quantum reports

MDPI

Article

Improved Quantum Algorithm for Solving Nonlinear Burgers' Equation

Xiaoqi Liu [1], Ruiqi Zhang [1], Cunqi Jia [2], Jing Wang [1], Lei Zhang [3,4] and Ming Li [1,*]

[1] College of Science, China University of Petroleum, Qingdao 266580, China; 20060001@upc.edu.cn (J.W.)
[2] Physical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia
[3] National Key Laboratory of Deep Oil and Gas, China University of Petroleum, Qingdao 266580, China; zhlei84@163.com
[4] School of Petroleum Engineering, China University of Petroleum, Qingdao 266580, China
* Correspondence: liming@upc.edu.cn

Abstract

Quantum computing has been proven to have greater computational power than classical computing for solving certain problems. In particular, significant progress in quantum computing has also offered a novel approach to solving partial differential equations. We propose an improved quantum algorithm to solve the nonlinear Burgers' equation. Built on the principle of qubit reuse, the algorithm effectively reduces the qubit resource overhead required for multi-step evolution, and further optimizes the number of qubits to be measured to obtain the target quantum state such that it no longer grows with increasing system dimension. Additionally, the algorithm reduces the total number of ancilla qubits required and correspondingly lowers the number of control qubits needed for multi-controlled quantum gates. The numerical simulations of the inviscid Burgers' and viscous Burgers' equation have confirmed the feasibility of the proposed algorithm.

Keywords: quantum algorithm design; linear combination of unitaries method; Burgers' equation

1. Introduction

Quantum computing has advanced considerably and holds promise for alleviating specific bottlenecks in scientific computing. Currently, a variety of efficient quantum algorithms have emerged for specific computational problems, such as Shor's algorithm for factoring large numbers [1], Grover's algorithm for unstructured search [2], quantum algorithms for solving linear systems of equations [3–6], and various quantum machine learning algorithms [7–9]. In addition, the notable advancements in quantum computing have brought about a new perspective for tackling partial differential equations (PDEs). PDEs form an essential bridge between mathematics, natural sciences and engineering with a broad spectrum of applications. Particularly in the field of fluid mechanics, they act as a core mathematical tool for accurately characterizing the governing laws of fluid motion. Computational fluid dynamics (CFD) is the discipline that studies flow problems by solving these equations numerically [10]. However, as problem dimensionality increases and the scale of discretization grows, solving PDEs on classical computers becomes extremely challenging.

Quantum algorithms have been discussed for addressing the issue of limited computing resources in CFD. Currently, most quantum methods for solving linear PDEs are

Check for updates

Academic Editor: Vladimir M. Stojanović

Received: 9 October 2026
Revised: 22 November 2026
Accepted: 26 November 2026
Published: 28 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

based on quantum linear equation solvers [3–6], which aim to provide potential quantum acceleration schemes for such problems. Among these, variational quantum linear solvers are hybrid quantum-classical algorithms specifically designed to solve systems of linear equations on near-term quantum hardware [11–14]. Many quantum algorithms have been developed to solve linear systems in CFD [15–19]. Ye et al. introduced a CFD framework that integrates quantum and classical computing by transforming nonlinear fluid dynamics problems into linear systems [20]. Furthermore, in recent years, quantum algorithms based on lattice Boltzmann methods have also been proposed [21–23]. Such algorithms typically model the collision operator through block encoding techniques or the linear combination of unitaries (LCU) method and employ quantum walks to carry out the corresponding streaming steps, thereby efficiently describing and solving fluid dynamics processes within the framework of quantum computing. Truncated Taylor [24], linear combination of Hamiltonian simulation method [25], Schrödingerization [26], and time-marching strategy [27–29] as well as other innovative methods have been developed recently.

Nonlinear problems remain one of the most challenging frontier issues in current scientific research. Due to the significant increase in complexity and difficulty of handling caused by the presence of nonlinear terms, the research on quantum algorithms related to nonlinear problems is still relatively limited compared to linear problems. Currently, designing variational quantum algorithms to solve relevant nonlinear PDEs represents a crucial approach [30,31]. Carleman linearization introduces higher-order power terms of variables into a finite-dimensional nonlinear differential equation system, transforming it into an infinite-dimensional linear system with infinitely many coupled linear ordinary differential equations, and truncates this infinite system within a preset allowable error bound to obtain a finite system of linear ordinary differential equations, making it an important method in this field [32,33]. As mentioned in [34], the aforementioned algorithms do not mention how to directly handle the nonlinear terms in the nonlinear PDEs of fluid dynamics. To address this issue, the authors in [34] provided a detailed summary of nonlinear problems and proposed a novel pure quantum algorithm that directly calculates the nonlinear term by employing multiple copies of the state vector. The algorithm requires an increasing number of qubits with each iteration, making it crucial to explore ways to further reduce the number of qubits for conserving quantum resources. Here, we propose an improved quantum algorithm for nonlinear Burgers' equation that retains the advantages of the original algorithm while reducing the number of ancilla qubits and significantly relaxes the conditions for obtaining the target quantum state.

The remainder of this paper is organized as follows: In Section 2, we review the mathematical framework for the nonlinear Burgers' equation. In Section 3, the LCU method and the Hadamard product are introduced. In Section 4, quantum algorithm for solving nonlinear Burgers' equation is introduced and numerical simulations are conducted. The complexity of the quantum algorithm is introduced in Section 5. Section 6 concludes the paper with final remarks.

2. Mathematical Framework for Nonlinear Burgers' Equation

In this section, we mainly introduce nonlinear Burgers' equation and derive the discretized control equations in space and time using the finite difference method (FDM).

The Burgers' equation describes the interplay between convection and diffusion in fluid flow problems and can be used to model physical phenomena governed by convection–diffusion processes in engineering applications [35,36]. The general form contains a nonlinear convective term and a viscous diffusion term controlled by the viscosity coefficient, leading to viscous and inviscid Burgers' equations defined by whether the diffusion term is retained. Both forms can be expressed in differential form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}, \tag{2}$$

where u, x, t and μ represent the velocity vector, spatial coordinate, time and the viscosity, respectively.

Similar to [34], the Burgers' equation is discretized using the FDM. Therefore, the inviscid Burgers' equation and viscous Burgers' equation can be transformed into a set of algebraic equations as follows:

$$u_i^{t+1} = u_i^t - \Delta t u_i^t g_i^t, \tag{3}$$

$$u_i^{t+1} = u_i^t - \Delta t u_i^t g_i^t + \mu \Delta t h_i^t, \tag{4}$$

where $g_i^t = \frac{u_i^t - u_{i-1}^t}{\Delta x}$ and $h_i^t = \frac{u_{i-1}^t - 2u_i^t + u_{i+1}^t}{\Delta x^2}$ are the first and second derivatives of u at node i at time t. According to the discretized control equations, the calculation of the first and second derivatives can be expressed in the following form:

$$\begin{aligned}\begin{bmatrix} g_0^t \\ g_1^t \\ g_2^t \\ \vdots \\ g_{N-1}^t \end{bmatrix} &= \frac{1}{\Delta x}\begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ -1 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_0^t \\ u_1^t \\ u_2^t \\ \vdots \\ u_{N-1}^t \end{bmatrix} \\ &= \frac{1}{\Delta x} \cdot D \cdot \begin{bmatrix} u_0^t \\ u_1^t \\ u_2^t \\ \vdots \\ u_{N-1}^t \end{bmatrix},\end{aligned} \tag{5}$$

$$\begin{aligned}\begin{bmatrix} h_0^t \\ h_1^t \\ h_2^t \\ \vdots \\ h_{N-1}^t \end{bmatrix} &= \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & \cdots & 0 & 1 \\ 1 & -2 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & \cdots & 1 & -2 \end{bmatrix} \cdot \begin{bmatrix} u_0^t \\ u_1^t \\ u_2^t \\ \vdots \\ u_{N-1}^t \end{bmatrix} \\ &= \frac{1}{\Delta x^2} \cdot D_2 \cdot \begin{bmatrix} u_0^t \\ u_1^t \\ u_2^t \\ \vdots \\ u_{N-1}^t \end{bmatrix}.\end{aligned} \tag{6}$$

3. Preliminaries

In order to build upon previous research and ensure the logical coherence and consistency of this study, this section provides a brief introduction to the LCU method and quantum Hadamard product.

