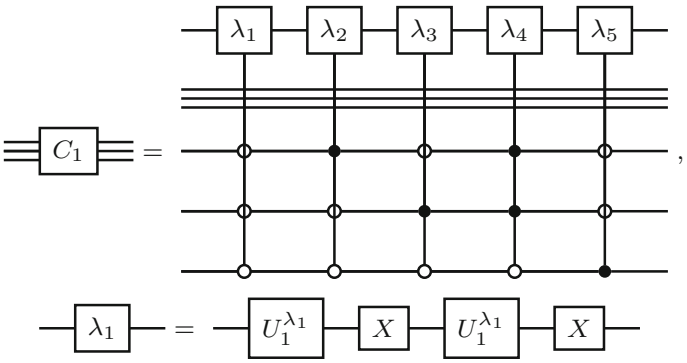

Fig. 7 Quantum circuit for implementing the operator C_1 in case of the D2Q5 LBM

are not taken into account, having their values a set to 1.0. In the circuit terminology, these states are introduced with the identity operator.

For the propagation step, there are now four distribution functions that need to be shifted in four different directions with the velocity vectors defined by Eq. (4), and one distribution function assigned to the rest particle which is excluded from this process. To implement this step into the quantum circuit, two operators R (Fig. 5) and L (Fig. 6) from the 1D case are utilized, where some additional modification in terms of controlled operations are introduced to account the 2D case. The resulting quantum circuit is shown in Fig. 8. It can be seen in Fig. 8 that the whole propagation process refers to the $|0\rangle_a$ state, meaning that the quantum states encoding the valid distribution functions are concentrated in the sub-state when the ancilla is in $|0\rangle$, while the state $|1^{\perp}_{\phi}\rangle$ is of lesser interest and do not contains any relevant information. However, this $|1^{\perp}_{\phi}\rangle$ will be used for the procedure of addition as a state that will receive the switched sub-state from the orthogonal counterpart, forming, therefore, the required state configuration for the addition to be performed.

To calculate the macroscopic variables based on Eq. 7, point-wise addition between quantum sub-states corresponding to the particular distribution function needs to be performed. This can be done by switching the quantum sub-states which need to be added between the states $|0\rangle_a$ and $|1\rangle_a$ by applying the SWAP gate, and then reordering them if it is necessary to obtain the right state configuration for the addition protocol. Finally, the post-selection of the state $|0\rangle_a|00\psi\rangle_q$ followed by normalization with the factor $2\|\phi\|/\sqrt{2}$ gives as a result the desired solution of the variable $\phi(x, y)$ for the next time level $t + 1$. The corresponding quantum circuit for determining the macroscopic variables of the D2Q5 lattice Boltzmann model is given in Fig. 8.

3.3 Complexity

The complexity of the proposed algorithm can be analyzed through four major steps, encoding, collision, propagation and macros calculation. The main question is: how does the resolution of the computational domain relate to the depth of quantum algorithm? For the encoding step, according to Shende et al. [48], $2^n - 1$ state

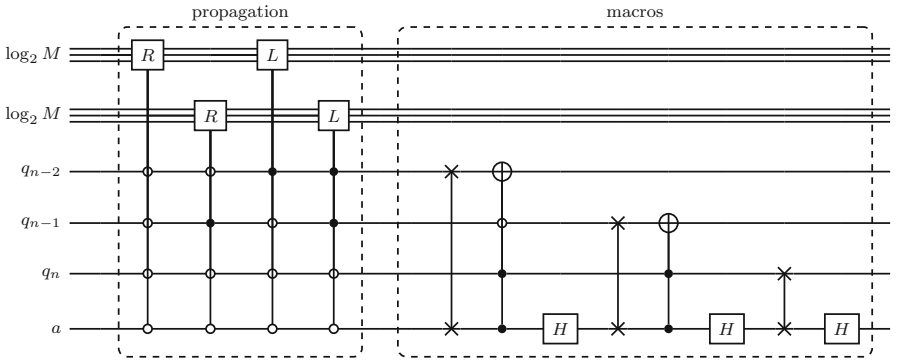

Fig. 8 Quantum circuit for the propagation step and calculation of the macroscopic variables in case of D2Q5 LBM

