

Variational Quantum Framework for Nonlinear PDE Constrained Optimization Using Carleman Linearization

Abeynaya Gnanasekaran ^{1*}†, Amit Surana ^{2*}† and Hongyu Zhu ²

¹ SRI International, Menlo Park, CA 94025, USA

² RTX Technology Research Center, 411 Silver Lane, East Hartford, CT 06108, USA

† Equal contribution

* Corresponding authors: abeynaya.gnanasekaran@sri.com, amit.surana@rtx.com

Received date: 7 March 2025; Accepted date: 6 May 2025; Published online: 6 June 2025

Abstract: We present a novel variational quantum framework for nonlinear partial differential equation (PDE) constrained optimization problems. The proposed work extends the recently introduced bi-level variational quantum PDE constrained optimization (BVQPCO) framework for linear PDE to a nonlinear setting by leveraging Carleman linearization (CL). CL framework allows one to transform a system of polynomial ordinary differential equations (ODE), i.e., ODE with polynomial vector field, into a system of infinite but linear ODE. For instance, such polynomial ODEs naturally arise when the PDE is semi-discretized in the spatial dimensions. By truncating the CL system to a finite order, one obtains a finite system of linear ODE to which the linear BVQPCO framework can be applied. In particular, the finite system of linear ODE is discretized in time and embedded as a system of linear equations. The variational quantum linear solver (VQLS) is used to solve the linear system for given optimization parameters and evaluate the design cost/objective function, and a classical black box optimizer is used to select the next set of parameter values based on this evaluated cost. We present detailed computational errors and complexity analysis and prove that under suitable assumptions, our proposed framework can provide potential advantages over classical techniques. We implement our framework using the PennyLane library and apply it to solve inverse Burgers' problem. We also explore an alternative tensor product decomposition which exploits the sparsity/structure of linear system arising from PDE discretization to facilitate the computation of VQLS cost functions.

Keywords: PDE constrained optimization; Carleman linearization; variational algorithms; quantum computing

1. Introduction

We present a variational quantum framework for optimization problems constrained by nonlinear partial differential equations (PDE). This extends the recently proposed bi-level variational quantum PDE constrained optimization (BVQPCO) framework for linear PDEs in [1] to a nonlinear setting. This is an important extension as many science and engineering applications such as aerodynamics, computational fluid dynamics (CFD), material science, computer vision, and inverse problems necessitate the solution of optimization problems constrained by a system of nonlinear PDE. Examples include Euler and Navier-Stokes PDEs and heat equation in fluid dynamics, combustion, and weather forecasting; magnetohydrodynamics and Vlasov-Maxwell PDEs in plasma dynamics and astrophysics; wave equation in structural mechanics; Black-Scholes PDE in finance; and reaction-diffusion PDEs in chemistry, biology, and ecology to name a few. Closed-form solutions are generally unavailable for PDE constrained optimization problems, necessitating the development of numerical methods [2–4]. A variety of classical gradient-based and gradient-free numerical methods have been developed, and rely on repeated calls to the underlying PDE solver. Since PDE simulations are computationally expensive, using them within the design/optimization loop can become a bottleneck.

To extend the BVQPCO framework for nonlinear PDE we employ the Carleman linearization (CL) framework. The CL framework allows one to transform polynomial ordinary differential equations (ODE), i.e., ODE with polynomial vector field, into a system of infinite but linear ODE [5]. For instance, such polynomial ODE naturally arises when PDE, e.g., as mentioned above, are semi-discretized in spatial dimensions. By truncating the CL system