3.1. Linear Combination of Unitaries Method

In this subsection, we introduce the LCU method [37,38] for computing operators D and D_2. This method can precisely decompose the target nonunitary operator into a linear

combination of weighted unitary operators, and the quantum algorithm in this work is implemented based on this property. For nonunitary operations that can be decomposed into linear combinations of unitary operations, ancillary qubits can be introduced to realize their equivalent implementation probabilistically. Suppose the operator A acts on an n-qubit target quantum state $|\varphi\rangle$, and it satisfies

$$A = \sum_{j=0}^{M-1} \alpha_j U_j, \tag{7}$$

where $\alpha_j \geq 0$ and U_j are unitary operators, $j = 0, \cdots, M-1$. Therefore, we have

$$A|\varphi\rangle = \sum_{j=0}^{M-1} \alpha_j U_j |\varphi\rangle. \tag{8}$$

This process requires two registers: register R is an n-qubit register used to store the quantum state $|\varphi\rangle$, and register anc serves as an ancilla register with n_{anc} qubits ($n_{anc} = \lceil \log M \rceil$). The initial state of the system formed by the two registers is

$$|0\rangle_{anc}^{\otimes n_{anc}} \otimes |\varphi\rangle_R, \tag{9}$$

where $|0\rangle_{anc}^{\otimes n_{anc}}$ denotes the state of the ancilla qubit labeled by the subscript "anc", with $|0\rangle_{anc_j}$, $j = 0, \cdots, n_{anc}$, denoting the state of a single qubit.

The detailed procedure for the LCU method is presented below. Firstly, applying the preparation operator U_{PREP} to the ancilla qubits $|0\rangle_{anc}^{\otimes n_{anc}}$, the resulting state is

$$\begin{aligned} &U_{PREP} \otimes I_N |0\rangle_{anc}^{\otimes n_{anc}} \otimes |\varphi\rangle_R \\ =&\frac{1}{\sqrt{\sum_{j=0}^{M-1} \alpha_j}} \sum_{j=0}^{M-1} \sqrt{\alpha_j} |j\rangle_{anc} \otimes |\varphi\rangle_R, \end{aligned} \tag{10}$$

where I_N is the $N \times N$ identity matrix, $N = 2^n$.

Controlled by the ancilla qubit $|j\rangle_{anc}$, U_j is conditionally applied to $|\varphi\rangle_R$, thereby implementing the selection operator U_{SEL}, then we have

$$U_{SEL} |j\rangle_{anc} \otimes |\varphi\rangle_R = |j\rangle_{anc} \otimes U_j |\varphi\rangle_R. \tag{11}$$

Finally, the operator $U_{PREP}^\dagger$ is applied to register anc, and we have

$$\begin{aligned} &(U_{PREP}^\dagger \otimes I_N) U_{SEL} (U_{PREP} \otimes I_N) |0\rangle_{anc}^{\otimes n_{anc}} |\varphi\rangle_R \\ =&\frac{1}{\sum_{j=0}^{M-1} \alpha_j} \sum_{j=0}^{M-1} \alpha_j |0\rangle_{anc}^{\otimes n_{anc}} U_j |\varphi\rangle_R + |\perp\rangle |\cdots\rangle_R. \end{aligned} \tag{12}$$

If $|0\rangle_{anc}^{\otimes n_{anc}}$ is measured, the state will be projected onto $\frac{1}{\sum_{j=0}^{M-1} \alpha_j} A|\varphi\rangle$, thereby enabling the nonunitary operation A to be applied to $|\varphi\rangle$.

Taking $M = 3$ as an example, that is, $A = \sum_{j=0}^{3} \alpha_j U_j$ and $U_{SEL} = \tilde{U}_0 \tilde{U}_1 \tilde{U}_2 \tilde{U}_3$, where

$$U_{SEL} = |00\rangle\langle 00| \otimes U_0 + |01\rangle\langle 01| \otimes U_1 + |10\rangle\langle 10| \otimes U_2 + |11\rangle\langle 11| \otimes U_3, \tag{13}$$

$$\tilde{U}_0 = |00\rangle\langle 00| \otimes U_0 + (I - |00\rangle\langle 00|) \otimes I, \tag{14}$$

$$\tilde{U}_1 = |01\rangle\langle 01| \otimes U_1 + (I - |01\rangle\langle 01|) \otimes I, \tag{15}$$

$$\tilde{U}_2 = |10\rangle\langle 10| \otimes U_2 + (I - |10\rangle\langle 10|) \otimes I, \tag{16}$$

$$\tilde{U}_3 = |11\rangle\langle 11| \otimes U_3 + (I - |11\rangle\langle 11|) \otimes I. \tag{17}$$

As shown in Figure 1, the initial state of the circuit is

$$|00\rangle_{anc}|\varphi\rangle_R, \tag{18}$$

and

$$U_{PREP} = \begin{bmatrix} \frac{\sqrt{\alpha_0}}{\sqrt{\sum_{j=0}^{3}\alpha_j}} & * & * & * \\ \frac{\sqrt{\alpha_1}}{\sqrt{\sum_{j=0}^{3}\alpha_j}} & * & * & * \\ \frac{\sqrt{\alpha_2}}{\sqrt{\sum_{j=0}^{3}\alpha_j}} & * & * & * \\ \frac{\sqrt{\alpha_3}}{\sqrt{\sum_{j=0}^{3}\alpha_j}} & * & * & * \end{bmatrix}. \tag{19}$$

According to the LCU method, we have

$$\frac{1}{\sum_{j=0}^{3}\alpha_j}|00\rangle_{anc}(\alpha_0 U_0 + \alpha_1 U_1 + \alpha_2 U_2 + \alpha_3 U_3)|\varphi\rangle_R + |\perp\rangle|\cdots\rangle_R, \tag{20}$$

where $|\perp\rangle$ satisfies $\langle 00|\perp\rangle = 0$.

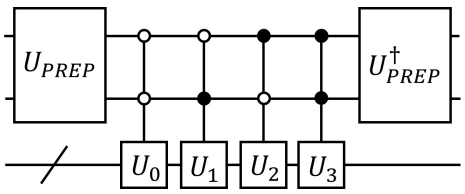


Figure 1. The quantum circuit of the LCU method when $M = 3$.

Since D and D_2 are circulant matrices, they are easily implemented in [34] using block encoding technique [39]. Block encoding is a technique for embedding a properly scaled nonunitary matrix A into a unitary matrix U_A, i.e.,

$$U_A = \begin{bmatrix} \frac{1}{\alpha}A & * \\ * & * \end{bmatrix}, \tag{21}$$

$$\frac{1}{\alpha}A = (\langle 0| \otimes I)U_A(I \otimes |0\rangle), \tag{22}$$

where $\alpha \geq \|A\|_2$ is a subnormalization factor and $\|\cdot\|_2$ is the spectral norm. The LCU method provides a standard block encoding form [40], namely

$$A = \alpha\langle 0| \otimes I(U_{PREP}^{\dagger} \otimes I_N)U_{SEL}(U_{PREP} \otimes I_N)I \otimes |0\rangle, \tag{23}$$

where $\alpha = \sum_{j=0}^{M-1}\alpha_j$.

In this work, we employ the LCU method to implement D and D_2. Taking D_2 as an example, one has

$$\begin{aligned} D_2 =& \begin{bmatrix} -2 & 1 & \cdots & 0 & 1 \\ 1 & -2 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & \cdots & 1 & -2 \end{bmatrix} \\ =& R - 2I + L, \end{aligned} \tag{24}$$

where

$$R = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{25}$$

which can be achieved by applying multi-controlled X gates and X gates as shown in Figure 2a,b.

