Varitional Quantum Algorithms for Non-Linear PDEs

Arunava Majumder
Indian Institute of Technology Kharagpur, Kharagpur 721302, India and
McMahon Laboratory, Cornell University, Ithaca, NY 14850

Mudassir Moosa
Department of physics and astronomy, Purdue university, USA and
McMahon Laboratory, Cornell University, Ithaca, NY 14850

Thomas Watts
McMahon Laboratory, Cornell University, Ithaca, NY 14850

Peter L. McMahon*
School of Applied and Engineering Physics, Cornell University, Ithaca, NY 14853, USA

(Collaborating with the members of the Variational-PDE sub-group)

(Dated: Wednesday 24th August, 2022)

Applications such as simulating large quantum systems or solving large-scale linear algebra problems are immensely challenging for classical computers due their extremely high computational cost. Quantum computers promise to unlock these applications, although fault-tolerant quantum computers will likely not be available for several years. Currently available quantum devices have serious constraints, including limited qubit numbers and noise processes that limit circuit depth. Variational Quantum Algorithms (VQAs), which employ a classical optimizer to train a parametrized quantum circuit, have emerged as a leading strategy to address these constraints We develop an efficient variational quantum algorithm for solving nonlinear partial differential equations (PDEs) with periodic boundary conditions based on the theoretical groundwork developed by Lubasch et al. in their paper Variational quantum algorithms for nonlinear problems [1]. This algorithm involves recasting the problem of solving a PDE as a optimization problem that is tackled by a classical computer and quantum computer in tandem. We demonstrate the mechanism and efficiency of this approach to solving nonlinear PDEs by considering the problem of solving the inviscid Burger's Equation $u_t = -uu_x$ and the viscid Burger's Equation $u_t = \nu u_{xx} - uu_x$ where ν is the diffusion coefficient.

CONTENTS

* pmcmahon@cornell.edu

I. INTRODUCTION

Nonlinear partial differential equations are notoriously hard to solve and appear in mathematical models for everything from the complex flows of fluids to the rapid fluctuations of the prices of financial derivatives. Traditional computational methods for solving PDEs generally involve discretizing the functions and approximating derivatives using finite-difference methods [2]. However, these methods are limited by numerical instability, high error propagation, and computational intractability [3]. Quantum computing offers exponential data compression, which lends itself to reducing the complexity of classically intractable problems. A subclass of quantum algorithms, known as variational algorithms, have become popular recently due to their power, and ability to be executed on Near-term Intermediate Scale Quantum (NISQ) devices [4]. Variational algorithms consist of parameterizing a quantum circuit to represent candidate solutions to a problem, and utilizing a classical subroutine to optimize the parameters. As such, they are hybrid quantum/classical algorithms, taking advantage of the exponential data compression of quantum states while not requiring as many qubits as strictly quantum algorithms. Recent quantum computing literature have attempted to apply variational methods to the problem of solving PDEs [1] [5] [6] [7]. Here, we present a robust variational algorithm along with a powerful ansatz capable of finding solutions to both linear and non-linear PDEs. We further show experimental results of the algorithm applied to solving the inviscid Burger's equation and the viscid Burger's equation for a variety of initial conditions.

A. The General 1-D Partial Differential Equation

Consider the problem of solving the following partial differential equation in one dimension with periodic

boundary conditions on the interval $[0, 1]$.

$$\frac{\partial u}{\partial t} = \hat{O}u \tag{1}$$

$u(0, t) = u(1, t)$ (Boundary Condition)

$u(x, 0) = u_0(x)$ (Initial Condition)

where $\hat{O}$ is a potentially non-linear differential operator consisting of partial derivatives with respect to the x variable (e.g. $u\frac{\partial u}{\partial x}, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \cdots$).

B. Classical Finite Difference Methods

One classical approach to solve this problem numerically is to make use of finite difference approximations for the various partial derivatives with respect to the spatial variable x that appear in $\hat{O}$. In order to define these approximations, we must first partition the interval $[0, 1]$ into a discrete mesh of N equally spaced gird points $x_k = \frac{k}{N}$ where $0 \leq k \leq N - 1$. Define the vector $U(t) \in \mathbb{R}^N$ that contains the function $u(x, t)$ evaluated at each grid point in our mesh x_k.

$$U(t) = \begin{pmatrix} u(x_0, t) \\ u(x_1, t) \\ \vdots \\ u(x_{N-1}, t) \end{pmatrix}$$

Recall that we are solving Eq. (1) with periodic boundary conditions, thus $u(x_0, t) = u(x_N, t)$, so no need to consider x_N grid point. Then, we use central differences to approximate the spacial derivatives at grid point x_k.

$$\frac{\partial u}{\partial x} = \frac{u(x_{k+1}, t) - u(x_{k-1}, t)}{2\Delta x} + \mathcal{O}(\Delta x^2) \tag{3}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x_{k+1}, t) - 2u(x_k, t) + u(x_{k-1}, t)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{4}$$

From these finite difference approximations, we can derive a matrix representation of the differential operator $\hat{O} \in \mathbb{R}^{N \times N}$ (classical matrix representation). Once the appropriate finite difference approximations are chosen, we have a choice between two approaches to solving to evolving Eq. (1) in time by a discrete time step τ.