preparation steps and 1 diagonal operator are used, where the final CNOT count is $2 \times 4^n - (2n + 3) \times 2^n + 2n$. For the collision step, it can be seen that both for the 1D and 2D cases, change in number of qubits to achieve a higher resolution of the computational model doesn't affect the size of ancillary register, while in the q register the number of multi-qubit gate operations scales with the number of qubits n as $\mathcal{O}(1)$. The reason for this lies in dependency between the multi-qubit gate operations and the total number of distribution function α being used, which, consequently imposes relation $\mathcal{O}(\alpha)$, while the number of controls per gate scales as $\mathcal{O}(\lceil \alpha/2 \rceil)$.

For the propagation step, the number of multi-qubit gate operations scales with the number of qubits for each link α as $\mathcal{O}(\log_2 D)$, where for the D1Q2 model $D = 2M$, while for the D2Q5 case $D = 5M^2$. The propagation for each link is implemented by applying one of the operators R or L, where the entry of control operations per gate, excluding the ancilla register and the qubit q_n introduced for the collision purposes only, scales as $\mathcal{O}(\log_2 \alpha)$.

The last segment is used for the calculation of the macroscopic variables by performing the point-wise addition of the distribution function encoded as the sub-states of the quantum state ψ. This part of the algorithm is composed of blocks each having one Hadamard, SWAP and multi-control gate, which primarily depends on the number of links α as $\mathcal{O}(\alpha - 1)$. However, increasing the number of qubits for particular model does not affect the circuit in terms of quantum gates, i.e., the macros block scales with the number of qubits n as $\mathcal{O}(1)$. Finally, the complexity of the whole algorithm, excluding the encoding step, can be defined as $\mathcal{O}(\log_2(\alpha D))$.

4 Validation and numerical simulation

To demonstrate the proposed quantum algorithm for solving the unsteady 1D and 2D advection–diffusion equation by using the LBM as a numerical procedure, transport of conservative tracer, having the concentration C in a schematic channel is considered. For both the 1D and 2D cases, a periodic boundary condition on closed boundaries is imposed, while for the initial concentration, $C(x, 0)$, some arbitrary Gauss like dis-

tribution is specified. The unsteady simulation is then performed by using Qiskit [45] platform, where the *'statevector simulator'* backend as a part of the high-performance simulator framework Aer is used.

4.1 The motion of 1D Gaussian Hill—D1Q2 model

The movements of 1D Gaussian hills in a uniform flow are simulated. In the computations, 64 lattice cells are used with $\Delta x = 1.0$, $\Delta t = 1.0$, $u = 0.2$, $\omega = 1$ and $D = 0.5$, all in lattice units. Equilibrium function is calculated according to Eq. (3) as $f_1^{eq}(\mathbf{x}, t) = 0.6\phi(\mathbf{x}, t)$ and $f_2^{eq}(\mathbf{x}, t) = 0.4\phi(\mathbf{x}, t)$, which is further used for the calculation of the operators C_1 and C_2. For this purpose, 7 qubits in the q register, and one qubit in ancillary register are utilized. The initial concentration is 0.1 everywhere, except for the point at $x = 12$ with a unit point source of $C = 0.2$. The results shown in Fig. 9 demonstrate good agreements with the analytical solutions.

$$C_1 = \begin{bmatrix} e^{i\overbrace{\log(0.6+0.8i)}^{\lambda_1}} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & e^{i\log(0.6+0.8i)} & 0 & 0 & \vdots \\ \vdots & \ddots & 0 & e^{i\overbrace{\log(0.4+0.916515i)}^{\lambda_2}} & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 0 & e^{i\log(0.4+0.916515i)} \end{bmatrix},$$

$$C_2 = \begin{bmatrix} e^{i\overbrace{\log(0.6-0.8i)}^{\lambda_1}} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & e^{i\log(0.6-0.8i)} & 0 & 0 & \vdots \\ \vdots & \ddots & 0 & e^{i\overbrace{\log(0.4-0.916515i)}^{\lambda_2}} & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 0 & e^{i\log(0.4-0.916515i)} \end{bmatrix}.$$

