Solving Differential Equations using Quantum Computation

R.Vishwanath (CE20B024)

Indian Institute of Technology Tirupati, India

Abstract

Differential equations are ubiquitous in the field of engineering and sciences. Solving differential equations is of particular importance in the field of engineering and sciences. Methods based on finite difference and spectral methods are popularly used to solve differential equations numerically. In recent years, few quantum algorithms have been shown to offer exponential speed up in the time complexity of steps required to solve the system numerically. In this work, we briefly review quantum algorithms employed to solve differential equations and their classical counterparts. Also, the procedure for a finite difference-based method to solve differential equations is presented, and quantum circuit implementation is shown. The procedure is applied to solve the 1-D transient heat equation.

Keywords: Quantum computation, Differential equations, Nonlinear dynamics, Heat equation

1. Introduction

Quantum computers represent a recent class of computing technology that operates fundamentally differently than traditional/classical computers. Rather than utilizing standard bits for data processing, they employ entities known as quantum bits or qubits which are governed by the principles of quantum mechanics, these qubits can simultaneously occupy various states due to a phenomenon known as superposition. Consequently, quantum computers can process numerous potential outcomes at once, unlike classical computers that holds one value at a time.

Another fascinating aspect of quantum computers is entanglement, where qubits become interconnected so that the state of one qubit is dependent on the state of another, no matter how far apart they are. This enables quantum computers to perform complex calculations and solve problems in ways that classical computers can't match. Because of these unique properties, quantum computers have the potential to tackle specific problems much faster and more efficiently than classical computers. This includes solving complex equations like partial differential equations, which are crucial in many scientific and engineering fields. By harnessing the power of quantum computation, researchers aim to unlock new levels of computational speed and capability, paving the way for groundbreaking discoveries and technological advancements.

Solving partial differential equations (PDEs) is a difficult challenge; these equations describe how things change over space and time, such as temporal and spatial variations of temperature in a room or wave motion

in water. They're tricky because they often don't have simple closed-form relations. Instead, numerical methods are commonly used, breaking down the problem into smaller pieces and approximating the solution locally. This can be time-consuming and prone to errors, especially for equations involving many different factors or dimensions and furthermore, ensuring that our solutions satisfy the imposed specific boundary and initial conditions. Even then, minor errors in calculations can significantly off-put the solution.

Quantum algorithms offer promising advantages over classical algorithms for solving partial differential equations (PDEs) by leveraging the unique principles of quantum mechanics. Unlike classical computers, quantum computers can explore multiple solutions simultaneously, thanks to their ability to process information in quantum bits or qubits, which exist in superposition states. This parallelism allows quantum algorithms to tackle the high dimensionality of PDEs more efficiently, potentially leading to exponential speedups in solution times. Moreover, quantum algorithms can further exploit quantum interference effects to enhance computational efficiency, enabling them to navigate complex solution spaces more precisely. Additionally, quantum algorithms hold promise for addressing challenges associated with numerical stability in classical approaches, potentially offering more robust and accurate solutions for PDEs. By harnessing the power of quantum computation, researchers aim to revolutionize PDE-solving methodologies, unlocking new frontiers in computational efficiency and accuracy that could have profound implications across various scientific and engineering domains.

2. Literature review of quantum algorithms for solving DEs

Pioneering work on the efficient simulation of Hamiltonian evolution through quantum computation performed by Berry et al. (2006) exhibited the prospect of solving differential equations through quantum algorithms, following the development of the HHL algorithm (Harrow et al. (2009)), a quantum algorithm for solving linear system of equations. The quantum algorithm showcased an exponential speedup in computational time complexity (w.r.t dimension of the system of equations) compared to the best classical algorithm known thus far. This has motivated researchers to integrate HHL algorithms or procedures within HHL algorithms to achieve the same exponential speedup in solving differential equations (Berry (2014)). More sophisticated methods inspired by classical methods like spectral methods (Childs and Liu (2020)) and finite element methods (Montanaro and Pallister (2016)) are employed to improve further time complexity associated with other variables in the system. Recently, the prospect and the possible utility of solving nonlinear differential equations were investigated (Liu (2022)), and a particular advantage in terms of time complexity is expected for solving nonlinear differential equations. As the current infrastructure of quantum computing hardware limits the practical implementation of the highly optimal models, Wei et al. (2023) suggested using the more primitive finite difference method-based algorithm to solve partial differential equations. Though

the time complexity of this algorithm scales sub-optimally with the parameters, it requires fewer gates to implement for very small system dimensions in comparison to other aforementioned methods.