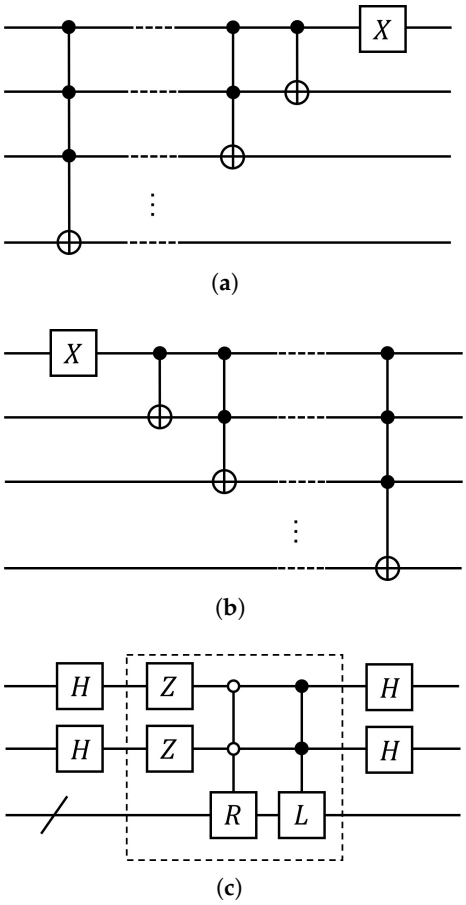


Figure 2. (**a**) R gate. (**b**) L gate. (**c**) Schematic of the quantum circuit for the LCU method on D_2.

According to the LCU method, we obtain:

$$\begin{aligned} & U^{D_2}{}^{\dagger}_{PREP} U^{D_2}_{SEL} U^{D_2}_{PREP} |0\rangle^{\otimes 2} |\varphi\rangle \\ =& (H^{\otimes 2} \otimes I) U^{D_2}_{SEL} (H^{\otimes 2} \otimes I) |0\rangle |\varphi\rangle \\ =& \frac{1}{4} (|00\rangle (R - 2I + L) |\varphi\rangle + |\perp\rangle |\varphi\rangle) \\ =& \frac{1}{4} (|00\rangle D_2 |\varphi\rangle + |\perp\rangle |\varphi\rangle), \end{aligned} \tag{26}$$

where $U^{D_2}_{SEL} = |00\rangle\langle 00| \otimes L + |11\rangle\langle 11| \otimes R - |01\rangle\langle 01| \otimes I - |10\rangle\langle 10| \otimes I$. The quantum circuit diagram is shown in Figure 2c.

In contrast to [34], which encodes the nonzero elements of matrix D_2 in register using controlled R_y, our method avoids this operation while also reducing the number of qubits required. Additionally, $(H^{\otimes 2} \otimes I) U^{D_2}_{SEL} (H^{\otimes 2} \otimes I)$ can be considered as a block encoding

strategy for D_2 [41]. Furthermore, the factors $\frac{1}{\Delta x}$ and $\frac{1}{\Delta x^2}$ involved in operators g_i^t and h_i^t can be introduced through the gate $R_y(\theta)$, where

$$R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}. \tag{27}$$

3.2. The Hadamard Product

The calculation of the nonlinear term $u\frac{\partial u}{\partial x}$ is a key challenge in the design of the quantum algorithm for the Burgers' equation. In this subsection, we will introduce how to implement the nonlinear term through quantum circuit.

The quantum Hadamard product technique enables element-wise multiplication of two vectors [34,42]. Define operation of element-wise multiplication as

$$u \odot g = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} \odot \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{N-1} \end{bmatrix} = \begin{bmatrix} u_0 g_0 \\ u_1 g_1 \\ u_2 g_2 \\ \vdots \\ u_{N-1} g_{N-1} \end{bmatrix}, \tag{28}$$

where u (g) is an N-dimensional column vector, where $N = 2^n$. Obviously, $u \odot g$ is contained within the tensor product $u \otimes g$, where

$$u \otimes g = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} \otimes \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{N-1} \end{bmatrix} = \begin{bmatrix} u_0 g_0 \\ u_0 g_1 \\ \vdots \\ u_0 g_{N-1} \\ \vdots \\ u_{N-1} g_0 \\ u_{N-1} g_1 \\ \vdots \\ u_{N-1} g_{N-1} \end{bmatrix}. \tag{29}$$

Although $u \otimes g$ contains all entries of $u \odot g$, these entries are interleaved with undesired entries. Our goal is to obtain a vector whose upper block is exactly $u \odot g$. To this end, we construct a permutation matrix to rearrange the basis so that $u_i g_i$ is arranged contiguously, namely

$$\begin{bmatrix} u_0 g_0 \\ u_0 g_1 \\ \vdots \\ u_0 g_{N-1} \\ \vdots \\ u_{N-1} g_0 \\ u_{N-1} g_1 \\ \vdots \\ u_{N-1} g_{N-1} \end{bmatrix} \xrightarrow{rearrangement} \begin{bmatrix} u_0 g_0 \\ u_1 g_1 \\ \vdots \\ u_{N-2} g_{N-2} \\ u_{N-1} g_{N-1} \\ \hline \vdots \\ \vdots \\ \vdots \end{bmatrix} = \left[\begin{array}{c} u \odot g \\ \hline \vdots \end{array}\right]. \tag{30}$$

We can achieve Equation (30) through n parallel CNOT gates. Specifically, each control qubit j in register R_1 is connected to its corresponding target qubit $j + n$ in register R_2. As

shown in Figure 3, the procedure of obtaining $u \odot g$ is as follows. The amplitudes of the vectors u and g are encoded as $|u\rangle$ and $|g\rangle$, that is

$$|u\rangle = \frac{1}{\|u\|} \sum_{i=0}^{N-1} u_i |i\rangle, \tag{31}$$

$$|g\rangle = \frac{1}{\|g\|} \sum_{j=0}^{N-1} g_j |j\rangle, \tag{32}$$

where $\|u\| = \sqrt{\sum_{j=0}^{N-1} (u_i)^2}$ and $\|g\| = \sqrt{\sum_{j=0}^{N-1} (g_i)^2}$.

Initialize quantum states $|u\rangle$ and $|g\rangle$ in registers R_1 and R_2, respectively, so that

$$\begin{aligned} |u\rangle_{R_1} \otimes |g\rangle_{R_2} =& \left(\frac{1}{\|u\|} \sum_{i=0}^{N-1} u_i |i\rangle\right) \otimes \left(\frac{1}{\|g\|} \sum_{j=0}^{N-1} g_j |j\rangle\right) \\ =& \frac{1}{\|u\|\|g\|} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} u_i g_j |i\rangle_{R_1} |j\rangle_{R_2}. \end{aligned} \tag{33}$$

The CNOT gates are applied to the corresponding qubits of register R_1 and register R_2. Then we have

$$\begin{aligned} & \frac{1}{\|u\|\|g\|} \sum_{i=0}^{N-1} u_i g_i |i\rangle_{R_1} |0\rangle_{R_2} + |\cdots\rangle_{R_1} |\bar{0}\rangle_{R_2} \\ =& \frac{1}{\|u\|\|g\|} |u \odot g\rangle_{R_1} |0\rangle_{R_2} + |\cdots\rangle_{R_1} |\bar{0}\rangle_{R_2}. \end{aligned} \tag{34}$$

We note that $|u \odot g\rangle := \sum_{i=0}^{N-1} u_i g_i |i\rangle$ is not a quantum state. In order to facilitate the description of the algorithm in subsequent sections, we will use this symbol as an expression.

