Application of a variational hybrid quantum-classical algorithm to heat conduction equation

Y. Y. Liu[a], Z. Chen[b], C. Shu[a,*], S. C. Chew[c] and B. C. Khoo[a]

[a] Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260

[b] School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, China 200240

[c] National University of Singapore, Singapore 117411, Republic of Singapore

Abstract

The prosperous development of both hardware and algorithms for quantum computing (QC) potentially prompts a paradigm shift in scientific computing in various fields. As an increasingly active topic in QC, the variational quantum algorithm (VQA) provides a promising direction for solving partial differential equations on Noisy Intermediate Scale Quantum (NISQ) devices. Although a clear perspective on the advantages of QC over classical computing techniques for specific mathematical and physical problems exists, applications of QC in computational fluid dynamics to solve practical flow problems, though promising, are still in an early

[*] Corresponding author, E-mail: mpeshuc@nus.edu.sg (C. Shu).

stage of development. To explore QC in practical simulation of flow problems, this work applies a variational hybrid quantum-classical algorithm, namely the variational quantum linear solver (VQLS), to solve heat conduction equation through finite difference discretization of the Laplacian operator. Details of VQLS implementation are discussed by various test instances of linear systems. Finally, the successful statevector simulations of the heat equation in one and two dimensions demonstrate the validity of the present algorithm by proof-of-concept results. In addition, the heuristic scaling for the heat conduction problem indicates the time complexity of the present approach with logarithmic dependence on the precision ε and linear dependence on the number of qubits n.

Keywords: quantum computing, quantum simulation, variational quantum linear solver, heat equation, near-term quantum device.

1. Introduction

The slower pace of benefits from Moore's law has prompted many discussions about its validity over the recent years [1]. Limited by the increasingly smaller sizes of silicon processes, groundbreaking advance of computational performance in classical computing framework is becoming more and more effortful. Different from the conventional computing paradigm, quantum computing (QC) performs the calculations through the physical manipulation of quantum systems, which is a potential game-changer in the scientific computing community and may unleash unprecedented growth in computational capabilities.

Recent decades have witnessed the rapid development of algorithms and hardware for QC technology. A quantum advantage has been demonstrated for the first time in a benchmark task of sampling the output of a pseudo-random quantum circuit [2]. Such a milestone inspires numerous explorations of QC in various scientific and engineering areas, such as material science, chemistry, machine learning and fluid dynamics, especially in current Noisy Intermediate Scale Quantum (NISQ) [3] era. In particular, the computational fluid dynamics (CFD), which is closely associated with the scientific computing techniques, naturally benefits from the promising computation power of QC [4,5]. The essence of CFD is to discretize the partial differential equations (PDEs) describing fluid phenomena into a set of linear algebraic equations which can be numerically resolved on computational platforms. Applications of CFD to eningeering problems place high demands in computational scale and efficiency, many of which are still beyond current computational capabilities. Thus, for sophisticated CFD missions, QC indeed provides a potential alternative to classical computing framework [6-9].

Within the context of quantum computing for CFD (QCFD), many quantum algorithms have been developed. In the literature, typical approaches with practical implementations on the quantum simulators or real devices include quantum machine learning (QML) [10-12], Harrow-Hassidim-Lloyd (HHL) algorithm [13-17] and variational quantum algorithms [18-20].

Among them, the HHL algorithm was initially proposed as a heuristic algorithm to solve the linear problem of $A\mathbf{x} = \boldsymbol{b}$ where A denotes a Hermitian sparse N × N matrix and $\boldsymbol{b}$ is a unit vector. This work [13] reported an exponential speedup for the HHL algorithm. Subsequently, many research applied it as a subroutine to solve PDEs [14-17,21,22]. However, it remains unclear whether the utilization of the HHL algorithm can achieve the exponential speedup for practical fluid problems involving a much larger size and more complex structure of the matrix A. At present, it has been demonstrated on the circuit quantum computers that the HHL-type linear solvers are limited to 4 × 4 matrices [23,24]. Apart from the limited computational scale, in the NISQ era, short coherence time of noisy quantum devices which are not fault-tolerant and cannot perfectly control the qubits could be the major obstacle to the application of the HHL algorithm.

Unlike the HHL-type algorithms, the VQA algorithms adapt to the NISQ hardware and are becoming realistic strategies to execute the actual quantum advantage. In practice, the successful application of VQA algorithms to solve linear systems refers to the variational quantum linear solver (VQLS) [25-28]. The VQLS algorithm can also serve as a subroutine similar to the HHL algorithm. Specifically, the system of linear equations obtained by discretizing PDEs with proper numerical schemes like the finite difference method (FDM) [31,32] and finite element method (FEM) [33] is solved by the VQLS algorithm. Such a technique has been applied to solve the Poisson equation [20,34,35]. However, practical applications of the VQLS algorithm to flow problems of engineering interest remain scarce. Further explorations are needed to quantify the actual quantum advantage of a heuristic quantum technology over the classical numerical methods.

Here, we demonstrate a practical utility of the VQLS algorithm by solving the heat conduction problem governed by a Laplace's equation. The Laplace's equation is a special case of the Poisson's equation and has been broadly used in fluid dynamics. Although there exist several studies of such heat transfer applications [15,36,37] using other non-variational quantum algorithms, in the NISQ era, those methods may be less competitive than the incorporation with the VQLS algorithm which employs a shallower-depth quantum circuit for efficient evaluation of a cost function. Inspired by the aforementioned works [20,34,35] about the Poisson's equation and based on the fact of relatively easy implementations for the Laplace's equation, we can focus on assessing the performance of the VQLS algorithm in this practical problem. This work may lay a foundation for further applications of the VQLS algorithm to more complicated fluid flow problems.

The remainder of this paper is organized as follows. Section 2 states the heat equation and FDM algorithm for its discretization. The VQLS algorithm is then introduced. In Section 3, technical details of the VQLS implementation are discussed, followed by its application to solve the heat conduction problem in Section 4. Section 5 concludes this paper with some remarks.

2. Methods

2.1. Governing equations and finite difference discretization

In this paper, the steady state heat transfer problem which can be described by a systems of linear PDEs is considered via the following d-dimensional Laplace's equation without a source term:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x_1^2} + \cdots \frac{\partial^2 T}{\partial x_d^2} = 0, \tag{1}$$

where T denotes temperature. Dirichlet boundary conditions are enforced with the temperature T_1 for the bottom boundary and T_2 for the upper boundary. The other boundaries are periodic. After discretizing Eq. (1) by the FDM, a linear system can be obtained. Specifically, in two dimensions, we have the following expression.

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0. \tag{2}$$

Discretizing Eq. (2) with the central difference yields

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} = 0. \tag{3}$$

When the uniform mesh is used, i.e., the mesh spacings $\Delta x = \Delta y$, the following equation can be further obtained.

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0. \tag{4}$$

The subscripts i and j in above equations indicate the grid indices in the x- and y- axes, respectively. Considering that there are $N + 2$ grids in both directions, i and j can be 1, 2, …, $N + 2$. For illustration purposes, Fig. 1 presents a schematic diagram of the heat conduction problem on the uniform mesh.

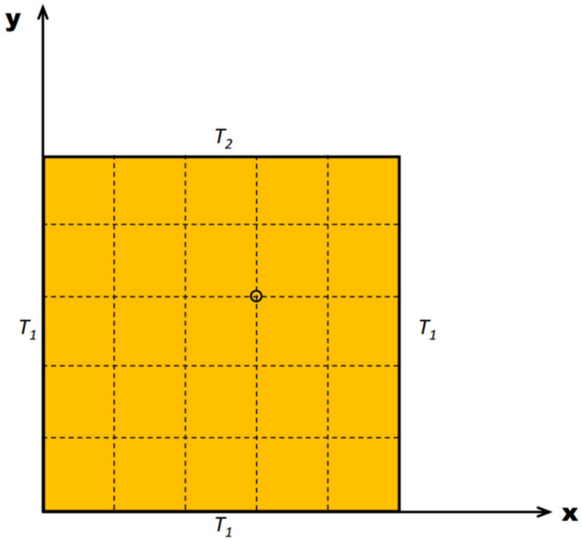


Fig. 1. Schematic diagram of heat conduction problem.