to a finite order, one obtains a finite system of linear ODE, and under suitable conditions, the impact of truncation error can be characterized [6,7]. CL has become an important tool in developing quantum algorithms for simulating polynomial ODE on quantum devices since the seminal work [8]. Specifically, assuming $R_2 < 1$, where R_2 is a parameter characterizing the ratio of the nonlinearity and forcing to the linear dissipation, the query/gate complexity of the proposed algorithm takes the form $\frac{sT^2q}{\epsilon} \text{poly}(\log T, \log n, \log 1/\epsilon)$, where ϵ is desired solution accuracy, n is system dimension, $q = \|\mathbf{x}(0)\|/\|\mathbf{x}(T)\|$ measures the decay of the solution $\mathbf{x}(t) \in \mathbb{R}^n$, T is the period of integration, and s is the sparseness of system matrices describing the ODE. This algorithm/analysis was extended for systems of k -th degree polynomial ODE for arbitrary (finite) values of k in [9]. Recent work has shown that by transforming the problem or working with a different formulation of conservation laws in CFD applications, e.g. lattice Boltzmann method (LBM), one may be able to more readily satisfy such conditions [10]. Furthermore, [11] developed detailed quantum resource estimates for implementing CL-based LBM simulation of incompressible CFD on a fully fault-tolerant quantum computer. While the motivation of exploring quantum algorithms in [11] is to accelerate simulation-based design which results in a PDE constrained optimization, there was no algorithmic treatment of that problem. The application of CL for solving polynomial ODE in a variational quantum setting has also been numerically investigated in [12], in which QLSA is replaced by the variational quantum linear solver (VQLS) [13]. This work was also restricted to applying CL+VQLS for PDE simulation and did not consider any associated design/optimization problem. Furthermore, the work does not make any computational error/complexity assessments for solving polynomial ODE by the CL+VQLS approach.

The main contributions of this work are as follows:

- We formulate a bi-level variational quantum framework, referred to as nonlinear BVQPCO (nBVQPCO), for solving nonlinear PDE constrained optimization problems as discussed above. While the original optimization problem is formulated in terms of an unnormalized PDE solution, we show how the problem can be transformed under certain conditions into an equivalent form which depends only on the normalized PDE solution. This is significant as VQLS generates only a normalized solution of the linear system as typically quantum algorithms do, and would have otherwise required a substantial overhead to obtain the unnormalized solution via additional estimation of the norm of the solution.
- We use the sigma basis [14], an alternative tensor product decomposition that exploits the sparsity/structure of linear system arising from PDE discretization to facilitate VQLS computations. This results in a substantial reduction in the number of terms in the linear combination of unitary (LCU) decomposition necessary for VQLS cost function evaluations, and thus leads to a significant computational advantage compared to the Pauli basis traditionally used for the LCU decomposition.
- We present a detailed computational error analysis for CL+VQLS-based solution of polynomial ODE and prove that under suitable assumptions the CL+VQLS approach can generate a normalized solution of a given ODE to an arbitrary accuracy. To the best of our knowledge such a rigorous analysis of CL+VQLS approach has not been presented before in the literature. Furthermore, by combining this error analysis with the empirically known results for the run time of VQLS, we assess potential utility of our nBVQPCO framework over equivalent classical methods.
- We implement the nBVQPCO framework using the PennyLane library and apply it to solve a prototypical inverse problem applied to nonlinear Burgers' PDE which involves calibrating the PDE model parameter to match the given measurements.

The paper is organized as follows: We start with mathematical preliminaries in Section 2 and introduce the nonlinear PDE constrained optimization problem in Section 3.1. We describe the CL+VQLS procedure in Section 3.2 and develop the nBVQPCO framework extension including an outline of pseudo-code for its implementation in Section 3.3. Complexity and error analysis of the proposed framework are covered in Section 4. In Section 5 we formulate the inverse PDE problem in our nBVQPCO framework, apply an alternative tensor product decomposition for computing VQLS cost function, and describe the implementation details and numerical results using the PennyLane library. We conclude in Section 7 with avenues for future work.

2. Mathematical Preliminaries

Let $\mathbb{N} = \{1, 2, \dots\}$, \mathbb{R} , and \mathbb{C} be the sets of positive integers, real numbers, and complex numbers, respectively. We will denote vectors by small bold letters and matrices by capital bold letters. \mathbf{A}^\dagger and \mathbf{A}' will denote the

vector/matrix complex conjugate and vector/matrix transpose, respectively. $Tr(\mathbf{A})$ will denote the trace of a matrix. We will represent an identity matrix of size $s \times s$ by \mathbf{I}_s .