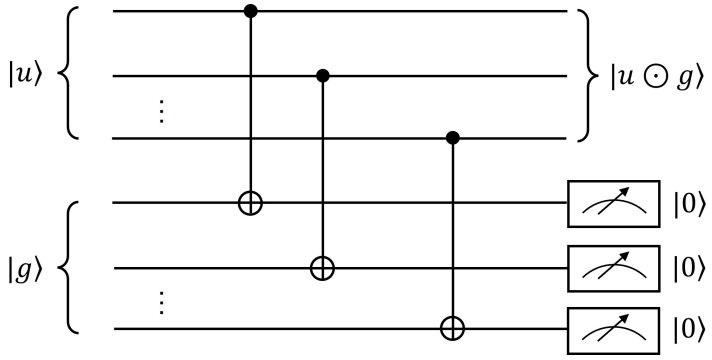


Figure 3. Quantum circuit for Hadamard product.

4. Quantum Algorithms for Solving the Burgers' Equation

Sections 4.1 and 4.2 describe the quantum algorithms for solving inviscid Burgers' equation and viscous Burgers' equation, respectively, and present the corresponding quantum circuits. Section 4.3 presents the numerical simulation results for algorithms.

The quantum algorithm for solving the Burgers' equation requires three registers: the main register R_2 is used to encode the velocity, register R_1 encodes a copy of the velocity, and register *anc* is an ancilla register.

In practical applications, amplitude encoding has become a widely used encoding method [43]. Without loss of generality, let $u^t = (u_0^t, u_1^t, \cdots, u_{N-1}^t)^T$ be a vector of $N = 2^n$

numbers, where n represents the number of qubits. In fact, when $\log N$ is not an integer, fill the remaining space with zeros. The corresponding quantum state is

$$\left|u^t\right\rangle = \frac{1}{\left\|u^t\right\|}\sum_{i=0}^{N-1} u_i^t|i\rangle, \tag{35}$$

where $\left\|u^t\right\| = \sqrt{\sum_{i=0}^{N-1}(u_i^t)^2}$. Therefore, the initial state is

$$|0\rangle_{anc}^{\otimes n_{anc}}|0\rangle_{R_1}^{\otimes n}|0\rangle_{R_2}^{\otimes n}, \tag{36}$$

where n_{anc} represents the number of ancilla qubits required by the LCU method.

4.1. Quantum Circuit for Inviscid Burgers' Equation

In this subsection, we present the algorithm for the inviscid Burger's equation. The equation is discretized using the FDM, and its corresponding discrete formulation is presented in Equation (3). As mentioned earlier, due to the presence of the $u\frac{\partial u}{\partial x}$ term, two copies of the velocity are required. Two ancilla qubits for the LCU method need to be introduced. Therefore, the initial quantum state is

$$|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|u^t\rangle_{R_2}. \tag{37}$$

The quantum circuit structure for the algorithm is illustrated in Figure 4. Below we describe the five core components that make up the single-step evolution process:

Step 1: The initial velocity state is prepared in the register R_2, and ancilla qubit $|\cdot\rangle_{anc_0}$ generates a weighted superposition state via $R_y(\alpha_1)$ rotation, where $\alpha_1 = 2\arccos\frac{\frac{1}{\|u^0\|}}{\sqrt{\frac{1}{\|u^0\|^2}+4(\frac{\Delta t}{\Delta x})^2}}$. The quantum state is

$$\cos\frac{\alpha_1}{2}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2} + \sin\frac{\alpha_1}{2}|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2}. \tag{38}$$

Step 2: The state $D\left|u^0\right\rangle$ is generated with ancilla qubit $|\cdot\rangle_{anc_0}$ as the control condition. After the controlled differential operation, the state becomes

$$\begin{aligned}&\cos\frac{\alpha_1}{2}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2}\\&+\frac{1}{2}\sin\frac{\alpha_1}{2}|1\rangle_{anc_0}[-|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}D\left|u^0\right\rangle_{R_2} + |0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}(I+R)\left|u^0\right\rangle_{R_2}].\end{aligned} \tag{39}$$

The success probability of post selection is directly related to the choice of LCU method, and different LCU strategies yield distinct impacts on this probability. Unlike the method in [34], this work adopts a distinct LCU strategy to implement nonunitary operator, moderately amplifying the effective amplitude per step.

Step 3: The quantum Hadamard product is implemented via CNOT gates. The element-wise entanglement of the two registers is achieved through controlled CNOT gates, which converts the velocity state and derivative state into an element-wise product state. Following the Hadamard product operation, the state takes the form

$$\begin{aligned}&\cos\frac{\alpha_1}{2}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2} + \frac{1}{2}\sin\frac{\alpha_1}{2}|1\rangle_{anc_0}[-|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\odot Du^0\right\rangle_{R_2}\\&+|1\rangle_{anc_1}|\bar{0}\rangle_{R_1}^{\otimes n}\left|\overline{u^0\odot Du^0}\right\rangle_{R_2} + |0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}(I+R)|u_0\rangle_{R_2}].\end{aligned} \tag{40}$$

To facilitate the subsequent execution of the algorithm, CNOT gate is applied, and the corresponding quantum state reads

$$\begin{aligned}&\cos\frac{\alpha_1}{2}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2}+\frac{1}{2}\sin\frac{\alpha_1}{2}|1\rangle_{anc_0}[-|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\left|u^0\odot Du^0\right\rangle_{R_2}\\&+|0\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}\left|\overline{u^0\odot Du^0}\right\rangle_{R_2}+|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}(I+R)|u_0\rangle_{R_2}].\end{aligned}\tag{41}$$

Step 4: Hadamard gate is applied to the LCU ancilla qubit to form quantum state superposition. The state is expressed as

$$\begin{aligned}&\frac{1}{\sqrt{2}}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}(\cos\frac{\alpha_1}{2}\left|u^0\right\rangle_{R_2}-\frac{1}{2}\sin\frac{\alpha_1}{2}\left|u^0\odot Du^0\right\rangle_{R_2})\\&+|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}+|1\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\&+|0\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}+|0\rangle_{anc_0}|0\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\&+|1\rangle_{anc_0}|0\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2},\end{aligned}\tag{42}$$

where $|\cdots\rangle_{R_2}$ is irrelevant and is omitted from further derivation.

Step 5: To relax the conditions for obtaining the target solution state, the multi-controlled X is applied. The quantum state is

$$\begin{aligned}&\frac{1}{\sqrt{2}}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}(\cos\frac{\alpha_1}{2}\left|u^0\right\rangle_{R_2}-\frac{1}{2}\sin\frac{\alpha_1}{2}\left|u^0\odot Du^0\right\rangle_{R_2})\\&+|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}+|1\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\&+|0\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}+|0\rangle_{anc_0}|1\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\&+|1\rangle_{anc_0}|1\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\=&|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}\frac{1}{c_1}\left\|u^1\right\|\left|u^1\right\rangle_{R_2}+|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\&+|1\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}+|0\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\&+|0\rangle_{anc_0}|1\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}+|1\rangle_{anc_0}|1\rangle_{anc_1}\left|\bar{0}\right\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2},\end{aligned}\tag{43}$$

where $\frac{1}{c_1}=\frac{1}{\sqrt{2}\left\|u^0\right\|^2\sqrt{\frac{1}{\left\|u^0\right\|^2}+4(\frac{\Delta t}{\Delta x})^2}}$.

