# Quantum Calculation of Classical Kinetic Equations: A Novel Approach for Numerical Analysis of 6D Boltzmann-Maxwell Equations in Collisionless Plasmas Using Quantum Computing

**H. Higuchi[1], J. W. Pedersen[2], A. Yoshikawa[3]**

[1] *Graduate School of Science, Kyushu University, Motooka, Nishi-ku, Fukuoka 819-0395, Japan*
[2] *Graduate School of Arts and Sciences, University of Tokyo, Komaba, Meguro-ku, Tokyo 153-8902, Japan*
[3] *Faculty of Science, Kyushu University, Motooka, Nishi-ku, Fukuoka 819-0395, Japan*

**Key Points:**

- A future fault-tolerant large-scale quantum computer speeds up simulations of the 6D collisionless Boltzmann equation in nonlinear plasmas.
- Future first principles simulators will have a huge number of lattices, leading to more advanced understanding and prediction of physics.
- To solve nonlinear PDEs using quantum computation, we used the method of amplitude embedding and quantum walk.

Corresponding author: Hayato Higuchi, `higuchi.hayato.007@s.kyushu-u.ac.jp`

**Abstract**

A novel quantum algorithm for solving the Boltzmann-Maxwell equations of the 6D collisionless plasma is proposed. The equation describes the kinetic behavior of plasma particles in electromagnetic fields and is known for the classical first-principles equations in various domains, from space to laboratory plasmas. We have constructed a quantum algorithm for a future fault-tolerant large-scale quantum computer to accelerate its costly computation. This algorithm consists mainly of two routines: the Boltzmann solver and the Maxwell solver. Quantum algorithms undertake these dual procedures, while classical algorithms facilitate their interplay. Each solver has a similar structure consisting of three steps: Encoding, Propagation, and Integration. We conducted a preliminary implementation of the quantum algorithm and performed a parallel validation against a comparable classical approach. IBM Qiskit was used to implement all quantum circuits.

## 1 Introduction

The space plasma environment, extending from the Sun to the magnetosphere-ionosphere-atmosphere, includes regions of frozen conditions, zones of anomalous resistance caused by electromagnetic turbulence, interconnected regions characterized by weakly ionized gas systems in strong magnetic fields, coupled neutral-atmosphere chemical processes, and pure neutral-atmosphere collision systems. Owing to their complex interactions, an inclusive understanding and forecasting of the space environment remains an elusive goal, even with the advancements in high-performance instrumentation and in-situ observation of satellites. Therefore, it is imperative to develop space plasma simulations capable of providing comprehensive insights, ranging from local spatial domains to the global schematic.

Historically, the development of space plasma simulations has been constrained by computational time, memory capacity, and data storage limitations, resolving complex phenomena with restricted physics at local space scales. In light of these constraints, space plasma simulations can be divided into two principal scale hierarchies. One approach endeavors to reproduce Macroscopic phenomena using a coarse approximation, whereas the other aims to recreate Microscopic phenomena derived from first principles. Examples of the former include magnetohydrodynamics (MHD), while the latter include techniques such as particle-in-cell (PIC) or the Vlasov equation (hereafter referred to as the collisionless Boltzmann equation). The choice between global simulation and comprehensive simulation of physical processes depends on the required space and time scales. However, several thematic concerns have emerged that require simulation via coupling between scale hierarchies. For example, we describe the plasma instability of the current sheet and the initiation mechanism of magnetic reconnection. The importance of kinetic effects resulting from ion-electron dynamics during the onset of magnetic reconnection has been demonstrated (Daughton, 2003; Moritaka & Horiuchi, 2008). To emulate this, a multi-hierarchical simulation with inter-domain coupling of MHD and PIC has been developed, which allows to analyze the influence of macroscopic dynamics on the microscopic physics of magnetic reconnection (Usami et al., 2009, 2014).

In contrast, the collisionless Boltzmann equation requires advanced numerical computations of the 6D distribution function in both space (3D) and velocity (3D) of the particles, and has traditionally been limited to the analysis of low-dimensional, low-resolution or microscopic phenomena. Given the susceptibility of direct methods to numerical diffusion, the more accurate electromagnetic Vlasov method has been designed and implemented(Umeda, 2008; Umeda et al., 2009; Minoshima et al., 2011; Umeda et al., 2012). The considerable progress in its research has allowed the elucidation of numerous authentic physical phenomena through the use of full electromagnetic Vlasov simulation, notwithstanding certain limitations regarding dimensionality and lattice number, which depend on the availability of computational resources(Umeda, Miwa, et al.,

2010; Umeda, Togano, & Ogino, 2010; Umeda et al., 2011; Umeda, 2012; Umeda et al., 2013, 2014). Theoretically, the integration of a collision term into the Boltzmann-Maxwell equations provides a comprehensive representation of the collision effects present in the complex coupled magnetosphere-ionosphere-atmosphere system of the Earth.

However, the current state of simulation technology is such that the fluid equations incorporating these collision effects have not yet been successfully modeled. The effects resulting from ionospheric collisions affect a variety of facets, ranging from auroras to magnetospheric dynamics (e.g. Yoshikawa et al. (2013)), and further lead to the manifestation of complex phenomena (e.g. Ohtani and Yoshikawa (2016)). Consequently, the collisionality Boltzmann-Maxwell equations encompass a plethora of significant phenomena within their domain of interest that are relevant to space-earth electromagnetics. In an idealized scenario, the entirety of these phenomena could be computed using the collisional Boltzmann-Maxwell equations, eliminating the need for scaling factorial coupled simulations and the reliance on a variety of assumptions. However, performing high-order numerical computations for the first-principles collisional Boltzmann-Maxwell equation requires the establishment of extremely precise numerical methods, coupled with an enormous computational burden $O(L^6)$ (where $L$ is the number of lattices per spatial degree of freedom), which is currently unattainable even with the computational power of today's supercomputers.

In recent years, advances in quantum computing, both software and hardware, have demonstrated numerous advantages of quantum algorithms, such as those represented by (Shor, 1994). Following Google's achievement of quantum supremacy in 2019 (Arute et al., 2019), the pragmatic implementation of quantum computing in plasma simulation, weather forecasting, fluid simulation, and various fields is attracting interest. In numerical computation, the first paper on solving linear equations with quantum computer, the so-called the HHL algorithm (Harrow et al., 2009), was published. Subsequently, a quantum algorithm for linear ordinary differential equations (ODE)(Berry et al., 2017) and for partial differential equations(PDE)(Childs et al., 2021), and many for fluid simulations have been reported in recent years (Mezzacapo et al., 2015; Budinski, 2022; Steijl & Barakos, 2018; Steijl, 2019, 2023; Arrazola et al., 2019; Cao et al., 2013; Wang et al., 2020; Gaitan, 2020, 2021). The employed methodologies vary considerably. Some use quantum computational versions of the lattice gas model (Yepez, 1998, 2001) or the lattice Boltzmann method (Miller et al., 2001), some use quantum Fourier transforms to solve the Poisson equation, some use HHL algorithms and Hamiltonian simulations and Some combine it with the HHL algorithm and Hamiltonian simulations, others reduce from PDEs to ODEs to solve nonlinear ODEs, and so on. Among them, the quantum lattice Boltzmann method is constructed by considering the streaming operation as Quantum Walk (Aharonov et al., 1993)(Succi et al., 2015). Similarly, a quantum algorithm for the Dirac equation was proposed (Fillion-Gourdeau et al., 2017), using the similarity of a sequence of time-evolving operations to Quantum Walk. And Todorova et al. developed a quantum algorithm for the collisionless Boltzmann equation that performs discrete real and discrete velocity space propagation by Quantum Walk using a discrete-velocity method (Todorova & Steijl, 2020). We consider that this method has an advantage over other quantum differential equation solving methods in that it is easier to introduce first-principles collision terms.

- Collisionless Boltzmann-Maxwell equations with $\boldsymbol{u}$(:velocity) constant and the electromagnetic field $\boldsymbol{E}, \boldsymbol{B}$ under vacuum conditions acting one way:

$$\frac{\partial f}{\partial t} + \boldsymbol{u}_{const} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{u}_{const} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0,$$

$$\nabla^2 \boldsymbol{E} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = 0,$$

$$\nabla^2 \boldsymbol{B} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{B}}{\partial t^2} = 0.$$

We developed a quantum algorithm for the 6D Boltzmann-Maxwell equations for collisionless plasmas under the above conditions based on the efficient quantum walk circuit(Douglas & Wang, 2009). In this process, we calculated the time evolution problem of the 6D distribution function with the addition of velocity space, referring to the quantum algorithms for the the discrete velocity method in the the Boltzmann equation(Todorova & Steijl, 2020) and the Macro step in the Navier-Stokes equations(Budinski, 2022). Thus, the implementation of the collision term, which is the final goal of our project, is much easier and can be developed step by step. Furthermore, according to our quantum algorithm, it is simpler and computationally less expensive to solve all regions with the collisionless Boltzmann-Maxwell equations than with Macro-Micro's hierarchically coupled simulators. The quantum computer's most important advantage, the lattice information in the spatial direction, is parallelized into a single state function by encoding amplitude embedding. The results show that the order of the Quantum Volume as the scale of the quantum circuit is $O\left(N_t \left(\log_2(L)\right)^2\right)$, which is an improvement over the order of the computational volume $O\left(N_t L^6\right)$ of a similar classical algorithm.

In the future, we will develop a quantum algorithm for the collisional Boltzmann-Maxwell equations and apply it to the plasma region from the sun to the Earth's magnetosphere-ionosphere-atmosphere. Thus, this will provide a framework in order to understand and fully predict the space plasma environment. At that time, we expect the device to be used is a future fault-tolerant large-scale quantum computer. This paper develops the first quantum algorithm for this purpose and summarizes the methodology and verification results.

This paper is organized as follows: Section 1.1 and 1.2 describe the model of numerical computation, Section 2 describes our Quantum Algorithm of Boltzmann solver, and Section 3 compares and verifies the results of the quantum algorithm with similar classical algorithms. In Section 4, we discuss current issues and future solutions.

### 1.1 Governing equations

We employ the collisionless plasma Boltzmann and Maxwell equations within an electromagnetic field as governing equations. Specifically, these equations are given by

- The collisionless plasma Boltzmann equation with an electromagnetic field:

$$\frac{\partial f}{\partial t} + \boldsymbol{u}_{const} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{u}_{const} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0, \tag{1}$$

- Wave equation for the electric field $\boldsymbol{E}$ in vacuum:

$$\nabla^2 \boldsymbol{E} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = 0, \tag{2}$$

- Wave equation for the magnetic field $\boldsymbol{B}$ in vacuum:

$$\nabla^2 \boldsymbol{B} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{B}}{\partial t^2} = 0. \tag{3}$$

Where $f$ is the distribution function of the plasma particles, $\boldsymbol{u}$ is the fluid velocity of the plasma, which we assume to be constant, $q/m$ is the charge to mass ratio of the particles and $\boldsymbol{E}$ and $\boldsymbol{B}$ are the electromagnetic fields. The Maxwell equations can be rewritten in the form of wave equations for the electric and magnetic fields respectively, as above, to implement the quantum algorithms more efficiently.

### 1.2 Numerical simulation method

For the execution of nonlinear partial differential equations (1,2,3) on quantum computers, these equations require discretization by methods such as the finite difference technique or the finite element method. In the following discourse, the finite difference approach is adopted for the Boltzmann-Maxwell equation, resulting in difference equations that are implementable on quantum circuits.

Proceeding with the application of the Forward Time Centered Space(FTCS) scheme, we differentiate the Boltzmann equations for collisionless plasma and derive a discretized representation. The differencing equation for the governing equation (1) is given by

$$\begin{aligned} f(x,y,z,v_x,v_y,v_z;t+\Delta t) &= f - \frac{u_x\Delta t}{2\Delta x}(f_{x+\Delta x} - f_{x-\Delta x}) - \frac{u_y\Delta t}{2\Delta y}(f_{y+\Delta y} - f_{y-\Delta y}) \\ &\quad - \frac{u_z\Delta t}{2\Delta z}(f_{z+\Delta z} - f_{z-\Delta z}) \\ &\quad - \frac{q(\boldsymbol{E}+\boldsymbol{u}_{const}\times\boldsymbol{B})_x\Delta t}{2m\Delta v_x}(f_{v_x+\Delta v_x} - f_{v_x-\Delta v_x}) \\ &\quad - \frac{q(\boldsymbol{E}+\boldsymbol{u}_{const}\times\boldsymbol{B})_y\Delta t}{2m\Delta v_y}(f_{v_y+\Delta v_y} - f_{v_y-\Delta v_y}) \\ &\quad - \frac{q(\boldsymbol{E}+\boldsymbol{u}_{const}\times\boldsymbol{B})_z\Delta t}{2m\Delta v_z}(f_{v_z+\Delta v_z} - f_{v_z-\Delta v_z}), \end{aligned} \tag{4}$$

where the value of $f(x,y,z,v_x,v_y,v_z;t)$, namely the distribution function at the reference point $x,y,z,v_x,v_y,v_z$ and time $t$, is simply denoted as $f$, and the same at the point deviating by one unit distance in each direction is denoted with subscripts:
(e.g.) $f_{x+\Delta x} := f(x+\Delta x,y,z,v_x,v_y,v_z;t)$.