4.2 Movements of 2D Gaussian Hill - D2Q5 model

For the 2D case, the evolution of the Gaussian hill in a square domain with a 16×16 lattice configuration is considered. The following parameters set has been adopted: $\Delta x = 1.0$, $\Delta t = 1.0$, $u = 0.2$, $v = 0.15$, $\omega = 1$ and $D = 0.166\dot{7}$, all in lattice units. The numerical simulation is conducted using an instantaneous drop of pollution at point $C(x = 4, y = 4) = 0.3$, while the concentration of 0.1 is imposed along the rest of the domain. A periodic boundary condition is assigned to all four boundaries. For this purpose, 11 qubits in the q register and one qubit in ancillary register are utilized.

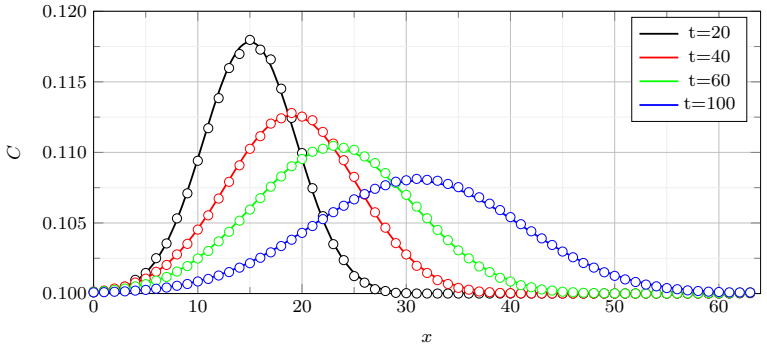

Fig. 9 Comparison of the numerical results (○) to the analytical solution (−) for the 1D ADE simulated with the D1Q2 LBM

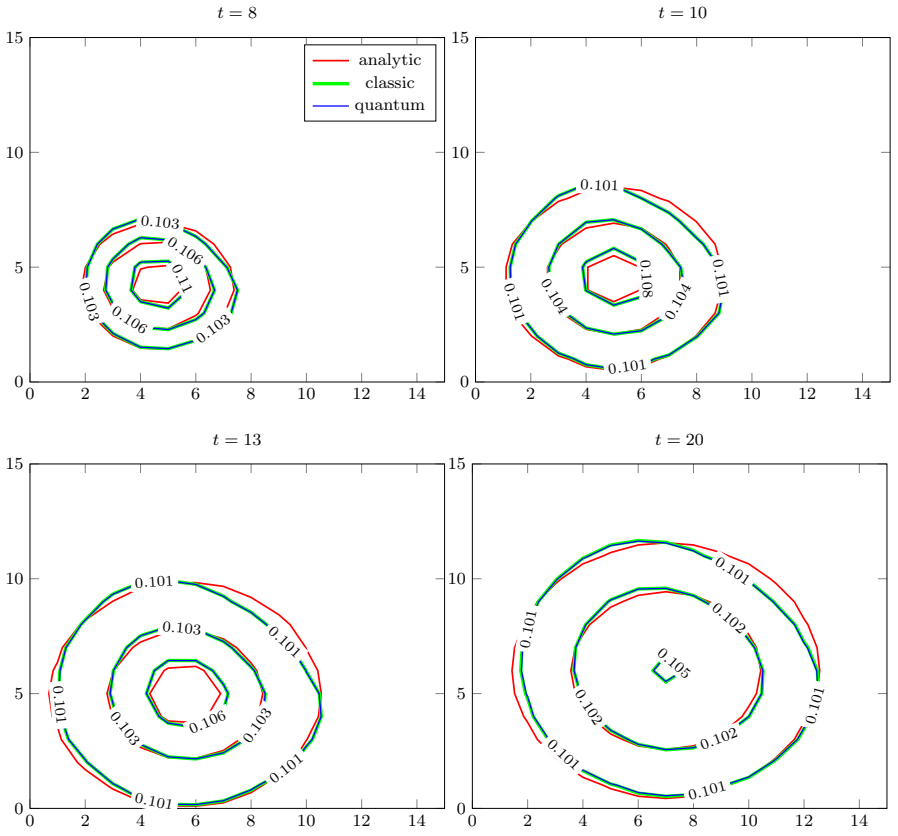

Fig. 10 Comparison of the numerical results obtained by the quantum algorithm, the classical FORTRAN code, and the analytical solution of the 2D ADE simulated with the D2Q5 LBM