Kronecker product will be denoted by \otimes . For any pair of vectors, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$, their Kronecker product $\mathbf{w} \in \mathbb{R}^{nm}$ is

$$\mathbf{w} = \mathbf{x} \otimes \mathbf{y} = (x_1 y_1, x_1 y_2, \dots, x_1 y_m, \dots, x_2 y_1, \dots, x_n y_1, \dots, x_n y_m)^T.$$

Similarly, for $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$, their Kronecker product is defined as

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \vdots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix}, \quad (1)$$

where $\mathbf{C} \in \mathbb{R}^{mp \times nq}$. The Kronecker power is a convenient notation to express all possible products of elements of a vector up to a given order, and is denoted by

$$\mathbf{x}^{[i]} = \underbrace{\mathbf{x} \otimes \mathbf{x} \cdots \otimes \mathbf{x}}_{i\text{-times}}, \quad (2)$$

for any $i \in \mathbb{N}$ with the convention $\mathbf{x}^{[0]} = 1$. Moreover, $\dim(\mathbf{x}^{[i]}) = n^i$, and each component of $\mathbf{x}^{[i]}$ is of the form $x_1^{\omega_1} x_2^{\omega_2} \cdots x_n^{\omega_n}$ for some multi-index $\omega \in \mathbb{N}^n$ of weight $\sum_{j=1}^n \omega_j = i$. Similarly, we denote the matrix Kronecker power by,

$$\mathbf{A}^{[i]} = \underbrace{\mathbf{A} \otimes \mathbf{A} \cdots \otimes \mathbf{A}}_{i\text{-times}}. \quad (3)$$

The standard inner product between two complex vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ will be denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$. The l_p -norm in the Euclidean space \mathbb{R}^n will be denoted by $\|\cdot\|_p, p = 1, 2, \dots, \infty$ and is defined as follows

$$\|\mathbf{x}\|_p = \left(\sum_{j=1}^n |x_j|^p \right)^{1/p}. \quad (4)$$

For the norm of a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, we use the induced norm, namely

$$\|\mathbf{A}\|_p = \max_{\mathbf{x} \in \mathbb{R}^m} \frac{\|\mathbf{A}\mathbf{x}\|_p}{\|\mathbf{x}\|_p}. \quad (5)$$

We will use $p = 2$, i.e., l_2 norm for vectors and spectral norm for matrices unless stated otherwise. The spectral radius of a matrix denoted by $\rho(\mathbf{A})$ is defined as

$$\rho(\mathbf{A}) = \max\{|\lambda| : \lambda \text{ eigenvalues of } \mathbf{A}\}. \quad (6)$$

Trace norm of a matrix \mathbf{A} is defined as $\|\mathbf{A}\|_{Tr} = Tr(\sqrt{\mathbf{A}^* \mathbf{A}})$ which is the Schatten q -norms with $q = 1$.

Let \mathcal{F} be a vector space of real vector valued functions $\mathbf{u}(t) : [0, T] \rightarrow \mathbb{R}^n$ defined on $[0, T]$. Then the inner product between $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{F}$ is defined as

$$\langle \mathbf{u}_1, \mathbf{u}_2 \rangle_T = \int_0^T \langle \mathbf{u}_1(t), \mathbf{u}_2(t) \rangle dt. \quad (7)$$

We will use standard bracket notation, i.e. $|\psi\rangle$ and $\langle\psi|$ in representing the quantum states [15]. The inner product between two quantum states $|\psi\rangle$ and $|\phi\rangle$ will be denoted by $\langle\psi|\phi\rangle$. For a vector \mathbf{x} , we denote by $|\mathbf{x}\rangle = \frac{\mathbf{x}}{\|\mathbf{x}\|}$ as the corresponding quantum state. The trace norm $\rho(\psi, \phi)$ between two pure states $|\psi\rangle$ and $|\phi\rangle$ is defined as

$$\rho(\psi, \phi) = \frac{1}{2} \|\langle\phi|\phi\rangle - |\psi\rangle\langle\psi|\|_{Tr} = \sqrt{1 - |\langle\phi|\psi\rangle|^2}. \quad (8)$$