We simplify the difference Boltzmann equation with the following assumption:

$$\frac{u_x\Delta t}{2\Delta x} = \frac{u_y\Delta t}{2\Delta y} = \frac{u_z\Delta t}{2\Delta z} = 1. \tag{5}$$

Similarly, the difference equations for the electric and magnetic fields are given as

$$\begin{aligned} \boldsymbol{E}(x,y,z;t+\Delta t) &= \left(2 - 2\frac{\Delta t^2}{\mu_0\epsilon_0}\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}\right)\right)\boldsymbol{E} - \boldsymbol{E}_{t-\Delta t} \\ &\quad + \frac{\Delta t^2}{\mu_0\epsilon_0\Delta x^2}\left(\boldsymbol{E}_{x+\Delta x}+\boldsymbol{E}_{x-\Delta x}\right) + \frac{\Delta t^2}{\mu_0\epsilon_0\Delta y^2}\left(\boldsymbol{E}_{y+\Delta y}+\boldsymbol{E}_{y-\Delta y}\right) \\ &\quad + \frac{\Delta t^2}{\mu_0\epsilon_0\Delta z^2}\left(\boldsymbol{E}_{z+\Delta z}+\boldsymbol{E}_{z-\Delta z}\right), \end{aligned} \tag{6}$$

$$\begin{aligned} \boldsymbol{B}(x,y,z;t+\Delta t) &= \left(2 - 2\frac{\Delta t^2}{\mu_0\epsilon_0}\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}\right)\right)\boldsymbol{B} - \boldsymbol{B}_{t-\Delta t} \\ &\quad + \frac{\Delta t^2}{\mu_0\epsilon_0\Delta x^2}\left(\boldsymbol{B}_{x+\Delta x}+\boldsymbol{B}_{x-\Delta x}\right) + \frac{\Delta t^2}{\mu_0\epsilon_0\Delta y^2}\left(\boldsymbol{B}_{y+\Delta y}+\boldsymbol{B}_{y-\Delta y}\right) \\ &\quad + \frac{\Delta t^2}{\mu_0\epsilon_0\Delta z^2}\left(\boldsymbol{B}_{z+\Delta z}+\boldsymbol{B}_{z-\Delta z}\right), \end{aligned} \tag{7}$$

where quantities such as $\boldsymbol{E}$ and $\boldsymbol{B}$ are defined in the same manner as $f$ above.

Furthermore, for simplicity of notation, we set hereafter as the Lorentz force term as

$$\boldsymbol{F} := \boldsymbol{u}\times\boldsymbol{B}. \tag{8}$$

Also, the speed of light $c$ in equation (2,3) is rewritten here using the permittivity and the permeability ($\epsilon_0$ and $\mu_0$) in the vacuum. Similar to the Boltzmann equation example, we make the following assumption:

$$\frac{\Delta t^2}{\mu_0\epsilon_0\Delta x^2} = \frac{\Delta t^2}{\mu_0\epsilon_0\Delta y^2} = \frac{\Delta t^2}{\mu_0\epsilon_0\Delta z^2} = 1. \tag{9}$$

Under the postulates of this manuscript, no velocity is obtained from the first-order velocity moment of the distribution function. Given the use of uniform velocities in both the temporal and spatial domains, the discretized magnetic field equation transforms into the propagation equation of the Lorentz force term.

As a result, we obtain the discretized Botzmann-Maxwell equation to be implemented as follows:

$$\begin{aligned} f(x,y,z,v_x,v_y,v_z;t+\Delta t) &= f-(f_{x+\Delta x}-f_{x-\Delta x})-(f_{y+\Delta y}-f_{y-\Delta y}) \\ &\quad -(f_{z+\Delta z}-f_{z-\Delta z}) \\ &\quad -\frac{q(\boldsymbol{E}+\boldsymbol{F})_x\Delta t}{2m\Delta v_x}(f_{v_x+\Delta v_x}-f_{v_x-\Delta v_x}) \\ &\quad -\frac{q(\boldsymbol{E}+\boldsymbol{F})_y\Delta t}{2m\Delta v_y}(f_{v_y+\Delta v_y}-f_{v_y-\Delta v_y}) \\ &\quad -\frac{q(\boldsymbol{E}+\boldsymbol{F})_z\Delta t}{2m\Delta v_z}(f_{v_z+\Delta v_z}-f_{v_z-\Delta v_z}), \qquad (10) \\ \boldsymbol{E}(x,y,z;t+\Delta t) &= -4\boldsymbol{E}-\boldsymbol{E}_{t-\Delta t}+(\boldsymbol{E}_{x+\Delta x}+\boldsymbol{E}_{x-\Delta x}) \\ &\quad +(\boldsymbol{E}_{y+\Delta y}+\boldsymbol{E}_{y-\Delta y})+(\boldsymbol{E}_{z+\Delta z}+\boldsymbol{E}_{z-\Delta z})\,, \qquad (11) \\ \boldsymbol{F}(x,y,z;t+\Delta t) &= -4\boldsymbol{F}-\boldsymbol{F}_{t-\Delta t}+(\boldsymbol{F}_{x+\Delta x}+\boldsymbol{F}_{x-\Delta x}) \\ &\quad +(\boldsymbol{F}_{y+\Delta y}+\boldsymbol{F}_{y-\Delta y})+(\boldsymbol{F}_{z+\Delta z}+\boldsymbol{F}_{z-\Delta z})\,. \qquad (12) \end{aligned}$$

This allows us to evolve the values of $f$ and , ($\boldsymbol{E}$, $\boldsymbol{B}$) independently. We call the quantum routines that perform this evolution the Boltzmann solver and the Maxwell solver, respectively. For the evolution of $f$ (Boltzmann solver), we need the values of $\boldsymbol{E}$ and $\boldsymbol{F}$ at each time step as they appear in the right-hand side of the equation (10), so we use the values obtained by the Maxwell solver.

## 2 Quantum Algorithm

In this section, a quantum algorithm based on the discretized Boltzmann-Maxwell equations (4,6,7) is constructed and implemented on quantum circuits. This quantum algorithm can be divided into two independent routines: the Boltzmann solver and the Maxwell solver. They take an initial function of $f$ and $(\boldsymbol{E}, \boldsymbol{B})$ as input, respectively. Both routines fix time and output physical quantities that evolve in one time step according to difference equations (11,12). By iterating this one-step evolution many times, we can obtain the value of a physical quantity that has evolved for an arbitrary time step.

The electric and magnetic fields derived by Maxwell solver are incorporated into the Propagation circuit of the Boltzmann solver as shown in the FIG. 1, thereby coupling each routine. The quantum calculations in this paper are carried out exactly in a way that deals with state vectors using a classical simulator provided by IBM Qiskit. It is straightforward to construct an authentic quantum algorithm based on measurements.

### 2.1 Boltzmann

Our Boltzmann solver can be segmented into three principal steps: Encoding, Propagation and Integration.

#### *2.1.1 Encoding*

First of all, it is necessary to encode the classical information of the physical quantities into the amplitudes of quantum states. Fixing the number of lattice sites in all spatial and velocity directions to be $L$, $f$ will have $V := L^6$ degrees of freedom. In the encoding step, we associate each of these degrees of freedom with one computational ba-

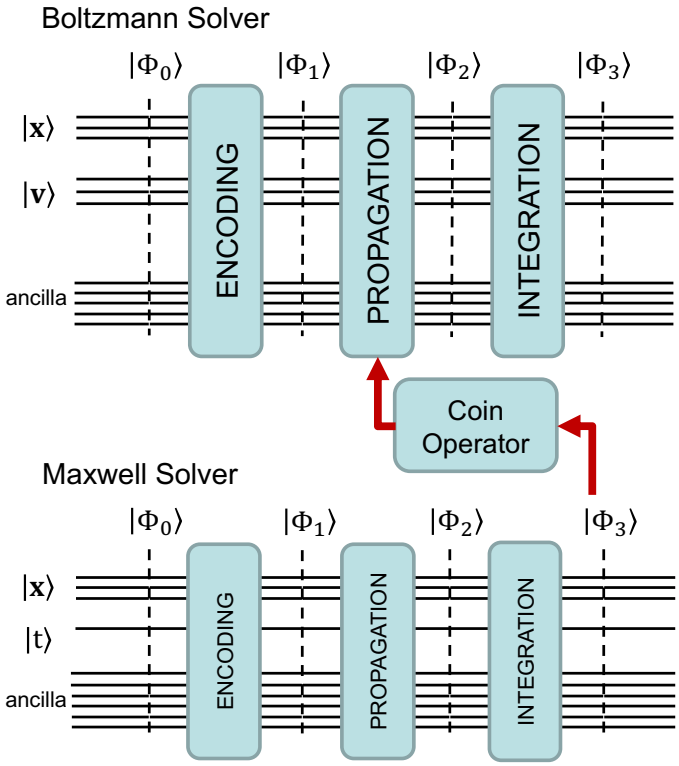



**Figure 1.** A Schematic of the quantum circuit of our algorithm for solving the Boltzmann-Maxwell equations. They consist of two routines that operate on the coin operator.

sis and encode the value of $f$ in the amplitude of the corresponding quantum state. Thus, a total of $V$ bases must be prepared in total, requiring $\lceil \log_2 V \rceil$ qubits. This method of encoding classical information into quantum information amplitudes is commonly referred to as the amplitude embedding technique.

To elucidate the relationship between physical quantities and probability amplitudes, the following conversion from a function $f(\boldsymbol{x}, \boldsymbol{v}; t)$ to a vector $f_i$ , $(0 \leq i \leq V-1)$ is implemented. The subscripts $i$ specify a point in the 6D lattice space. For example, $i = 0$ corresponds to the origin point $(\boldsymbol{x}, \boldsymbol{v}) = (0, 0, 0, 0, 0, 0)$, and i = 1 represents the value of the distribution function moved by one lattice point in the x direction: $(\boldsymbol{x}, \boldsymbol{v}) = (\Delta x, 0, 0, 0, 0, 0)$. Namely, the amount of $f_i$ follows

$$\text{(e.g.)} \quad f_0 = f(0, 0, 0, 0, 0, 0; t = t_r), \tag{13}$$

$$f_1 = f(\Delta x, 0, 0, 0, 0, 0; t = t_r). \tag{14}$$

Note that the quantum state does not contain any information about time, since the propagation takes place with fixed time. We will assume $L = 2^{N_L}$ in the following. As evidenced in Section 3, our actual numerical calculations are executed with $N_L = 3$ ($L = 8$).

The first important algorithm in the Encoding step is with a given distribution function at a fixed $t = t_r$ to prepare a quantum state, which we name $|\phi_0\rangle_{\text{phys}}$, with these values in its amplitudes:

$$|\phi_0\rangle_{\text{phys}} = \sum_{i=0}^{V-1} \tilde{f}_i |i\rangle, \tag{15}$$

where $\tilde{f}$ is the normalized distribution function as follows:

$$\tilde{f}_i = C f_i \quad , \quad C = \left( \sum_{i=0}^{V-1} |f_i|^2 \right)^{-1/2} \quad . \tag{16}$$

At the initial time step of $t = 0$, an arbitrary distribution can be designated as an initial function. Post the second step, the distribution function generated by the Boltzmann solver in the prior step ought to be provided as input. This iterative process allows for the computation of the distribution function at any desired time step. This procedure of state preparation can be executed in alignment with Appendix B.

It should be noted that, within the context of this manuscript, we have formulated the algorithm in a manner that measures $f$ post each step and re-encodes it in the subsequent step, in order to circumvent excessive enlargement of the quantum circuit's depth. This design necessitates $O(V)$ measurements at every time step, failing the advantage of the quantum algorithm. However, it is straightforward to connect each time step seamlessly. Namely, any measurements are required between each time step, implying that such a design will be beneficial when managing large-scale quantum apparatuses in the future. Further discussion on quantum advantage will be given in later sections.

The qubits prepared within this context are termed as the physical qubits, denoted as $|\text{phys}\rangle$. Looking more closely, $|\text{phys}\rangle$ is prepared by a total of 6 closed Hilbert spaces corresponding to spatial and velocity degrees of freedom, each having $N_L (= \log_2 L)$ qubits. Namely, we write it as

$$|\text{phys}\rangle = |\text{phys};x\rangle|\text{phys};y\rangle|\text{phys};z\rangle|\text{phys};v_x\rangle|\text{phys};v_y\rangle|\text{phys};v_z\rangle \tag{17}$$

Subsequent to the Propagation step, the ensuing quantum algorithms necessitate an additional qubit, which depending on their role, is identified as either subnode qubits $|\text{sub}\rangle$ or ancilla qubits $|\text{ancilla}\rangle$. As will explaind later the number of subnode and ancilla qubits are fixed to 4 and 1, respectively, regardless of the parameters and physical setup. Thus, the numbers of qubits required by the Boltzmann solver are

$$N_{\text{phis}} = 6N_L \ , \ N_{\text{sub}} = 4 \ , \ N_{\text{anc}} = 1, \tag{18}$$

and the following quantum state is prepared and output in after this Encoding step:

$$|\phi_1\rangle = |\phi_0\rangle_{\text{phys}} \otimes |0\rangle_{\text{sub}} \otimes |0\rangle_{\text{ancilla}}, \tag{19}$$

$$= \sum_{i=0}^{V-1} \tilde{f}_i |i\rangle_{\text{phys}} |0\rangle_{\text{sub}} |0\rangle_{\text{ancilla}}. \tag{20}$$

### *2.1.2 Propagation*

In the Propagation step, we partially utilize the tequniques of quantum algorithm method (Douglas & Wang, 2009) and implement an algorithm that multiplies each probability amplitude of $\phi_1$ by the coefficient of each term in the discretized equation.

The subnode bases and their corresponding physical quantities $f, \epsilon, and \sigma$ denote the normalized distribution encoded as the amplitude of the associated state, the coefficients to be incorporated via the coin operator, and the sign to be multiplied during the integration step, respectively. To solve the evolution equation (10), we need to prepare and add up all the terms that arise in the equation such as:

$$f, \ \mp f_{x \pm \Delta x}, \cdots, \mp \frac{q(\boldsymbol{E} + \boldsymbol{F})_x \Delta t}{2m \Delta v_x} f_{v_x \pm \Delta v_x}, \cdots .$$

After passing through the encoding step, we are now in possession of a quantum state $|\phi_1\rangle$, within which the data of the distribution function are encoded in the amplitude.