3. Solving 1-D heat equation

In this work, we use the approach presented by Wei et al. (2023) to solve a 1-D heat equation numerically. The 1-D transient heat equation is presented below,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = k\frac{\partial^2 u}{\partial x^2}, \tag{1}$$

where U, t, x and k refer to the temperature, time, position coordinate and thermal diffusivity, respectively. To numerically solve any differential equations, we require information regarding boundary conditions (to obtain a unique solution) and the initial conditions. This can be understood by considering Picard's successive approximation scheme, which is the general procedure to solve ODEs and involves the integration of the differential function to advance the solution numerically. Thus, boundary conditions are needed to perform the operation. The operation procedure of Picard's successive approximation in solving a 1^{st} order differential equation ($\dot{x} = f(x)$) is shown below,

$$x_{i+1}(t) = x_i(t_0) + \int_{t_0}^{t} f(x_i(s), s)\mathrm{d}s, \tag{2}$$

where $x_{i+1}(t)$ denote the approximate value corresponding to $i + 1^{th}$ iteration of the procedure. In eq. (2), we can observe that we need a boundary condition ($x(t_0)$) or a starting condition (x_0) to advance the system. As Picard's successive approximation scheme describes the general procedure involved in solving DEs, all numerical schemes, including the quantum algorithms, require boundary conditions to solve it numerically. Depending on the number of dimensions of the system and type of boundary conditions, the number of boundary conditions required for a unique solution might vary.

As the eq. (1) is a partial differential equation (involves derivative w.r.t two independent variables) and involves 2^{nd} order derivative w.r.t x, the system has three arbitrary constants. It requires at least three boundary conditions to solve it. To solve the 1-D heat equation, in this work, we impose the following boundary conditions,

$$u(x = x_0, t) = C_1, \tag{3a}$$

$$u(x = x_n, t) = C_2, \tag{3b}$$

$$u(x, t = 0) = f(x), \tag{3c}$$

where C_1, C_2 are constant values imposed at positions x_0 and x_n, which are assumed as endpoints in this work, also the initial temperature profile (at time $t = t_0$. Without loss of generality, t_0 is treated as zero

in this work) is specified through the function $f(x)$. It is to be noted that $f(x)$ must also satisfy eqs. (3a) and (3b).

To simplify the system into a form which is easily implementable, we assume a function $v(x,t)$ with the following structure,

$$u(x,t) = v(x,t) + (x - x_0).\frac{C_2 - C_1}{x_n - x_0} + C_1, \tag{4}$$

substituting eq. (4) on the 1-D heat equation (eq. (1)), we realize that the variable $v(x,t)$ also satisfies the 1-D transient heat equation as shown below,

$$\frac{\mathrm{d}v}{\mathrm{d}t} = k\frac{\partial^2 v}{\partial x^2}, . \tag{5}$$

Owing to the structure of the variable $v(x,t)$ as per eq. (4), we realize that the boundary conditions are now modified as per the following form,

$$v(x = x_0, t) = 0, \tag{6a}$$

$$v(x = x_n, t) = 0, \tag{6b}$$

$$v(x, t = 0) = f(x) - (x - x_0).\frac{C_2 - C_1}{x_n - x_0} - C_1. \tag{6c}$$

Now let us consider only the variable $v(x,t)$ instead of $u(x,t)$ ($u(t,x)$ can always be inferred as per eq. (4)) and consider $v(x,t)$ as the modified temperature variable. Though the boundary conditions at the endpoints (eqs. (6a) and (6b) are now simplified. It is particularly important to express the initial state (eq. (6c)) in the quantum representation so as to advance the solution.

3.1. Discretization of solution space

As we employ a finite difference scheme, it is necessary to discretize the space and solve over the discretized time intervals. Let us assume that it is necessary to identify the temperature profile at time T. So, the time domain is split into m equal intervals given by [0,t_1], [t_1,t_2], ... and [t_{m-1},t_m]. In a similar manner, the space domain is also split into N equal intervals given by [x_0,x_1],[x_1,x_2]... and [x_{n-1},x_n]. The corresponding solutions at these points are denoted by v_j^i, where superscript (i$\in \{0, 1..., m\}$) denotes time index and subscript (j$\in \{0, 1..., n\}$) corresponds to length index. A visual sketch of the discretized space is provided in fig. 1.