3. Variational Quantum Framework for Nonlinear PDE Constrained Optimization

3.1. Nonlinear PDE Constrained Optimization Problem

We consider a general class of PDE constrained design optimization problems of the form [16]

$$\min_{\mathbf{p}, \mathbf{u}} C_d(\mathbf{u}, \mathbf{p}) \quad (9)$$

$$\text{s.t. } \mathcal{F}(\mathbf{u}, \mathbf{p}, t) = 0, \quad (10)$$

$$\mathcal{B}(\mathbf{u}, \mathbf{p}, t) = 0, \quad (11)$$

$$g_i(\mathbf{p}) \leq 0, i = 1, \dots, n_c, \quad (12)$$

where $t \in [0, T]$ for given $T \in \mathbb{R}$, $\mathbf{p} \in \mathbb{R}^{n_p}$ is the vector of design variables (e.g., material properties, aerodynamic shape), C_d is the cost function (e.g., heat flux, drag/ lift), and $\mathcal{F}(\mathbf{u}, \mathbf{p}, t)$ are the PDE constraints (e.g., conservation laws, constitutive relations) with $\mathbf{u}(\mathbf{x}, t; \mathbf{p})$ being the solution of the PDE defined as a function of space \mathbf{x} and time t for given parameters \mathbf{p} , $\mathcal{B}(\mathbf{u}, \mathbf{p}, t)$ to capture the constraints coming from the PDE boundary and initial conditions, and $g_i(\mathbf{p})$, $i = 1, \dots, n_c$ are constraints on the parameters. We make the following assumptions.

Assumption 1. When semi-discretized in space, the PDE (10) and associated boundary/initial conditions (11) results in a polynomial ODE of the form

$$\dot{\mathbf{u}}(t) = \mathbf{F}_0(t, \mathbf{p}) + \mathbf{F}_1(t, \mathbf{p})\mathbf{u} + \dots + \mathbf{F}_k(t, \mathbf{p})\mathbf{u}^{[k]}, \quad (13)$$

$$\mathbf{u}(0) = \mathbf{u}_0(\mathbf{p}) \in \mathbb{R}^n,$$

where $k \in \mathbb{N}$ is the order of the polynomial, $\mathbf{u}(t) \in \mathbb{R}^n$, $t \in [0, T]$ (with slight abuse of notation) is an n -dimensional solution vector and $\mathbf{F}_i(t, \mathbf{p}) \in \mathbb{R}^{n \times n^i}$ are in general time dependent matrices. We assume that the boundary conditions have already been accounted for in this ODE representation, see Section 5.1 for an example.

Assumption 2. The cost function is a quadratic function of the solution $\mathbf{u}(t)$, $t \in [0, T]$, i.e.

$$C_d(\mathbf{u}, \mathbf{p}) = f(w_1 \langle \mathbf{u}, \mathbf{H}\mathbf{u} \rangle_T + w_2 \langle \mathbf{u}, \mathbf{h} \rangle_T, \mathbf{p}), \quad (14)$$

where $w_1, w_2 \in \mathbb{R}$, $\mathbf{H} \in \mathbb{R}^{n \times n}$ and $\mathbf{h} \in \mathbb{R}^n$.

Under these assumptions, the optimization problem (9) can be represented as

$$\min_{\mathbf{p}, \mathbf{u}} C_d(\mathbf{u}, \mathbf{p}) \equiv f(w_1 \langle \mathbf{u}, \mathbf{H}\mathbf{u} \rangle_T + w_2 \langle \mathbf{u}, \mathbf{h} \rangle_T, \mathbf{p}) \quad (15)$$

$$\text{s.t. } \dot{\mathbf{u}}(t) = \mathbf{F}_0(t, \mathbf{p}) + \mathbf{F}_1(t, \mathbf{p})\mathbf{u} + \dots + \mathbf{F}_k(t, \mathbf{p})\mathbf{u}^{[k]}, \quad (16)$$

$$\mathbf{u}(0) = \mathbf{u}_0(\mathbf{p}), \quad (17)$$

$$g_i(\mathbf{p}) \leq 0, i = 1, \dots, n_c. \quad (18)$$

Remark 1. Assumption 1 is fairly general as several PDEs arising in engineering and science result in polynomial ODEs when semi-discretized in space, see [9] and references therein. Examples include Euler and Navier-Stokes PDEs and heat equation in fluid dynamics, combustion, and weather forecasting; magnetohydrodynamics and Vlasov-Maxwell PDEs in plasma dynamics and astrophysics; wave equation in structural mechanics; Black-Scholes PDE in finance; and reaction-diffusion PDEs in chemistry, biology, and ecology to name a few. Furthermore, polynomial ODEs also arise in mechanics, molecular dynamics, chemical kinetics, epidemiology, social dynamics, and biological networks.