**Table 1.** The subnode bases and their corresponding physical quantities. $f, \epsilon$, and $\sigma$ respectively represent the (unnormalized) distribution function associated with each basis state, the coefficients to be incorporated via the coin operator, and the sign to be multiplied during the integration step. These are the quantities that appear on the right-hand side of the difference equation (10).

| $j$ | $\|j\rangle_{\text{sub}}$ | $f_j$ | $\epsilon_j$ | $\sigma_j$ |
|---|---|---|---|---|
| 0 | $\|0000\rangle$ | $f(x,y,z,v_x,v_y,v_z)$ | 1 | $+1$ |
| 1 | $\|0001\rangle$ | $f(x+\Delta x,y,z,v_x,v_y,v_z)$ | 1 | $-1$ |
| 2 | $\|0010\rangle$ | $f(x-\Delta x,y,z,v_x,v_y,v_z)$ | 1 | $+1$ |
| 3 | $\|0011\rangle$ | $f(x,y+\Delta y,z,v_x,v_y,v_z)$ | 1 | $-1$ |
| 4 | $\|0100\rangle$ | $f(x,y-\Delta y,z,v_x,v_y,v_z)$ | 1 | $+1$ |
| 5 | $\|0101\rangle$ | $f(x,y,z+\Delta z,v_x,v_y,v_z)$ | 1 | $-1$ |
| 6 | $\|0110\rangle$ | $f(x,y,z-\Delta z,v_x,v_y,v_z)$ | 1 | $+1$ |
| 7 | $\|0111\rangle$ | $f(x,y,z,v_x+\Delta v_x,v_y,v_z)$ | $q\frac{E_x(x,y,z)+F_x(x,y,z)\Delta t}{2m\Delta v_x}$ | $-1$ |
| 8 | $\|1000\rangle$ | $f(x,y,z,v_x-\Delta v_x,v_y,v_z)$ | $q\frac{E_x(x,y,z)+F_x(x,y,z)\Delta t}{2m\Delta v_x}$ | $+1$ |
| 9 | $\|1001\rangle$ | $f(x,y,z,v_x,v_y+\Delta v_y,v_z)$ | $q\frac{E_y(x,y,z)+F_y(x,y,z)\Delta t}{2m\Delta v_y}$ | $-1$ |
| 10 | $\|1010\rangle$ | $f(x,y,z,v_x,v_y-\Delta v_y,v_z)$ | $q\frac{E_y(x,y,z)+F_y(x,y,z)\Delta t}{2m\Delta v_y}$ | $+1$ |
| 11 | $\|1011\rangle$ | $f(x,y,z,v_x,v_y,v_z+\Delta v_z)$ | $q\frac{E_z(x,y,z)+F_z(x,y,z)\Delta t}{2m\Delta v_z}$ | $-1$ |
| 12 | $\|1100\rangle$ | $f(x,y,z,v_x,v_y,v_z-\Delta v_z)$ | $q\frac{E_z(x,y,z)+F_z(x,y,z)\Delta t}{2m\Delta v_z}$ | $+1$ |
| 13 | $\|1101\rangle$ | 0 | 0 | $-1$ |
| 14 | $\|1110\rangle$ | 0 | 0 | $+1$ |
| 15 | $\|1111\rangle$ | 0 | 0 | $-1$ |

Therefore, by considering an algorithm that multiplies each coefficient such as $\frac{q(\boldsymbol{E}+\boldsymbol{F})_x\Delta t}{2m\Delta v_x}$ by the corresponding state, the amplitudes of all states are updated to the state with the appropriate coefficient appearing in equation (10). We will deal with the explicit sign in the equation later. The values of $\boldsymbol{E}$ and $\boldsymbol{F}$ at the certain time step are obtained from Maxwell solver.

Subnodes serve to identify the terms that arise at a specific time step, namely $f, f_{x\pm\Delta x}, \cdots, f_{v_x\pm\Delta v_x}\cdots$. In total, there are 13 ($= 1+2\times6$) terms: one term $f$, which precedes propagation, and terms propagated by each $\pm1$ unit for each of the six directions in space and velocity. Hence, 4 ($= \lceil 13 \rceil$) qubits are necessitated as a subnode. It should be noted that this number remains uninfluenced by physical quantities like volume. For simplicity, we have associated them as depicted in TABLE 1. Here, $\epsilon_j$ is the coefficient applied to each term, and $\sigma_j$ is the sign explicitly attributed to each term in TABLE 1. In fact, both $\epsilon_j$ and $\sigma_j$ are coefficients in the difference equation (10), so it is possible to define epsilon to include the sign of $\sigma_j$. However, we choose to distinguish between them because $\epsilon_j$ represents a quantity that depends on a specific assumption as indicated by the assumption (5,9), while $\sigma_j$ is a universally determined quantity. By making this distinction, we think we can minimize the part that we need to be modified based on different assumptions.

As elucidated below, the coin operator is accountable for the multiplication of these coefficients, and the shift operator assumes responsibility for correlating each term with the basis of the subnode.

We can create the appropriate coefficients by first make the subnodes in superpotition using the H-gate. Then apply the diagonal matrix with $\{\epsilon\}$ as components:

$$\Lambda := \text{diag}(\epsilon_0, \epsilon_1, \cdots, \epsilon_{15}). \tag{21}$$

The operation with this diagonal matrix is not a unitary and thus it must be embedded in a unitary matrix of larger size. Since the coefficients are real, this procedure can be done easily as explained in the Appendix Appendix B. Here, we use the ancilla qubit $|a_0\rangle$ to create a unitary matrix of larger size. We call this whole operator acting on the subnode (and the ancilla qubit) the "coin operator" according to the terminology of quantum walk. As a result, we obtain the state after operating the coin operator as follows:

$$\begin{aligned} U_{\mathrm{Coin}}|\phi_1\rangle &= \sum_{i=0}^{V-1} U_{\mathrm{Coin}}\tilde{f}_i|i\rangle_{\mathrm{phys}}|0\rangle_{\mathrm{sub}}|0\rangle_{\mathrm{ancilla}}, \\ &= \sum_{i=0}^{V-1}\sum_{j=0}^{15} \tilde{f}_i\tilde{\epsilon}_j|i\rangle_{\mathrm{phys}}|j\rangle_{\mathrm{sub}}|0\rangle_{\mathrm{ancilla}} + |*\rangle|1\rangle_{\mathrm{ancilla}}, \end{aligned} \tag{22}$$

where $\tilde{\epsilon}$ represents a normalized quantity. $|*\rangle$ represents the computationally unnecessary states, which are identified by the ancilla qubit being $|1\rangle_{\mathrm{ancilla}}$.

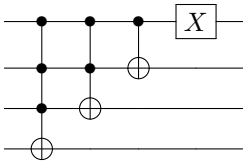
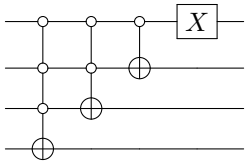
Next, so-called increment/decrement gates are applied on both subnode and physical qubits to associate the basis of subnode and physical amount at different points. The increment/decrement gates are operators that shift one computational basis, respectively. Specifically, those operator satisfy

$$\begin{aligned} U_{\mathrm{Incr.}}|i\rangle &= |i+1\rangle, \\ U_{\mathrm{Decr.}}|i\rangle &= |i-1\rangle. \end{aligned} \tag{23}$$

Suppose the periodic boundary condition on the $N$-qubits system:

$$\begin{aligned} U_{\mathrm{Incr.}}|2^N-1\rangle &= |0\rangle, \\ U_{\mathrm{Decr.}}|0\rangle &= |2^N-1\rangle, \end{aligned} \tag{24}$$

those operator follow the relation:$U^{\dagger}_{\mathrm{Incr.}} = U_{\mathrm{Decr.}}$. The increment circuit can be specifically configured as follows.

$U_{\mathrm{Incr.}} =$  , $U_{\mathrm{Decr.}} =$  (25)

By performing controlled-Increment/Decrement gates on the subnode as control registers and the physical qubits as target registers, we can map the subnode to a physical quantity on each lattice point. We call this sequential operations as the "shift operator". The circuit of the shift operator is shown in FIG.2.

As a result, after applying both the coin operator and the shift operator, we obtain the following state as a final output of this propagation step:

$$|\phi_2\rangle = \sum_{i=0}^{V-1}\sum_{j=0}^{15} \tilde{\epsilon}_j\tilde{f}_{i,j}|i\rangle_{\mathrm{phys}}|j\rangle_{\mathrm{sub}}|0\rangle_{\mathrm{ancilla}} + |*\rangle|1\rangle_{\mathrm{ancilla}}. \tag{26}$$

We can articulate the exact correlation between $\tilde{f}i$ and $\tilde{f}i,j$ as outlined herein. Initially, we had the capacity to signify the index $i$ as $i = sL + t$, $(0 \leq s < 6,\ 0 \leq t < L)$, which, for instance, correlates with the direction $x$ when $s = 0$, $y$ when $s = 1$, and so forth, and the coordinates of the corresponding directions are symbolized by $t$. The shift

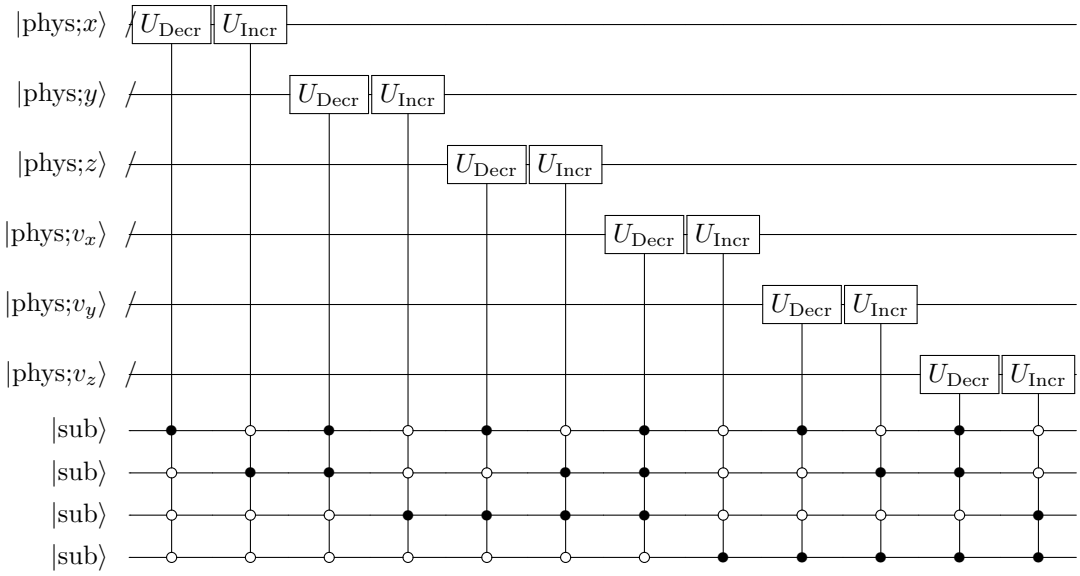



**Figure 2.** A Quantum circuit for the shift operators. Increment and Decrement operators controlled by subnodes are aligned according to the order of TABLE 1.

operator moves computational bases in each subspace by $\pm 1$, respecting periodic boundary conditions in each orientation. This $\pm 1$ direction is specified by the index $j$ as shown in TABLE 1. Therefore, $\tilde{f}_{i,j}$ can be represented as follows:

$$\tilde{f}_{i,j} = \tilde{f}_{sL+(t+(-1)^j) \bmod L}, \tag{27}$$

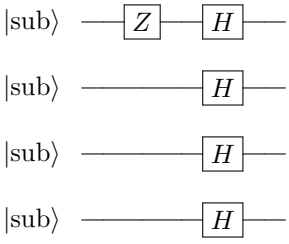
when $i = sL + t,\ (0 \leq s < 6,\ 0 \leq t < L)$.

### *2.1.3 Integration*

Passing through the encoding and propagation steps so far, we obtain a state in which the all 13 terms arising in the right-hand side of the equation (10) for a fixed time step under are encoded in the amplitude of each basis state. In this step, we perform a superposition of subnode states to compute the sum of all terms and collect them into the amplitude of a single state$|0000\rangle_{\text{sub}}$. However, As a preprocessing step, we need to invert the phases of certain states as explained below.

The amplitude of each basis are multiplied by the coefficients in the difference equation 10, excluding the explicit sign, which is denoted by sigma in TABLE 1. Therefore, we need to inverse the phase of corresponding state for the terms with a minus sign. This process is also very simple and only requires one application of Z gate as shown in circuit 28 before applying H gates.

Finally, we superimpose all sunode states by applying H as shown in circuit(28).




(28)

As a result, the amplitudes of the states from $|0000\rangle_{\rm sub}$to$|1111\rangle_{\rm sub}$ are summed and gathered as the amplitude of $|0000\rangle_{\rm sub}$ state with equal weighting of 1/4. Therefore, we finally obtain the following state

$$|\phi_3\rangle \quad = \quad \frac{1}{4}\sum_{i=0}^{V}\sum_{j=0}^{12}\sigma_j\tilde{\epsilon}_j\tilde{f}_{i,j}|i\rangle_{\rm phys}|0000\rangle_{\rm sub}|0\rangle_{\rm ancilla} + |*\rangle. \tag{29}$$

With more clear form, we can write:

$$\begin{aligned}\sum_{j=0}^{12}\sigma_j\tilde{\epsilon}_j\tilde{f}_{i,j} \quad &\sim \quad f-(f_{x+\Delta x}-f_{x-\Delta x})-(f_{y+\Delta y}-f_{y-\Delta y})-(f_{z+\Delta z}-f_{z-\Delta z}) \\ &\quad -\frac{q(\boldsymbol{E}+\boldsymbol{F})_x\Delta t}{2m\Delta v_x}(f_{v_x+\Delta v_x}-f_{v_x-\Delta v_x})-\frac{q(\boldsymbol{E}+\boldsymbol{F})_y\Delta t}{2m\Delta v_y}(f_{v_y+\Delta v_y}-f_{v_y-\Delta v_y}), \\ &= \quad f(x,y,z,v_x,v_y,v_z;t+\Delta t),\end{aligned} \tag{30}$$

where the distribution function is at the corresponding point of $(x, y, z, v_x, x_y, v_z)$ to the index $i$. Since the normalizing factors of $f$ and $\epsilon$ are involved here, the relation is denoted as "$\sim$".

According to the resultant state $|\phi_3\rangle$, we can measure the physical and subnode qubits and focus on the $|0\rangle_{\rm sub}$ to obtain a distribution function that is one time step evolved according to the Boltzmann-Maxwell equation. For further time steps, we can use this distribution function as an initial value to input to the first encoding step, and further time evolution can be implemented by performing similar steps.