Under the condition that all ancilla qubits have reached the zero state, the quantum state on the register can be obtained as $\left|u^{t+1}\right\rangle_{R_2}$. For the design of multi-step evolution circuits, the approach proposed by [34] can be employed. We define operator U_1 as the unitary operation implemented by the Hadamard product and the LCU method within a single time step. Accordingly, the recursive relation for multi-step evolution is given by $U_t=U_1U_{t-1}^{\otimes 2}$, where U_{t-1} denotes the evolution operator for step $t-1$ that generates multiple intermediate state copies required for subsequent steps, and U_t denotes the full evolution operator for step t. Incorporating the design principle of qubit reuse, the quantum state evolution over two time steps can be expressed as follows:

$$\cdots |\psi_t\rangle_{R_1} \cdots |\psi_t\rangle_{R_1} |\psi_t\rangle_{R_2} \tag{44}$$

$$\xrightarrow{U_1} \cdots |\psi_t\rangle_{R_1} \cdots \underbrace{|0\rangle_{R_1}^{\otimes n}}_{reuse} \underbrace{|\psi_{t+1}\rangle_{R_2}}_{target} \tag{45}$$

$$\xrightarrow{CU_t} \cdots |\psi_t\rangle_{R_1} \cdots |\psi_t\rangle_{R_1} \underbrace{|\psi_{t+1}\rangle_{R_2}}_{target} \tag{46}$$

$$\xrightarrow{U_1} \cdots \underbrace{|0\rangle_{R_1}^{\otimes n}}_{reuse} \cdots |\psi_{t+1}\rangle_{R_1} \underbrace{|\psi_{t+1}\rangle_{R_2}}_{target} \tag{47}$$

$$\xrightarrow{U_1} \cdots \underbrace{|0\rangle_{R_1}^{\otimes n}}_{reuse} \cdots \underbrace{|0\rangle_{R_1}^{\otimes n}}_{reuse} \underbrace{|\psi_{t+2}\rangle_{R_2}}_{target}. \tag{48}$$

This method significantly reduces the qubit resource overhead in multi-step simulations while maintaining continuous unitary evolution, and provides a feasible circuit implementation framework for quantum simulation of nonlinear fluid equation. It circumvents the process of converting the output of the previous step into classical data through measurement and then reinitializing the corresponding quantum state. It is noted that at the beginning of the circuit, we need to use the $R_y(\alpha_t)$ gate to achieve the superposition of quantum states. For each iteration, the rotation $R_y(\alpha_{t+1})$ depends solely on Δt, Δx and c_t from the previous step, where $c_0 = \|u^0\|$, and it can be calculated as follows:

$$\cos\frac{\alpha_{t+1}}{2} = \frac{\frac{1}{c_t}}{\sqrt{\frac{1}{{c_t}^2} + 4(\frac{\Delta t}{\Delta x})^2}}, \tag{49}$$

$$sin\frac{\alpha_{t+1}}{2} = \frac{2\frac{\Delta t}{\Delta x}}{\sqrt{\frac{1}{{c_t}^2} + 4(\frac{\Delta t}{\Delta x})^2}}. \tag{50}$$

The factor c_t at each iteration can be expressed in a general form as

$$c_{t+1} = \sqrt{2}{c_t}^2 \sqrt{\frac{1}{{c_t}^2} + 4(\frac{\Delta t}{\Delta x})^2}. \tag{51}$$

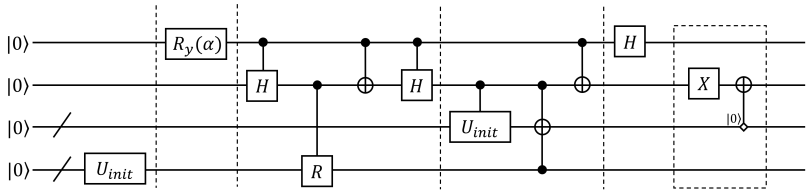


Figure 4. The circuit diagram for one iteration of the inviscid Burgers' equation.

The final state of the quantum algorithm is a quantum state encoding the solution to the problem. The final step is to extract relevant information from it. However, the readout problem in quantum algorithms is not an easy one. Classical shadow provides an efficient method for predicting many properties of a subsystem from a small number of measurements [44]. Furthermore, Williams et al. proposed a method for addressing the readout problem using quantum learning tools, which are employed to extract relevant features [45]. As nonunitary differential operators are realized by the LCU method with post-selection

for valid branch filtering, the success probability of the target quantum state decays with growing time steps, which can be mitigated using quantum amplitude amplification.

4.2. Quantum Circuit for Viscous Burgers' Equation

In this subsection, based on the quantum algorithm for inviscid Burgers' equation, we design the quantum algorithm for viscous Burgers' equation by merely adding one more ancilla qubit. Based on the FDM, the governing equation is discretized in both space and time. The resulting discrete scheme is presented in Equation (4). The viscous Burgers' equation differs from inviscid Burgers' equation only by including the second derivative of velocity. Accordingly, we introduce an additional ancilla qubit on the basis of quantum circuit for inviscid Burgers' equation to construct the superposition of quantum states. Here, the initial quantum state is

$$|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}\left|u^t\right\rangle_{R_2}. \tag{52}$$

Figure 5 shows the quantum circuit used for one iteration of viscous Burgers' equation, and the evolution of the quantum states depicted therein is described in detail below.

Step 1: The system register is prepared in the initial velocity state, and the LCU ancilla qubit is rotated by $R_y(\beta_1)$ to generate the superposition state, where $\beta_1 = 2\arccos(\frac{1}{\sqrt{1+16(\mu\frac{\Delta t}{\Delta x^2}a_1\frac{1}{\|u^0\|})^2}})$, and $a_1 = \frac{1}{\sqrt{2}\sqrt{\frac{1}{\|u^0\|^2}+4(\frac{\Delta t}{\Delta x})^2}}$. The resulting quantum state is

$$\cos\frac{\beta_1}{2}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2} + \sin\frac{\beta_1}{2}|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2}. \tag{53}$$

Step 2: With the first ancilla qubit as the control condition, operation of the inviscid Burgers'equation is applied to the $|0\rangle_{anc_0}$ branch. The resulting state reads

$$\begin{aligned}
&\cos\frac{\beta_1}{2}|0\rangle_{anc_0}[|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}a_1(\frac{1}{\|u^0\|}\left|u^0\right\rangle_{R_2} - \frac{\Delta t}{\Delta x}\left|u^0\odot Du^0\right\rangle_{R_2})\\
&+|1\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2} + |1\rangle_{anc_1}|1\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\
&+|0\rangle_{anc_1}|1\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2} + |0\rangle_{anc_1}|1\rangle_{anc_2}|\overline{0}\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\
&+|1\rangle_{anc_1}|1\rangle_{anc_2}|\overline{0}\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}] + \sin\frac{\beta_1}{2}|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}\left|u^0\right\rangle_{R_2}.
\end{aligned} \tag{54}$$

Step 3: With the first LCU ancilla qubit as the control condition, the second-order differential operator D_2 is applied to the $|1\rangle_{anc_0}$ branch. The resulting quantum state takes the form

$$\begin{aligned}
&\cos\frac{\beta_1}{2}|0\rangle_{anc_0}[|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}a_1(\frac{1}{\|u^0\|}\left|u^0\right\rangle_{R_2} - \frac{\Delta t}{\Delta x}\left|u^0\odot Du^0\right\rangle_{R_2})\\
&+|1\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2} + |1\rangle_{anc_1}|1\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\
&+|0\rangle_{anc_1}|1\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2} + |0\rangle_{anc_1}|1\rangle_{anc_2}|\overline{0}\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\
&+|1\rangle_{anc_1}|1\rangle_{anc_2}|\overline{0}\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}]\\
&+\sin\frac{\beta_1}{2}|1\rangle_{anc_0}[|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}\frac{1}{4}D_2\left|u^0\right\rangle_{R_2}\\
&+|0\rangle_{anc_1}|1\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2} + |1\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}\\
&+|1\rangle_{anc_1}|1\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}|\cdots\rangle_{R_2}].
\end{aligned} \tag{55}$$