C. Variational quantum algorithm

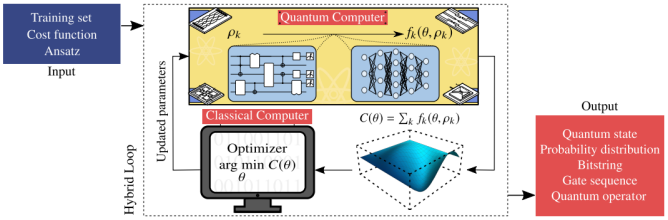


FIG. 1. **Schematic diagram of a Variational Quantum Algorithm (VQA).** The inputs to a VQA are: a cost function $C(\theta)$ which encodes the solution to the problem, an ansatz whose parameters are trained to minimize the cost, and (possibly) a set of training data used during the optimization. At each iteration of the loop one employs a quantum computer to efficiently estimate the cost (or its gradients). This information is fed into a classical computer that leverages the power of optimizers to navigate the cost landscape and solve the optimization problem

1. *Forwards Euler's Method*

The first method for solving Eq. (1) we have at our disposal is Forwards Euler's method. Forwards Euler's method gives us the following prescription for obtaining $U(t + \tau)$.

$$U(t + \tau) = (\mathbf{1} + \tau\hat{O})U(t) + \mathcal{O}(\tau^2) \tag{5}$$

This method is an explicit finite difference method that is known to be numerically stable and convergent when $\tau = \mathcal{O}(N^{-2})$. The numerical error per times step as calculated using Forward Euler's method is $\mathcal{O}(N^{-2}) + \mathcal{O}(\tau)$. (Discuss local and global truncation error)

Eq. (5) can be rephrased as the problem of minimizing a cost function, where

$$\mathcal{C}(U(t + \tau)) = \|U(t + \tau) - (\mathbf{1} + \tau\hat{O})U(t)\|^2 \tag{6}$$

2. Backwards Euler's Method

The second method for solving Eq .(1) we have at our disposal is Backwards Euler's method. Backwards Euler's method gives us the following prescription for obtaining $U(t+\tau)$.

$$(\mathbf{1} - \tau\hat{O})U(t+\tau) = U(t) + \mathcal{O}(\tau^2) \tag{7}$$

This method is an implicit finite difference method that is known to be always numerically stable and convergent but is more computationally intensive given that it requires solving a linear system of equations for $U(t+\tau)$. The numerical error per times step as calculated using Forward Euler's method is $O(N^{-2}) + O(\tau)$. (Discuss local and global truncation error)

Similarly, Eq. (7) can be rephrased as the problem of minimizing a cost function $\mathcal{C}$, where

$$\mathcal{C}(U(t+\tau)) = \|(\mathbf{1} - \tau\hat{O})U(t+\tau) - U(t)\|^2 \tag{8}$$

II. QUANTUM FINITE DIFFERENCE METHODS

A. Quantizing the Solution to PDEs

In order to solve our PDEs using a variational algorithm, we must quantize our general form of a 1-D PDE, namely Eq. (1), which we do as such:

$$\frac{\partial}{\partial t}|u(t)\rangle = \hat{O}|u(t)\rangle \tag{9}$$

where $\hat{O}$ is now a quantum operator acting on our Hilbert space, and $|u(t)\rangle = \sum_{k=0}^{N-1} u(x_k, t)|k\rangle$ corresponds to our function represented in Hilbert space by the computational basis. The central idea of variational algorithms is to represent our candidate solution function $|u(t)\rangle$ by a parameterized gate $\hat{U}(\boldsymbol{\lambda})$, known as the Ansatz, which acts on the quantum register to produce a quantum state $|\psi(\boldsymbol{\lambda})\rangle$ whose amplitudes encode the solution to a given problem for a specific set of parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots)$. The goal of the algorithm is then to find those specific set of parameters $\boldsymbol{\lambda}$ which correspond to our solution. In the case of our algorithm, the $\hat{U}(\boldsymbol{\lambda})$ allow us to store the solution $u(x,t)$ as a quantum state whose amplitudes correspond to the solution $u(x,t)$ evaluated at each of the grid points x_k of our mesh

$$|u(t)\rangle = \lambda_0\hat{U}(\boldsymbol{\lambda})|0\rangle^{\otimes n} = \lambda_0|\psi(\boldsymbol{\lambda})\rangle$$

Notice that we must introduce a real scalar $\lambda_0 \in \mathbb{R}$ given that the inner product $\langle u(t)|u(t)\rangle \neq 1$ in general. We consider this scalar λ_0 to be an additional variational parameter. Furthermore, the number of grid points in our mesh now directly corresponds to the number of qubits n we use in our quantum register $|\mathbf{0}\rangle$, that is to say $N = 2^n$. This is one of the major advantages of this algorithm given that our mesh become exponentially more fine as we increase the number of qubits we use in our computation, causing our spatial error to become exponentially small $\mathcal{O}(2^{-n})$. In order to solve Eq . (10) using Forwards or Backwards Euler's Method, we must quantize the cost functions given in Eq .(6) and Eq .(8) respectively.

B. Quantizing the Cost Function

For the sake of brevity, we consider only the Forwards Euler's Method here, though the same procedure can be repeated with the Backwards Euler's Method (see supplementary document). Suppose we knew the specific set of parameters $\tilde{\lambda}_0, \tilde{\boldsymbol{\lambda}}$ such that $|u(t)\rangle = \tilde{\lambda}_0\hat{U}(\tilde{\boldsymbol{\lambda}})|\mathbf{0}\rangle = \tilde{\lambda}_0|\tilde{\psi}\rangle$ and we wished to find the next set of parameters $\lambda_0, \boldsymbol{\lambda}$ for which $|u(t+\tau)\rangle = \lambda_0\hat{U}(\boldsymbol{\lambda})|\mathbf{0}\rangle = \lambda_0|\psi(\boldsymbol{\lambda})\rangle$. In order to find the next set of parameters $\lambda_0, \boldsymbol{\lambda}$, we can quantize Eq .(6), and rewrite it in terms of an inner product on Hilbert space

$$\begin{aligned}
\mathcal{C}(\lambda_0, \boldsymbol{\lambda}) &= \|u(t+\tau) - (\mathbf{1} + \tau\hat{O})u(t)\|^2 \\
&= \langle u(t+\tau) - (\mathbf{1} + \tau\hat{O})u(t) | u(t+\tau) - (\mathbf{1} + \tau\hat{O})u(t)\rangle \\
&= \langle \lambda_0\psi(\boldsymbol{\lambda}) - (\mathbf{1} + \tau\hat{O})\tilde{\lambda}_0\tilde{\psi} | \lambda_0\psi(\boldsymbol{\lambda}) - (\mathbf{1} + \tau\hat{O})\tilde{\lambda}_0\tilde{\psi}\rangle \\
&= \lambda_0^2 - 2\lambda_0\tilde{\lambda}_0\mathfrak{Re}\{\langle\psi(\boldsymbol{\lambda})|\mathbf{1} + \tau\hat{O}|\tilde{\psi}\rangle\} + \text{const.}
\end{aligned}$$

by direct expansion, where $\hat{O}$ is now an operator on our Hilbert space. We can therefore compute our cost function efficiently on a quantum computer by calculating $\langle\psi(\boldsymbol{\lambda})|\mathbf{1}+\tau\hat{O}|\tilde{\psi}\rangle$ with quantum circuits, and find the optimal parameters $(\lambda_0, \boldsymbol{\lambda})$ with a classical optimization subroutine.