Here are remarks on this algorithm, most of what is touched on here will be discussed more comprehensively in the Section 4. First, we asserted that the measurement of the state delivers the value of the distribution function; however, what is specifically attained is the square of the absolute value of the distribution function. Nevertheless, given that the value of the distribution function $f$ is consistently real and non-negative, the precise value of $f$ can be accurately recovered from the measurements. On the other hand, $\boldsymbol{E}$ and $\boldsymbol{B}$ handled by Maxwell solver in Appendix Appendix A are real but also have negative values, so not exactly the same algorithm can be used. However, during computation with real quantum algorithms, there isn't a genuine necessity to measure the values of $\boldsymbol{E}$ and $\boldsymbol{B}$. The primary function of the Maxwell solver is simply to convey these values to the Boltzmann solver within the quantum circuit, hence this does not present a significant issue. If one want to measure $\boldsymbol{E}$ and $\boldsymbol{B}$ values as well, a further ancilla node that identifies the sign must be prepared, and an additional quantum oracle is also needed.

Next, Actually measuring $f$ does not lead to quantum advantage. This is because $f$ still has $O(V = L^6)$ degrees of freedom, and it is inevitable to measure it $O(V)$ times in order to obtain full information. However, this problem can be avoided because what we are physically interested in is not $f$ itself, but the velocity moment quantity obtained by integrating $f$ with respect to velocity $v$. If we could implement this integral, i.e., just a sum in the discrete system, in an efficient quantum algorithm, the computational complexity would be superior to that of a naive classical algorithm. Furthermore, we believe that it is possible to reduce the Hilbert space to be measured based on physical conditions such as uniformity with respect to a certain spatial direction, limiting the measurement to the physical space of interest, etc.

## 3 Comparison

In this paper, all quantum circuits were exactly simulated by dealing directly with statevectors. Thus it is expected that the results will be in exact agreement with numerical calculations using conventional classical algorithms. We prepared $L = 8$ lattice sites

in each spatial and velocity direction and calculated with the volume $V = 8^6$. As for the quantum algorithm $6 \times \lceil \log_2 L \rceil = 18$ qubits were used as $|\mathrm{phys}\rangle$.

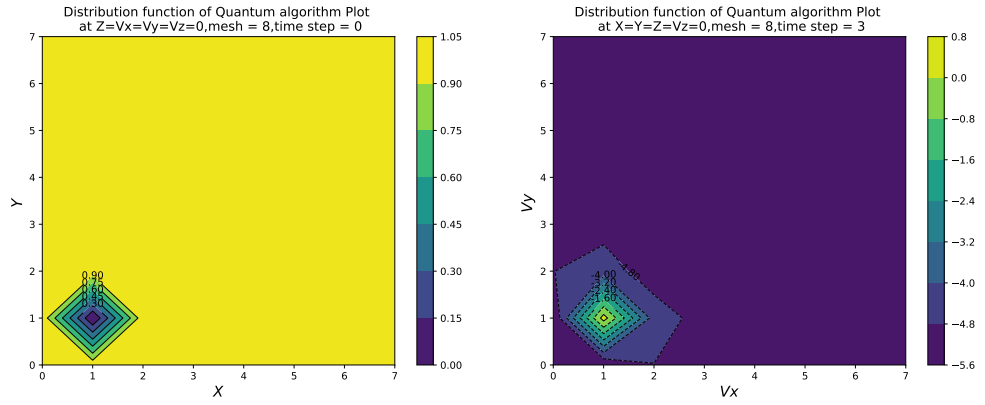
And we set $\Delta x = \Delta y = \Delta z = 30\mathrm{m}, \Delta t = 10^{-7}\mathrm{s}$, satisfying the assumption (9). Thereby, $v_x = v_y = v_z = 3 \times 10^8 \mathrm{m/s}$ is constant at the speed of light. The plasma particles are assumed to be positrons and set $e = 1.6 \times 10^{-19}\mathrm{C}, m_e = 9.1 \times 10^{-31}\mathrm{kg}$, so we put $\Delta v_x = \Delta v_y = \Delta v_z = 10^5 \mathrm{m/s}$. In this section, for simplicity, we re-scale variables $x, y, \cdots$ dividing by the unit $\Delta x, \Delta y, \cdots$ and denote them as coordinates on a lattice space. That is, $x = n$ denotes the point where $x = n\Delta x$ physically.

### 3.1 Initial condition

As the initial distribution function, we employed a simple setup: we set 0 for $(x = 1, y = 1)$ or $(v_x = 1, v_y = 1)$, and set 1 for the other spaces. Namely,

$$\begin{aligned} f(x, y, z, v_x, v_y, v_z; t = 0)|_{x=1 \cap y=1} &= 0, \\ f(x, y, z, v_x, v_y, v_z; t = 0)|_{v_x=1 \cap v_y=1} &= 0, \\ f(x, y, z, v_x, v_y, v_z; t = 0) &= 1 \text{ (otherwise)}. \end{aligned}$$

This is a simple setup to compare the agreement with the classical algorithm, and in practice it is necessary to give a suitable initial condition corresponding to considering physical phenomena such as plasma.



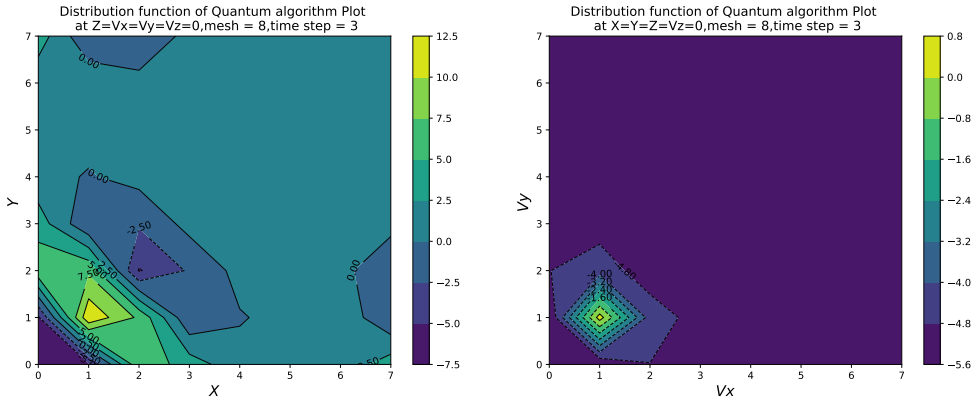

(a) (b)

**Figure 3.** The initial distribution function in the space for (a) the $x$ - $y$ subplane with $z = v_x = v_y = v_z = 0$, and (b) the $v_x$ - $v_y$ subplane with $x = y = z = v_z = 0$. This makes it possible to check the influence of electromagnetic fields on propagation in velocity space as well as in real space.

Since we implemented the Increment/Decrement circuits periodic (24), the simulation results are also periodic so that the 0-th and $L$-th lattice points are identical for all directions.

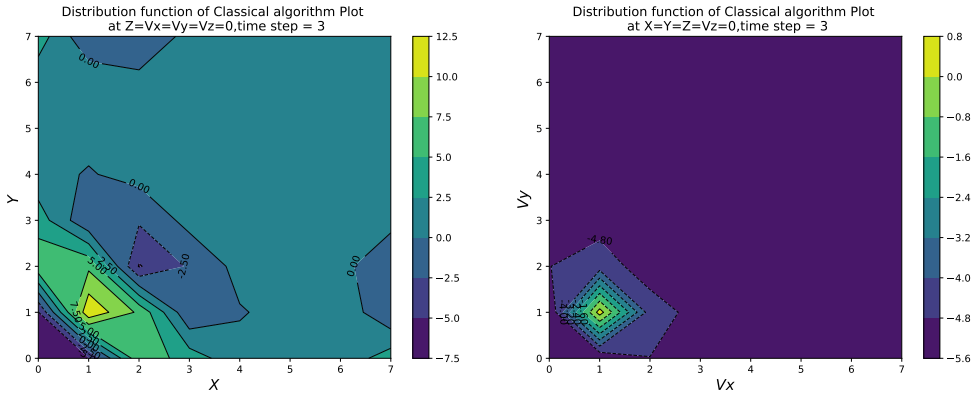
### 3.2 Simulation result

We implemented our quantum algorithm with the input conditions and advanced time evolution from time step = 0 to time step = 3.




(a) (b)

**Figure 4.** The results show (a) real space propagation at $z = v_x = v_y = v_z = 0$ and (b) velocity space propagation at $x = y = z = v_z = 0$ with time evolution to time step = 3 using our quantum algorithm.




(a) (b)

**Figure 5.** The results are based on a classical algorithm of the time evolution of the difference equations (4,6,7) using the same FTCS scheme as in this paper, with similar initial and boundary conditions. (a) shows real space propagation at $z = v_x = v_y = v_z = 0$ and (b) shows velocity space propagation at $x = y = z = v_z = 0$ with time evolution to time step = 3

Comparing FIG. 4 and FIG. 5, the simulation results of the quantum algorithm perfectly match those of the classical algorithm with similar conditions and methods. This is because we are simulating exactly with statevector in this case, and the actual results based on measurements will have statistical errors depending on the number of shots.

Although $f$ should take values between 0 and 1, this is not the case in FIG. 4 and FIG. 5. This is a consequence of numerical diffusion due to discretization using the FTCS scheme, which occurs universally in classical algorithms. As noted in the discussion, the numerical diffusion is reduced by $O(\Delta t)$ in the time direction and $O\left((\Delta x)^2\right)$ in the space direction , so it is guaranteed to give correct results if the calculation is performed on a sufficiently large system.

The propagation in real space and velocity space is different, showing that it is acted upon by the electromagnetic field solved with the Maxwell solver. We achieved one of our goals in this paper, that is, the coupling of the Boltzmann equation and the Maxwell equation. However, note that this is a unilateral interaction from the Maxwell equation, since the assumption of uniform velocity and vacuum condition is used.

## 4 Discussion

Our plasma simulator is not yet able to cover generic phenomena according to the governing equations (1,2,3). This paper is in the middle stage of our project. This means that our plasma simulator does not yet account for velocity inhomogeneity in the convective term of the distribution function, the interaction between electromagnetic fields and plasma particles, and the collisional effects. To add these physical effects, new quantum algorithms must be developed.

- Self-consistent collisionless Boltzmann-Maxwell equations interacting with the electromagnetic field by calculating $\rho$ charge density, velocity, and $\boldsymbol{j}$ current density in moment quantities of the distribution function:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0,$$
$$\nabla^2 \boldsymbol{E} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \frac{1}{\epsilon_0}\nabla\rho + \mu_0 \frac{\partial \boldsymbol{j}}{\partial t},$$
$$\nabla^2 \boldsymbol{B} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{B}}{\partial t^2} = -\mu_0 \left(\nabla \times \boldsymbol{j}\right).$$

The next stage will be to improve the current quantum algorithm to the quantum algorithm for the collisionless Boltzmann-Maxwell equation described above. To do this, a quantum algorithm that calculates the amount of velocity moments in the distribution function should be developed. Thereby, the electromagnetic field and plasma particles can interact with each other via velocity inhomogeneity, charge density, and current density. This stage can simulate all the complex kinetic effects of collisionless plasma in an electromagnetic field; it simulates macroscopic MHD phenomena that reflect kinetic effects as Micro phenomena. In other words, even macroscopic phenomena can fall back to microscopic phenomena, thus contributing to the complete understanding of the physical process and to the prediction. The domain covers space plasmas in space planetary science, such as the solar surface, and the earth's magnetosphere and astrophysics, such as black hole accretion disks and interstellar winds.

- Self-consistent collisional Boltzmann-Maxwell equations interacting with an electromagnetic field, with the addition of a first-principles collision term:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = Col(f, f'),$$
$$\nabla^2 \boldsymbol{E} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \frac{1}{\epsilon_0}\nabla\rho + \mu_0 \frac{\partial \boldsymbol{j}}{\partial t},$$
$$\nabla^2 \boldsymbol{B} - \frac{1}{c^2}\frac{\partial^2 \boldsymbol{B}}{\partial t^2} = -\mu_0 \left(\nabla \times \boldsymbol{j}\right).$$

Furthermore, in the final stage, this quantum algorithm will be improved to a quantum algorithm for computing the collision term from the distribution function. By adding a first-principles collision term, the domain of coverage is further extended. It covers the highly complex collisional effects of space plasma versus neutral atmospheres, simulating the ionospheric dynamics of various planetary systems; except for Maxwell solver, it calculates non-equilibrium states of rarefied gases first principles; apply Boltzmann solver and it solves problems of neutrinos and bubble structure in the universe.

We used a finite difference FTCS scheme as our numerical model; the FTCS scheme has numerical errors on the order of $O(\Delta t)$, $O(\Delta x_i^2)$ and $O(\Delta v_{x_i}^2)$ per time evolution. Previously, 6D Vlasov simulation research using classical computers has been able to allocate only $L \sim 100$ ($L$: lattices per spatial degree of freedom), even using supercomputers. Therefore, simple numerical methods such as the FTCS scheme are not very appropriate for classical algorithms because of the large numerical errors. However, in the case of quantum computation with a large-scale quantum computer in a domain that is impossible with a classical computer, the number of lattices per spatial degree of freedom ($\gg 100$ lattices) is a very large quantity, and thus the numerical error is inevitably very small. For example, we estimate that $L > 10^6$ is needed to simulate the auroral electron acceleration problem in the magnetosphere-ionosphere. For that very large $L$, the numerical error from the FTCS scheme is small enough. Moreover, since $L$ increases exponentially with the line increase in hardware logical qubits, the speed of expansion and growth of the computational domain and the speed of improvement in accuracy become exponential.

The greatest advantage of quantum algorithms over classical algorithms is massively parallelization. We estimate the Quantum Volume of our quantum algorithm and describe the quantum advantage of the Boltzmann-Maxwell equation. Simply, we will call Quantum Volume=width(number of qubits)$\times$ depth(number of gates) in our quantum algorithm. The width of this quantum algorithm is $6\log_2(L)+6$ where $L$ denotes the number of lattice points in each direction. Comparing to the classical algorithm $O(L)$ computational complexity of the classical algorithm, the fact that it can be expressed in $\log_2(L)$ qubits is a quantum advantage. On the other hand, the measured quantum circuits for $L = 2$, $L = 4$, and $L = 8$, were found to be approximately $600 \times \log_2(L)$ per time evolution. In case of time evolution to Time step $= N_t$, the approximated Quantum Volume would be $3600 \times N_t \log_2(L)\,(\log_2(L) + 1)$. This is of the order of of the scale: $O\left(N_t\,(\log_2(L))^2\right)$. Compared to the computational volume of a similar classical algorithm $O\left(N_t L^6\right)$, the order is improved by compression of 6D spatial information. Thus, the larger $L$ is, the higher the quantum superiority.