To validate the proposed algorithm, comparison with analytical solution and results obtained by the 'classical' FORTRAN [52] code for the D2Q5 model [37] is shown in Fig. 10. Overlapping of the results of the FORTRAN code solution and the proposed quantum algorithm solution is demonstrated, while some discrepancies can be seen compared to the analytic solution. However, these deviations are not produced by the quantum algorithm solely, which is demonstrated previously, but by the low resolution of the computational mesh as one of the major issues when numerical methods for solving differential equations are used. Of course, to increase the accuracy of the model finer mesh with higher resolution and, hence, a greater number of qubits is required.

5 Conclusion

In this paper, a novel quantum algorithm for solving the advection–diffusion equation by using the full lattice Boltzmann method as a numerical scheme is proposed and validated. For this purpose, the unsteady form of the one-dimensional and two-dimensional ADE discretized with the D1Q2 and D2Q5 lattice Boltzmann model, respectively, is utilized. Due to the basic nature of the LBM, the proposed quantum algorithm consists of three major segments, collision, propagation, and calculation of the macroscopic variables, for which a corresponding circuit configuration using the gate based quantum computing platform Qiskit has been developed and tested. The evolution of the 1D and 2D Gaussian Hill is used for validation of the algorithm, while the analytic solution and the 'classic' FORTRAN code are exploited for the comparison. The obtained results show excellent agreement with the compared values. Furthermore, the complexity analysis shows that for the particular model collision the macros segment of the algorithm is not affected by the change in the size of the computational domain. Developing quantum algorithms for other, more complex physical processes related to fluid dynamics using the lattice Boltzmann method is the major goal of the forthcoming work, where nonlinearity in the equilibrium function is one of the main obstacles that need to be overcome.

Compliance with ethical standards

Conflicts of interest Not applicable.

Funding Not applicable.

Availability of data and material Not applicable.

Code availability Not applicable.

References

1. Grover, L.: A fast quantum mechanical algorithm for database search. In: Proceedings, 28th Annual ACM Symposium on the Theory of Computing, p. 212 (1996)
2. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information (Repr. ed.). Cambridge Univ. Press (2001)