3.2. State preparation

To encode information regarding the temperature at each discretised position coordinate, we encode the discretized position coordinate through basis encoding and the corresponding temperature (modified temperature) is encoded through amplitude encoding. The index of each position coordinate is encoded as basis state through basis encoding, i.e. position at index i as $|i\rangle$. For amplitude encoding, the corresponding

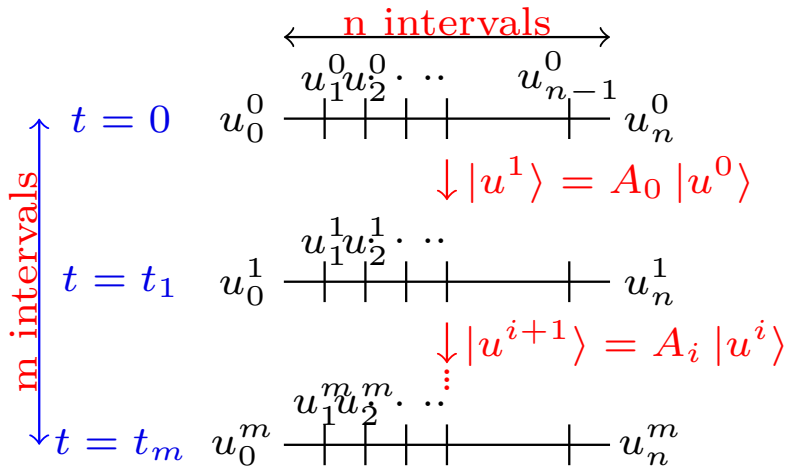

Figure 1: Schematic of discretized solution space

amplitude of each basis state is proportional to the modified temperature at that point, so measurement of the state would provide a particular basis, and the probability of such a measurement is proportional to the modified temperature corresponding to that basis/state. Through this encoding method, we end up with the following form,

$$|\psi^i\rangle = \frac{\sum_{j=0}^{n} v(x_j^i)\,|j\rangle}{C}, \tag{7}$$

where C is the normalization coefficient.

3.3. FDM formulation

Through the finite difference scheme, the eq. (5) is discretized in the following manner,

$$v_j^{i+1} - v_j^i \approx \frac{T}{n}\frac{\mathrm{d}v_j}{\mathrm{d}t} \tag{8a}$$

$$= \frac{k.T}{n}\frac{\mathrm{d}^2 v^i}{\mathrm{d}x^2} \tag{8b}$$

$$\approx k\frac{T}{n}\frac{\left(v_{j+1}^i + v_{j-1}^i - 2v_j^i\right)}{2(\frac{x_m - x_0}{n})^2} \tag{8c}$$

$$= k\frac{T}{n}\frac{\left(v_{j+1}^i + v_{j-1}^i - 2v_j^i\right)}{2(\frac{x_m - x_0}{n})^2} \tag{8d}$$

$$v_j^{i+1} \approx v_j^i + \alpha(v_{j+1}^i + v_{j-1}^i - 2v_j^i) \tag{8e}$$

$$v_j^{i+1} \approx (1 - 2\alpha)v_j^i + \alpha(v_{j+1}^i + v_{j-1}^i) \tag{8f}$$

In eq. (8f), $\alpha = \frac{nT}{2(x-x_0)^2}$. We can observe that representation of eq. (8f) in the matrix formation would lead to a tri-diagonal formation shown below,

$$\begin{Bmatrix} v_0^{i+1} \\ v_1^{i+1} \\ v_2^{i+1} \\ . \\ . \\ v_{n-1}^{i+1} \\ v_n^{i+1} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & ... & 0 & 0 \\ \alpha & 1-2\alpha & \alpha & ... & 0 & 0 \\ 0 & \alpha & 1-2\alpha & ... & 0 & 0 \\ . & . & . & . & . & \\ . & . & . & . & . & . \\ 0 & 0 & 0 & ... & 1-2\alpha & \alpha \\ 0 & 0 & 0 & ... & \alpha & 1 \end{bmatrix} \begin{Bmatrix} v_0^{i} \\ v_1^{i} \\ v_2^{i} \\ . \\ . \\ v_{n-1}^{i} \\ v_n^{i} \end{Bmatrix} \tag{9}$$

In eq. (9), the matrix is referred to as $\boldsymbol{A}$ in further discussions. Thus, to simulate the evolution of the state, the above operation has to be performed. The procedure is summarized below,

$$|\psi^i\rangle = \frac{\sum_{j=0}^{n} v(x_j^i)\,|j\rangle}{C} \tag{10}$$

$$|\psi^0\rangle = \sum_{j=0}^{n} (f(x_j) - (x_j - x_0).\frac{C_2 - C_1}{x_n - x_0} - C_1)\,|j\rangle)/C^* \tag{11}$$

$$|\psi^{i+1}\rangle = \boldsymbol{A}\,|\psi^i\rangle \tag{12}$$

Through this method, we would be able to advance the solution state through time stepping, provided we are able to simulate the matrix $\boldsymbol{A}$ and also the initial state as per eqs. (6c) and (11).