Step 4: Hadamard gate is applied to the LCU ancilla qubit $|\cdot\rangle_{anc_0}$, such that the two terms form additive and subtractive superposition results in different computational basis subspaces of the ancilla register. The corresponding state vector is given by

$$\begin{aligned}
&\frac{1}{\sqrt{2}}|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n} \\
&\otimes[\cos\frac{\beta_1}{2}a_1(\frac{1}{\|u^0\|}\left|u^0\right\rangle_{R_2}-\frac{\Delta t}{\Delta x}\left|u^0\odot Du^0\right\rangle_{R_2})+\frac{1}{4}\sin\frac{\beta_1}{2}D_2\left|u^0\right\rangle_{R_2}] \\
&+|0\rangle_{anc_0}|0\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2}+|0\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
&+|0\rangle_{anc_0}|1\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2}+|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
&+|1\rangle_{anc_0}|0\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2}+|1\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
&+|1\rangle_{anc_0}|1\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
=&|0\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|0\rangle_{R_1}^{\otimes n}\frac{1}{c_1}\left\|u^1\right\|\left|u^1\right\rangle_{R_2}+|0\rangle_{anc_0}|0\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
&+|0\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{anc_2}|\cdots\rangle_{R_1R_2}+|0\rangle_{anc_0}|1\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
&+|1\rangle_{anc_0}|0\rangle_{anc_1}|0\rangle_{anc_2}|\cdots\rangle_{R_1R_2}+|1\rangle_{anc_0}|0\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2} \\
&+|1\rangle_{anc_0}|1\rangle_{anc_1}|0\rangle_{anc_2}|\cdots\rangle_{R_1R_2}+|1\rangle_{anc_0}|1\rangle_{anc_1}|1\rangle_{anc_2}|\cdots\rangle_{R_1R_2},
\end{aligned} \tag{56}$$

where $\frac{1}{c_1}=\frac{a_1}{\sqrt{2}\|u^0\|^2\sqrt{1+16(\mu\frac{\Delta t}{\Delta x^2}a_1\frac{1}{\|u^0\|})^2}}$.

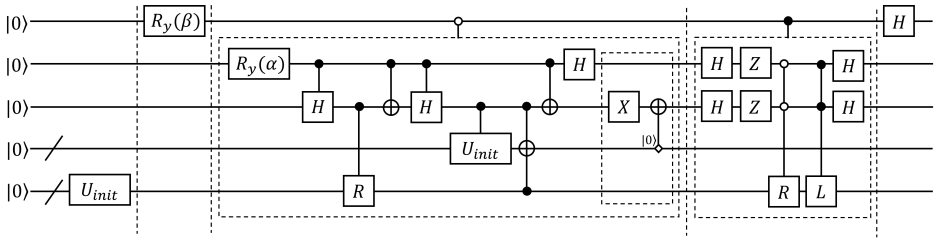


Figure 5. The circuit diagram for one iteration of the viscous Burgers' equation.

The difference from inviscid Burgers' equation lies in that viscous Burgers' equation includes the term $\mu\frac{\partial^2 u}{\partial x^2}$. For each iteration, the rotation $R_y(\beta_{t+1})$ depends solely on Δt, Δx and c_t from the previous step, where $c_0=\|u^0\|$, and it can be calculated as follows:

$$\cos\frac{\beta_{t+1}}{2}=\frac{1}{\sqrt{1+16(\mu\frac{\Delta t}{\Delta x^2}a_{t+1}\frac{1}{c_t})^2}}, \tag{57}$$

$$sin\frac{\beta_{t+1}}{2}=\frac{4\mu\frac{\Delta t}{\Delta x^2}a_{t+1}\frac{1}{c_t}}{\sqrt{1+16(\mu\frac{\Delta t}{\Delta x^2}a_{t+1}\frac{1}{c_t})^2}}, \tag{58}$$

$$a_{t+1}=\frac{1}{\sqrt{2}\sqrt{\frac{1}{{c_t}^2}+4(\frac{\Delta t}{\Delta x})^2}}. \tag{59}$$

The factor c_t at each iteration can be expressed in a general form as

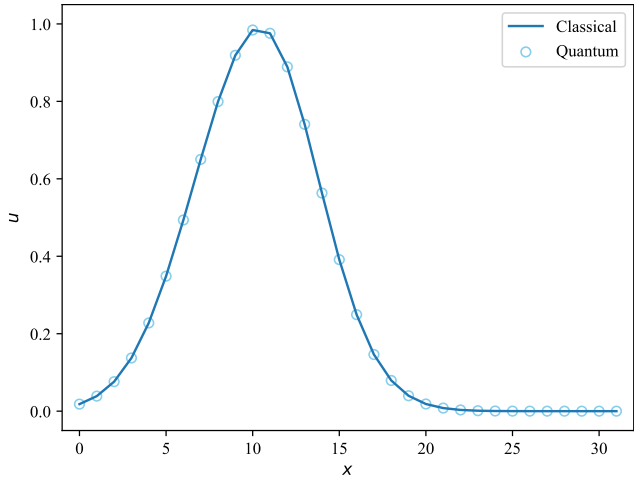
$$c_{t+1}=2{c_t}^2\sqrt{\frac{1}{{c_t}^2}+4(\frac{\Delta t}{\Delta x})^2+8(\mu\frac{\Delta t}{\Delta x^2}\frac{1}{c_t})^2}. \tag{60}$$

4.3. Numerical Simulation

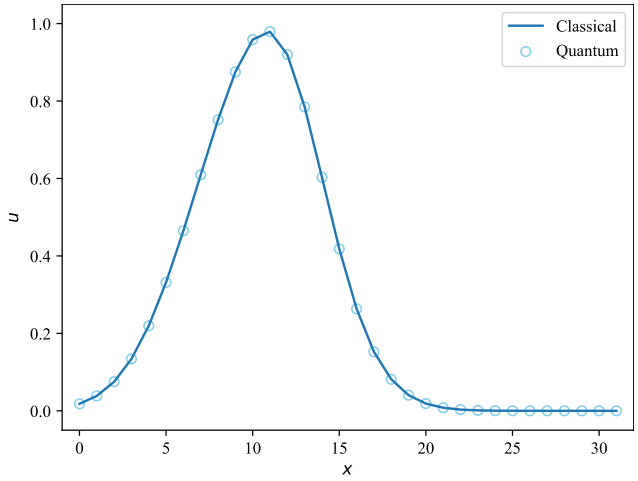
In this subsection, numerical simulations of the Burgers' equation are performed using the *qiskit* package. Classical and quantum numerical results are obtained to demonstrate the feasibility of the proposed algorithm.

4.3.1. Inviscid Burgers' Equation

In this part, for inviscid Burgers' equation, classical numerical results are compared with those obtained from the proposed quantum algorithm. The computational domain is discretized into 32 nodes with a constant spacing of $\Delta x = 1$, and the time step is fixed at $\Delta t = 0.4$. Initialization is performed using the exponential function $u(x, 0) = e^{-(\frac{x}{5}-2)^2}$. Figure 6 shows the numerical results obtained by solving this equation, and the comparisons are made at $t = 0.4$ and $t = 0.8$ respectively.



(**a**)



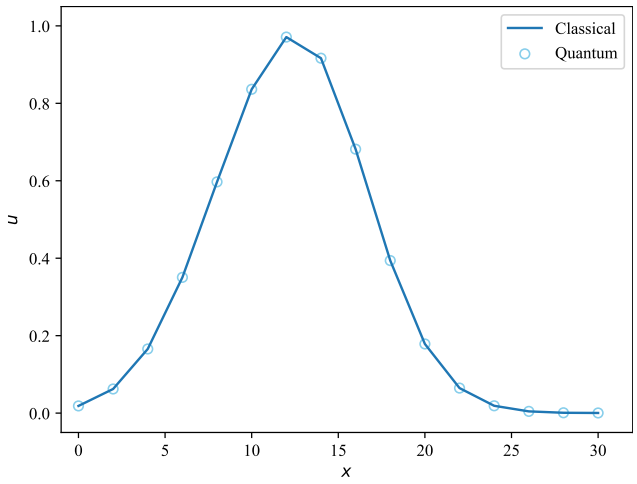
(**b**)

Figure 6. Comparison of numerical results between the classical discretization method and the numerical simulation of the quantum algorithm with (**a**) $t = 0.4$ and (**b**) $t = 0.8$ respectively for the inviscid Burgers' equation. The computational domain is divided into 32 nodes.