Our quantum algorithms are intended for a future large-scale quantum computer, but there remain several issues in terms of efficient algorithms. There is a problem of the efficient preparation of the initial distribution function on quantum circuits. The Encoding step Appendix B method has the exponential complexity $O\left(2^N\right)$ of preparing arbitrary quantum states in a $2^N$-dimensional Hilbert space with an $N$ qubit(Zalka, 1998; Georgescu et al., 2014). This problem is an important topic in quantum computation, and various efficient methods have been proposed. For example, Georgescu et al. developed an efficient method to prepare quantum states with polynomial complexity in a number of qubits(Georgescu et al., 2014), and other efficient quantum state initialization methods such as log-concave. Other efficient methods for specific cases, such as log-concave probability distribution functions, have been reported as well(Grover & Rudolph, 2002). Although the initial distribution function varies depending on the physical phenomenon to be simulated, the Maxwell velocity distribution function, for example, is a log-concave probability distribution function and may be efficiently prepared(Todorova & Steijl, 2020).

Our quantum algorithm is more efficient than the classical algorithm in spatial information, but not in the time direction. The reason for this is that the finite difference method of a numerical computation does not allow time information to enter the width of quantum circuits. The finite difference method is a time-marching-based method for classical numerical calculations using the forward term on the left side of the difference equation. Due to its nature, one of the degrees of freedom must always be in the depth when implemented in a quantum computer. Variables that are not set to width are not accelerated, so there are restrictions on the number of lattices with respect to the number of degrees of freedom that can be set to depth, even for large-scale quantum computation. One simple way to improve this is to rewrite the difference equation of the fi-

nite difference method so that the smallest number of lattice degrees of freedom is the evolution parameter instead of time. Although only one degree of freedom is restricted, this method can keep the depth relatively small.

A common problem in quantum differential equation solving is the problem of vanishing time-marching-based measurement probabilities. In general terms, quantum linear system algorithms have an exponentially decreasing measurement probability with respect to the time step, depending on the number of time steps. The quantum algorithm in this study suffers from the same problem. The first possible solution to this problem is the application of the compression gadget proposed by Fang et al(Fang et al., 2023). This is a time-marching-based quantum differential equation solving method that is independent of time steps by repeating uniform singular value amplification.They verified their implementation on linear ODEs, but it may be applicable to our PDEs. Next, we also consider the use of different quantum differential equation solving methods as a solution. Hamiltonian simulations are a common method for solving quantum differential equations, and the Vlasov-poisson and Vlasov-Maxwell equations have already been used (Toyoizumi et al., 2023; Engel et al., 2019). While it is easy to implement the compression gadget (Fang et al., 2023) within a Hamiltonian simulation, we consider that it is difficult to implement the nonlinear Boltzmann-Maxwell equations with first-principles collision terms in a Hamiltonian simulation.

## 5 Summary

In this paper, a novel quantum algorithm for solving the Boltzmann-Maxwell equation for collisionless plasmas has been formulated; both the Boltzmann and Maxwell equation solvers were structured with a similar quantum circuit. To confirm the validity of our quantum algorithm, we performed simulations of the distribution function propagation process under the background electromagnetic field propagation using the Qiskit platform. We compared the results of the quantum calculation with the results of the parallel classical calculation and found perfect agreement between them. This completes the framework for efficiently solving nonlinear problems in various plasmas, such as space plasmas. Prospective endeavors may cultivate the development of a more generalized quantum algorithm for the Boltzmann-Maxwell equation for collisional plasmas, wherein the vacuum condition is eliminated and first-principles collision terms are incorporated.

## 6 Acknowledgment

Discussions during the Yukawa Institute for Theoretical Physics (YITP) summer school YITP-W-22-13 on "A novel numerical approach to quantum field theories" were useful as we started this work. HH would like to acknowledge the financial support of the Kyushu University Innovator Fellowship Program (Quantum Science Area). The work of HH and AY is supported by JSPS KAKENHI Grant Numbers JP20H01961 and JP22K21345. The work of JWP is supported in part by the JSPS Grant-in-Aid for Research Fellow Number 22J14732 and the JST SPRING, Grant Number JPMJSP2108.

## Appendix A Maxwell solver

The basic structure of the Maxwell solver is almost identical to that of the Boltzmann solver. Similar to the Boltzmann solver, the Maxwell solver consists of three steps: encoding, propagation, and integration. The algorithm is briefly described, with special emphasis on the differences to the Boltzmann solver.

### A1 Encoding

In Maxwell solver, the physical quantities E and B are written together as g, and develop them simultaneously according to the equations (11,12). Since there are no velocity degrees of freedom, only $N_{\rm phys} = 3\lceil \log_2 L \rceil$ qubit are prepared for $|{\rm phys}\rangle$, and one additional qubit representing time is also prepared. $|{\rm sub}\rangle$ requires $N_{\rm sub} = 6$ qubit in this case. This is because we need $N_{\rm species} = 1$ qubit to distinguish the difference of the physical quantity, namely $\boldsymbol{E}$ or $\boldsymbol{B}$, $N_{\rm direction} = 2(= \lceil \log_2 3 \rceil)$ qubits to specify the elements of the vector for them as they are vector, and $N_{\rm term} = 3(= \lceil \log_2 8 \rceil)$ qubits to indicate 8 terms appearing the equations (11,12). Collectively, these are called subnodes, but their roles are actually divided as follows:

$$|{\rm sub}\rangle \to |{\rm species}\rangle|{\rm direction}\rangle|{\rm sub}\rangle. \tag{A1}$$

These correspondences are shown in Table A1 where $\epsilon$ and $\sigma$ represent the the coefficient and explicit sign of each term in the equations (11,12). Therefore, using exactly the same algorithm as the Boltzmann solver, we obtain the following state as the outcome of this encoding step:

$$|\phi_1\rangle = \sum_{i=0}^{V-1}\sum_{s=0}^{1}\sum_{d=0}^{2} \tilde{g}_{i,t,d}|i\rangle_{\rm phys}|0\rangle_{\rm time}|s\rangle_{\rm species}|d\rangle_{\rm direction}|0\rangle_{\rm term}|0\rangle_{\rm ancilla}, \tag{A2}$$

where the subscript i indicates a lattice point using the same rules as in the Boltzmann solver, $g_{i,t,d}$ are given in TABLE A1, and $\tilde{g}$ is normalized $g$. At the first time step we need to specify the initial values for $g$.

### A2 Propagation

The structure of the Propagation step in Maxwell solver is fundamentally a Quantum Walk, similar to the Propagation in Boltzmann solver. Thus we need to construct the coin operator and the shift operator. However, the elements of the Coin operator, the time qubits, and the type of subnodes are different. Furthermore, the time increment circuit is used only with respect to the state $|111\rangle_{\rm sub}$ to use the physical quantity of one previous time. Therefore, in this section, Propagation step generate the states corresponding to the terms propagated in space-time by using the increment and decrement circuits.

The coin operator acts on the subnodes.

$$U_{\rm coin}|s\rangle_{\rm species}|d\rangle_{\rm direction}|j\rangle_{\rm sub} = \tilde{\epsilon}_{s,d,j}|s\rangle_{\rm species}|d\rangle_{\rm direction}|j\rangle_{\rm sub}, \tag{A3}$$

where you can also find $\epsilon_{s,d,j}$ in TABLE A1 and $\tilde{\epsilon}$ is normalized $\epsilon$.

One difference from the Boltzmann solver is that the right-hand side of the expression (11,12) contains a term $g_{i,t-1,s,d}$ that also evolves in the time direction. This effect can be easily implemented by treating time as part of the spatial direction and applying the shift operator in the same way, but note that only the increment circuit is operated since the direction is only negative. After operating the coin and the shift operator, we obtain the following state as the outcome of this propagation step:

$$\begin{aligned}|\phi_2\rangle \quad = \quad & \sum_{i=0}^{V-1}\sum_{t=0}^{1}\sum_{s=0}^{1}\sum_{d=0}^{2}\sum_{j=0}^{7} \tilde{\epsilon}_{s,d,j}\tilde{g}_{i,t,s,d}|i\rangle_{\rm phys}|t\rangle_{\rm time}|s\rangle_{\rm species}|d\rangle_{\rm direction}|j\rangle_{\rm sub}|0\rangle_{\rm ancilla} \\ & +|*\rangle|1\rangle_{\rm ancilla},\end{aligned} \tag{A4}$$

where $\tilde{g}_{i,t,s,d}$ represents the shift of $\pm 1$ unit in each spatial and the temporal. As for the time direction, $|1\rangle_{\rm time}|111\rangle_{\rm sub}$ and the initial amplitude at $|0\rangle_{\rm time}|000\rangle_{\rm sub}$ are exchanged by the increment circuit (25).The reason for this exchange is because one previous time state is needed to generate a term that propagates in the time direction.

### A3 Integration

In contrast to the Boltzmann equation, the Maxwell equation is a second-order differential equation. As a result, the signs $\sigma_j$ that appear in the corresponding difference equation (10) differ from those in the Boltzmann equation (as shown in Table A1). In such cases, an controlled-inverse gate, which is shown as follows, should be applied prior to the superposition by the H gate:

$$U_{\text{Inv.}} = \begin{array}{l} |\text{sub}\rangle \\ |\text{sub}\rangle \\ |\text{sub}\rangle \end{array} \quad \begin{array}{c} \bullet \\ | \\ \bullet \end{array}$$

The rest of the integration step can use the same method as the Boltzmann solver, but this time we are dealing with different physical quantities,$\boldsymbol{E}$ and $\boldsymbol{B}$, in the same circuit, so we need to sum each of them and not confuse them. As a result, we can specify the spatial lattice point $(i)$ and the species, and obtain the time-evolved quantities $\boldsymbol{E}, \boldsymbol{B}$ developed in the amplitude of $|000\rangle_{\text{sub}}$.

**Table A1.** The subnode bases and their corresponding physical quantities. $g, \epsilon$, and $\sigma$ respectively represent the (unnormalized) electromagnetic fields associated with each basis state, the coefficients to be incorporated via the coin operator, and the sign to be multiplied during the integration step. These are the quantities that appear on the right side of the difference equations (11,12). Here we write only for $|\text{direction}\rangle = |00\rangle_{\text{direction}}$ as an example; $|01\rangle_{\text{direction}}$ and $|10\rangle_{\text{direction}}$ correspond to the $y$- and $z$- components of $\boldsymbol{E}$ and $\boldsymbol{F}$, respectively.

| $\lvert s\rangle_{\text{species}}\lvert d=0\rangle_{\text{direction}}\lvert j\rangle_{\text{sub}}$ | $g_{s,d=0,j}$ | $\epsilon_{s,d=0,j}$ | $\sigma_j$ |
|---|---|---|---|
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 000\rangle_{\text{sub}}$ | $E_x(x,y,z;t)$ | $-4$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 001\rangle_{\text{sub}}$ | $E_x(x+\Delta x,y,z;t)$ | $1$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 010\rangle_{\text{sub}}$ | $E_x(x-\Delta x,y,z;t)$ | $1$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 011\rangle_{\text{sub}}$ | $E_x(x,y+\Delta y,z;t)$ | $1$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 100\rangle_{\text{sub}}$ | $E_x(x,y-\Delta y,z;t)$ | $1$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 101\rangle_{\text{sub}}$ | $E_x(x,y,z+\Delta z;t)$ | $1$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 110\rangle_{\text{sub}}$ | $E_x(x,y,z-\Delta z;t)$ | $1$ | $+1$ |
| $\lvert 0\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 111\rangle_{\text{sub}}$ | $E_x(x,y,z;t-\Delta t)$ | $1$ | $-1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 000\rangle_{\text{sub}}$ | $F_x(x,y,z;t)$ | $-4$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 001\rangle_{\text{sub}}$ | $F_x(x+\Delta x,y,z;t)$ | $1$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 010\rangle_{\text{sub}}$ | $F_x(x-\Delta x,y,z;t)$ | $1$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 011\rangle_{\text{sub}}$ | $F_x(x,y+\Delta y,z;t)$ | $1$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 100\rangle_{\text{sub}}$ | $F_x(x,y-\Delta y,z;t)$ | $1$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 101\rangle_{\text{sub}}$ | $F_x(x,y,z+\Delta z;t)$ | $1$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 110\rangle_{\text{sub}}$ | $F_x(x,y,z-\Delta z;t)$ | $1$ | $+1$ |
| $\lvert 1\rangle_{\text{species}}\lvert 00\rangle_{\text{direction}}\lvert 111\rangle_{\text{sub}}$ | $F_x(x,y,z;t-\Delta t)$ | $1$ | $-1$ |

## Appendix B Construction of our coin operator

In this section we consider an algorithm to multiply a vector to each quantum basis. Let $\Lambda$ denote the multiplying vector:

$$\Lambda = (\lambda_0, \lambda_2, \cdots \lambda_{M-1}), \tag{B1}$$

where we suppose that $\{\lambda\}$ take real values and $\Lambda$ be normalized: $\sum_i \lambda_i^2 = 1$.

To implement this algorithm, we need operate a diagonal matrix $\mathcal{A}$ having entries corresponding to $\Lambda$ but this cannot be done directly because it is not unitary operator

in general. Thus we realized this non-unitary operation by using one ancilla qubit and embedding the matrix $\mathcal{A}$ in a unitary matrix with larger size, which is known as the block encoding method. As $\{\lambda\}$ are always real, this procedure can easily be implemented as follows:

$$U = \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B} & -\mathcal{A} \end{pmatrix}, \tag{B2}$$

with

$$\mathcal{A} = \mathrm{diag}\left(\lambda_1, \lambda_2, \cdots\right), \tag{B3}$$

$$\mathcal{B} = \mathrm{diag}\left(\sqrt{1-\lambda_1^2}, \sqrt{1-\lambda_2^2}, \cdots\right). \tag{B4}$$

After performing this unitary operation on an arbitrary state:

$$|\psi\rangle = \sum_i \alpha_i |i\rangle_{phys} |0\rangle_{anc}, \tag{B5}$$

we obtain the following state:

$$|\psi'\rangle = U|\psi\rangle, \tag{B6}$$

$$= \sum_i \lambda_i \alpha_i |i\rangle_{\mathrm{phys}} |0\rangle_{\mathrm{ancilla}} + |*\rangle |1\rangle_{\mathrm{ancilla}}, \tag{B7}$$

which we can distinguish desired/unnecessary states with $|0/1\rangle_{\mathrm{ancilla}}$.