3.4. Preperation of initial state

To simulate the initial state as per eq. (11), we adopt the procedure based on quantum networking (Kaye and Mosca, 2004). In which we assume the existence of an oracle which encodes the conditional probability of a qubit. It is based on the following concept,

$$p(x_1 \cap x_2 \cap x_3..x_n) = p(x_1)p(x_2/x_1)p(x_3/(x_2,x_1))...P(x_n/(x_1,x_2,x_3...x_{n-1})), \tag{13}$$

where $p(a \cap b)$ denotes the probability of an event a given b has occurred. Assuming a quantum register exists which is capable of encoding the conditional probability, it is possible to generate any arbitrary quantum state. The information is encoded into the system with a controlled operation of the register. The conditional probabilities are encoded by the quantum register in the following way,

$$|x_1\rangle\,|x_2\rangle\,...\,|x_{k-1}\rangle\,|0\rangle \xrightarrow{c-U^{\psi}_{x_1,x_2,...x_{k-1},0}} |x_1\rangle\,|x_2\rangle\,...\,|x_{k-1}\rangle\,(\frac{p_{x_1,x_2,...x_{k-1},0}}{p_{x1,x2,...x_{k-1}}}\,|0\rangle + \frac{p_{x_1,x_2,...x_{k-1},1}}{p_{x_1,x_2,...x_{k-1}}}\,|1\rangle) \tag{14}$$

$$|\bar{\psi}\rangle\,|0\rangle\,|x_1\rangle\,...\,|x_{k-1}\rangle \xrightarrow{U_k} |\bar{\psi}\rangle\,|\omega_k\rangle\,|x_1\rangle\,...\,|x_{k-1}\rangle \tag{15}$$

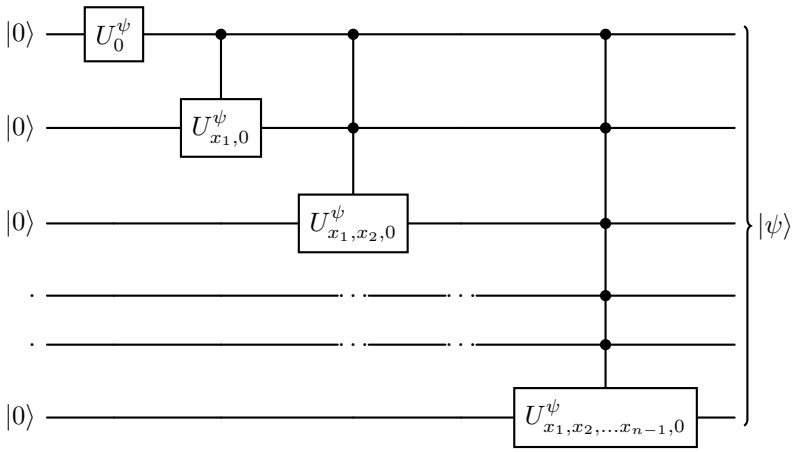

Figure 2: Quantum circuit implementation for preparing an arbitrary quantum state

$$cos^2(2\pi\omega_k) = (\frac{p_{x_1,x_2,\ldots x_{k-1},0}}{p_{x_1,x_2,\ldots x_{k-1}}})^2, \tag{16}$$

$$c - U^\psi_{x_1,x_2,\ldots x_{n-1},0}(c - U^\psi_{x_1,x_2,\ldots x_{n-2},0}(\ldots c - U^\psi_{x_1,0}(U_0^\psi |0\rangle)|0\rangle \ldots)|0\rangle)|0\rangle = |\psi\rangle\,. \tag{17}$$

The circuit implementation for preparing an arbitrary state using this method is presented in fig. 2. Using this procedure, the initial state (as per eq. (11)) can be prepared.