Given the Dirichlet boundary conditions and Eq. (3) for each interior grid point in the whole flow domain, such a problem can be equivalently solved via directly computing the solutions of a systems of coupled linear equations as follows,

$$A\mathbf{x} = \mathbf{b}, \tag{5}$$

where

$$A = \begin{bmatrix} -4 & 1 & 0 & & & 0 & 1 & 0 & & & 0 \\ 1 & -4 & \ddots & \ddots & & & \ddots & \ddots & \ddots & & \\ 0 & \ddots & \ddots & \ddots & \ddots & & & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & 1 & \ddots & & & \ddots & \ddots & 0 \\ & & \ddots & 1 & \ddots & 0 & \ddots & & & \ddots & 1 \\ 0 & & & \ddots & 0 & \ddots & 1 & \ddots & & & 0 \\ 1 & \ddots & & & \ddots & 1 & \ddots & \ddots & \ddots & & \\ 0 & \ddots & \ddots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & & & \ddots & \ddots & \ddots & 1 & 0 \\ & & \ddots & \ddots & \ddots & & & \ddots & 1 & -4 & 1 \\ 0 & & & 0 & 1 & 0 & & & 0 & 1 & -4 \end{bmatrix}_{N^2 \times N^2} \tag{6}$$

$$\mathbf{x} = \begin{bmatrix} T_{11} & T_{21} & \dots & T_{N1} & T_{12} & T_{22} & \dots & T_{N2} & \dots & T_{N-1N} & T_{NN} \end{bmatrix}^T_{N^2}$$

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & \dots & \dots & \dots & 0 & -1 & \dots & \dots & -1 & -1 \end{bmatrix}^T_{N^2}.$$

Clearly, A is a $N^2 \times N^2$ sparse matrix and **b** denotes the known vector by setting T_1

= 0 and T_2 = 1. Note that the number of interior grids is $N^2 = 2^n$ where n represents the number of qubits used.

In the similar form, the linear system for the one-dimensional case is cast as follows.

$$A = \begin{bmatrix} -2 & 1 & & & & & \\ 1 & & & & & & \\ & & \ddots & \ddots & & & \\ & & \ddots & & & & \\ & & & & \ddots & \ddots & \\ & & & & & \ddots & & 1 \\ & & & & & & 1 & -2 \end{bmatrix}_{N \times N} \tag{7}$$

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 & -1 \end{bmatrix}^T_N .$$

In this case, the number of interior grids is $N = 2^n$ with the number of qubits n.

Solutions of the above linear systems can be solved by the VQLS algorithm introduced in Subsection 2.2, which closes the loop of simulating the heat conduction problem.

2.2. Variational quantum linear solver to solve linear systems of equations

The VQLS is a variational quantum algorithm for solving linear systems of equations with superior efficiency to classical computational methods on NISQ quantum computers. In specific, for the linear systems in Eq. (5), the VQLS can find a normalized $|\mathbf{x}\rangle$ to fulfil the relationship $A|\mathbf{x}\rangle = |\mathbf{b}\rangle$ where $|\mathbf{b}\rangle$ denotes the quantum state from the known vector **b**. Fig. 2 depicts an overview of the VQLS algorithm. As illustrated, the input to this algorithm are the matrix A given as a linear combination of unitary matrices A_m with the coefficients c_m and a short-depth quantum circuit U such that $|\mathbf{b}\rangle = U|0\rangle$. This process can be completed through state preparation. Then the cost function $C(\boldsymbol{\alpha})$ is constructed and evaluated with a devised parameterized ansatz $V(\boldsymbol{\alpha})$. Through the hybrid quantum-classical optimization loop, the optimal parameters $\boldsymbol{\alpha}^*$ for the ansatz circuit can be found when the cost function $C(\boldsymbol{\alpha})$ achieves the convergence criterion. At the feedback loop termination, the ansatz $V(\boldsymbol{\alpha}^*)$ prepares the quantum state $|\mathbf{x}^*\rangle$ that is proportional to the solution $\mathbf{x}$, as the final output. Here, for clarity purposes, only three processes, namely, state preparation, ansatz and cost function are illustrated. Interesting readers may refer to Ref. [25] for more details of the VQLS algorithm.

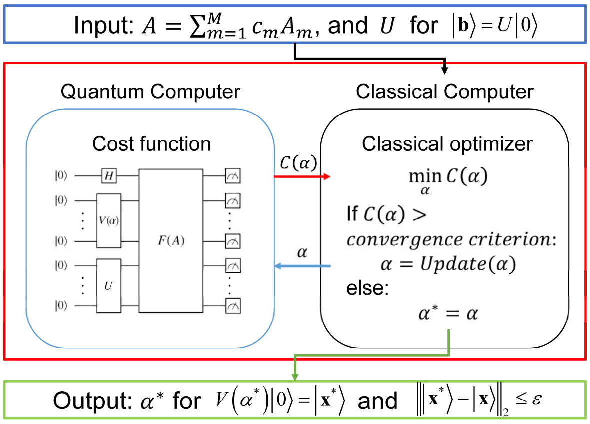

Input: $A = \sum_{m=1}^{M} c_m A_m$, and U for $|\mathbf{b}\rangle = U|0\rangle$
Quantum Computer
Classical Computer
Cost function
$|0\rangle$
H
$V(\alpha)$
U
$F(A)$
$C(\alpha)$
α
Classical optimizer
$\min_{\alpha} C(\alpha)$
If $C(\alpha) >$
$convergence\ criterion$:
$\alpha = Update(\alpha)$
else:
$\alpha^* = \alpha$
Output: α^* for $V(\alpha^*)|0\rangle = |\mathbf{x}^*\rangle$ and $\left\| |\mathbf{x}^*\rangle - |\mathbf{x}\rangle \right\|_2 \leq \varepsilon$

Fig. 2. Basic VQLS algorithm schematic diagram.

A. State preparation

The input to the VQLS algorithm requires the quantum states representing the matrix A and the vector $\mathbf{b}$. In practical implementation, the matrix A is decomposed into a linear combination of unitary matrices A_m with c_m complex coefficients as follows:

$$A = \sum_{m=1}^{M} c_m A_m. \tag{8}$$

For the decomposition, one popular approach [20,25,27,34] is based on the identity I and the following Pauli gates basis of X, Y and Z.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{9}$$

For example, for a matrix A with the size 2^n × 2^n, the Pauli basis can be selected as $\sigma_n = \{P_1 \otimes \cdots \otimes P_l\}^n$, $P_l \in \{I, X, Y, Z\}$. And A_m will belong to σ_n. Its corresponding coefficients c_m are determined via $c_m = \mathrm{Tr}(A_m \cdot A)/2^n$.

A normalized complex vector of a quantum state $|\mathbf{b}\rangle$ should also be prepared. This can be fulfilled by applying a unitary operation U to the ground state $|0\rangle$, i.e.,

$$|\mathbf{b}\rangle = U|0\rangle. \tag{10}$$

The unitary operation U may be found by using the method proposed by Shende et al. [30].

B. Hardware-efficient ansatz

The VQLS employs an ansatz for the gate sequence $V(\boldsymbol{\alpha})$ which simulates a potential solution $|\mathbf{x}\rangle = V(\boldsymbol{\alpha})|0\rangle$. In fact, it is free to choose the ansatz for specific problems, while we will follow the popular choice of using the fixed structure hardware-efficient ansatz [25,29]. As illustrated in Fig. 3, such an ansatz consists of multiple layers (*a layer is identified in the green square*) of controlled-Z gates across alternating pairs of neighboring qubits entangled by $R_y(\alpha)$ rotation gates. In Fig. 3, all parameters α are set to 1 for illustration purposes. Noteworthily, during every run of the quantum circuit, the structure of quantum gates remains the same and only the parameters α in R_y gates change.

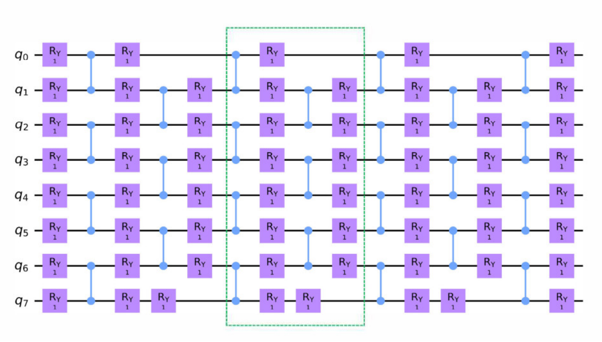


Fig. 3. An example of the fixed structure hardware-efficient ansatz.

C. Cost function

With the ansatz $V(\boldsymbol{\alpha})$, the state $\left|\psi(\boldsymbol{\alpha})\right\rangle = V(\boldsymbol{\alpha})\left|0\right\rangle$ can be prepared. Hereinafter, $\left|\psi(\boldsymbol{\alpha})\right\rangle$ is denoted as "$\left|\psi\right\rangle$" for simplicity. Since the VQLS algorithm aims to minimize the cost function, when the state $\left|\Phi\right\rangle = A\left|\psi\right\rangle$ is nearly proportional to $|\mathbf{b}\rangle$, the output of the cost function should be very small. When the vectors are close to being orthogonal, the output of the cost function will be very large. Thus, the following cost function is introduced

$$\begin{aligned} C_{Gp} &= \left\langle \Phi \middle| H_P \middle| \Phi \right\rangle \\ &= \left\langle \Phi \middle| \left(I - \left|\mathbf{b}\right\rangle\left\langle\mathbf{b}\right| \right) \middle| \Phi \right\rangle \\ &= \left\langle \Phi | \Phi \right\rangle - \left\langle \Phi | \mathbf{b} \right\rangle \left\langle \mathbf{b} | \Phi \right\rangle, \end{aligned} \tag{11}$$

with the projection Hamiltonian

$$H_P = I - |\mathbf{b}\rangle\langle\mathbf{b}|. \tag{12}$$

Generally, normalizing the cost function is necessary to increase the accuracy of the algorithm. Practically, by replacing $|\Phi\rangle$ with $|\Phi\rangle/\sqrt{\langle\Phi|\Phi\rangle}$ where $\sqrt{\langle\Phi|\Phi\rangle}$ denotes the norm of $|\Phi\rangle$, $||\Phi||$:

$$C_p = \frac{C_{Gp}}{\langle\Phi|\Phi\rangle} = 1 - \frac{|\langle\mathbf{b}|\Phi\rangle|^2}{\langle\Phi|\Phi\rangle}. \tag{13}$$

The minimization of C_p with respect to the variational parameters should lead towards the problem solution.