3. Coppersmith, D.: An Approximate Fourier Transform Useful in Quantum Factoring. Technical Report. IBM, New York (1994)
4. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994)
5. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Phys. Rev. Lett. **103**(15) (2009). ISSN 1079-7114. https://doi.org/10.1103/physrevlett.103.150502
6. Ambainis, A.: Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations (2010)
7. Qian, P., Huang, W.-C., Long, G.-L.: A quantum algorithm for solving systems of nonlinear algebraic equations (2019)
8. Wang, H., Xiang, H.: Quantum algorithm for total least squares data fitting. Phys. Lett. A **383**(19), 2235–2240 (2019)
9. Doronin, S.I., Feldman, E.B., Zenchuk, A.I.: Solving systems of linear algebraic equations via unitary transformations on quantum processor of IBM quantum experience. Quant. Inform. Process. **19**(68) (2020)
10. Cao, Y., Romero, J., Olson, J.P., Degroote, M., Johnson, P.D., Kieferov, M., Kivlichan, I.D., Menke, T., Peropadre, B., Sawaya, N.P.D., Sim, S., Veis, L., Aspuru-Guzik, A.: Quantum chemistry in the age of quantum computing. Chem. Rev. **119**(19), 10856–10915 (2019)
11. Kerenidis, I.S., Prakash, A.: Quantum gradient descent for linear systems and least squares. Phys. Rev. A **101**(2), 022316 (2020)
12. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm (2014)
13. Schuld, M., Petruccione, F.: Supervised Learning with Quantum Computers. Springer Nature, Switzerland (2018)
14. Garg, S., Ramakrishnan, G.: Advances in quantum deep learning: an overview (2020)
15. Sharma, S.: Qeml (quantum enhanced machine learning): using quantum computing to enhance ml classifiers and feature spaces (2020)
16. Farhi, E., Neven, H.: Classification with quantum neural networks on near term processors (2018)
17. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. Phys. Rev. A **101**(3), 2020. ISSN 2469-9934. https://doi.org/10.1103/physreva.101.032308
18. Killoran, N., Bromley, T.R., Arrazola, J.M., Schuld, M., Quesada, N., Lloyd, S.: Continuous-variable quantum neural networks. Phys. Rev. Res. **1**(3), 2019. ISSN 2643-1564. https://doi.org/10.1103/physrevresearch.1.033063
19. Mari, A., Bromley, T.R., Izaac, J., Schuld, M., Killoran, N.: Transfer learning in hybrid classical-quantum neural networks (2019)
20. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., OBrien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature Commun. **5** (2014). https://doi.org/10.1038/ncomms5213
21. Prieto, C.B., LaRose, R., Cerezo, M., Subasi, Y., Cincio, L., Coles, P.J.: Variational quantum linear solver (2019)
22. Huang, H.Y., Bharti, K., Rebentrost, P.: Near-term quantum algorithms for linear systems of equations (2019)
23. Stokes, J., Izaac, J., Killoran, N., Carleo, G.: Quantum natural gradient. Quantum **4**, 269, 2020. ISSN 2521-327X. https://doi.org/10.22331/q-2020-05-25-269
24. Sweke, R., Wilde, F., Meyer, J., Schuld, M., Fhrmann, P. K., Piganeau, B. M., Eisert, J.: Stochastic gradient descent for hybrid quantum-classical optimization, (2019)
25. Yamamoto, N.: On the natural gradient for variational quantum eigensolver (2019)
26. Berry, D.W.: High-order quantum algorithm for solving linear differential equations. J. Phys. A: Math. Theor. **47**(10), 105301 (2014). https://doi.org/10.1088/1751-8113/47/10/105301. 10.1088%2F1751-8113%2F47%2F10%2F105301
27. Childs, A.M., Liu, J.P., Ostrander, A.: High-precision quantum algorithms for partial differential equations (2020)
28. Costa, P.C.S., Jordan, S., Ostrander, A.: Quantum algorithm for simulating the wave equation. Phys. Rev. A **99**, 012323 (2019). https://doi.org/10.1103/PhysRevA.99.012323
29. Cao, Y., Papageorgiou, A., Petras, I., Traub, J., Kais, S.: Quantum algorithm and circuit design solving the Poisson equation. New J. Phys. **15**(1), 013021 (2013). https://doi.org/10.1088/1367-2630/15/1/013021