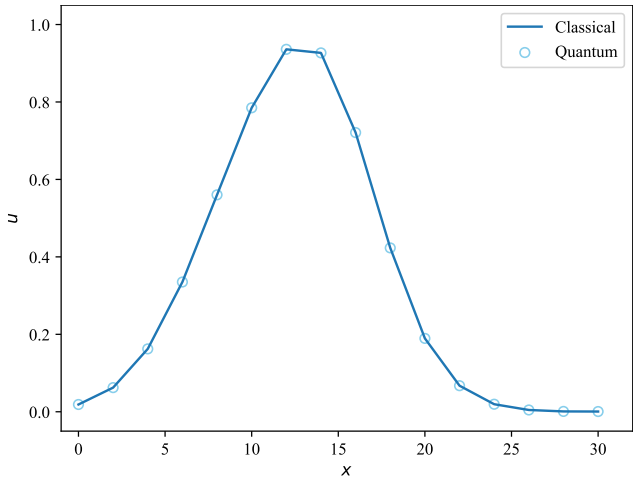
4.3.2. Viscous Burgers' Equation

In this part, we conduct the numerical simulation of the quantum algorithm for the viscous Burgers' equation. The computational domain is discretized into 16 nodes ($\Delta x = 2$). The initial value of the computational domain is set as $u(x, 0) = e^{-(\frac{x}{6}-2)^2}$, $\mu = 0.1$, and the solutions are obtained for three time steps with $\Delta t = 0.5$. Figure 7 shows good agreement

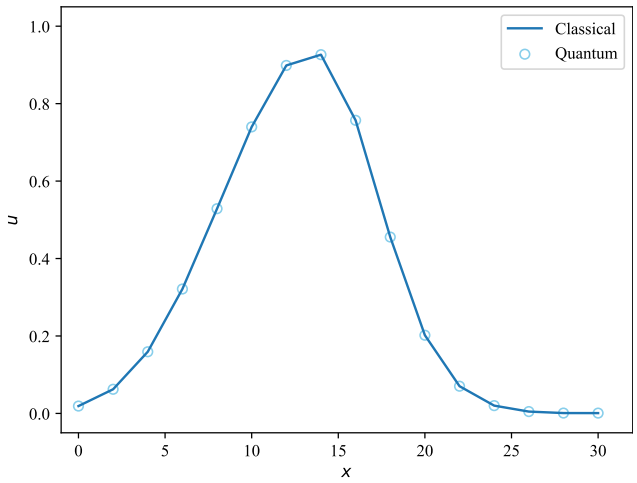
between the classical numerical results and the results obtained from the simulations of the quantum algorithm.



(**a**)



(**b**)



(**c**)

Figure 7. Comparison of numerical results between the classical discretization method and the numerical simulation of the quantum algorithm for the viscous Burgers' equation at (**a**) time step $t = 0.5$, (**b**) time step $t = 1$, (**c**) time step $t = 1.5$.

5. Complexity Analysis

In this section, the circuit complexity of the algorithm is analyzed in terms of the required number of qubits and quantum gates.

According to the superposition principle, as the problem space expands, the resources required by the quantum algorithm grow only logarithmically. In particular, if the velocity vector consists of N_x nodes, then encoding the velocity on a quantum computer requires $\log N_x$ qubits. Under the qubits reuse scheme [34], the total number of qubits in the velocity registers is $(N_t + 1) \log N_x$ for N_t steps. A single evolution step requires 3 ancilla qubits dedicated to the LCU module. Since each iteration is allocated with independent ancilla resources, the total qubit count of ancilla registers grows linearly with the number of time steps to $3N_t$. Therefore, under the same multi time step strategy, the improved algorithm reduces a total of $2N_t$ qubits.

In the implementation of multi-controlled gates, ancilla qubits serve as important control resources, and the number of ancilla qubits correlates with algorithmic complexity. Reducing the number of ancilla qubits can therefore effectively lower the computational overhead. Because operations D and D_2 are the primary factors affecting computational complexity and reflect the improvements in this work compared to [34], we will focus our analysis on D and D_2. Operations D and D_2 contain R gate and L gate, both of which can be decomposed into a sequence of multi-controlled Toffoli gates. For a k-controlled Toffoli gate, it can be decomposed into $2k - 4$ relative-phase Toffoli gates and a Toffoli gate using $k - 2$ ancilla qubits [17]. The algorithm involves $C^2 - R$, $C^3 - R$ and $C^3 - L$, where $C^k - U$ denotes the controlled-U gate with k control qubits. Because a operator $C^2 - R$ has k-controlled Toffoli gates for $3 \leq k \leq \log N_x + 1$ and a Toffoli gate, it can be expressed by $(\log N_x - 1) \log N_x$ relative-phase Toffoli gates and $\log N_x$ Toffoli gates. Similarly, $C^3 - R$ ($C^3 - L$) can be expressed by $\log N_x(\log N_x + 1)$ relative-phase Toffoli gates and $\log N_x$ Toffoli gates. Therefore, implementing operations D and D_2 requires a total of $3\log^2 N_x + \log N_x$ relative-phase Toffoli gates and $3 \log N_x$ Toffoli gates. And the quantum circuit requires $O(\log^2 N_x)$ depth. Compared to [34], our work reduces the number of relative-phase Toffoli gates by $\log^2 N_x + 11 \log N_x + 8$ and the number of Toffoli gates by $\log N_x + 4$. In addition, our algorithm relaxes the requirements for obtaining the target quantum state. More specifically, in [34], the condition required to obtain the target quantum state is that all ancilla qubits and register R_1 are in the zero state. However, in this paper, it is sufficient for the ancilla qubits to satisfy the constraint that they are in the zero state.

6. Conclusions

In this paper, we focus on designing quantum algorithm to solve the Burgers' equation. This equation is a fundamental nonlinear PDE applied in various fields, such as fluid dynamics. Due to the linear nature of quantum mechanics, designing quantum algorithms to solve nonlinear PDEs is quite challenging. This paper proposes an improved quantum algorithm that reduces the number of required qubits and relaxes the requirements for obtaining the target quantum state. Based on the qubit reuse principle, this algorithm eliminates redundant quantum state preparation and re-encoding in each evolution step. After a single evolution step, the quantum register encoding the solution vector can directly serve as the input state for subsequent steps. Meanwhile, the algorithm implements nonunitary operators via the LCU method, which further compresses the scale of ancilla qubits required for each iteration and accordingly reduces the number of control qubits needed for multi-controlled quantum gates.Building on the advantages of the algorithm proposed in [34], this work further optimizes the scheme for obtaining the target quantum state. The original stringent requirement that both copy registers and all ancilla qubits be

simultaneously projected onto the zero state is simplified to requiring only ancilla qubits to be projected onto the zero state, and the total number of required qubits is further reduced as a result. The number of qubits to be measured to obtain the target quantum state no longer grows with the system dimension, which effectively improves the applicability of the algorithm in high dimensional nonlinear system scenarios. The numerical simulation results show good agreement with the classical results.

Meanwhile, it should be clarified that subject to the fundamental constraints of the quantum no-cloning theorem and the post-selection-based realization of nonunitary operations, as stated in [42], the success probability decays gradually with the number of time steps. This is an intrinsic complexity characteristic shared by all such technical approaches. For practical applications, the proposed algorithm can leverage the advantage of dimensional quantum acceleration in short-time evolution simulations of high-dimensional systems. For longer time-scale evolution requirements, further optimization can be realized by adopting mature techniques such as amplitude amplification.