Remark 2. Assumption 2 commonly arises in many PDE constrained optimization problems. For instance in inverse problems the cost function is typically taken as a squared norm of difference between the solution and measured variables, which results in a quadratic form, see Section 5.1 for details. Other examples include PDE optimal control where the cost function is a weighted combination of deviation from a reference signal and squared norm of control input, aerodynamic design where the objective is to minimize heat transfer/drag or maximize lift, and computer vision

problems such as shape from shading, surface reconstruction from sparse data, and optical flow computation where the objective function depends on an appropriate form of squared norm.

The BVQPCO framework proposed in [1] cannot be applied directly to the optimization problem (15)-(18), as the constraints are nonlinear. To address this we employ the CL framework, whereby the polynomial ODEs are transformed into an infinite-dimensional system of linear ODEs and truncated. Since, it is always possible to map the k -th degree polynomial ODE (13) to a higher dimensional quadratic polynomial ODE, and then apply CL [9], without loss of generality we restrict $k = 2$ throughout this paper.

3.2. Solving Nonlinear ODEs via Carleman Linearization-Based VQLS

Consider a system of inhomogeneous quadratic polynomial ODE (16) with $k = 2$, i.e.

$$\begin{aligned}\dot{\mathbf{u}} &= \mathbf{F}_0(t, \mathbf{p}) + \mathbf{F}_1(t, \mathbf{p})\mathbf{u} + \mathbf{F}_2(t, \mathbf{p})\mathbf{u}^{[2]}, \\ \mathbf{u}(0) &= \mathbf{u}_0 \in \mathbb{R}^n.\end{aligned}\quad (19)$$

Below we describe the steps required to transform the solution of these nonlinear ODEs into a form amenable to VQLS, see Figure 1 for illustration of the steps involved.

Step 1: For CL we introduce variables $\mathbf{w}_i = \mathbf{u}^{[i]} \in \mathbb{R}^{n^i}$, $i \in \mathbb{N}$, which satisfy

$$\dot{\mathbf{w}}_i = \sum_{j=0}^2 \mathbf{A}_{i+j-1}^i(t, \mathbf{p})\mathbf{w}_{i+j-1}, \quad (20)$$

where $\mathbf{A}_{i+j-1}^i \in \mathbb{R}^{n^i \times n^{i+j-1}}$ is given by

$$\mathbf{A}_{i+j-1}^i(t, \mathbf{p}) = \sum_{p=1}^i \overbrace{\mathbf{I}_{n \times n} \otimes \cdots \otimes \mathbf{F}_j(t, \mathbf{p}) \otimes \cdots \otimes \mathbf{I}_{n \times n}}^{i \text{ factors}} \quad (21)$$

p -th position

with $0 \leq j \leq 2$. Note that (20) defines an infinite dimensional system of linear ODE with state $\mathbf{w}_\infty(t) = (\mathbf{w}_1^T(t), \mathbf{w}_2^T(t), \dots)^T$ and initial condition $\mathbf{w}_\infty(0) = (\mathbf{u}_0^T, (\mathbf{u}_0^{[2]})^T, \dots)^T$, and is known as CL. For any finite $N \in \mathbb{N}$ define

$$\mathbf{w}(t) = \begin{pmatrix} \mathbf{w}_1(t) \\ \mathbf{w}_2(t) \\ \mathbf{w}_3(t) \\ \vdots \\ \mathbf{w}_N(t) \end{pmatrix} \in \mathbb{R}^{N_c}, \quad (22)$$

where $N_c = \frac{n^{N+1}-n}{n-1}$. The CL when truncated to order N , results in a finite system of linear ODEs

$$\dot{\hat{\mathbf{w}}} = \mathbf{A}_N(t, \mathbf{p})\hat{\mathbf{w}} + \mathbf{b}(t, \mathbf{p}), \quad (23)$$

$$\hat{\mathbf{w}}(0) = (\mathbf{u}_0^T, (\mathbf{u}_0^{[2]})^T, \dots, (\mathbf{u}_0^{[N]})^T)^T, \quad (24)$$