C. Deriving the Operator $\hat{O}$

One detail which we ignored in the previous section is that we must construct the (quantum) differential operator $\hat{O}$. To do so, we must first quantize the finite difference approximation of the spatial derivative given in Eq. (3) and Eq. (4).

1. The Adder Operator

In order to quantize spatial derivatives, we introduce what Lubasch et. al. refer to as the Adder operator $\hat{A}$ [1]. The purpose of the Adder operator is to permute the entries of the state vector $|u(t)\rangle$ in the following way

$$\hat{A}\,|u(t)\rangle = \hat{A}\begin{pmatrix} u(x_0,t) \\ u(x_1,t) \\ \vdots \\ u(x_{N-1},t)\end{pmatrix} = \begin{pmatrix} u(x_1,t) \\ u(x_2,t) \\ \vdots \\ u(x_0,t)\end{pmatrix}$$

Refer to the supplementary section for the general form of the controlled adder $C\hat{A}$. Using this Adder operator, quantum finite difference operators can be derived from the classical finite difference approximations. Below we provide the quantum finite difference approximations for the first and second derivatives in the spatial variable x.

$$\hat{\nabla} = \frac{\hat{A}-\hat{A}^\dagger}{2N^{-1}}$$

$$\hat{\Delta} = \frac{\hat{A}+\hat{A}^\dagger-2\mathbf{1}}{N^{-2}}$$

2. The Diagonal Operator

The second ingredient for constructing the operator $\hat{O}$ is a diagonal operator $\hat{D}_{u(t)}$ that will allow us to represent non-linear differential operators. The matrix representation of $\hat{D}_{u(t)}$ is given by $\mathrm{diag}(u(x_0,t),\cdots,u(x_{N-1},t))$ and therefore acts on the state $|u(t)\rangle$ as follows.

$$\hat{D}_{u(t)}\,|u(t)\rangle = \sum_{k=0}^{N-1} u(x_k,t)^2\,|k\rangle = \lambda_0^2\sum_{k=0}^{N-1}\psi(x_k,t)^2\,|k\rangle$$

Notice that the operator $\hat{D}_{u(t)}$ is not unitary and thus we never construct this state explicitly. Instead, we construct the controlled diagonal operator $C\hat{D}_\psi$ which is unitary (refer to the supplementary section for how to construct $C\hat{D}_\psi$). Now we are ready to construct $\hat{O}$ for various PDEs. For the inviscid Burger's Equation $\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}$, $\hat{O} = -\tilde{\lambda}_0\hat{D}_{\tilde{\psi}}\hat{\nabla}$. For the viscid Burger's Equation $\frac{\partial u}{\partial t} = \nu\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x}$, we simply combine the $\hat{O}$ for the heat equation and the inviscid Burger's equation to obtain $\hat{O} = \nu\hat{\Delta} - \tilde{\lambda}_0\hat{D}_{\tilde{\psi}}\hat{\nabla}$.

D. Adder circuit

The Adder circuit is crucial to the algorithm, since this is what we use to get the Euler's derivative. There have been a few proposed designs of the adder circuit, and shown in Fig. 2 is an adder circuit design proposed by Lubasch, et al. [1] for a 6-qubit circuit with 4 ancilla qubits.

E. Computing expectation values in the cost function

In this section, we utilize low-depth quantum circuitry developed by Lubasch et. al. to compute the expectation $\langle\psi(\boldsymbol{\lambda})|\mathbf{1}+\tau\hat{O}|\tilde{\psi}\rangle$ that appears in the cost function $\mathcal{C}(\lambda_0,\boldsymbol{\lambda})$ by way of the Hadamard test [1]. Note that for each expectation we must construct two separate circuits: one for computing the real part $\mathfrak{Re}\{\langle\psi(\boldsymbol{\lambda})|\hat{O}|\tilde{\psi}\rangle\}$ and an-

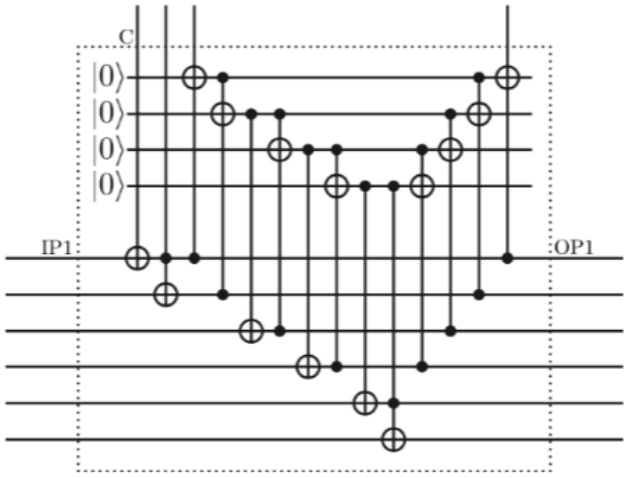
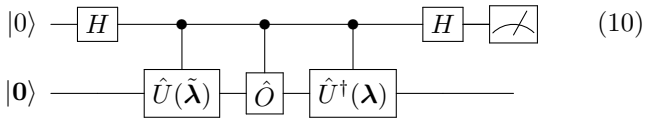
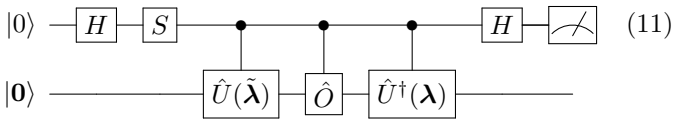


FIG. 2. A circuit for implementing the adder operator for 6 qubits

other for computing the imaginary part $\mathfrak{Im}\{\langle\psi(\boldsymbol{\lambda})|\hat{O}|\tilde{\psi}\rangle\}$. The following two circuits allow us to do so.