## References


Aharonov, Y., Davidovich, L., & Zagury, N. (1993, Aug). Quantum random walks. *Phys. Rev. A*, *48*, 1687–1690. Retrieved from `https://link.aps.org/doi/10.1103/PhysRevA.48.1687` doi: 10.1103/PhysRevA.48.1687

Arrazola, J. M., Kalajdzievski, T., Weedbrook, C., & Lloyd, S. (2019, Sep). Quantum algorithm for nonhomogeneous linear partial differential equations. *Phys. Rev. A*, *100*, 032306. Retrieved from `https://link.aps.org/doi/10.1103/PhysRevA.100.032306` doi: 10.1103/PhysRevA.100.032306

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J., Barends, R., ... Martinis, J. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, *574*, 505–510. Retrieved from `https://www.nature.com/articles/s41586-019-1666-5`

Berry, D. W., Childs, A. M., Ostrander, A., & Wang, G. (2017, oct). Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, *356*(3), 1057–1081. Retrieved from `https://doi.org/10.1007/Fs00220-017-3002-y` doi: 10.1007/s00220-017-3002-y

Budinski, L. (2022). Quantum algorithm for the navier–stokes equations by using the streamfunction-vorticity formulation and the lattice boltzmann method. *International Journal of Quantum Information*, *20*(02), 2150039. Retrieved from `https://doi.org/10.1142/S0219749921500398` doi: 10.1142/S0219749921500398

Cao, Y., Papageorgiou, A., Petras, I., Traub, J., & Kais, S. (2013, jan). Quantum algorithm and circuit design solving the poisson equation. *New Journal of Physics*, *15*(1), 013021. Retrieved from `https://doi.org/10.1088/F1367-2630/15/1/013021` doi: 10.1088/1367-2630/15/1/013021

Childs, A. M., Liu, J.-P., & Ostrander, A. (2021, nov). High-precision quantum algorithms for partial differential equations. *Quantum*, *5*, 574. Retrieved from `https://doi.org/10.22331/Fq-2021-11-10-574` doi: 10.22331/q-2021-11-10-574

Daughton, W. (2003, 07). Electromagnetic properties of the lower-hybrid drift instability in a thin current sheet. *Physics of Plasmas*, *10*(8), 3103-3119. Retrieved from https://doi.org/10.1063/1.1594724 doi: 10.1063/1.1594724

Douglas, B. L., & Wang, J. B. (2009, May). Efficient quantum circuit implementation of quantum walks. *Phys. Rev. A*, *79*, 052335. Retrieved from https://link.aps.org/doi/10.1103/PhysRevA.79.052335 doi: 10.1103/PhysRevA.79.052335

Engel, A., Smith, G., & Parker, S. E. (2019, Dec). Quantum algorithm for the vlasov equation. *Phys. Rev. A*, *100*, 062315. Retrieved from https://link.aps.org/doi/10.1103/PhysRevA.100.062315 doi: 10.1103/PhysRevA.100.062315

Fang, D., Lin, L., & Tong, Y. (2023, March). Time-marching based quantum solvers for time-dependent linear differential equations. *Quantum*, *7*, 955. Retrieved from https://doi.org/10.22331/q-2023-03-20-955 doi: 10.22331/q-2023-03-20-955

Fillion-Gourdeau, F. m. c., MacLean, S., & Laflamme, R. (2017, Apr). Algorithm for the solution of the dirac equation on digital quantum computers. *Phys. Rev. A*, *95*, 042343. Retrieved from https://link.aps.org/doi/10.1103/PhysRevA.95.042343 doi: 10.1103/PhysRevA.95.042343

Gaitan, F. (2020). Finding flows of a navier–stokes fluid through quantum computing. *npj Quantum Information*, *6*, 1-6.

Gaitan, F. (2021). Finding solutions of the navier-stokes equations through quantum computing—recent progress, a generalization, and next steps forward. *Advanced Quantum Technologies*, *4*(10), 2100055. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.202100055 doi: https://doi.org/10.1002/qute.202100055

Georgescu, I. M., Ashhab, S., & Nori, F. (2014, Mar). Quantum simulation. *Rev. Mod. Phys.*, *86*, 153–185. Retrieved from https://link.aps.org/doi/10.1103/RevModPhys.86.153 doi: 10.1103/RevModPhys.86.153

Grover, L., & Rudolph, T. (2002). *Creating superpositions that correspond to efficiently integrable probability distributions.*

Harrow, A. W., Hassidim, A., & Lloyd, S. (2009, Oct). Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, *103*, 150502. Retrieved from https://link.aps.org/doi/10.1103/PhysRevLett.103.150502 doi: 10.1103/PhysRevLett.103.150502

Mezzacapo, A., Sanz, M., Lamata, L., Egusquiza, I. L., Succi, S., & Solano, E. (2015, aug). Quantum simulator for transport phenomena in fluid flows. *Scientific Reports*, *5*(1). Retrieved from https://doi.org/10.1038/srep13153 doi: 10.1038/srep13153

Miller, W., Succi, S., & Mansutti, D. (2001, Apr). Lattice boltzmann model for anisotropic liquid-solid phase transition. *Phys. Rev. Lett.*, *86*, 3578–3581. Retrieved from https://link.aps.org/doi/10.1103/PhysRevLett.86.3578 doi: 10.1103/PhysRevLett.86.3578

Minoshima, T., Matsumoto, Y., & Amano, T. (2011). Multi-moment advection scheme for vlasov simulations. *Journal of Computational Physics*, *230*(17), 6800-6823. Retrieved from https://www.sciencedirect.com/science/article/pii/S0021999111003147 doi: https://doi.org/10.1016/j.jcp.2011.05.010

Moritaka, T., & Horiuchi, R. (2008, 09). Roles of ion and electron dynamics in the onset of magnetic reconnection due to current sheet instabilities. *Physics of Plasmas*, *15*(9). Retrieved from https://doi.org/10.1063/1.2979316 (092114) doi: 10.1063/1.2979316

Ohtani, S., & Yoshikawa, A. (2016). The initiation of the poleward boundary intensification of auroral emission by fast polar cap flows: A new interpretation based on ionospheric polarization. *Journal of Geophysical Re-*

*search: Space Physics*, *121*(11), 10,910-10,928. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016JA023143 doi: https://doi.org/10.1002/2016JA023143

Shor, P. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science* (p. 124-134). doi: 10.1109/SFCS.1994.365700

Steijl, R. (2019). Quantum algorithms for fluid simulations. In F. Bulnes, V. N. Stavrou, O. Morozov, & A. V. Bourdine (Eds.), *Advances in quantum communication and information* (chap. 3). Rijeka: IntechOpen. Retrieved from https://doi.org/10.5772/intechopen.86685 doi: 10.5772/intechopen.86685

Steijl, R. (2023). Quantum circuit implementation of multi-dimensional non-linear lattice models. *Applied Sciences*, *13*(1). Retrieved from https://www.mdpi.com/2076-3417/13/1/529 doi: 10.3390/app13010529

Steijl, R., & Barakos, G. N. (2018). Parallel evaluation of quantum algorithms for computational fluid dynamics. *Computers and Fluids*, *173*, 22-28. Retrieved from https://www.sciencedirect.com/science/article/pii/S0045793018301841 doi: https://doi.org/10.1016/j.compfluid.2018.03.080

Succi, S., Fillion-Gourdeau, F., & Palpacelli, S. (2015, may). Quantum lattice boltzmann is a quantum walk. *EPJ Quantum Technology*, *2*(1). Retrieved from https://doi.org/10.1140/epjqt/s40507-015-0025-1 doi: 10.1140/epjqt/s40507-015-0025-1

Todorova, B. N., & Steijl, R. (2020). Quantum algorithm for the collisionless boltzmann equation. *Journal of Computational Physics*, *409*, 109347. Retrieved from https://www.sciencedirect.com/science/article/pii/S0021999120301212 doi: https://doi.org/10.1016/j.jcp.2020.109347

Toyoizumi, K., Yamamoto, N., & Hoshino, K. (2023). *Hamiltonian simulation using quantum singular value transformation: complexity analysis and application to the linearized vlasov-poisson equation.*

Umeda, T. (2008, 07). A conservative and non-oscillatory scheme for vlasov code simulations. *Earth, Planets and Space*, *60*(7), 773-779. Retrieved from https://cir.nii.ac.jp/crid/1360855568571728128 doi: 10.1186/bf03352826

Umeda, T. (2012, 02). Effect of ion cyclotron motion on the structure of wakes: A vlasov simulation. *Earth, Planets and Space*, *64*(2), 231-236. Retrieved from https://cir.nii.ac.jp/crid/1360574095306421632 doi: 10.5047/eps.2011.05.035

Umeda, T., Ito, Y., & Fukazawa, K. (2013, aug). Global vlasov simulation on magnetospheres of astronomical objects. *Journal of Physics: Conference Series*, *454*(1), 012005. Retrieved from https://dx.doi.org/10.1088/1742-6596/454/1/012005 doi: 10.1088/1742-6596/454/1/012005

Umeda, T., Kimura, T., Togano, K., Fukazawa, K., Matsumoto, Y., Miyoshi, T., ... Ogino, T. (2011, 01). Vlasov simulation of the interaction between the solar wind and a dielectric body. *Physics of Plasmas*, *18*(1). Retrieved from https://doi.org/10.1063/1.3551510 (012908) doi: 10.1063/1.3551510

Umeda, T., Miwa, J.-i., Matsumoto, Y., Nakamura, T. K. M., Togano, K., Fukazawa, K., & Shinohara, I. (2010, 05). Full electromagnetic Vlasov code simulation of the Kelvin–Helmholtz instability. *Physics of Plasmas*, *17*(5). Retrieved from https://doi.org/10.1063/1.3422547 (052311) doi: 10.1063/1.3422547

Umeda, T., Nariyuki, Y., & Kariya, D. (2012). A non-oscillatory and conservative semi-lagrangian scheme with fourth-degree polynomial interpolation for solving the vlasov equation. *Computer Physics Communications*, *183*(5), 1094-1100. Retrieved from https://www.sciencedirect.com/science/article/pii/S0010465512000264 doi: https://doi.org/10.1016/j.cpc.2012.01.011

Umeda, T., Togano, K., & Ogino, T. (2009). Two-dimensional full-electromagnetic vlasov code with conservative scheme and its application to magnetic reconnection. *Computer Physics Communications*, *180*(3), 365-374. Retrieved from https://www.sciencedirect.com/science/article/pii/S0010465508003767 doi: https://doi.org/10.1016/j.cpc.2008.11.001

Umeda, T., Togano, K., & Ogino, T. (2010, 05). Structures of diffusion regions in collisionless magnetic reconnection. *Physics of Plasmas*, *17*(5). Retrieved from https://doi.org/10.1063/1.3403345 (052103) doi: 10.1063/1.3403345

Umeda, T., Ueno, S., & Nakamura, T. K. M. (2014, may). Ion kinetic effects on nonlinear processes of the kelvin–helmholtz instability. *Plasma Physics and Controlled Fusion*, *56*(7), 075006. Retrieved from https://dx.doi.org/10.1088/0741-3335/56/7/075006 doi: 10.1088/0741-3335/56/7/075006

Usami, S., Horiuchi, R., Ohtani, H., & Den, M. (2014, nov). Multi-hierarchy simulation of collisionless driven reconnection by real-space decomposition. *Journal of Physics: Conference Series*, *561*(1), 012021. Retrieved from https://dx.doi.org/10.1088/1742-6596/561/1/012021 doi: 10.1088/1742-6596/561/1/012021

Usami, S., Ohtani, H., Horiuchi, R., & Den, M. (2009). First demonstration of collisionless driven reconnection in a multi-hierarchy simulation. *Plasma and Fusion Research*, *4*, 049-049. doi: 10.1585/pfr.4.049

Wang, S., Wang, Z., Li, W., Fan, L., Wei, Z., & Gu, Y. (2020). Quantum fast poisson solver: the algorithm and complete and modular circuit design. *Quantum Information Processing*, *19*(6), 170. Retrieved from https://doi.org/10.1007/s11128-020-02669-7 doi: 10.1007/s11128-020-02669-7

Yepez, J. (1998). Lattice-gas quantum computation. *International Journal of Modern Physics C*, *09*, 1587-1596.