$$\mathbf{A}_N(t, \mathbf{p}) = \begin{pmatrix} \mathbf{A}_1^1 & \mathbf{A}_2^1 & 0 & \cdots & 0 \\ \mathbf{A}_1^2 & \mathbf{A}_2^2 & \mathbf{A}_3^2 & \cdots & 0 \\ 0 & \mathbf{A}_2^3 & \mathbf{A}_3^3 & \mathbf{A}_4^3 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \mathbf{A}_{N-1}^N & \mathbf{A}_N^N \end{pmatrix}, \hat{\mathbf{w}}(t) = \begin{pmatrix} \hat{\mathbf{w}}_1 \\ \hat{\mathbf{w}}_2 \\ \hat{\mathbf{w}}_3 \\ \vdots \\ \hat{\mathbf{w}}_N \end{pmatrix} \in \mathbb{R}^{N_c}, \mathbf{b}(t, \mathbf{p}) = \begin{pmatrix} \mathbf{F}_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (25)$$

where we have dropped dependence on t, \mathbf{p} in the RHS terms for brevity. Note that $\hat{\mathbf{w}}(t)$ approximates $\mathbf{w}(t)$ as defined in Eq. (22) and the error incurred as a function of the truncation level N is described by Lemma 1.

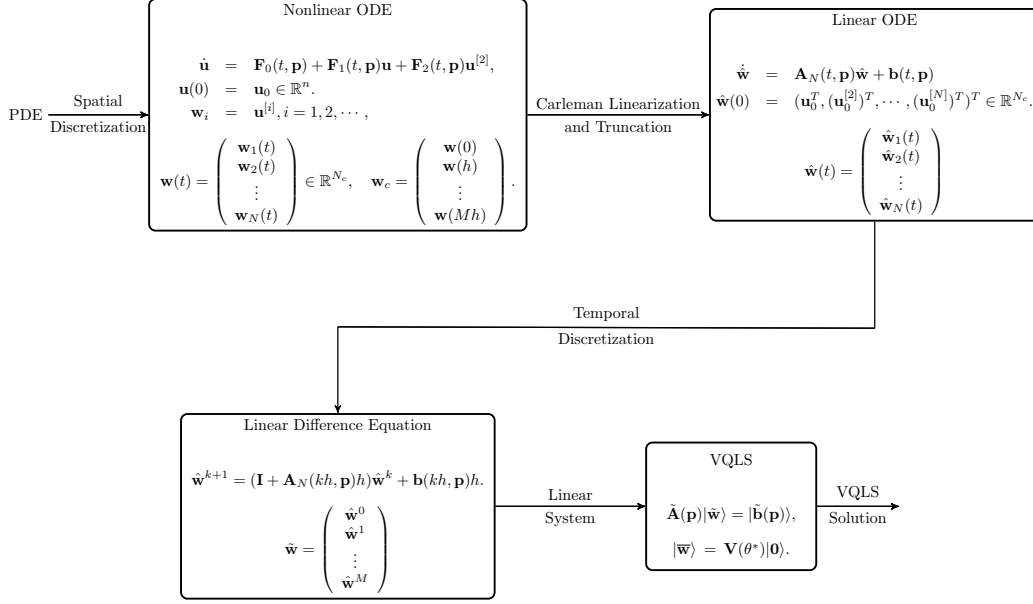


Figure 1. Illustration of steps involved in transforming a quadratic polynomial ODE (19) into a system of linear algebraic equations (30) amenable to VQLS. This involves CL and truncation resulting in a system of linear ODEs, time discretization of linear ODEs to generate a system of linear difference equations, and expressing difference equation into a single system of linear algebraic equations with normalized solution and right-hand side. While an explicit time discretization is shown in the figure, one can use implicit Euler discretization or other schemes as well.

Forward Euler:

Step 2: Applying the forward Euler method to the system (23) with step size $h = T/M$, $M \in \mathbb{N}$, results in a set of difference equations

$$\hat{\mathbf{w}}^{k+1} - (\mathbf{I} + \mathbf{A}_N(kh, \mathbf{p})h)\hat{\mathbf{w}}^k = \mathbf{b}(kh, \mathbf{p})h, \quad (26)$$

where $\hat{\mathbf{w}}^k$ approximates $\hat{\mathbf{w}}(kh)$ as defined in Eq. (25) for $k = \{0, \dots, M-1\}$ with $\hat{\mathbf{w}}^0 = \hat{\mathbf{w}}(0)$. The error introduced by Euler discretization is characterized by Lemma 2.