The preparation of an arbitrary 2-qubit state is explained to elucidate the model further. let us consider the synthesis of qubit state $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, where α, β, γ and δ are constants and satisfy the normalization constraint (i.e. $\alpha^2 + \beta^2 + \gamma^2 + \delta^2 = 1$). The following steps are to be followed to prepare this state, starting from the state $|00\rangle$,

$$|0\rangle|0\rangle \xrightarrow{U_0^\psi} (p_{x_1=0}|00\rangle + p_{x_1=1}|10\rangle) \xrightarrow{c-U^\psi_{x_1,0}} (\frac{p_{1,0}}{p_1}|10\rangle + \frac{p_{1,1}}{p_1}|11\rangle)p_{x_1=0} + (\frac{p_{0,0}}{p_0}|00\rangle + \frac{p_{0,1}}{p_0}|01\rangle)p_{x_1=0}. \tag{18}$$

In the above equation, we note that $\frac{p_{1,0}}{p_1} + \frac{p_{1,1}}{p_1} = 1$, $\frac{p_{0,0}}{p_0} + \frac{p_{0,1}}{p_0} = 1$ and $p_{x_1=0} + p_{x_1=1}$, thus we have three dependent variables in the final state so appropriate values can be adopted for the probability states to transform the qubit into any arbitrary state. Notably, here, as the probabilities are defined as positive, the states can only have positive values, but rotation operation can be performed to incorporate negative amplitudes.

3.5. FDM Matrix decomposition

It is necessary to break down the matrix into simpler matrices that can be simulated through quantum circuit implementation. For our case, as we have a tridiagonal matrix, it can be broken down into the

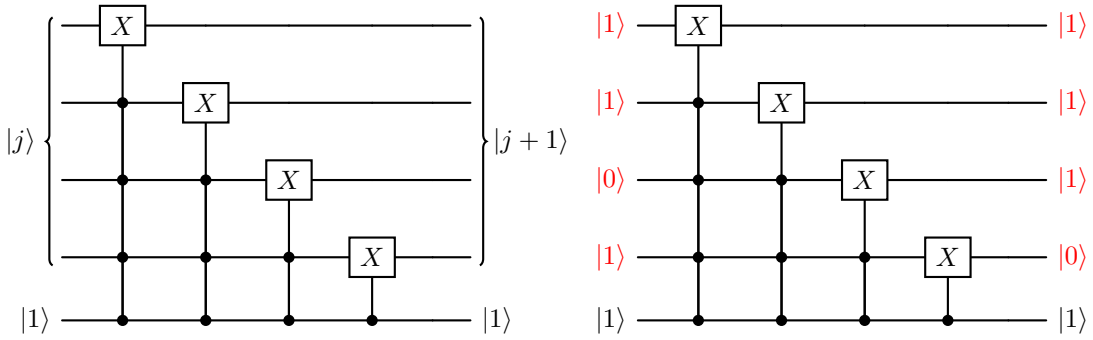

Figure 3: Quantum circuit implementation of matrix operation A_2; Example depicting shifting of basis state from $|13\rangle$ to $|14\rangle$

following forms,

$$\begin{bmatrix} 1 & 0 & 0 & \dots & . & . & 0 & 0 \\ \alpha & 1-2\alpha & \alpha & \dots & . & . & 0 & 0 \\ 0 & \alpha & 1-2\alpha & \dots & . & . & 0 & 0 \\ . & . & . & \dots & . & . & . & . \\ . & . & . & \dots & . & . & . & . \\ 0 & 0 & 0 & \dots & . & . & 0 & 1 \end{bmatrix}_{N\times N} \simeq (1-2\alpha)\boldsymbol{I}_{N\times N} + \alpha \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ . & . & . & . & \\ . & . & . & . & . \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{N\times N} + \alpha \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ . & . & . & . & \\ . & . & . & . & . \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{N\times N}, \tag{19}$$

the matrices are referred to as $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ and $\boldsymbol{A_3}$ respectively. Upon keen observation, we can note that the matrix decomposition is not exact in 4 instances, which are highlighted in red in the above equation. But as we know, those states corresponding to these rows are given by boundary conditions and are always zero in the modified temperature formulation. We can always correct these states after the time-stepping operation.

As the matrix $\boldsymbol{A_1}$ is the identity matrix, it does not require a gate to perform the operation; as for the other matrix, we need separate gates to implement it. For the matrix $\boldsymbol{A_2}$, we realize that it performs the operation of transferring the amplitude of the basis state $|j\rangle$ to the basis state $|j+1\rangle$, except the last basis state, whose amplitude is set as zero. Similarly, $\boldsymbol{A_3}$ performs the reverse operation, replacing the amplitude of basis state $|j+1\rangle$ with that of basis state $|j\rangle$, except the first basis state, which is set as zero. The decomposition of $\boldsymbol{A_2}$ is possible through the quantum circuit presented in fig. 3. In the figure, an example of shifting of basis state from $|13\rangle$ to $|14\rangle$ is also presented. As for the quantum circuit implementation of $\boldsymbol{A_3}$, we can use the same circuit used for simulating $\boldsymbol{A_2}$, but in the flipped orientation so that the state can be advanced from state $|j+1\rangle$ to state $|j\rangle$.