Yepez, J. (2001, Mar). Quantum lattice-gas model for computational fluid dynamics. *Phys. Rev. E*, *63*, 046702. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.63.046702 doi: 10.1103/PhysRevE.63.046702

Yoshikawa, A., Amm, O., Vanhamäki, H., & Fujii, R. (2013). Illustration of cowling channel coupling to the shear alfven wave. *Journal of Geophysical Research: Space Physics*, *118*(10), 6405-6415. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/jgra.50513 doi: https://doi.org/10.1002/jgra.50513

Zalka, C. (1998). Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1969), 313-322. Retrieved from https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1998.0162 doi: 10.1098/rspa.1998.0162

**Figure1.**

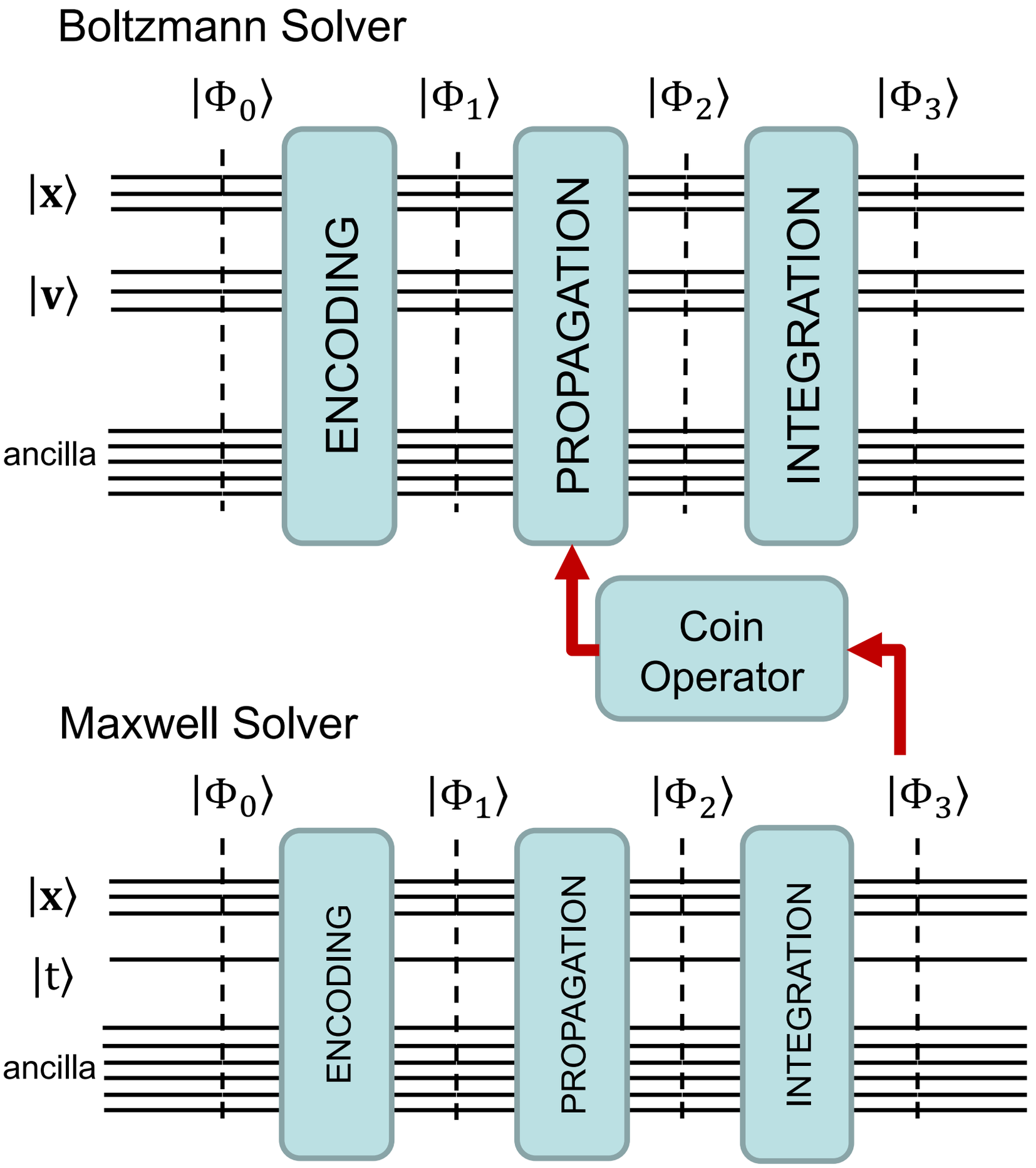

Boltzmann Solver
$|\Phi_0\rangle$
$|\Phi_1\rangle$
$|\Phi_2\rangle$
$|\Phi_3\rangle$
$|\mathbf{x}\rangle$
$|\mathbf{v}\rangle$
ancilla
ENCODING
PROPAGATION
INTEGRATION
Coin Operator
Maxwell Solver
$|\Phi_0\rangle$
$|\Phi_1\rangle$
$|\Phi_2\rangle$
$|\Phi_3\rangle$
$|\mathbf{x}\rangle$
$|\mathrm{t}\rangle$
ancilla
ENCODING
PROPAGATION
INTEGRATION

**Figure3-a.**

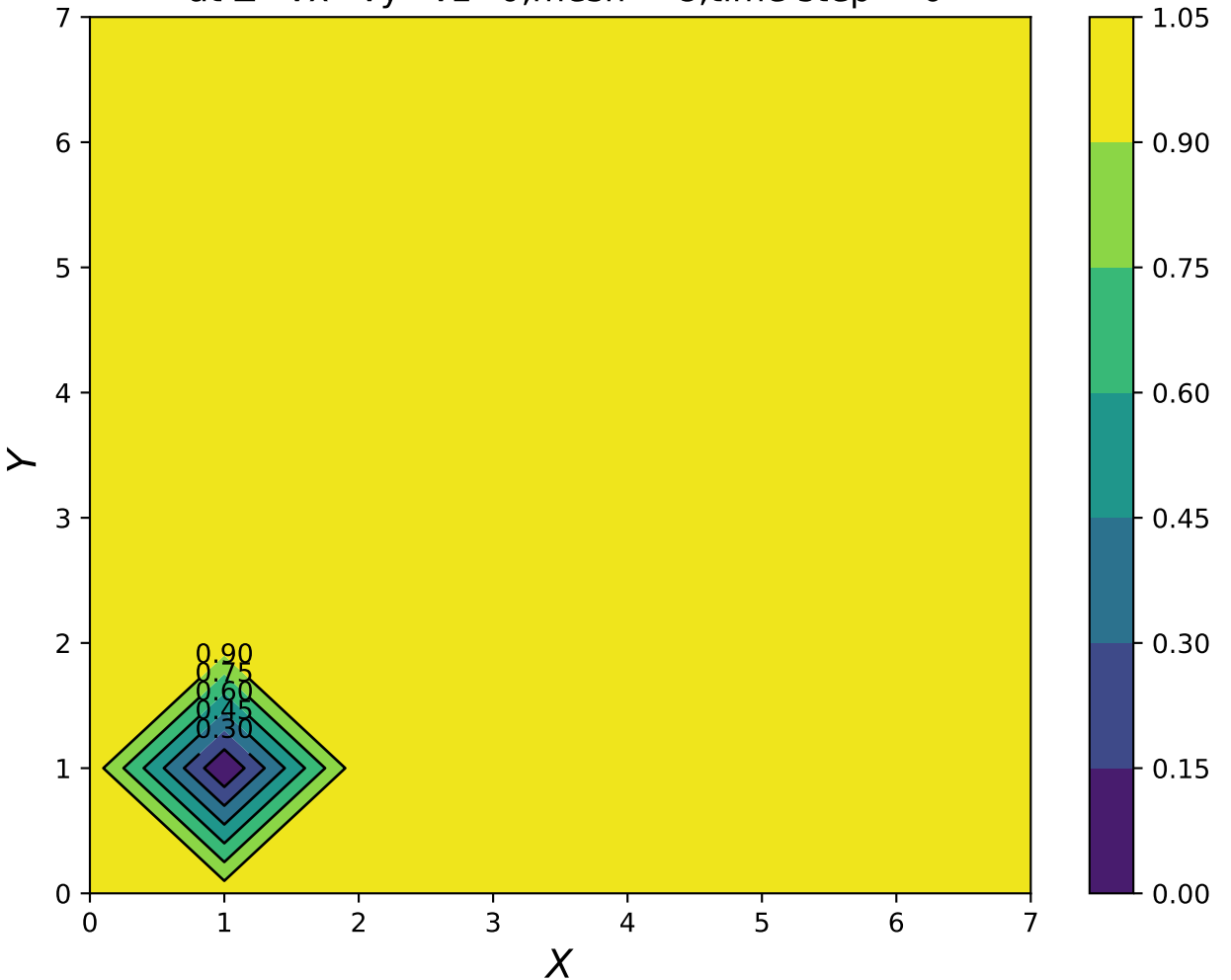

Distribution function of Quantum algorithm Plot
at Z=Vx=Vy=Vz=0,mesh = 8,time step = 0
0.90
0.75
0.60
0.45
0.30
Y
X
0
1
2
3
4
5
6
7
1.05
0.90
0.75
0.60
0.45
0.30
0.15
0.00

**Figure3-b.**

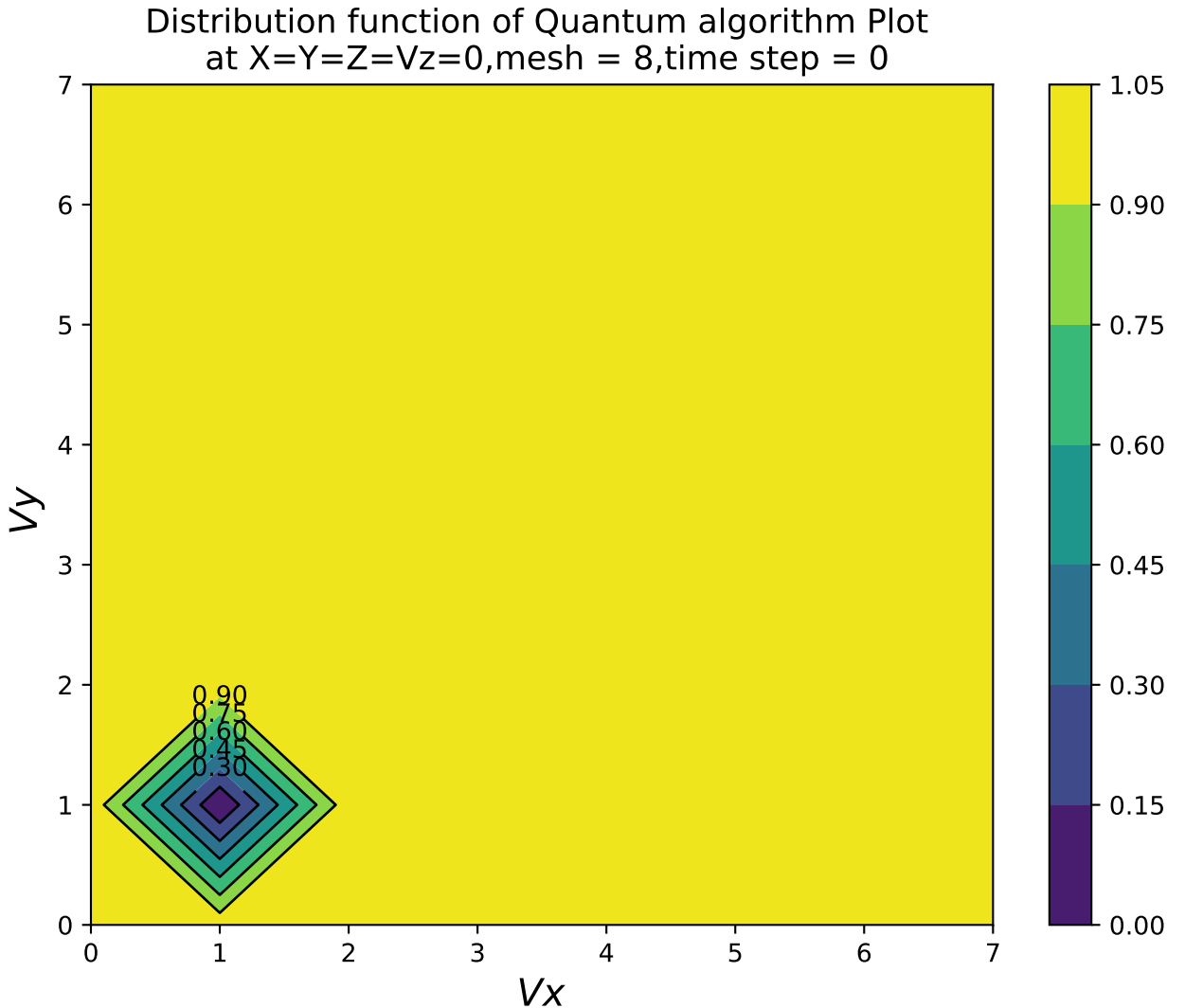

Distribution function of Quantum algorithm Plot
at X=Y=Z=Vz=0,mesh = 8,time step = 0
Vy
Vx
0.90
0.75
0.60
0.45
0.30
0
1
2
3
4
5
6
7
0.00
0.15
0.30
0.45
0.60
0.75
0.90
1.05

**Figure4-a.**

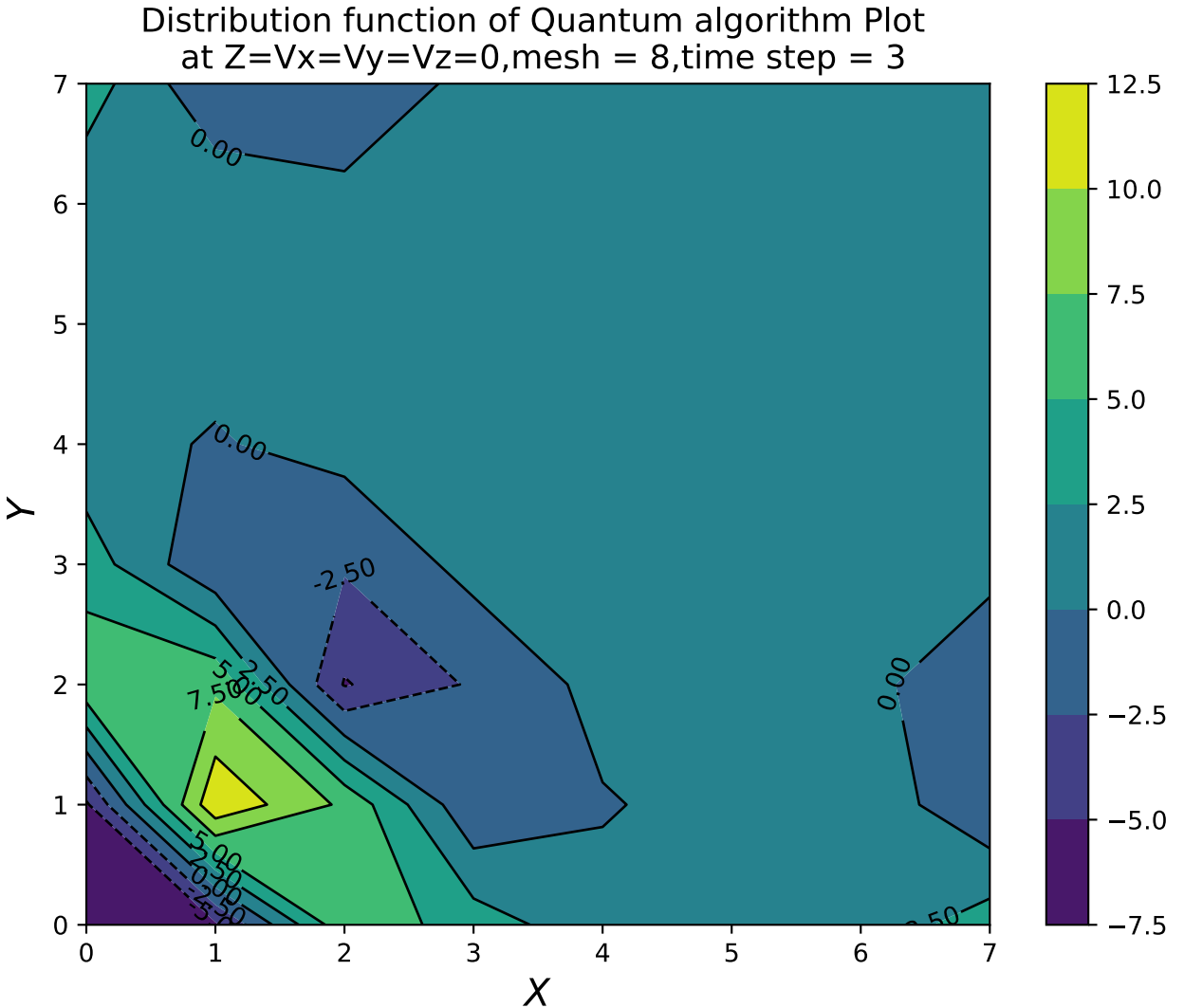

Distribution function of Quantum algorithm Plot
at Z=Vx=Vy=Vz=0,mesh = 8,time step = 3
Y
X
0.00
-2.50
5.00
2.50
7.50
-5.00
-7.50
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5