(10)



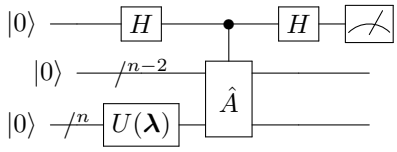
(11)

We use circuit (11) to compute $\mathfrak{Re}\{\langle\psi(\boldsymbol{\lambda})|\hat{O}|\tilde{\psi}\rangle\}$ and circuit (12) to compute $\mathfrak{Im}\{\langle\psi(\boldsymbol{\lambda})|\hat{O}|\tilde{\psi}\rangle\}$. Note that $\mathfrak{Im}\{\langle\psi(\boldsymbol{\lambda})|\hat{O}|\tilde{\psi}\rangle\}$ is required for computing the gradient of the cost function $\nabla\mathcal{C}(\lambda_0, \boldsymbol{\lambda})$. we have that $O \equiv \partial_{xx}$. We denote the corresponding finite approximation $\hat{\Delta} = \hat{O}$ with reference to the Laplacian(defined in eq.10). We first turn our attention to computing $\langle\psi(\boldsymbol{\lambda})|\hat{\Delta}|\psi(\boldsymbol{\lambda})\rangle$ and $\langle\psi(\boldsymbol{\lambda})|\hat{\Delta}^\dagger\hat{\Delta}|\psi(\boldsymbol{\lambda})\rangle$. Expanding these expectation values in terms of $\hat{A}$ yields

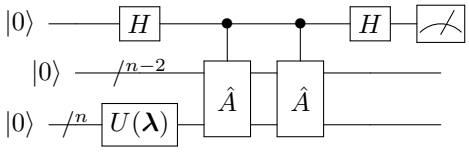
$$\langle\psi(\boldsymbol{\lambda})|\hat{\Delta}|\psi(\boldsymbol{\lambda})\rangle = h_N^{-2}(2\langle\psi(\boldsymbol{\lambda})|\hat{A}|\psi(\boldsymbol{\lambda})\rangle - 2),$$

$$\langle\psi(\boldsymbol{\lambda})|\hat{\Delta}^\dagger\hat{\Delta}|\psi(\boldsymbol{\lambda})\rangle$$
$$= h_N^{-4}(2\langle\psi(\boldsymbol{\lambda})|\hat{A}^2|\psi(\boldsymbol{\lambda})\rangle - 8\langle\psi(\boldsymbol{\lambda})|\hat{A}|\psi(\boldsymbol{\lambda})\rangle + 6).$$

Now we construct low-depth circuits for estimating $\langle\psi(\boldsymbol{\lambda})|\hat{A}^m|\psi(\boldsymbol{\lambda})\rangle$ for $m = 1, 2$. This is accomplished with the following two circuits based on the Hadamard test



$(m = 1)$



$(m = 2)$

We use circuits $(m = 1)$ and $(m = 2)$ to estimate $\langle\psi(\boldsymbol{\lambda})|\hat{A}^m|\psi(\boldsymbol{\lambda})\rangle$ by measuring the topmost *ancillary* qubit repeatedly. Let N_0 be the number of times we measure $|0\rangle$ and N_1 be the number of time we measure $|1\rangle$, then $\langle\psi(\boldsymbol{\lambda})|\hat{A}^m|\psi(\boldsymbol{\lambda})\rangle \approx \mathbb{P}[|0\rangle] - \mathbb{P}[|1\rangle] = (N_0 - N_1)/(N_0 + N_1)$. We can estimate $\langle\psi(\boldsymbol{\lambda})|\hat{A}^m|\psi(\boldsymbol{\lambda})\rangle$ with an error ϵ by measuring the ancillary qubit $O(\text{poly}(1/\epsilon))$. For each of the circuits (11) and (12), we measure the uppermost qubit m times and count the number of times we obtain zero $N_0^{(m)}$ and the number of times we obtain one $N_1^{(m)}$, then compute the expectation as $(N_1^{(m)} - N_0^{(m)})/m$.

F. Gradient of Forwards Euler's Method

While gradient-free optimization methods have been shown to be effective when optimizing the cost functions used in quantum variational algorithms [8], the cost function is best optimized by computing the gradient $\nabla\mathcal{C}(\lambda_0, \boldsymbol{\lambda})$ and using an optimization routine such as the Broyden–Fletcher–Goldfarb–Shannon algorithm (BFGS). In order to calculate the gradient of the cost function, we need to take partial derivatives of expectations $\langle\psi(\boldsymbol{\lambda})|\hat{O}|\tilde{\psi}\rangle$. Bravo-Prieto et al. provide a comprehensive discussion of how to compute derivatives of $\hat{U}(\boldsymbol{\lambda})$ (e.g. $\frac{\partial\psi(\boldsymbol{\lambda})}{\partial\lambda_i}$) in their paper Variational Quantum Linear Solver [9]. Then, we need only use the Hadamard test to compute each component of the gradient as given below.

$$\frac{\partial \mathcal{C}}{\partial \lambda_0} = 2\lambda_0 - 2\tilde{\lambda}_0 \mathfrak{Re}\{\langle\psi(\boldsymbol{\lambda})|\mathbf{1} + \tau\hat{O}|\tilde{\psi}\rangle\}$$

$$\frac{\partial \mathcal{C}}{\partial \lambda_i} = -2\lambda_0\tilde{\lambda}_0 \mathfrak{Re}\{\langle\frac{\partial\psi(\boldsymbol{\lambda})}{\partial\lambda_i}|\mathbf{1} + \tau\hat{O}|\tilde{\psi}\rangle\}$$