Clearly, there are two values to be calculated in order to evaluate the cost function, namely $|\langle\mathbf{b}|\Phi\rangle|^2$ and $\langle\Phi|\Phi\rangle$. A standard quantum computation technique called the Hadamard Test can estimate all expectation values of the previous expression. In quantum computation, the Hadamard test is a method used to create a random variable whose expected value is the expected real part of the observed value of a quantum state, i.e., $\mathrm{Re}\langle\varphi|U|\varphi\rangle$,with respect to some unitary operator. In this way, $\langle\Phi|\Phi\rangle$ can be computed by

$$\begin{aligned}\langle\Phi|\Phi\rangle &= (\mathrm{A}|\psi\rangle)^\dagger \mathrm{A}|\psi\rangle \\ &= \sum_m \sum_n \underline{\langle 0|V^\dagger} A_m{}^\dagger A_n \underline{V|0\rangle c_m{}^* c_n},\end{aligned} \tag{14}$$

and $|\langle\mathbf{b}|\Phi\rangle|^2$ can be got via

$$\begin{aligned}|\langle\mathbf{b}|\Phi\rangle|^2 &= |\langle\mathbf{b}|\mathrm{A}\psi\rangle|^2 \\ &= \sum_m \sum_n \underline{\langle 0|V^\dagger} A_m{}^\dagger \underline{U|0\rangle\langle 0|U^\dagger} A_n \underline{V|0\rangle} c_m{}^* c_n.\end{aligned} \tag{15}$$

In order to compute $\langle\Phi|\Phi\rangle$, every possible term $\underline{\langle 0|V^\dagger} A_m{}^\dagger A_n \underline{V|0\rangle}$ should be calculated using the Hadamard test. This requires the preparation of the state $V|0\rangle$

(corresponding to $|\varphi\rangle$ in the above introduction of Hadamard test), and controlled operations with some control-auxiliary qubits for the unitary matrices $A^{\dagger}{}_{m}$ and A_n (corresponding to U in the above introduction of Hadamard test). As illustrated in the fixed hardware ansatz, $V|0\rangle$ can be got. Similarly, in order to compute $|\langle\mathbf{b}|\Phi\rangle|^2$, every possible terms $\underline{\langle 0|V}^{\dagger}A_m{}^{\dagger}\underline{U|0\rangle}$ and $\underline{\langle 0|U}^{\dagger}A_n\underline{V|0\rangle}$ should be calculated using the Hadamard test. This requires the preparation of the state $V|0\rangle$, and controlled operations with some control-auxiliary qubits for the unitary matrices $A^{\dagger}{}_{m}$, A_n, U and V.

However, this global expression might be experimentally challenging in the Hardmard test since it requires to apply all the unitaries ($U^{\dagger}$, $A^{\dagger}{}_{m}$, A_n and V) in a controlled way, i.e., conditioned on the state of an ancillary qubit. The local cost function is to minimize a local version of the cost function which is easier to measure and, at the same time, leads to the same optimal solution. This local cost function can be obtained by replacing the projector $|0\rangle\langle 0|$ in the previous expression with the following positive operator:

$$P=\frac{1}{2}I+\frac{1}{2n}\sum_{j=0}^{n-1}Z_j. \tag{16}$$

where Z_j is the Pauli Z operator locally applied to the jth qubit. This gives a new $|\langle\mathbf{b}|\Phi\rangle|^2$:

$$|\langle\mathbf{b}|\Phi\rangle|^2=\sum_m\sum_n\underline{\langle 0|V^{\dagger}}A_m{}^{\dagger}\underline{UPU^{\dagger}}A_n\underline{V|0\rangle}c_m{}^{*}c_n, \tag{17}$$

Thus, the local cost function will be

$$C=1-\frac{\sum_{m}\sum_{n}\underline{\langle 0|V^{\dagger}}A_{m}{}^{\dagger}\underline{UPU^{\dagger}}A_{n}\underline{V|0\rangle}c_{m}{}^{*}c_{n}}{\sum_{m}\sum_{n}\underline{\langle 0|V^{\dagger}}A_{m}{}^{\dagger}A_{n}\underline{V|0\rangle}c_{m}{}^{*}c_{n}}, \tag{18}$$

and so we can solve our problem by minimizing C. Substituting the definition of P into the expression for C, we get:

$$C=\frac{1}{2}-\frac{1}{2n}\frac{\sum_{j=0}^{n-1}\sum_{m}\sum_{n}c_{m}{}^{*}c_{n}u_{m,n,j}}{\sum_{m}\sum_{n}c_{m}{}^{*}c_{n}u_{m,n,-1}}, \tag{19}$$

which can be computed whenever we are able to measure the following coefficients

$$u_{m,n,j}=\langle 0|V^{\dagger}A_{m}{}^{\dagger}UZ_{j}U^{\dagger}A_{n}V|0\rangle, \tag{20}$$

where we used the convention that if j=−1, Z_{-1} is replaced with the identity, i.e.,

$$u_{m,n,-1}=\langle 0|V^{\dagger}A_{m}{}^{\dagger}A_{n}V|0\rangle. \tag{21}$$

Also in this case the complex coefficients $u_{m,n,j}$ can be experimentally measured with a Hadamard test.

3. Implementation Tests of Variational Quantum Linear Solver

Based on the theoretical background of the VQLS algorithm illustrated in Subsection 2.2, this section applies this algorithm to solve a series of test problem instances on a quantum simulator as a proof-of-concept first. Generally, there are three factors affecting the convergence [34], i.e., the number of shots which determines numerical noise on evaluation process, the ansatz as the number of qubit increases and optimizer as the number of variables rises. In this work, we employ a hardware-efficient ansatz $V(\boldsymbol{\alpha})$ as described in Subsection 2.2. For the selection of classical optimizer, the comparison of various optimizers reported in Ref. [39] offers important guidance. Here, the gradient-descent optimizer with momentum [40,41] is utilized in the current study. It is known that the initialization parameters of the ansatz can have a great impact on the optimization of its parameters, and accordingly the initial parameters have been set randomly without loss of generality.

Details of implementation about the number of shots and heuristic scaling are discussed with the simulated results. Although these discussions basically are problem-specific, the analyses are deemed relevant and indeed provide proper guidance for other applications. All quantum simulations are implemented using the Xanadu's Pennylane open-source library [38] with a statevector simulator as a backend. In the comparison hereinafter, the results obtained by the classical solver and VQLS solver are noted as "classic" and "VQLS", respectively.

3.1 Definition of the test case

Here, the illustrative test instance is given by a matrix A of size $N \times N$ with a state vector $|\mathbf{b}\rangle$ as follows.

$$\begin{aligned} A &= c_0 A_0 + c_1 A_1 + c_2 A_2 \\ &= c_0 I^{\otimes n} + 0.2 X_0 Z_1 + 0.2 X_0, \\ |\mathbf{b}\rangle &= U|0\rangle = \prod_{j=0}^{n} H_j |0\rangle. \end{aligned} \tag{22}$$

where n represents the number of qubits used and $N = 2^n$. H is the Hadamard gate which is represented by the following Hadamard matrix.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{23}$$

The coefficients c_0 , c_1 and c_2 in Eq. (22) determine the matrix property. In this test, c_1

and c_2 are fixed with $c_1 = c_2 = 0.2$, while we adjust c_0 to change the condition number k for the matrix A. Accordingly, the matrix A has the following expression.

$$A = \begin{array}{r|cccccccccc|l}
 & & & & & \text{col:}N/2+1 & & \text{col:}3N/4 & & & & \\
 & c_0 & & & & 0.4 & & & & & & \\
 & & \ddots & & & & \ddots & & & & & \\
 & & & & & & & 0.4 & & & & \\
 & & & & & & & & 0 & & & \\
\text{row:}N/2+1 & 0.4 & & & & \ddots & & & & \ddots & & \\
 & & \ddots & & & & \ddots & & & & 0 & \\
\text{row:}3N/4 & & & 0.4 & & & & & & & & \\
 & & & & 0 & & & & & & & \\
 & & & & & \ddots & & & & \ddots & & \\
 & & & & & & 0 & & & & c_0 & _{N\times N}
\end{array} . \qquad (24)$$

Clearly, the matrix A is restricted to be sparse. The condition number is fixed at 2.3333 when $c_0 = 1$ no matter how many numbers of qubits are used.