Step 3: The iterative system (26) can be expressed as a single system of linear equations

$$\overbrace{\begin{pmatrix} \mathbf{I} & 0 & 0 & \dots \\ -[\mathbf{I} + \mathbf{A}_N(0, \mathbf{p})h] & \mathbf{I} & 0 & \dots \\ 0 & 0 & \ddots & \ddots \\ 0 & 0 & -[\mathbf{I} + \mathbf{A}_N((M-1)h, \mathbf{p})h] & \mathbf{I} \end{pmatrix}}^{\tilde{\mathbf{A}}(\mathbf{p})} \overbrace{\begin{pmatrix} \hat{\mathbf{w}}^0 \\ \hat{\mathbf{w}}^1 \\ \vdots \\ \hat{\mathbf{w}}^M \end{pmatrix}}^{\tilde{\mathbf{w}}} = \overbrace{\begin{pmatrix} \hat{\mathbf{w}}(0) \\ h\mathbf{b}(0, \mathbf{p}) \\ \vdots \\ h\mathbf{b}((M-1)h, \mathbf{p}) \end{pmatrix}}^{\tilde{\mathbf{b}}(\mathbf{p})}. \quad (27)$$

Backward Euler:

Step 2: Similarly, applying the backward Euler method to the system (23) with step size $h = T/M$ yields

$$(\mathbf{I} - \mathbf{A}_N(kh, \mathbf{p})h)\hat{\mathbf{w}}^{k+1} - \hat{\mathbf{w}}^k = \mathbf{b}(kh, \mathbf{p})h, \quad (28)$$

where $\hat{\mathbf{w}}^k$ approximates $\hat{\mathbf{w}}(kh)$ for $k = \{0, \dots, M-1\}$.

Step 3: The iterative system (28) can similarly be expressed as a single system of linear equations

$$\overbrace{\begin{pmatrix} \mathbf{I} & 0 & 0 & \dots \\ -\mathbf{I} & [\mathbf{I} - \mathbf{A}_N(0, \mathbf{p})h] & 0 & \dots \\ 0 & 0 & \ddots & \ddots \\ 0 & 0 & -\mathbf{I} & [\mathbf{I} - \mathbf{A}_N((M-1)h, \mathbf{p})h] \end{pmatrix}}^{\tilde{\mathbf{A}}(\mathbf{p})} \overbrace{\begin{pmatrix} \hat{\mathbf{w}}^0 \\ \hat{\mathbf{w}}^1 \\ \vdots \\ \hat{\mathbf{w}}^M \end{pmatrix}}^{\tilde{\mathbf{w}}} = \overbrace{\begin{pmatrix} \hat{\mathbf{w}}_{in} \\ h\mathbf{b}(0, \mathbf{p}) \\ \vdots \\ h\mathbf{b}((M-1)h, \mathbf{p}) \end{pmatrix}}^{\tilde{\mathbf{b}}(\mathbf{p})}. \quad (29)$$

In the VQLS framework the linear system obtained using forward Euler (Eq. (27)) or backward Euler (Eq. (29)) is further transformed into the form

$$\tilde{\mathbf{A}}(\mathbf{p})|\tilde{\mathbf{w}}\rangle = |\tilde{\mathbf{b}}(\mathbf{p})\rangle, \quad (30)$$

where $|\tilde{\mathbf{b}}\rangle = \tilde{\mathbf{b}}/\|\tilde{\mathbf{b}}\|$ and $|\tilde{\mathbf{w}}\rangle = \tilde{\mathbf{w}}/\|\tilde{\mathbf{w}}\|$ are the normalized vectors. The VLQS algorithm then optimizes the parameter θ of the ansatz $\mathbf{V}(\theta)|\mathbf{0}\rangle$ such that

$$\tilde{\mathbf{A}}(\mathbf{p})\mathbf{V}(\theta)|\mathbf{0}\rangle = |\tilde{\mathbf{b}}(\mathbf{p})\rangle.$$

The optimal parameter θ^* then prepares a solution $|\tilde{\mathbf{w}}\rangle = \mathbf{V}(\theta^*)|\mathbf{0}\rangle$ which is an approximation to $|\tilde{\mathbf{w}}\rangle$. The approximation error is characterized by the Lemma 4.