G. Universal layered ansatz

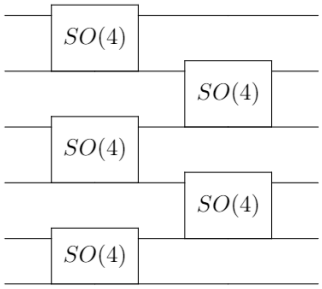
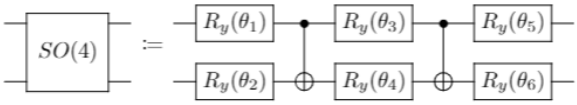


FIG. 3. Circuit shows one particular type of ansatz called universal layered ansatz for 6 qubits but it can be extended to 10 qubits as well as reduced to 4 qubit circuits. This is basically the similar one given in [1] to simulate general non-linear PDE's

Where the $SO(4)$ is of the form below



Where $R_y(\theta_i) = e^{-i\theta_i t}$

H. Results 1

We use the same ansatz given in Fig.3 to represent couple of functions using 12 qubits i.e. the above circuit is extended to 12 ansatz qubits. The proposed ansatz can approximate the following functions upto an L_2 norm of $\leq 10^-2$.

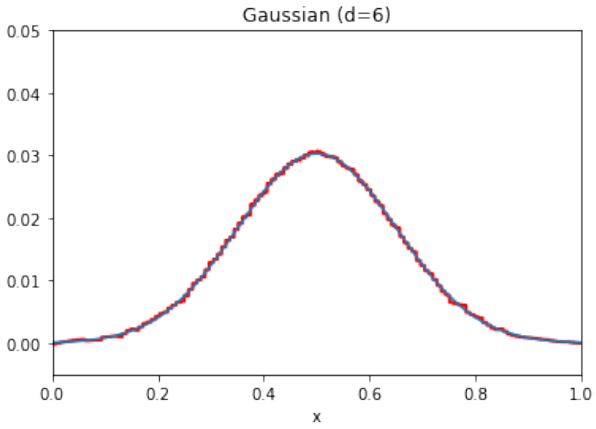


FIG. 4. Y- axis represents initial condition i.e. $u(x, t = 0)$ where the initial condition is chosen to be Gaussian with a total depth of the circuit d=6

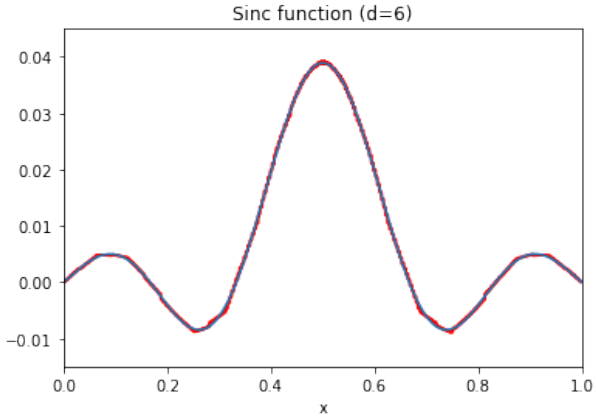


FIG. 5. Y- axis represents initial condition i.e. $u(x, t = 0)$ where the initial condition is chosen to be $sinc(x) = \frac{\sin(x)}{x}$ with a total depth of the circuit d=6

I. SIMULATION RESULTS

One practical problem in designing the quantum algorithm is finding the right ansatz design to simulate or optimize a system with the right optimizer function.
In this section, we will share some results on numerically testing the above ansatz in Fig.3 with a very specific optimizers called "Imfil" optimizer which doesn't require gradient descent to achieve the global minimum for a 4 qubit system to solve the Berger's equation with a total depth of d=4.

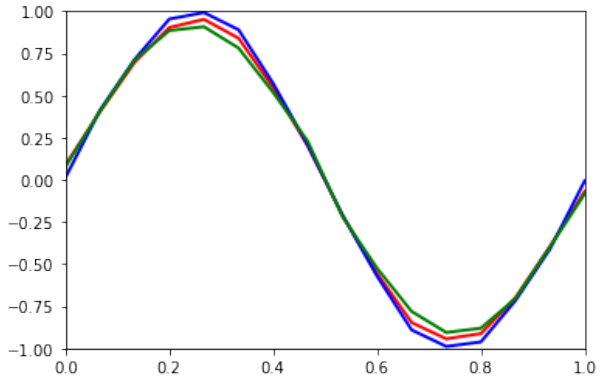


FIG. 6. Y- axis represents solution to the Berger's equation at time $t = 0$ i.e. $u(x, t = 0)$(**Blue** and **Green** for different initial guess of the parameters), The **Red** one represents the direct read-in of the initial condition $u(x, t = 0) = \sin(x)$

In Fig.6 we solve the Burger's equation using a variational quantum circuit and plot the resultant $u(x, t)$ at time $t = 0$

J. Ansatz design and Read-In

Now we want to scale our algorithm because for small number of qubits there would be no advantage but due to the hardware restrictions we have to be careful while designing a suitable ansatz to perform time evolution of the Berger's equation. One way to do this considering the below circuit.