3.6. Quantum circuit implementation for solving 1-D transient heat equation

An n qubit system has 2^n basis states, out of which we require only the first N basis states for encoding our discretized geometric domain. The first n qubit states are prepared as per eq. (11) using the quantum

networking procedure explained earlier, and the remaining states are set with an amplitude of 0. The total number of qubits required to represent the initial state is $N == 2^{[log_2(n+1)]}$. Now, as for the matrix A, we know that the matrix is an N-dimensional square matrix now implementing the approximate decomposition of A would lead to an additional error, excluding the four errors already mentioned; there will be an error in the amplitude of the basis state $|n+1\rangle$. As we know that the amplitude of the state is always zero, so we can correct it following the matrix operation. The quantum circuit implementation of the finite difference scheme is presented in fig. 4. The steps involved are discussed below, The initial state is as follows,

$$|\beta_0\rangle = |\psi^i\rangle \otimes |00\rangle \,. \tag{20}$$

The gate denoted by S is a gate meant to encode information regarding the coefficients of the matrices in eq. (19). The operator converts the state $|00\rangle$ into the following form

$$\boldsymbol{S}\,|00\rangle = \frac{((1-2\alpha)\,|00\rangle + \alpha\,|10\rangle + \alpha\,|01\rangle)}{D}, \tag{21}$$

where D is a normalization coefficient. Thus following the rotation gate, the qubit is in the following state,

$$|\beta_1\rangle = \boldsymbol{I} \otimes \boldsymbol{S}\,|\beta_0\rangle = |\psi^i\rangle \otimes \frac{((1-2\alpha)\,|00\rangle + \alpha\,|10\rangle + \alpha\,|01\rangle)}{D}. \tag{22}$$

Following the controlled operation of $\boldsymbol{A_1}$ with controls of qubit state $|00\rangle$ leads to the following state,

$$|\beta_2\rangle = \frac{\big((1-2\alpha)\boldsymbol{A_1}\,|\psi^i\rangle \otimes |00\rangle + \alpha\boldsymbol{I}\,|\psi^i\rangle \otimes |10\rangle + \alpha\boldsymbol{I}\,|\psi^i\rangle \otimes |01\rangle\big)}{D}. \tag{23}$$

Similarly following the controlled operation of gate $\boldsymbol{A_2}$ and $\boldsymbol{A_2}$ corresponding to qubit states $|10\rangle$ and $|01\rangle$ respectively leads to the following states in order

$$|\beta_3\rangle = \frac{\big((1-2\alpha)\boldsymbol{A_1}\,|\psi^i\rangle \otimes |00\rangle + \alpha\boldsymbol{A_2}\,|\psi^i\rangle \otimes |10\rangle + \alpha\boldsymbol{I}\,|\psi^i\rangle \otimes |01\rangle\big)}{D}, \tag{24}$$

$$|\beta_4\rangle = \frac{\big((1-2\alpha)\boldsymbol{A_1}\,|\psi^i\rangle \otimes |00\rangle + \alpha\boldsymbol{A_2}\,|\psi^i\rangle \otimes |10\rangle + \alpha\boldsymbol{A_3}\,|\psi^i\rangle \otimes |01\rangle\big)}{D}. \tag{25}$$

Following this, the Hadamard gates are applied on the ancillary qubits leading to the state,

$$|\beta_5\rangle = ((1-2\alpha)\boldsymbol{A_1}\,|\psi^i\rangle \otimes (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \tag{26}$$

$$+ \alpha\boldsymbol{A_2}\,|\psi^i\rangle \otimes (|00\rangle + |01\rangle - |10\rangle - |11\rangle) \tag{27}$$

$$+ \alpha\boldsymbol{A_3}\,|\psi^i\rangle \otimes (|00\rangle - |01\rangle + |10\rangle - |11\rangle))/D^*. \tag{28}$$

The measurements are performed on the ancillary qubits when the measurements correspond to state $|00\rangle$. The state evolves into the following form (the ancillary qubits are dropped in the equation),

$$|\beta_6\rangle = \frac{((1-2\alpha)\boldsymbol{A_1} + \alpha\boldsymbol{A_2} + \alpha\boldsymbol{A_3})\,|\psi^i\rangle}{D} = \boldsymbol{A}\,|\psi^i\rangle \,. \tag{29}$$