**Figure4-b.**

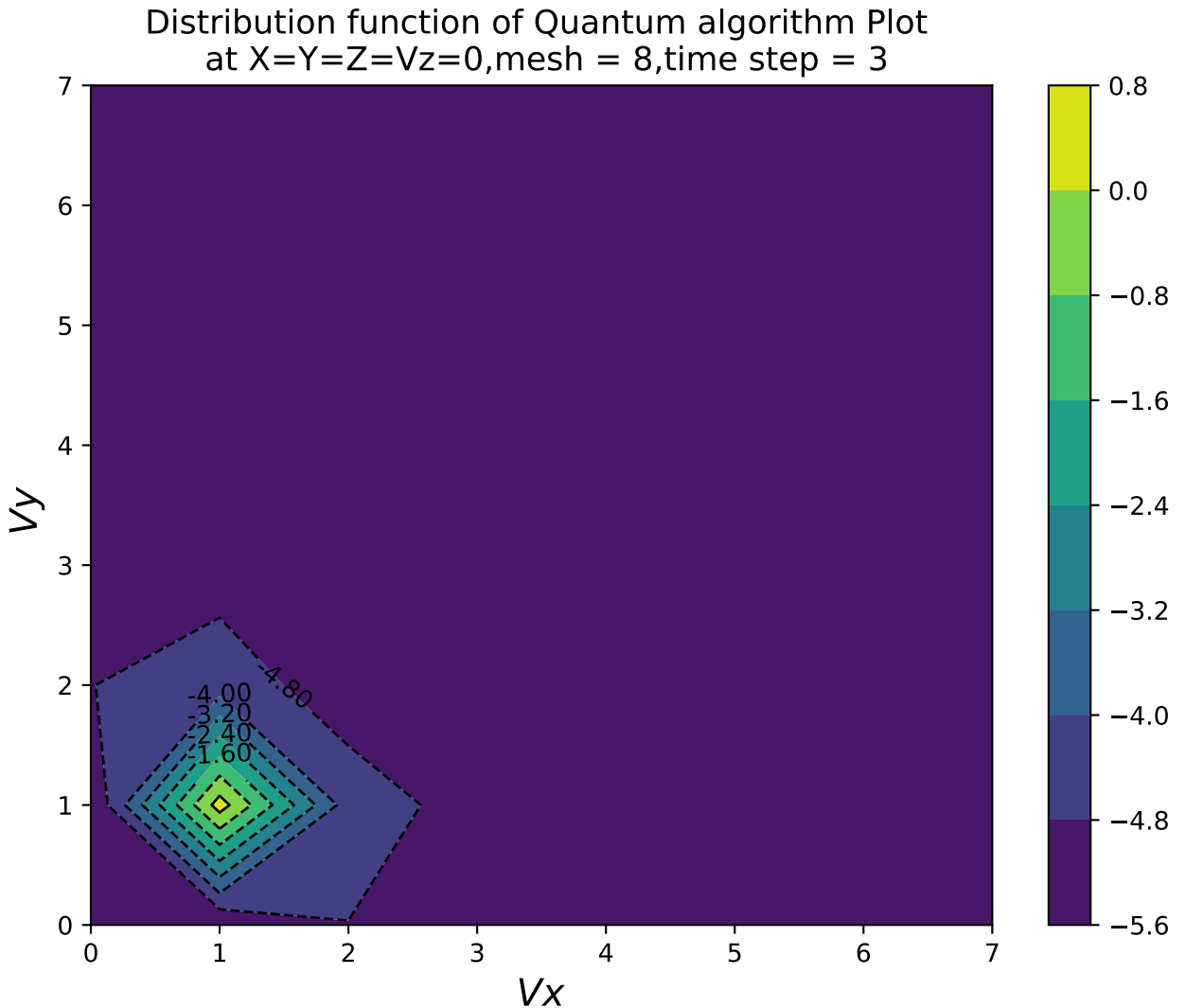

Distribution function of Quantum algorithm Plot
at X=Y=Z=Vz=0,mesh = 8,time step = 3
Vy
Vx
-4.00
-3.20
-2.40
-1.60
-4.80
0
1
2
3
4
5
6
7
0.8
0.0
−0.8
−1.6
−2.4
−3.2
−4.0
−4.8
−5.6

**Figure5-a.**

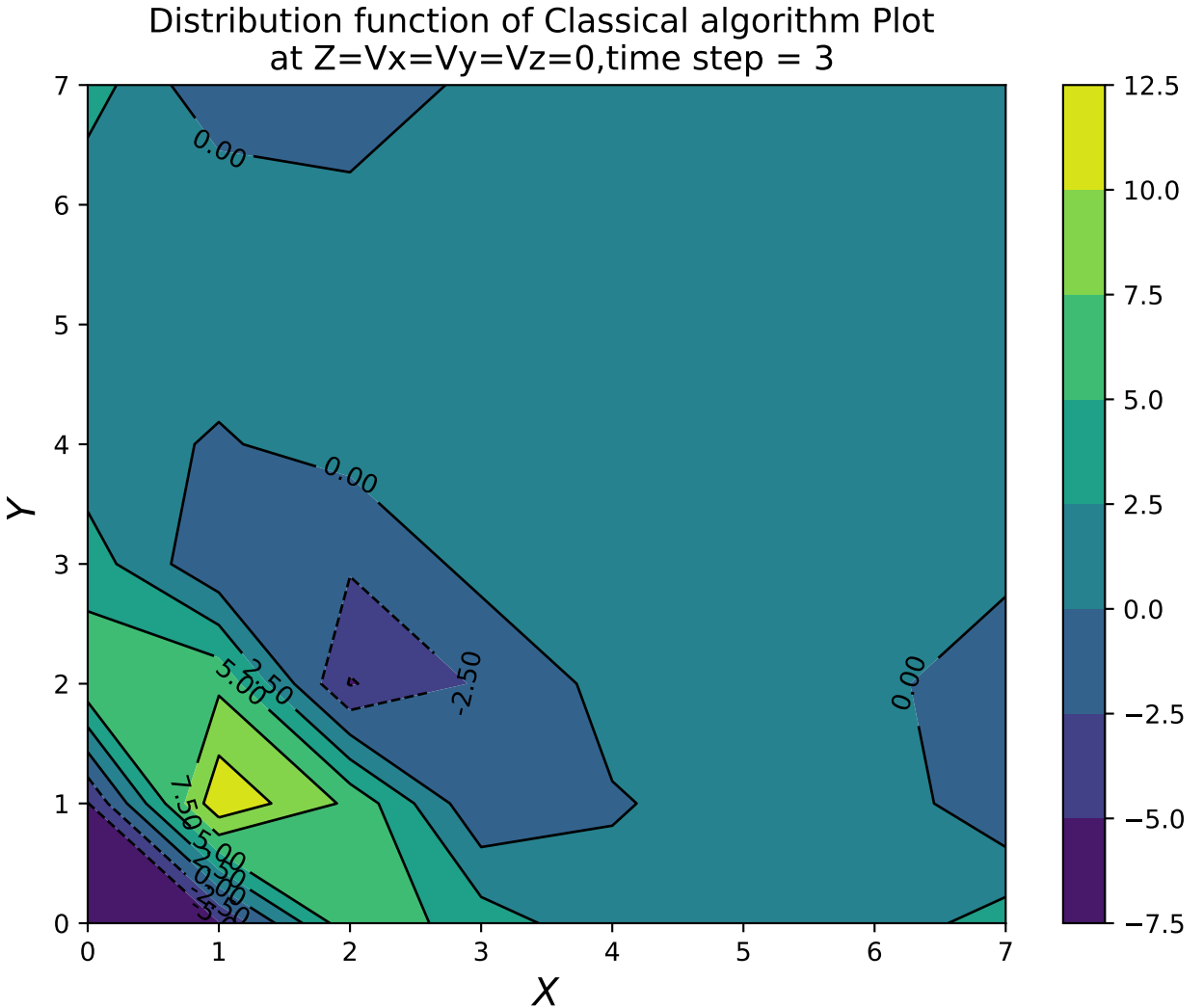
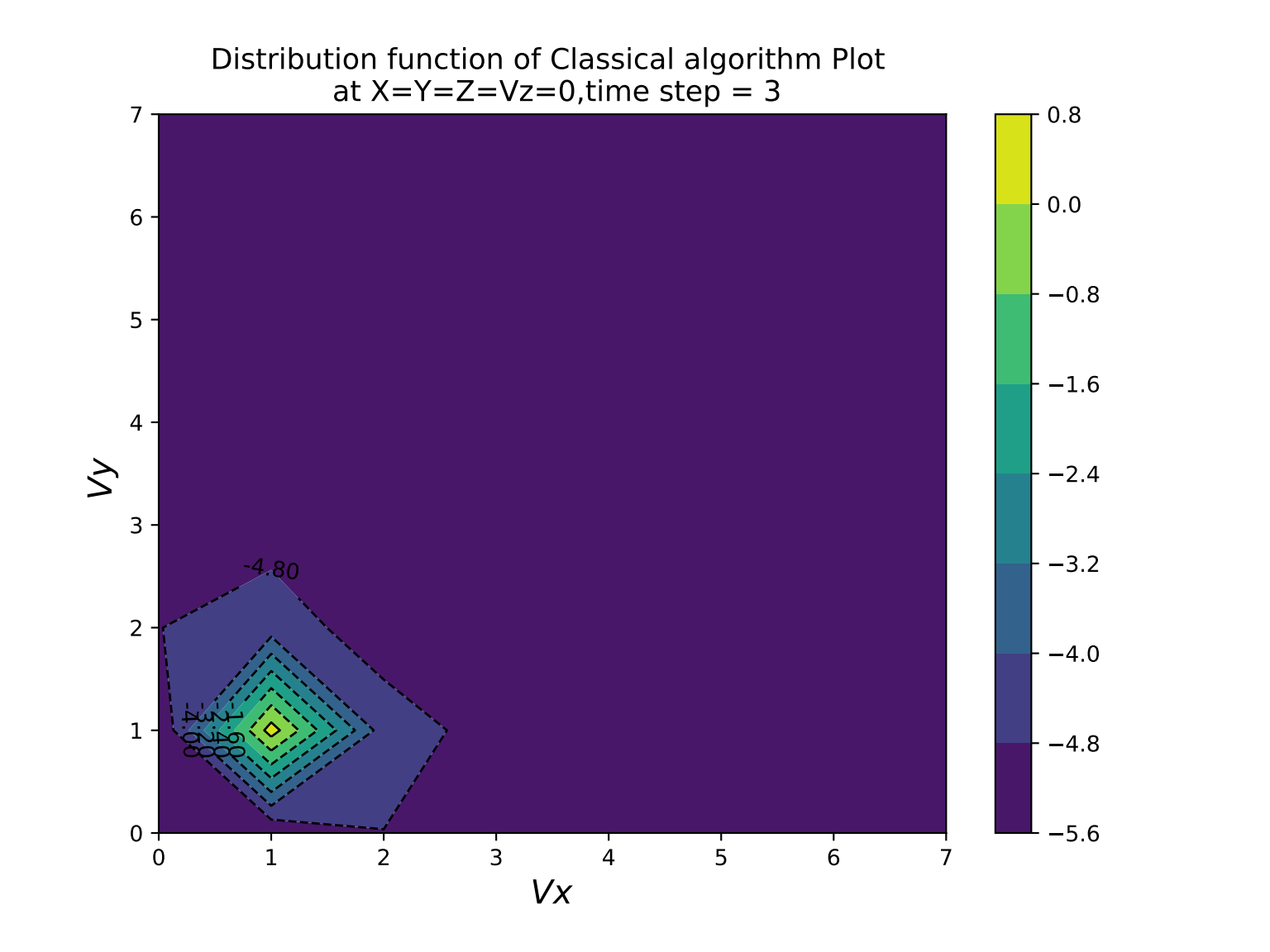

Distribution function of Classical algorithm Plot
at Z=Vx=Vy=Vz=0,time step = 3
X
Y
0.00
0.00
0.00
2.50
5.00
-2.50
7.50
5.00
2.50
0.00
-2.50
-5.00
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5

**Figure5-b.**


Distribution function of Classical algorithm Plot
at X=Y=Z=Vz=0,time step = 3
Vy
Vx
-4.80
0.8
0.0
−0.8
−1.6
−2.4
−3.2
−4.0
−4.8
−5.6