Author Contributions: Conceptualization, investigation, methodology, validation, supervision, writing and original draft preparation, X.L., R.Z., C.J., J.W., L.Z. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fundamental Research Funds for the Central Universities No. 25CX03004A, 26CX03002A, Qingdao Natural Science Foundation No. 25-1-1-163-zyyd-jch, and the Shandong Provincial Natural Science Foundation ZR2025LLZ005.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332.
2. Grover, L.K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* **1997**, *79*, 325.
3. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502.
4. Clader, B.D.; Jacobs, B.C.; Sprouse, C.R. Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.* **2013**, *110*, 250504.
5. Childs, A.M.; Kothari, R.; Somma, R.D. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comput.* **2017**, *46*, 1920–1950.
6. Wossnig, L.; Zhao, Z.; Prakash, A. Quantum linear system algorithm for dense matrices. *Phys. Rev. Lett.* **2018**, *120*, 050502.
7. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631–633.
8. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202.
9. Dunjko, V.; Briegel, H.J. Machine learning & artificial intelligence in the quantum domain: A review of recent progress. *Rep. Prog. Phys.* **2018**, *81*, 074001.
10. Zawawi, M.H.; Saleha, A.; Salwa, A.; Hassan, N.H.; Zahari, N.M.; Ramli, M.Z.; Muda, Z.C. A review: Fundamentals of computational fluid dynamics (CFD). In *AIP Conference Proceedings*; AIP Publishing LLC: Melville, NY, USA, 2018; Volume 2030, p. 020252.
11. Bravo-Prieto, C.; LaRose, R.; Cerezo, M.; Subasi, Y.; Cincio, L.; Coles, P. J. Variational quantum linear solver. *Quantum* **2023**, *7*, 1188.
12. Huang, H.Y.; Bharti, K.; Rebentrost, P. Near-term quantum algorithms for linear systems of equations with regression loss functions. *New J. Phys.* **2021**, *23*, 113021.
13. Xu, X.; Sun, J.; Endo, S.; Li, Y.; Benjamin, S.C.; Yuan, X. Variational algorithms for linear algebra. *Sci. Bull.* **2021**, *66*, 2181–2188.
14. Patil, H.; Wang, Y. Krstić, P.S. Variational quantum linear solver with a dynamic ansatz. *Phys. Rev. A* **2022**, *105*, 012423.
15. Liu, H.L.; Wu, Y.S.; Wan, L.C.; Pan, S.J.; Qin, S.J.; Gao, F.; Wen, Q.Y. Variational quantum algorithm for the Poisson equation. *Phys. Rev. A* **2021**, *104*, 022418.

16. Gnanasekaran, A.; Surana, A. Efficient variational quantum linear solver for structured sparse matrices. In Proceedings of the 2024 IEEE International Conference on Quantum Computing and Engineering (QCE), Montreal, QC, Canada, 15–20 September 2024, pp. 199–210
17. Sato, Y.; Kondo, R.; Koide, S.; Takamatsu, H.; Imoto, N. Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation. *Phys. Rev. A* **2021**, *104*, 052409.
18. Li, H.M.; Wang, Z.X.; Fei, S.M. Variational quantum algorithms for Poisson equations based on the decomposition of sparse Hamiltonians. *Phys. Rev. A* **2023**, *108*, 032418.
19. Linden, N.; Montanaro, A.; Shao, C. Quantum vs. classical algorithms for solving the heat equation. *Commun. Math. Phys.* **2022**, *395*, 601–641.
20. Ye, C.C.; An, N.B.; Ma, T.Y.; Dou, M.H.; Bai, W., Sun, D.J.; Chen, Z.Y.; Guo, G. P. A hybrid quantum-classical framework for computational fluid dynamics. *Phys. Fluids* **2024**, *36*, 127111.
21. Todorova, B.N., Steijl, R. Quantum algorithm for the collisionless Boltzmann equation. *J. Comput. Phys.* **2020**, *409*, 109347.
22. Budinski, L. Quantum algorithm for the advection–diffusion equation simulated with the lattice Boltzmann method. *Quantum Inf. Process.* **2021**, *20*, 57.
23. Xu, L.; Li, M.; Zhang, L.; Sun, H.; Yao, J. Improved quantum lattice Boltzmann method for advection-diffusion equations with a linear collision model. *Phys. Rev. E* **2025**, *111*, 045305.
24. 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. 2017*, **356**, 1057–1081.
25. An, D.; Childs, A.M.; Lin, L. Quantum Algorithm for Linear Non-unitary Dynamics with Near-Optimal Dependence on All Parameters. *Commun. Math. Phys.* **2026**, *407*, 19.
26. Jin, S.; Liu, N.; Yu, Y. Quantum simulation of partial differential equations: Applications and detailed analysis. *Phys. Rev. A* **2023**, *108*, 032603.
27. Brearley, P.; Laizet, S. Quantum algorithm for solving the advection equation using Hamiltonian simulation. *PPhys. Rev. A* **2024**, *110*, 012430.
28. Over, P.; Bengoechea, S.; Brearley, P.; Laizet, S.; Rung, T. Quantum algorithm for the advection-diffusion equation by direct block encoding of the time-marching operator. *Phys. Rev. A* **2025**, *112*, L010401.
29. Bengoechea, S.; Over, P.; Rung, T. Quantum Time-Marching Algorithms for Solving Linear Transport Problems Including Boundary Conditions. *Int. J. Numer. Methods Eng.* **2026** *127*, e70326.
30. Lubasch, M.; Joo, J.; Moinier, P., Kiffner, M.; Jaksch, D. Variational quantum algorithms for nonlinear problems. *Phys. Rev. A* **2020**, *101*, 010301.
31. Sarma, A.; Watts, T.W.; Moosa, M.; Liu, Y.; McMahon, P.L. Quantum variational solving of nonlinear and multidimensional partial differential equations. *Phys. Rev. A* **2024**, *109*, 062616.
32. Liu, J.P.; Kolden, H.Ø.; Krovi, H.K.; Childs, A.M. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2026805118.
33. Gonzalez-Conde, J.; Lewis, D.; Bharadwaj, S.S.; Sanz, M. Quantum Carleman linearization efficiency in nonlinear fluid dynamics. *Phys. Rev. Res.* **2025**, *7*, 023254.
34. Esmaeilifar, E.; Ahn, D.; Myong, R.S. Quantum algorithm for nonlinear Burgers' equation for high-speed compressible flows. *Phys. Fluids* **2024**, *36*, 106110.
35. Bonkile, M.P.; Awasthi, A.; Lakshmi, C.; Mukundan, V.; Aswin, V.S. A systematic literature review of Burgers' equation with recent advances. *Pramana* **2018**, *90*, 69.
36. Bateman, H. Some recent researches on the motion of fluids. *Mon. Weather Rev.* **1915**, *43*, 163–170.
37. Childs, A.M.; Wiebe, N. Hamiltonian simulation using linear combinations of unitary operations. *arXiv* **2012**, arXiv:1202.5822.
38. Berry, D.W.; Childs, A.M.; Cleve, R.; Kothari, R.; Somma, R.D. Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.* **2015**, *114*, 090502.
39. Camps, D.; Lin, L.; Van Beeumen, R.; Yang, C. Explicit quantum circuits for block encodings of certain sparse matrices. *SIAM J. Matrix Anal. Appl.* **2024**, *45*, 801–827.
40. Low, G.H.; Chuang, I.L. Hamiltonian simulation by qubitization. *Quantum* **2019**, *3*, 163.
41. Sturm, A.; Schillo, N. Efficient and Explicit Block Encoding of Finite Difference Discretizations of the Laplacian. *arXiv* **2025**, arXiv:2509.02429.
42. Koukoutsis, E.; Vahala, G.; Soe, M.; Hizanidis, K.; Vahala, L.; Ram, A.K. Time-Marching Quantum Algorithm for Simulation of Nonlinear Lorenz Dynamics. *Entropy* **2025**, *27*, 871.
43. McArdle, S.; Gilyén, A.; Berta, M. Quantum state preparation without coherent arithmetic. *Phys. Rev. Lett.* **2026**, *136*, 240603.

44. Huang, H.Y.; Kueng, R.; Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.* **2020**, *16*, 1050–1057.
45. Williams, C.A.; Scali, S.; Gentile, A.A.; Berger, D.; Kyriienko, O. Addressing the readout problem in quantum differential equation algorithms with quantum scientific machine learning. *arXiv* **2024**, arXiv:2411.14259.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.