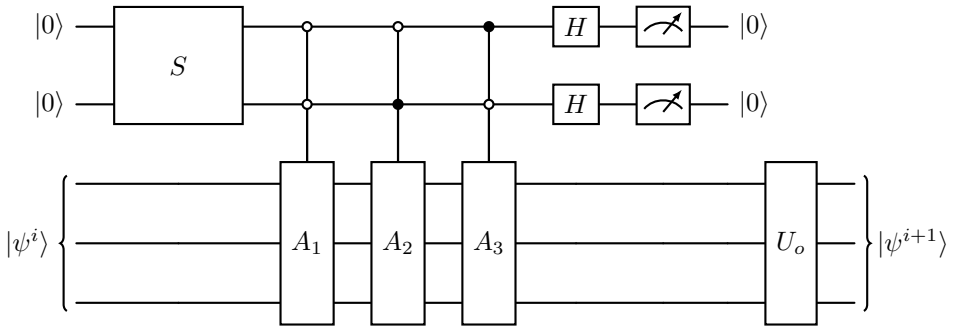

Figure 4: Quantum circuit implementation of the finite difference scheme

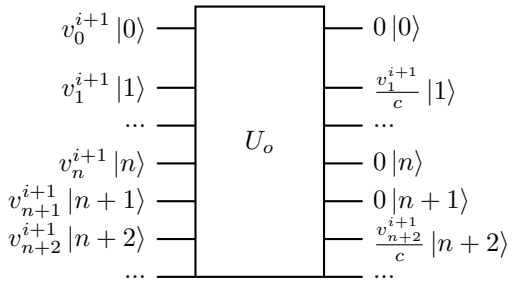

Figure 5: Illustration of the error correction gate

As discussed earlier, the solution has errors in three bases corresponding to the basis state $|0\rangle$, $|n\rangle$ and $|n+1\rangle$. As explained earlier, the amplitude value of these three basis states must be 0. The follwoing error correction gate can be implemented to perform this operation.

$$|\psi^{i+1}\rangle \xrightarrow{U_0} \frac{(\boldsymbol{I} - \sum_{p\in H} |p\rangle\langle p|)}{C} |\psi^{i+1}\rangle \,; H = \{0,\ n,\ n+1\}, \tag{30}$$

where C is the normalization coefficient. The existence of such a gate to perform this operation is assumed. It uniformly scales the remaining amplitudes so that the relative ratios dictating the probabilities or temperature are unaffected. An illustration of the action of the gate is shown in fig. 5. It is noteworthy that the operation of the error correction gate cannot be mimicked by the amplitude amplification algorithm as they do not uniformly scale the amplitude of the non-selected states, whereas uniform scaling is required for our operation. Following the error correction operation, we end up with the state with amplitudes encoding temperature corresponding to the discrete-time interval.

3.7. Extension to other differential equations

As the finite difference scheme can be performed for all differential equations, the sparsity of the system may vary; however, n-diagonal matrices can be encoded in a similar procedure by decomposition. This

method is general and can be applied to solve all problems. The drawback of this particular model is that it is not efficient, similar to the classical cases where multistep methods like Runge-kutta schemes or spectral methods are more efficient; in the quantum computation, there exist more efficient methods as explained in section 2.

4. Comparing classical and quantum algorithms

In this section, we provide a brief picture of the complexity of the best-known classical and quantum algorithms for solving general differential equations, Apart from particulars involving the structure of the problem, the main parameters needed to solve the differential equation numerically include the dimension of the system (N) corresponding to the number of elements in a finite element mesh or the order of a system, the timestep (Δt) employed and the desired range of error (ϵ). The time complexity for numerically solving differential equations is presented in table 1. Notably, there is an exponential improvement in terms of time complexity with respect to the dimension of the system. Thus far, the authors are not aware of any algorithm which has showcased any exponential improvement in the error parameter. As for the timestep, an upper bound exists based on the timestep employed. The best-known methods for solving differential equations numerically are based on Hamiltonian evolution methods, but as a general, Hamiltonian can't be evolved in sub-linear time as per the no-fast-forwarding theorem (Childs and Kothari (2010)). No algorithm exists that can simulate every differential equation in sub-linear times for a general differential equation. The evolution can be fast-tracked provided we inscribe the mathematical structure of the particular differential equation, but this is impossible for a general differential equation. To offer some intuition regarding the no fast forwarding theorem, we can consider a dynamical system (e.g. a 1-DoF pendulum); it is required to calculate the position at a given time, provided knowledge about the behaviour of the pendulum, the response can be estimated using analytical expressions but if no information about the behaviour of the pendulum can be inferred, it can only be observed by simulating the pendulum till the particular time and measuring it. In a similar sense, for a problem where the mathematical structure is not specially encoded within the algorithm (which would not be possible in an algorithm meant for all differential equations in general), no-fast forwarding of computational time complexity concerning the time step parameter is possible.