30. Wang, S., Wang, Z., Li, W., Fan, L., Wei, Z., Yongjian, G.: Quantum fast Poisson solver: the algorithm and complete and modular circuit design. Quant. Inform. Process. (2020). https://doi.org/10.1007/s11128-020-02669-7
31. Berry, D.W., Childs, A.M., Ostrander, A., Wang, G.: Quantum algorithm for linear differential equations with exponentially improved dependence on precision. Commun. Math. Phys. **356**, 1057–1081 (2017)
32. Xin, T., Wei, S., Cui, J., Xiao, J., Arrazola, I., Lamata, L., Kong, X., Dawei, L., Solano, E., Long, G.: Quantum algorithm for solving linear differential equations: theory and experiment. Phys. Rev. A **101**, 032307 (2020). https://doi.org/10.1103/PhysRevA.101.032307
33. Leyton, S.K., Osborne, T.J.: A quantum algorithm to solve nonlinear differential equations (2008)
34. Rivet, J.P., Boon, J.P.: Lattice Gas Hydrodynamics. Cambridge University Press, London (2001)
35. Rothman, D.H., Zaleski, S.: Lattice-Gas Cellular Automata Simple Models of Complex Hydrodynamics. Cambridge University Press, London (1996)
36. Chen, S., Doolen, G.D.: Lattice Boltzmann method for fluid flows. Ann. Rev. Fluid Mech. **30**, 329–364 (1998)
37. Mohamad, A.A.: Lattice Boltzmann Method–Fundamentals and Engineering Applications with Computer Codes. Springer, London (2011)
38. Yepez, J.: Quantum lattice-gas model for computational fluid dynamics. Phys. Rev. E **63**, 046702 (2001)
39. Berman, G.P., Ezhov, A.A., Kamenev, D.I., Yepez, J.: Simulation of the diffusion equation on a type-ii quantum computer. Phys. Rev. E **66**, 012310 (2002). https://doi.org/10.1103/PhysRevA.66.012310
40. Micci, M.M., Yepez, J.: Measurement-based quantum lattice gas model of fluid dynamics in 2+1 dimensions. Phys. Rev. E **92**, 033302 (2015). https://doi.org/10.1103/PhysRevE.92.033302
41. Yepez, J.: Type-ii quantum computers. Int. J. Mod. Phys. C **12**(09), 1273–1284 (2001). https://doi.org/10.1142/S0129183101002668
42. Todorova, B.N., Steijl, R.: Quantum algorithm for the collisionless Boltzmann equation. J. Comput. Phys. **409**, (2020)
43. Gaitan, F.: Finding flows of a Navier–Stokes fluid through quantum computing. NPJ Quant. Inform. (2020). https://doi.org/10.1038/s41534-020-00291-0
44. Low, G.H., Chuang, I.L.: Hamiltonian simulation by qubitization. Quantum **3**, 163 (2019). ISSN 2521-327X. https://doi.org/10.22331/q-2019-07-12-163
45. Abraham, H., Offei, A., Akhalwaya, I.Y., Aleksandrowicz, G., Alexander, T., Arbel, E., Asfaw, A., Azaustre, C., Ngoueya, A., Barkoutsos, P., Barron, G., Bello, L., Ben-Haim, Y., Bevenius, D., Bishop, L.S., Bolos, S., Bosch, S., Bravyi, S., Bucher, D., Burov, A., Cabrera, F., Calpin, P., Capelluto, L., Carballo, J., Carrascal, G., Chen, A., Chen, C.-F., Chen, R., Chow, J.M., Claus, C., Cocking, R., Cross, A.J., Cross, A.W., Cross, S., Cruz-Benito, J., Culver, C., Córcoles-Gonzales, A.D., Dague, S., El Dandachi, T., Dartiailh, M., Frr, D., Davila, A.R., Dekusar, A., Ding, D., Doi, J., Drechsler, E., Drew, Dumitrescu, E., Dumon, K., Duran, I., EL-Safty, K., Eastman, E., Eendebak, P., Egger, D., Everitt, M., Fernández, P.M., Ferrera, A.H., Chevallier, F., Frisch, A., Fuhrer, A., George, M., Gacon, J., Gago, B.G., Gambella, C., Gambetta, J.M., Gammanpila, A., Garcia, L., Garion, S., Gilliam, A., Gomez-Mosquera, J., de la Puente González, S., Gorzinski, J., Gould, I., Greenberg, D., Grinko, D., Guan, W., Gunnels, J.A., Haglund, M., Haide, I., Hamamura, I., Hamido, O.C., Havlicek, V., Hellmers, J., Herok, L., Hillmich, S., Horii, H., Howington, C., Hu, S., Hu, W., Imai, H., Imamichi, T., Ishizaki, K., Iten, R., Itoko, T., Seaward, J., Javadi, A., Jessica, Jivrajani, M., Johns, K., Jonathan-Shoemaker, Kachmann, T., Kanazawa, N., Kang-Bae, Karazeev, A., Kassebaum, P., King, S., Knabberjoe, Kobayashi, Y., Kovyrshin, A., Krishnakumar, R., Krishnan, V., Krsulich, K., Kus, G., LaRose, R., Lacal, E., Lambert, R., Latone, J., Lawrence, S., Li, G., Liu, D., Liu, P., Maeng, Y., Malyshev, A., Manela, J., Marecek, J., Marques, M., Maslov, D., Mathews, D., Matsuo, A., McClure, D.T., McGarry, C., McKay, D., McPherson, D., Meesala, S., Metcalfe, T., Mevissen, M., Mezzacapo, A., Midha, R., Minev, Z., Mitchell, A., Moll, N., Mooring, M.D., Morales, R., Moran, N., MrF, Murali, P., Müggenburg, J., Nadlinger, D., Nakanishi, K., Nannicini, G., Nation, P., Navarro, E., Naveh, Y., Neagle, S.W., Neuweiler, P., Niroula, P., Norlen, H., O'Riordan, L.J., Ogunbayo, O., Ollitrault, P., Oud, S., Padilha, D., Paik, H., Perriello, S., Phan, A., Piro, F., Pistoia, M., Piveteau, C., Pozas-iKerstjens, A., Prutyanov, V., Puzzuoli, D., Pérez, J., Quintiii, Ramagiri, N., Rao, A., Raymond, R., Martín-Cuevas Redondo, R., Reuter, M., Rice, J., Rodríguez, D.M., Karur, R., Rossmannek, M., Ryu, M., Tharrmashastha, S.A.P.V., Ferracin, S., Sandberg, M., Sargsyan, H., Sathaye, N., Schmitt, B., Schnabel, C., Schoenfeld, Z., Scholten, T.L., Schoute, E., Schwarm, J., Sertage, I.F., Setia, K., Shammah, N., Shi, Y., Silva, A., Simonetto, A., Singstock, N., Siraichi, Y., Sitdikov, I., Sivarajah, S., Sletfjerding, M.B., Smolin, J.A., Soeken, M., Sokolov, I.O.,