3.2 Results and discussion

First, the test instance of matrix size 1024 × 1024, i.e., the number of qubits n = 10, is solved first to validate the VQLS algorithm for large matrix problems. Fig. 4 plots the obtained solutions, in which the x-axis denotes the index of the solution point and the y-axis represents the value of the solution. Note that the perfect simulation with the statevector simulator outputs the analytical state of the solution without any noise. These quantum solutions are extracted when the local cost reaches the convergence threshold of 10^{-9}. Clearly, good agreement between the results computed by the VQLS and classical solver are shown, which indicates the validity of the VQLS algorithm in solving a large linear system.

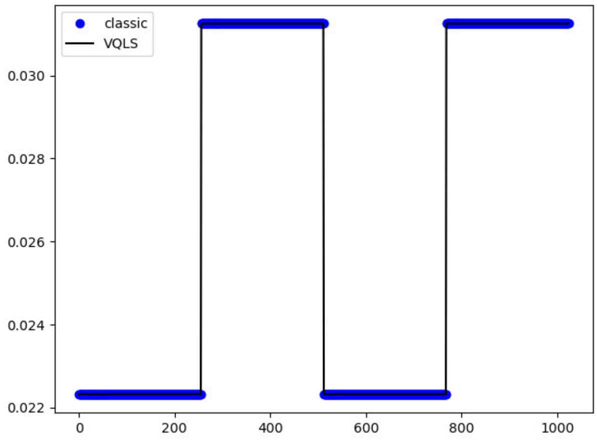
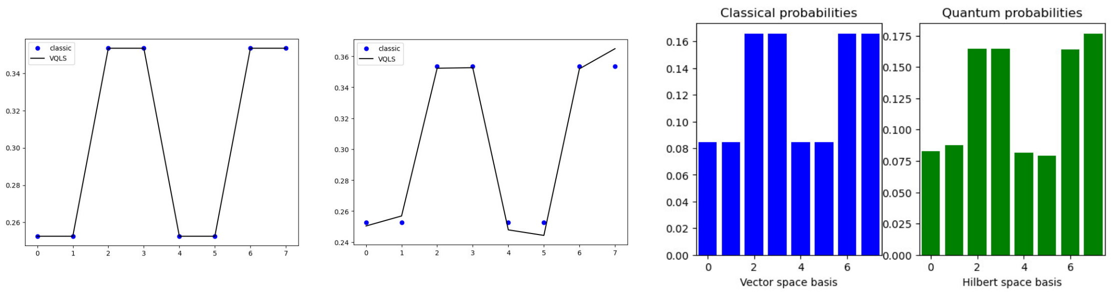


Fig. 4. Analytical state obtained by using the VQLS algorithm with $n = 10$.

In fact, although the statevector simulator which conducts the perfect simulations is used here, it is not realistic as a real quantum device would have to sample the circuit many times to generate these probabilities. Thus, the simulations have to be run with a circuit that samples the circuit instead of calculating the probabilities numerically. Theoretically, this simulation would only converge somewhat well for a very large number of shots (runs of the circuit, in order to calculate the probability distribution of outcomes). Thus the effect of the noise resulted from the number of shots is analyzed. As shown from Fig. 5 to Fig. 8, when the number of shots is selected as 3000, the quantum solutions after sampling differ from the analytical state. As the matrix size increases (from 8 × 8 to 256 × 256), the difference becomes larger. The variance of the state probability shows the same phenomena.

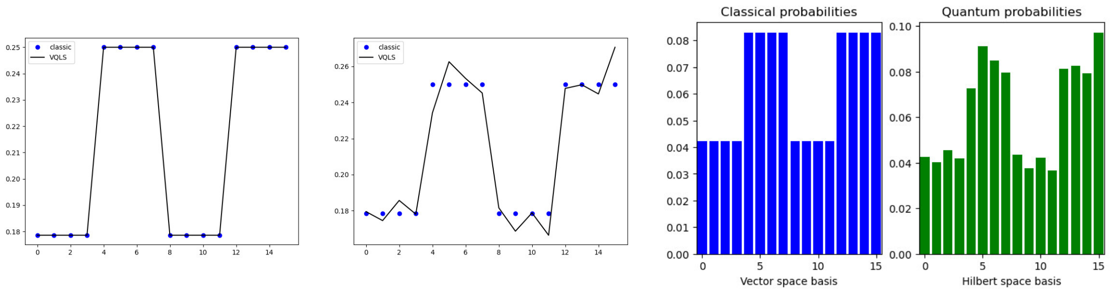


(a) Analytical state

(b) Quantum solutions with 3000 shots

(c) Quantum state probabilities with 3000 shots

Fig. 5. Results obtained by using the VQLS algorithm with $n = 3$: (a) Analytical state, (b) quantum solutions with 3000 shots and (c) quantum state probabilities with 3000 shots.

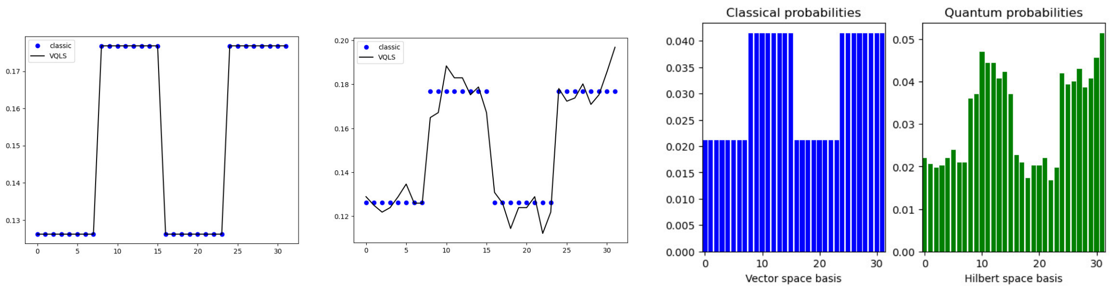


(a) Analytical state

(b) Quantum solutions with 3000 shots

(c) Quantum state probabilities with 3000 shots

Fig. 6. Results obtained by using the VQLS algorithm with $n = 4$: (a) Analytical state, (b) quantum solutions with 3000 shots and (c) quantum state probabilities with 3000 shots.

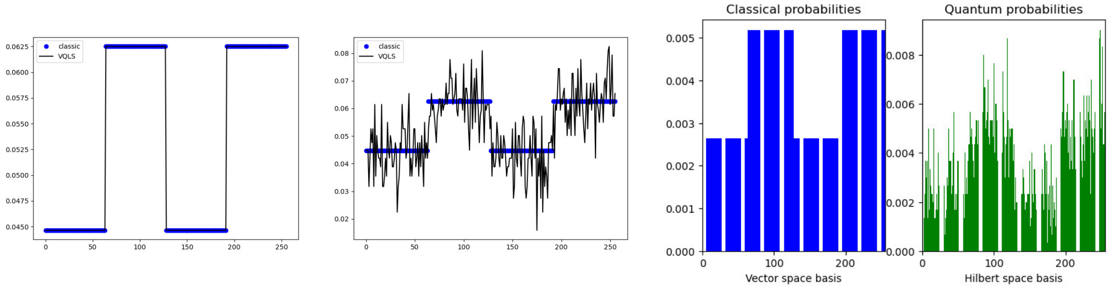


(a) Analytical state

(b) Quantum solutions with 3000 shots

(c) Quantum state probabilities with 3000 shots

Fig. 7. Results obtained by using the VQLS algorithm with n = 5: (a) Analytical state, (b) quantum solutions with 3000 shots and (c) quantum state probabilities with 3000 shots.



(a) Analytical state

(b) Quantum solutions with 3000 shots

(c) Quantum state probabilities with 3000 shots

Fig. 8. Results obtained by using the VQLS algorithm with n = 8: (a) Analytical state, (b) quantum solutions with 3000 shots and (c) quantum state probabilities with 3000 shots.

Then the dependence of the precision ε on the number of shots is further evaluated. Fig. 9 plots the relationships between the number of shots and the precision. For each data point, we implemented and averaged over 10 executions of the VQLS solver. In the log-log scale, it is clear that the dependence on the number of shots for the precision appears to be linear. As the number of qubit increases, the condition number of the matrix A becomes large and more variables should be optimized. Thus, the high fidelity requires more number of shots.