4.1. Non-linear systems

Based on the exponential improvement in the time complexity for simulating differential equations, quantum algorithms can be levied upon to solve nonlinear differential equations where the improvement can be practically realized (Liu, 2022). To elucidate it, let us consider the following problem,

$$\dot{x} = f(x), \tag{31}$$

Table 1: Comparing complexity of best-known classical and quantum algorithms

Most efficient classical algorithm (known thus far)	Most efficient quantum algorithm (known thus far)
∼O(poly(N))	∼ O(poly(log(N)))
∼O(poly(log($1/\epsilon$)))	∼O(poly(log($1/\epsilon$)))
∼O(poly(Δt)	∼O(poly(Δt)

where f(x) is a nonlinear and C_∞ function of x, thus Taylor series expansion of such a function is possible, transforming the equation into the following form,

$$\dot{x} = a_0 + a_1 x + a_2 x^2 ... \tag{32}$$

Let us consider the case where $a_{i\neq 2} = 0$ and $a_2 = 1$. We can use the Carlemann linearization procedure to express the nonlinear ODE as a system of first-order linear ODEs, as shown below,

$$\dot{x} = x^2 \tag{33}$$

$$\dot{y_i} = i y_{i+1} \quad \forall\ i \geq 0 \tag{34}$$

$$\{\dot{y}_\infty\} = \boldsymbol{A}\{y_\infty\}, \tag{35}$$

where A is a constant matrix and $y_i = x^i$. Now, in both classical and quantum computational procedures, we can truncate the infinite-order matrix to a finite dimension depending on the desired error range. Now, in the case of classical computation, the time complexity scales as a polynomial function of the dimension of the system, whereas it scales as a polynomial function of the logarithm of the dimension of the system. It can be noticed that such a system can be formed for any continuously smooth and analytic function in eq. (31).

5. Conclusions

In this paper, we provided a brief review of the quantum algorithms used to solve differential equations and their classical counterparts. We explained a finite different scheme-based quantum algorithm used to solve the heat equation. We also provide a comparison between the time complexity associated with classical and quantum algorithms and explain potential applications in solving nonlinear differential equations.

References

1. Berry DW, Ahokas G, Cleve R, Sanders BC. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics* 2006;270(2):359–371. URL: http://dx.doi.org/10.1007/s00220-006-0150-x. doi:10.1007/s00220-006-0150-x.

2. Harrow AW, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. *Phys Rev Lett* 2009;103:150502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502. doi:10.1103/PhysRevLett.103.150502.
3. Berry DW. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical* 2014;47(10):105301. URL: http://dx.doi.org/10.1088/1751-8113/47/10/105301. doi:10.1088/1751-8113/47/10/105301.
4. Childs AM, Liu JP. Quantum spectral methods for differential equations. *Communications in Mathematical Physics* 2020;375(2):1427–57. URL: https://doi.org/10.1007/s00220-020-03699-z. doi:10.1007/s00220-020-03699-z.
5. Montanaro A, Pallister S. Quantum algorithms and the finite element method. *Phys Rev A* 2016;93:032324. URL: https://link.aps.org/doi/10.1103/PhysRevA.93.032324. doi:10.1103/PhysRevA.93.032324.
6. Liu JP. Ph.D. thesis; 2022.
7. Wei SJ, Wei C, Lv P, Shao C, Gao P, Zhou Z, Li K, Xin T, Long GL. A quantum algorithm for heat conduction with symmetrization. *Science Bulletin* 2023;68(5):494–502. URL: https://www.sciencedirect.com/science/article/pii/S2095927323001147. doi:https://doi.org/10.1016/j.scib.2023.02.016.
8. Kaye P, Mosca M. Quantum networks for generating arbitrary quantum states. 2004. arXiv:quant-ph/0407102.
9. Childs AM, Kothari R. *Quantum Information and Computation* 2010;10(7 8). URL: http://dx.doi.org/10.26421/QIC10.7-8. doi:10.26421/qic10.7-8.