Thomas, S., Starfish, Steenken, D., Stypulkoski, M., Sun, S., Sung, K.J., Takahashi, H., Tavernelli, I., Taylor, C., Taylour, P., Thomas, S., Tillet, M., Tod, M., Tomasik, M., de la Torre, E., Trabing, K., Treinish, M., Pe, T., Turner, W., Vaknin, Y., Valcarce, C.R., Varchon, F., Vazquez, A.C., Villar, V., Vogt-Lee, D., Vuillot, C., Weaver, J., Wieczorek, R., Wildstrom, J.A., Winston, E., Woehr, J.J., Woerner, S., Woo, R., Wood, C.J., Wood, R., Wood, S., Wood, S., Wootton, J., Yeralin, D., Yonge-Mallo, D., Young, R., Yu, J., Zachow, C., Zdanski, L., Zhang, H., Zoufal, C., Zoufalc, a matsuo, adekusar drl, bcamorrison, brandhsn, chlorophyll zz, dekel.meirom, dekool, dime10, drholmie, dtrenev, elfrocampeador, faisaldebouni, fanizzamarco, gadial, gruu, jagunther, jliu45, kanejess, klinvill, kurarrr, lerongil, ma5x, merav aharoni, michelle4654, ordmoj, rmoyard, saswati qiskit, sethmerkel, strickroman, sumitpuri, tigerjack, toural, vvilpas, welien, willhbang, yang.luh, yotamvakninibm, and Mantas Čepulkovskis. Qiskit: An open-source framework for quantum computing (2019)

46. Bhatnagar, P.L., Gross, E.P., Krook, M.: A model for collision processes in gases. i: Small amplitude processes in charged and neutral one-component system. Phys. Rev. **94**, 511–525 (1954)
47. Zhou, J.G.: Macroscopic lattice Boltzmann method (maclab) (2019)
48. Shende, V.V., Bullock, S.S., Markov, I.L.: Synthesis of quantum-logic circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **25**(6), 1000–1010 (2006). ISSN 1937-4151.https://doi.org/10.1109/TCAD.2005.855930
49. Kay, A.: Tutorial on the quantikz package (2020)
50. Childs, A.M.: Universal computation by quantum walk. Phys. Rev. Lett. **102**(18) (2009). ISSN 1079-7114. https://doi.org/10.1103/PhysRevLett.102.180501
51. Shakeel, A.: Efficient and scalable quantum walk algorithms via the quantum Fourier transform (2019)
52. Hahn, X.: Fortran for visual studio code (2015). https://marketplace.visualstudio.com/items?itemName=Gimly81.fortran