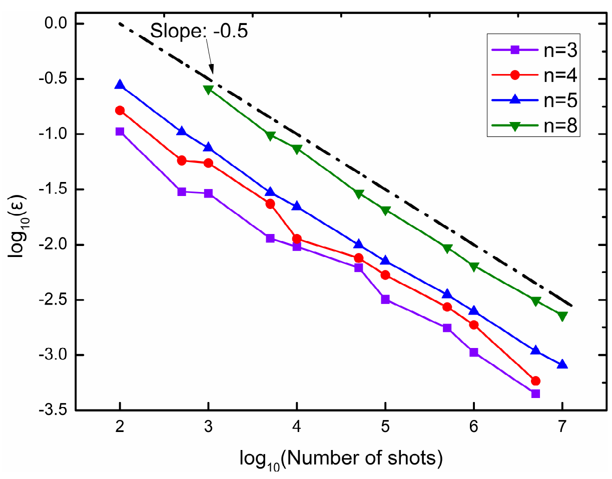
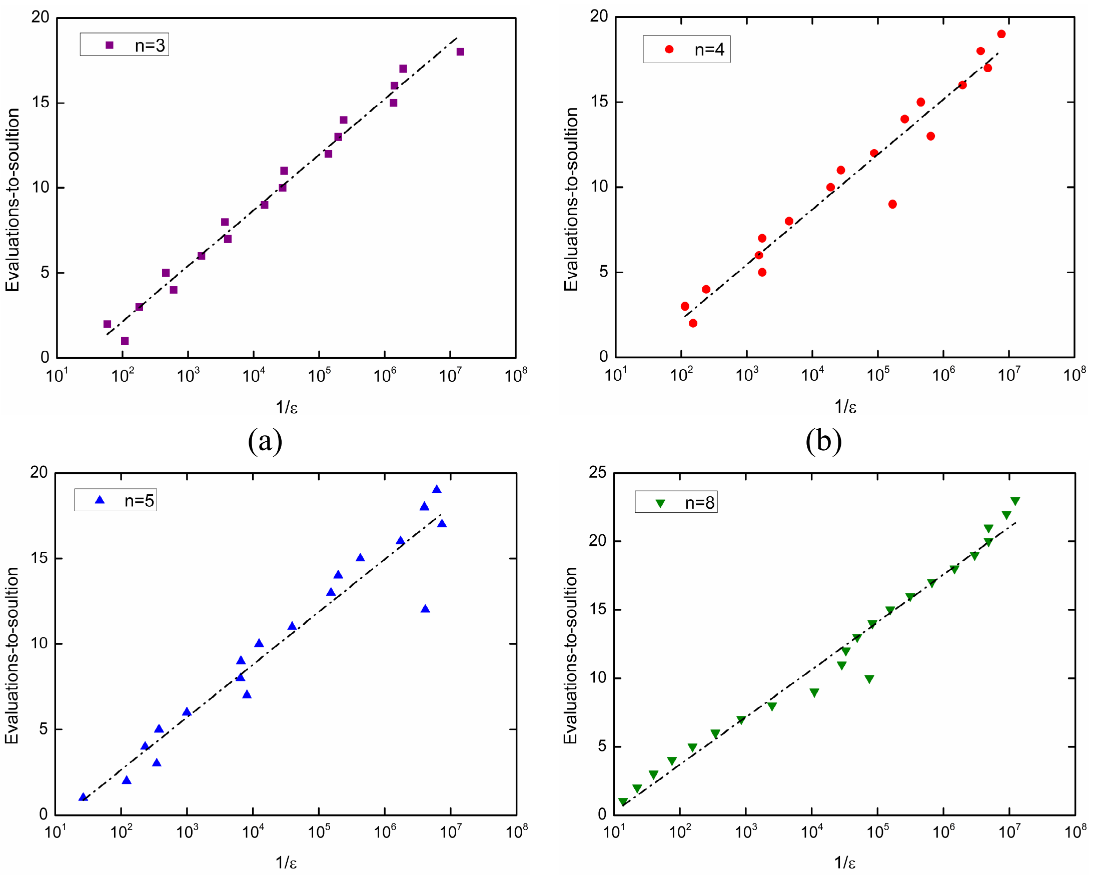


Fig. 9. Logarithmic plots of precision (ε) versus number of shots.

The time complexity of the VQLS is explored as well. We increase the number of qubits and analyses the scaling of VQLS with n. To conduct the heuristic scaling of the condition number, we change the coefficients c with n = 4. Note that for the heuristic scaling, the perfect simulation is conducted without finite sampling. Furthermore, the run time of the solver is quantified with the evaluations-to-solution, which refers to the number of exact cost function evaluations during the optimization need to guarantee that the precision ε is below a specific criteria.

In Fig. 10, the change of evaluations-to-solution with number of shots is found to be linear where the x-axis is plotted in the logarithmic scale, implying that the 1/ε scaling is logarithmic. In addition, the dependences on the number of qubits (or the matrix size) and the condition number are heuristically scaled in Fig. 11 and Fig. 12, respectively. They both appear to be linear for this case. These results confirm the competitive efficiency of the VQLS compared with the classical methods.



(a)

(b)

(c)

(d)

Fig. 10. VQLS heuristic scaling for the matrices generated according to Eq. (22). The evaluations-to-solution is the number of executions needed to guarantee a desired precision *ε*. Evaluations-to-solution versus 1/*ε* for a system of (a) *n* = 3 qubits, (b) *n* = 4 qubits, (c) *n* = 5 qubits and (d) *n* = 8 qubits. The *x* axis is shown in a log scale. For all values of *n* the data were fitted with a linear function, implying that the 1/*ε* scaling is logarithmic.

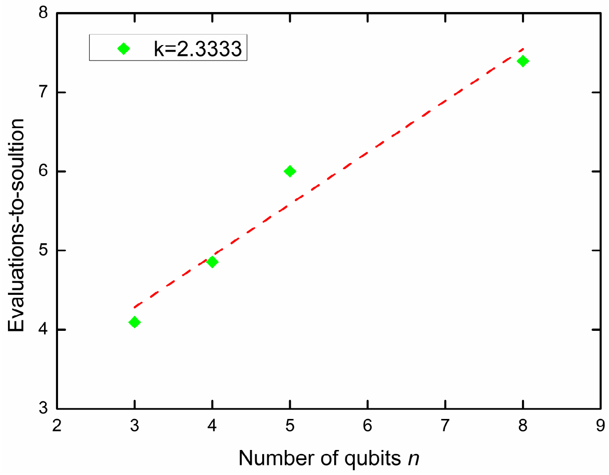


Fig. 11. VQLS heuristic scaling for the matrices generated according to Eq. (22). The evaluations-to-solution is the number of executions needed to guarantee a desired precision $\varepsilon = 0.001$. Evaluations-to-solution versus number of qubits n for systems of the condition number $k = 2.3333$. The dependence on n appears to be linear (logarithmic in N) for this case.

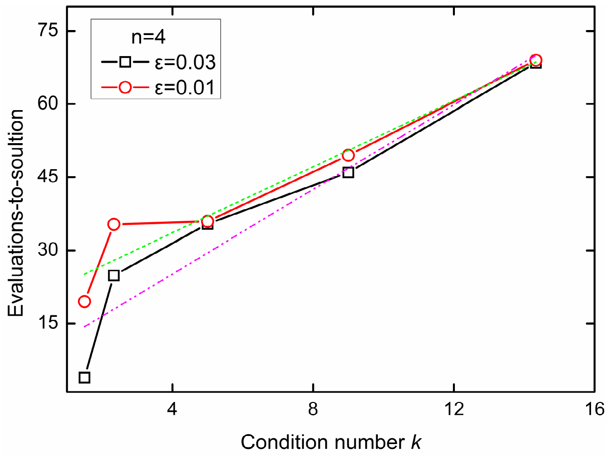


Fig. 12. VQLS heuristic scaling for the matrices generated according to Eq. (22). The evaluations-to-solution is the number of executions needed to guarantee a desired precision ε. Evaluations-to-solution versus condition number k for a system of the number of qubits $n = 4$.

4. Applications to Heat Conduction Problem

The preceding analysis can be extended to simulate the heat conduction problem with one and higher dimensions. Consider the cast linear systems presented in Eqs. (6) and (7). The corresponding implementation details and results with discussions are presented in this section. Note that the discussions hereinafter only include the perfect simulations with the statevector quantum simulator. The analysis of heuristic scaling only include converged results and each instances are run 10 times to get the averaged data.