The final and most important step of the algorithm is choosing a suitable variational ansatz $\hat{U}(\boldsymbol{\lambda})$ given in Fig.1. It is called QFT(Quantum Fourier transform) based universal layered ansatz in Fig.7. The ansatz must be chosen in such a manner so that the number of gates i.e. the amount of error must remain as minimum as possible. In the circuit the universal layer consists of SU(4) gates with depth=2, the SU(4) is defines as

Where A_1, A_2, A_3, A_4 are operators from SU(2) groups. The general SU(2) operator is defined as

$$SU(2) = R_z(\alpha)R_y(\beta)R_z(\gamma) \tag{12}$$

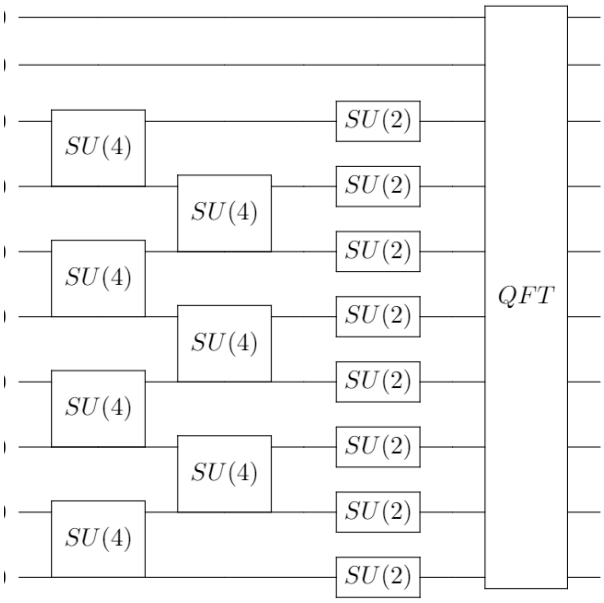


FIG. 7. Universal Layered QFT based ansatz

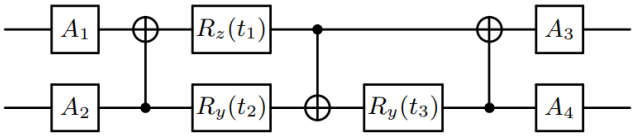


FIG. 8. A circuit for implementing a transform in SU(4)

Where α, β and γ are variational parameters, and as a total SU(4) contains 15 parameters. Note that in our universal layered ansatz the layer of SU(4) is applied to the last 8 qubits twice and the fist two are kept as ancillas, same for the SU(2) layer but applied only once. After the variational part the QFT is applied to all the qubits to get the ultimate problem specific output.
The term *read-in* refers to the determination of the initial set of parameters $(\lambda_{0,0}, \boldsymbol{\lambda}_0)$ that generate the initial condition vector $|u(0)\rangle$. Another interesting property of our variational ansatz (in Fig.1) is that it can approximate many functions with good enough accuracy.

K. Results 2

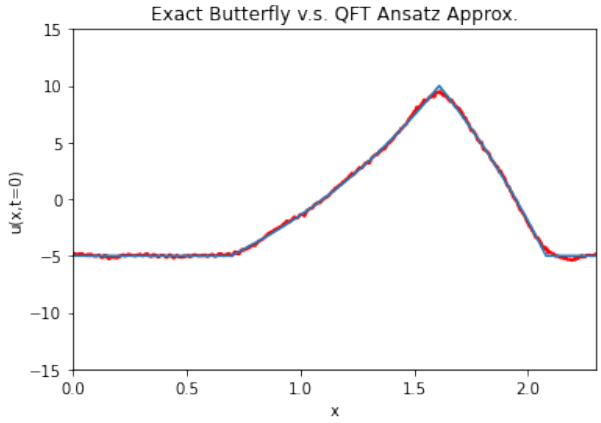


FIG. 9. Approximating Black–Scholes initial put condition.

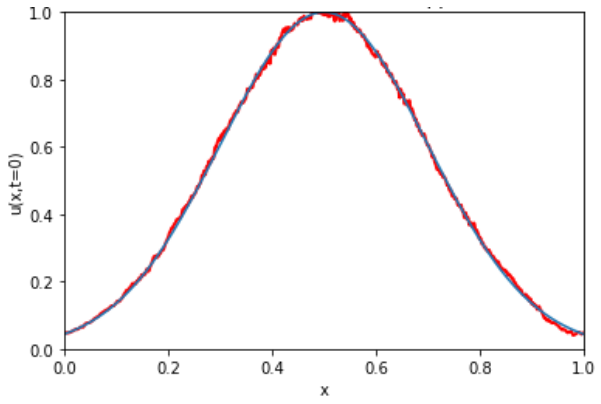


FIG. 10. Approximating Gaussian function.

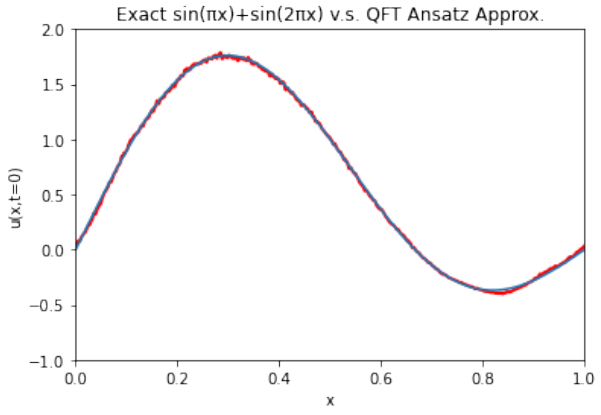


FIG. 11. This plot is for 10 qubits variaitonal circuit where we approximate the initial condition $\sin \pi x + \sin 2\pi x$.

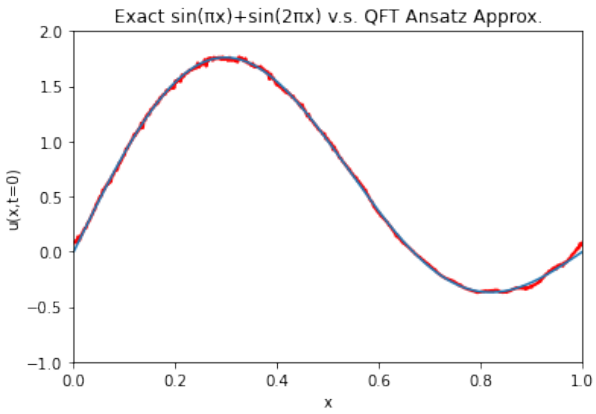


FIG. 12. This plot is for 12 qubits variaitonal circuit, the same structure given in sec.II J with two additional ancilla qubits in the beginning, where we approximate the initial condition $\sin \pi x + \sin 2\pi x$.