4.1. One-dimensional heat conduction

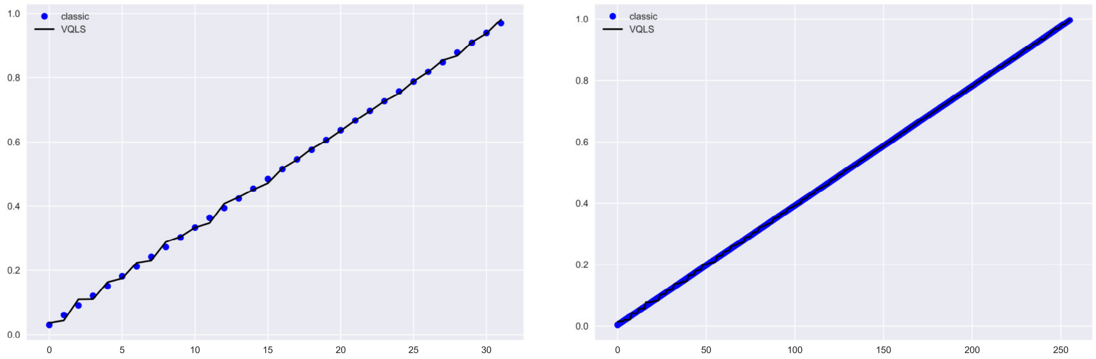
In one-dimensional case, the linear system is a strictly tridiagonal matrix. Given the boundary conditions and applying a proper normalization, the quantum states for the matrix *A* and vector **b** can be obtained. For example when *N* = 8 grids are used to discretize the computational domain, correspondingly *n* = 3, the matrix A can be linearly decomposed to 8 items as follows.

$$\begin{aligned} A &= \sum_{k=0}^{7} c_k A_k \\ &= 2(I - 0.125X_0X_1X_2 + 0.125X_0Y_1Y_2 - 0.125Y_0X_1Y_2 - \\ &\quad 0.125Y_0Y_1X_2 - 0.25X_1X_2 - 0.25Y_1Y_2 - 0.5X_2). \end{aligned} \tag{25}$$

The state for the vector **b** is given by

$$\left|\mathbf{b}\right\rangle = X_0X_1X_2\left|0\right\rangle. \tag{26}$$

Two simulated results of *N* = 32 and 256 are present in Fig. 13. When the cost function plateaus, the results obtained by the quantum VQLS solver are in acceptable agreement with solutions of the classical solver applied to this problem. The largest variance occurs when $N = 32$ which is to be expected because of our use of a random initial parameters for ansatz. When optimal parameters are given, the quantum and classical solvers may produce comparable errors.

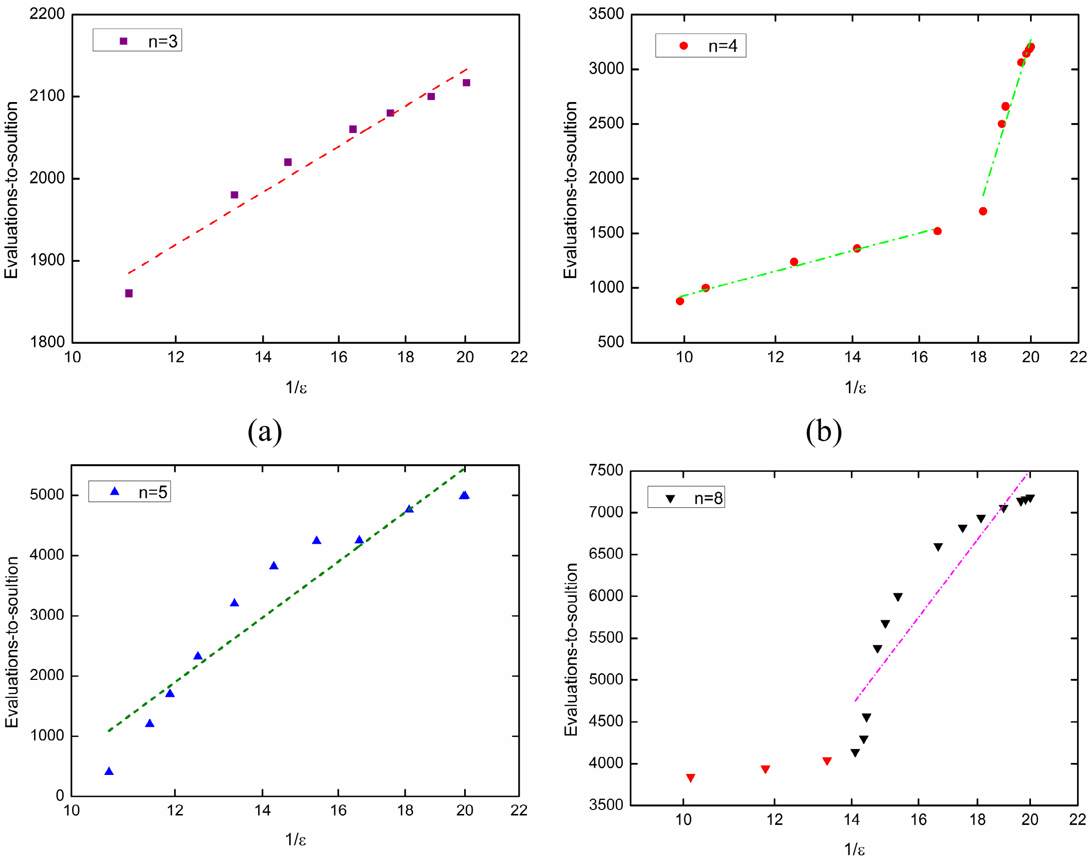


(a) Convergence rate when $N = 32$

(b) Solutions of temperature when $N = 256$

Fig. 13. Results of one-dimensional heat conduction problem: (a) Comparison of solutions when $N = 32$, (b) comparison of solutions when $N = 256$.

Similar to the analyses in Section 3, the time complexity is discussed via the heuristic scaling. Since the condition number changes with the size of the matrix (namely the system size N), we cannot scale the time complexity solely depending on the number of qubits or the condition number. Thus we combine them and conduct the heuristic scaling for the one-dimensional problem. The relationship for the evaluations-to-solution versus $1/\varepsilon$ is determined. Furthermore, the scaling for evaluations-to-solution versus the system size N guaranteeing a desired ε precision is found. The corresponding results are shown in Fig. 14 and Fig. 15. Obviously, for all values of n, the data were nearly fitted with a linear function, implying that the $1/\varepsilon$ scaling is logarithmic. From the heuristic scaling with number of qubits n for $\varepsilon = 0{:}05$. the dependence appears to be linear (logarithmic in N) for this example.



(a)

(b)

(c)

(d)

Fig. 14. Heuristic scaling for one-dimensional heat conduction problem. The evaluations-to-solution versus $1/\varepsilon$ for $n = 3; 4; 5$ and 8. The x-axis is shown in a log scale. For all values of n, the data were nearly fitted with a linear function, implying that the $1/\varepsilon$ scaling is logarithmic.

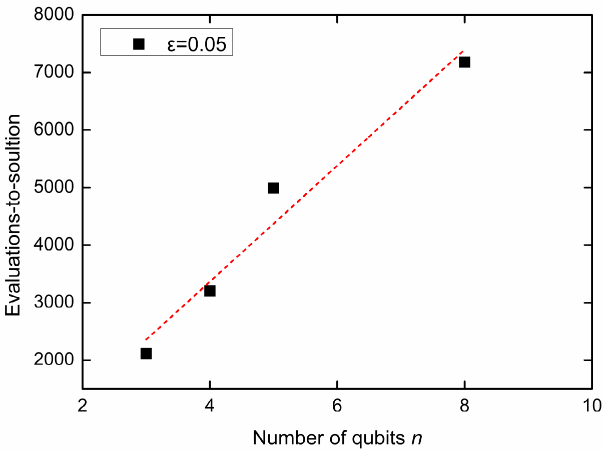


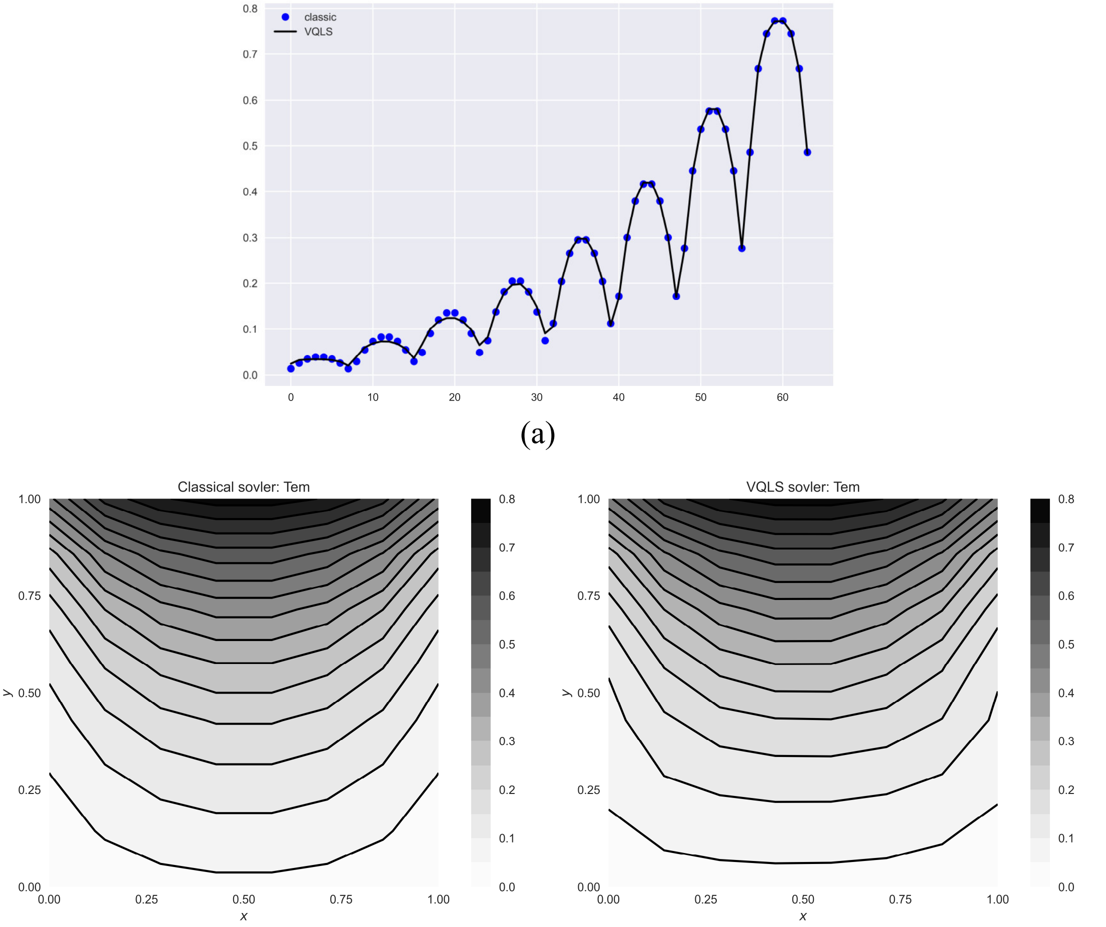
Fig. 15. Heuristic scaling with number of qubits n when $\varepsilon = 0{:}05$ for one-dimensional heat conduction problem. The dependence on n appears to be linear (logarithmic in N) for this example.

4.2. Two-dimensional heat conduction

When a higher dimension is considered for the heat conduction problem, as shown in Eq. (6), the structure of the linear system becomes a block symmetric banded/Toeplitz matrix. Similar to the one-dimensional case, the quantum states for the matrix A and vector **b** should be prepared via quantum logic operators for execution in quantum circuits. When $n = 6$ qubits are used for the ansatz, i.e., $N = 8$ interior grids in each direction, the matrix A can be linearly decomposed to 15 items. For the case of $n = 8$ and $N = 16$, there are 31 items for linear combination of unitaries for the matrix A.

Fig. 16 and Fig. 17 plot the results obtained by the VQLS solver applied to this problem. The reference data are computed by a classical solver with the direct method. As we see from the comparisons of the temperature solutions and contours, excellent agreements have been achieved.

Fig. 18 and Fig. 19 depict the heuristic scaling for the dependence on ε and the number of qubits n, respectively. It can be observed that, similar to the conclusion in the one-dimensional case, the $1/\varepsilon$ scaling is logarithmic and the dependence on n appears to be linear (logarithmic in N).

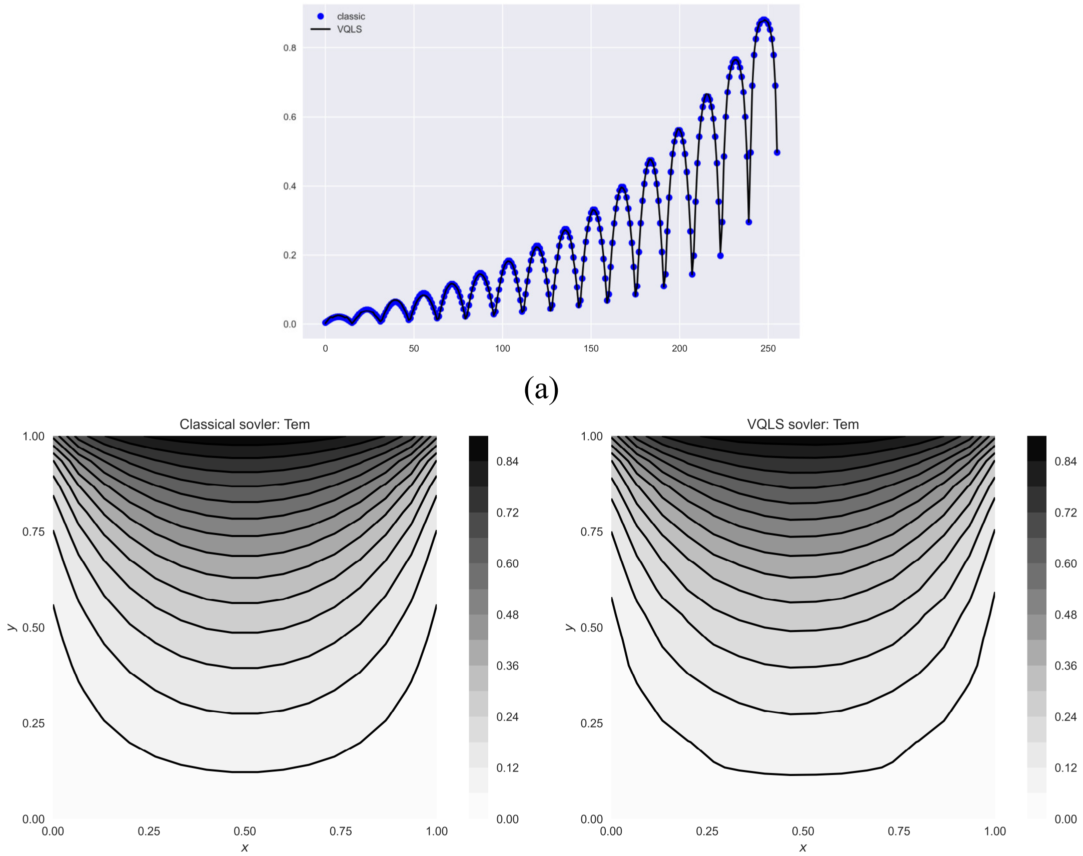


(a)

(b) Classical solver

(c) VQLS solver

Fig. 16. Results of two-dimensional heat conduction problem when $N = 8$ and $\varepsilon = 0.05$: (a) comparison of solutions; (b) the temperature contours obtained by the classical solver and (c) the temperature contours obtained by the VQLS solver.

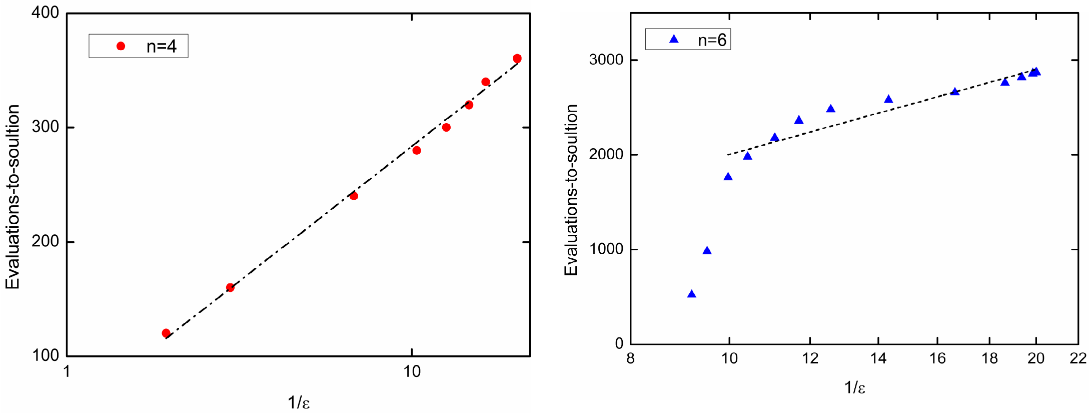


(a)

(b) Classical solver

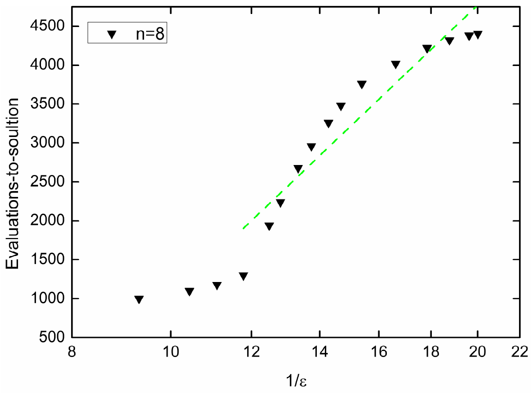
(c) VQLS solver

Fig. 17. Results of two-dimensional heat conduction problem when $N = 16$ and $\varepsilon = 0.05$: (a) comparison of solutions; (b) the temperature contours obtained by the classical solver and (c) the temperature contours obtained by the VQLS solver.



(a)

(b)



(c)

Fig. 18. Heuristic scaling for two-dimensional heat conduction problem. The evaluations-to-solution versus 1/*ε* for *n* = 4; 6 and 8. The *x*-axis is shown in a log scale. For all values of *n*, the data were nearly fitted with a linear function, implying that the 1/*ε* scaling is logarithmic.