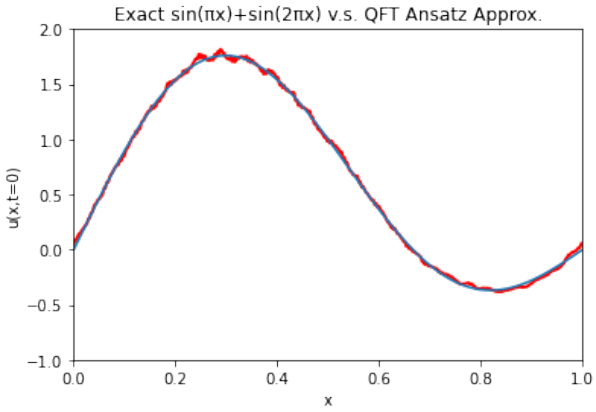


FIG. 13. This plot is for 15 qubits variaitonal circuit, the same structure given in sec.II J with 5 additional ancilla qubits in the beginning and a total of 7 ancilla qubits, where we approximate the initial condition $\sin \pi x + \sin 2\pi x$.

We use the same ansatz to approximate the above functions that are given. The proposed ansatz can approximate the following functions upto an L_2 norm of 0.3-0.4.

L. Zalka-Grover-Rudolph (ZGR) ansatz

We can derive a better variational ansatz for real functions using the ideas from Zalka [10], and Grover and Rudolph [11], to discretize non-negative probability distributions in a quantum register. In their work they

showed that probability distributions could be approximated by conditional rotations of the least significant qubits based on the state of all previous qubits. The overview of the ansatz is shown in the Fig.14

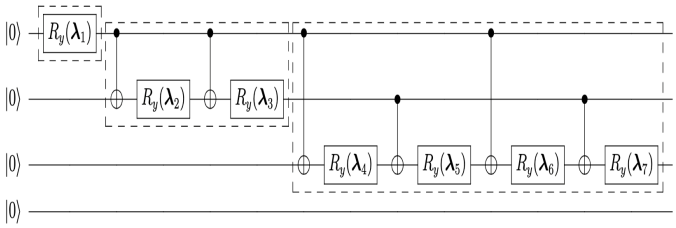


FIG. 14. ZGR ansatz

The problem with this ansatz is that the total number of parameters required for such variational ansatz is exponentiation in the number of qubits i.e. if we choose to apply this on a n qubit system then the total number of parameters would be $2^n - 1$

1. *Function representation using ZGR ansatz*

Given a function $f(x)$ defined on a domain [0, L], our goal is to (approximately) represent it as the following quantum state of n qubits.

$$|f\rangle_n = \sum_{l=0}^{2^n-1} f(x_l)|l\rangle_n, \tag{13}$$

Taking Fourier series expansion of our function $f(x)$

$$f(x) = \sum_{k=0}^{\infty}(a_k \cos(\frac{2\pi kx}{L}) + b_k \sin(\frac{2\pi kx}{L})) \tag{14}$$

where a_k and b_k are Fourier coefficients.

Now, we should approximate the function $f(x)$ by only considering the first 2^m terms in eq.14.

$$f(x) = \sum_{k=0}^{2^m-1}(a_k \cos(\frac{2\pi kx}{L}) + b_k \sin(\frac{2\pi kx}{L})) \tag{15}$$

If $f_m(x)$ is a good approximation of $f(x)$, then the state $|f_m\rangle_n$ is a good approximation of $|f\rangle_n$, where $|f_m\rangle_n$ means we are having a n qubit system but the ZGR ansatz is only applied to last m qubits and this is because to keep the number of parameters as low as possible.

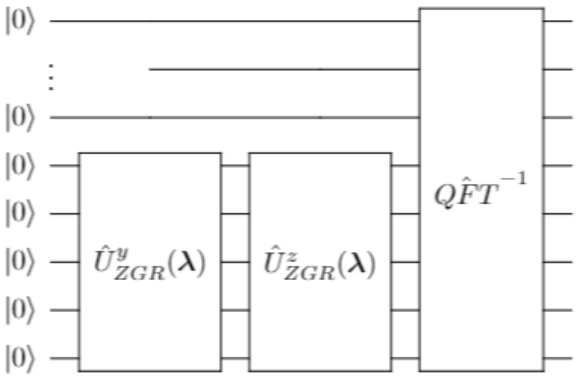


FIG. 15. The ZGR-QFT ansatz.

Where Where the $\hat{U}^{y/z}_{ZGR}(\lambda)$ is the ansatz given in Fig.14 and the parameters will be the Fourier coefficients in the eq.15 and it will be the exact representation of the function $f(X)$ but the only issue with such ansatz is that it is not hardware efficient because controlling qubits which are far apart is apparently very difficult in real experimental setup.

M. Results 3

Using the ansatz in Fig.15 we will show some results where we tried to find the exact representation of function using the above ansatz