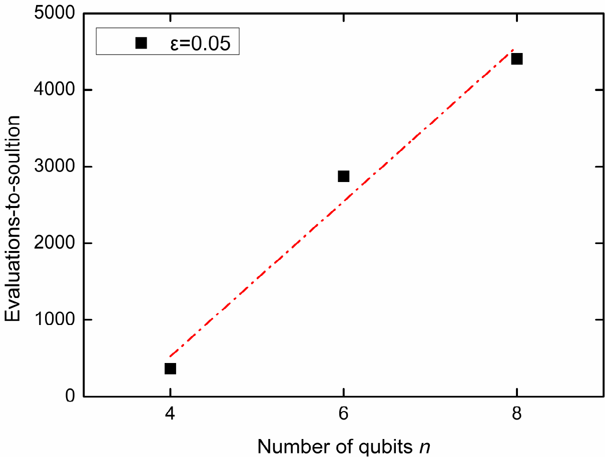


Fig. 19. Heuristic scaling with number of qubits *n* when ε = 0:05 for two-dimensional heat conduction problem. The dependence on n appears to be linear (logarithmic in *N*) for this example.

5. Conclusions and Perspectives

This paper presents a practical application of the variational quantum linear solver for heat equations which include a Laplacian operator to be solved. Using statevector simulations, it is demonstrated that the variational quantum algorithms can be useful in numerical resolution of partial differential equations. The relationship between the shots and variance is revealed. By doing comprehensive assessment of various parameters, the time complexity of the VQLS algorithm is demonstrated to scale efficiently in the precision, the condition number and the size of the linear system. In addition, with the successful simulations of the one- and two-dimensional heat conduction problem, the present VQLS based method is well validated in terms of accuracy and efficiency. Based on agreeable results, the heuristic scaling shows the nearly logarithmic dependence on $1/\varepsilon$ and linear dependence on the number of qubits (or logarithmic dependence on the system size).

In this study, due to computational limitations, only the relatively small sizes of linear systems are considered. Moreover, the ansatz and optimizer may be non-optimal. Although expected results for the heat conduction problem are obtained by the present quantum solver, it should be admitted that solving linear system of equations in general which results from PDEs is still challenging, and future discussion about the potential speed-up for the restricted and practical tasks would be an interesting research direction.

References

[1] P. Givi, A.J. Daley, D. Mavriplis, M. Malik, Quantum speedup for aeroscience and engineering, AIAA Journal 58, 3715 (2020).

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G. Brandao, D.A. Buell, B. Burkett, Quantum supremacy using a programmable superconducting processor, Nature 574, 505 (2019).

[3] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum 2, 79 (2018).

[4] S.S. Bharadwaj, K.R. Sreenivasan, Quantum computation of fluid dynamics, (2020) arXiv preprint arXiv:2007.09147.

[5] W. Itani, Fluid Dynamicists Need to Add Quantum Mechanics into their Toolbox, (2021).

[6] N. Ray, T. Banerjee, B. Nadiga, S. Karra, Towards solving the Navier-Stokes equation on quantum computers, (2019) arXiv preprint arXiv:1904.09033.

[7] C.C. Chen, S.Y. Shiau, M.F. Wu, Y.R. Wu, Hybrid classical-quantum linear solver using Noisy Intermediate-Scale Quantum machines, Sci. Rep. 9, 1 (2019).

[8] Steijl, R., Barakos, G. N. (2018). Parallel evaluation of quantum algorithms for computational fluid dynamics. Computers & Fluids, 173, 22-28.

[9] Zylberman, J., Di Molfetta, G., Brachet, M., Loureiro, N.F. and Debbasch, F., (2022). Hybrid Quantum-Classical Algorithm for Hydrodynamics. arXiv preprint arXiv:2202.00918.

[10] J. D. Martín-Guerrero, L. Lamata, Quantum Machine Learning: A tutorial, Neurocomputing, 470, 457 (2022).

[11] C.B. Góes, T.O. Maciel, G.G. Pollachini, R. Cuenca, J.P. Salazar, E.I. Duzzioni, QBoost for regression problems: solving partial differential equations, (2021) arXiv preprint arXiv:2108.13346.

[12] O. Kyriienko, A.E. Paine, V. E. Elfving, Solving nonlinear differential equations with differentiable quantum circuits, Phys. Rev. A 103, 052416 (2021).

[13] A.W. Harrow, A. Hassidim, S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. 103, 150502 (2009).

[14] A., Montanaro, S. Pallister, Quantum algorithms and the finite element method, Phys. Rev. A 93, 032324 (2016).

[15] C. Lu, Z. Hu, B. Xie, N. Zhang, Quantum CFD simulations for heat transfer applications, In ASME International Mechanical Engineering Congress and Exposition (Vol. 84584, p. V010T10A050), American Society of Mechanical Engineers, (2020).

[16] Z.Y. Chen, C. Xue, S.M. Chen, B.H. Lu, Y.C. Wu, J.C. Ding, S.H. Huang, G.P. Guo, Quantum finite volume method for computational fluid dynamics with classical input and output, (2021) arXiv preprint arXiv:2102.03557.

[17] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei, Y. Gu, Quantum fast Poisson solver: the algorithm and complete and modular circuit design, Quantum Inf. Process. 19, 1 (2020).

[18] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, D. Jaksch, Variational quantum

algorithms for nonlinear problems, Phys. Rev. A 101, 010301 (2020).

[19] Y. Yang, Z. Shan, B. Zhao, L. Xu, Variational Quantum Algorithm and Its Application on Non-Linear Equations, In Journal of Physics: Conference Series (Vol. 1883, No. 1, p. 012007), IOP Publishing (2021, April).

[20] H.L. Liu, Y.S. Wu, L.C. Wan, S.J. Pan, S.J. Qin, F. Gao, Q.Y. Wen, Variational quantum algorithm for the Poisson equation, Phys. Rev. A 104, 022418 (2021).

[21] D.W. Berry, A.M. Childs, A. Ostrander, G.M. Wang, Quantum Algorithm for Linear Differential Equations with Exponentially Improved Dependence on Precision, Commun. Math. Phys. 356, 1057 (2017).

[22] P.C. Costa, S. Jordan, A. Ostrander, Quantum algorithm for simulating the wave equation, Phys. Rev. A 99, 012323 (2019).

[23] Y. Lee, J. Joo, S. Lee, Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience, Sci. Rep. 9, 1 (2019).

[24] F. Gao, G. Wu, M. Yang, W. Cui, F. Shuang, A hybrid algorithm to solve linear systems of equations with limited qubit resources, Quantum Inf. Process. 21, 1 (2022).

[25] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, P.J. Coles, Variational quantum linear solver, (2019) arXiv preprint arXiv:1909.05820.

[26] H. Patil, Y. Wang, P.S. Krstić, Variational quantum linear solver with a dynamic ansatz, Phys. Rev. A 105, 012423 (2022).

[27] X. Xu, J. Sun, S. Endo, Y. Li, S.C. Benjamin, X. Yuan, Variational algorithms for linear algebra, Sci. Bull. 66, 2181 (2021).

[28] H.Y. Huang, K. Bharti, P. Rebentrost, Near-term quantum algorithms for linear systems of equations with regression loss functions, New J. Phys. 23, 113021 (2021).

[29] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, P.J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, Nat. Commun. 12, 1791 (2021).

[30] V.V. Shende, S.S. Bullock, I.L. Markov, Synthesis of quantum-logic circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25, 1000 (2006).

[31] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Comput. Struct. 11, 83 (1980).

[32] M.N. Özişik, H.R. Orlande, M.J. Colaco, R.M. Cotta, Finite difference methods in heat transfer. CRC press, (2017).

[33] O.C. Zienkiewicz, R.L Taylor, J.Z. Zhu, The finite element method: its basis and fundamentals, Elsevier (2005).

[34] E. Cappanera, Variational Quantum Linear Solver for Finite Element Problems: a Poisson equation test case, (2021).

[35] Y. Sato, R. Kondo, S. Koide, H. Takamatsu, N. Imoto, Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation, Phys. Rev. A 104, 052409 (2021).

[36] G.G. Pollachini, J.P. Salazar, C.B. Góes, T.O. Maciel, E.I. Duzzioni, Hybrid classical-quantum approach to solve the heat equation using quantum annealers,

Phys. Rev. A 104, 032426 (2021).

[37] N. Linden, A. Montanaro, C. Shao, Quantum vs. classical algorithms for solving the heat equation, (2020) arXiv preprint arXiv:2004.06516.

[38] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M.S. Alam, S. Ahmed, J.M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, PennyLane: Automatic differentiation of hybrid quantum-classical computations, (2018) arXiv:1811.04968.

[39] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, F. Petruccione, A comparison of various classical optimizers for a variational quantum linear solver, Quantum Inf. Process. 20, 1 (2021).

[40] P. Kingma Diederik, J.B. Jimmy, ADAM: A Method for Stochastic Optimization, (2014) arXiv:1412.6980.

[41] A. Borle, V. Elfving, S.J. Lomonaco, Quantum approximate optimization for hard problems in linear algebra, SciPost Physics Core 4, 031 (2021).