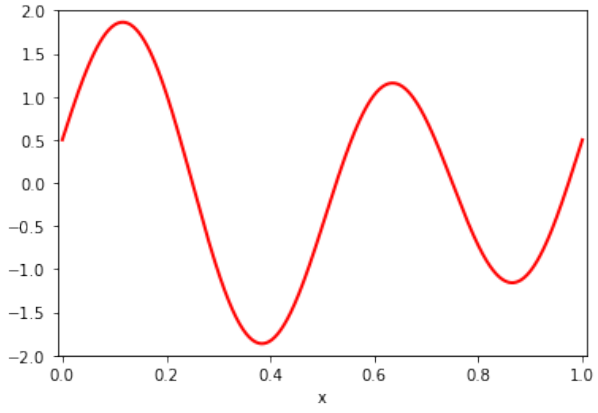


FIG. 16. Representing target $f(x) = 1.5\sin(4\pi x) + 0.5\cos(2\pi x)$

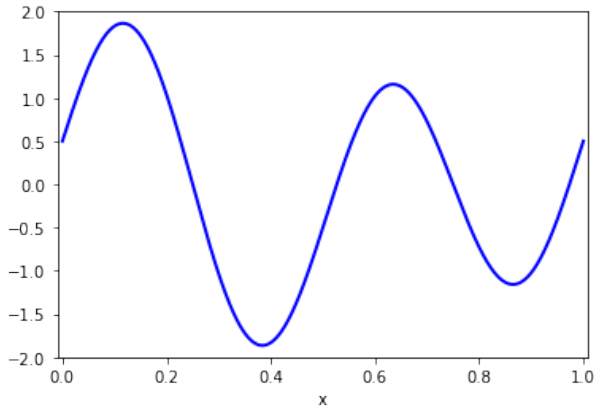


FIG. 17. Representing resultant $f(x) = 1.5\sin(4\pi x) + 0.5\cos(2\pi x)$

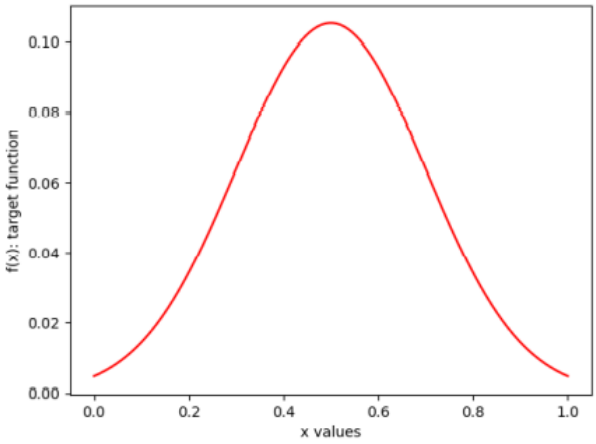


FIG. 18. Representing target Gaussian

III. CONCLUSION: NEXT STEPS

In this work, we show the efficiency of the PDE solving algorithm first introduced by Lubasch *et all* to solve the nonlinear Partial differential Equations. With ansatz for the read-in method proposed in this paper, we can efficiently read-in different initial conditions to a quantum-computer to solve even Linear Black-scholes equation as well as Berger's equation. For the next step we are trying to deal with the time evolution part. We currently have some good designs of ansatz up to 15 qubits which can produce desired output up to some L_2 norm close to zero. We will still look for more better designs to get almost zero L_2 norm.

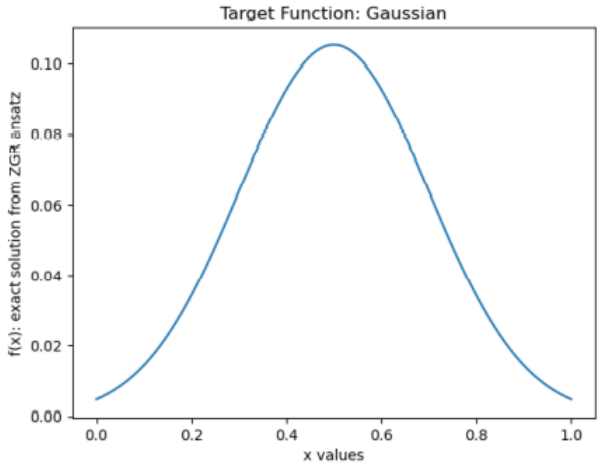


FIG. 19. Representing resultant Gaussian

ACKNOWLEDGMENTS

I would like the thank professor Peter McMahon for giving me the opportunity to work in his lab and all of the members of the Variational-PDE sub group Team for their consistent hard work.

[1] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Variational quantum algorithms for nonlinear problems, Physical Review A **101**, 10.1103/physreva.101.010301 (2020).

[2] J. Blazek, Chapter 3 - principles of solution of the governing equations, in *Computational Fluid Dynamics: Principles and Applications (Third Edition)*, edited by J. Blazek (Butterworth-Heinemann, Oxford, 2015) third edition ed., pp. 29 – 72.

[3] L. Yam and W. Cheng, Dynamic response of a cantilever timoshenko column due to almost-axial impact — a comparison between continuous and 2 dof models, in *Advances in Engineering Plasticity and its Applications*, edited by W. LEE (Elsevier, Oxford, 1993) pp. 461 – 468.

[4] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Science and Technology **4**, 043001 (2019).

[5] F. Fontanela, A. Jacquier, and M. Oumgari, A quantum algorithm for linear pdes arising in finance (2019), arXiv:1912.02753 [q-fin.CP].

[6] O. Kyriienko, A. E. Paine, and V. E. Elfving, Solving nonlinear differential equations with differentiable quantum circuits (2020), arXiv:2011.10395 [quant-ph].

[7] H. Liu, Y. Wu, L. Wan, S. Pan, S. Qin, F. Gao, and Q. Wen, Variational quantum algorithm for poisson equation (2020), arXiv:2012.07014 [quant-ph].

[8] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, Classical optimizers for noisy intermediate-scale quantum devices (2020), arXiv:2004.03004 [quant-ph].

[9] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, Variational quantum linear solver (2020), arXiv:1909.05820 [quant-ph].

[10] C. Zalka, Simulating quantum systems on a quantum computer, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**, 313 (1998).

[11] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, arXiv preprint quant-ph/0208112 (2002).

[12] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Physical Review A **98**, 10.1103/physreva.98.032309 (2018).

[13] A. Holmes and A. Y. Matsuura, Efficient quantum circuits for accurate state preparation of smooth, differentiable functions (2020), arXiv:2005.04351 [quant-ph].

[14] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[15] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028 [quant-ph].

[16] A. Muriel, Exact solution of the navier stokes equation with periodic boundary conditions (2018).

[17] L. K. Kuiper, Fourier solution of two-dimensional navier stokes equation with periodic boundary conditions and incompressible flow (2016), arXiv:1607.00566 [math.DS].

[18] F. Vatan and C. Williams, Optimal quantum circuits for general two-qubit gates, Physical Review A **69**, 032315 (2004).

[19] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Variational quantum algorithms, arXiv preprint arXiv:2012.09265 (2020).