Different cost functions have been proposed for the VQLS algorithm [13]. This includes global cost function $C_{ug}(\theta)$ and its normalized version $C_g(\theta)$

$$C_{ug}(\theta) = \text{Tr}(|\phi\rangle\langle\phi|(\mathbf{I} - |\tilde{\mathbf{b}}\rangle\langle\tilde{\mathbf{b}}|)) = \langle\psi|\mathbf{H}_g|\psi\rangle, \quad C_g(\theta) = \frac{C_{ug}}{\langle\phi|\phi\rangle} = 1 - \frac{|\langle\tilde{\mathbf{b}}|\phi\rangle|^2}{\langle\phi|\phi\rangle} \quad (31)$$

where $|\phi\rangle = \tilde{\mathbf{A}}|\psi(\theta)\rangle$, $|\psi(\theta)\rangle = \mathbf{V}(\theta)|\mathbf{0}\rangle$ and $\mathbf{H}_g = \tilde{\mathbf{A}}^*\mathbf{U}_b(\mathbf{I} - |\mathbf{0}\rangle\langle\mathbf{0}|)(\mathbf{U}_b)^*\tilde{\mathbf{A}}$, and similarly unnormalized and normalized local cost functions

$$C_{ul}(\theta) = \langle\psi|\mathbf{H}_l|\psi\rangle, \quad C_l(\theta) = \frac{C_{ul}}{\langle\phi|\phi\rangle}, \quad \mathbf{H}_l = \tilde{\mathbf{A}}^*\mathbf{U}_b\left(\mathbf{I} - \frac{1}{n}\sum_{i=1}^n|0_k\rangle\langle 0_k| \otimes \mathbf{I}_{\tilde{k}}\right)(\mathbf{U}_b)^*\tilde{\mathbf{A}} \quad (32)$$

respectively, with $|0_k\rangle$ being a zero state on qubit k , $\mathbf{I}_{\tilde{k}}$ being identity on all qubits except the qubit k , and the unitary \mathbf{U}_b prepares $|\tilde{\mathbf{b}}\rangle = \mathbf{U}_b|\mathbf{0}\rangle$. Computation of these cost functions can be accomplished via the Hadamard test circuit and its variations (see [13] for the details) and rely on the linear combination of unitaries (LCU) decomposition of the matrix $\tilde{\mathbf{A}}$.

Linear Combination of Unitaries Decomposition: The square matrix $\tilde{\mathbf{A}}$ can be decomposed as a linear combination of unitary operators $\tilde{\mathbf{A}}_i$ with complex scalar coefficients α_i as,

$$\tilde{\mathbf{A}} = \sum_{i=1}^{n_l} \alpha_i \tilde{\mathbf{A}}_i. \quad (33)$$

One popular approach for unitary operators is based on the Pauli basis formed from the identity \mathbf{I} and the Pauli gates \mathbf{X} , \mathbf{Y} , and \mathbf{Z} . A matrix $\tilde{\mathbf{A}}$ with the size $2^n \times 2^n$ can then be written as a linear combination of elements selected from the basis set

$$\mathcal{P}_n = \{\mathbf{P}_1 \otimes \mathbf{P}_2 \cdots \otimes \mathbf{P}_n : \mathbf{P}_i \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}\}.$$

On this basis, each $\tilde{\mathbf{A}}_i \in \mathcal{P}_n$ and its corresponding coefficient α_i can be determined via different numerical approaches [17]. More recently, alternative tensor product decomposition has been proposed which uses different basis elements to better exploit the underlying structure and sparsity of matrices and results in more efficient decomposition [14,18]. For example, for the matrices arising from discretization of Poisson equation [18] and heat equation [14] the number of LCU terms under sigma basis only scale logarithmically, i.e., as $\mathcal{O}(n)$ with the matrix size $2^n \times 2^n$, compared to the Pauli basis for which the number of terms can vary from $\mathcal{O}(2^n)$ (diagonal matrices) to $\mathcal{O}(2^{2n})$ (dense matrices).

The sigma basis set S comprises of following operators

$$S = \{\sigma_+, \sigma_-, \sigma_+\sigma_-, \sigma_+\sigma_-, \mathbf{I}\}, \quad (34)$$

where $\sigma_+ = |0\rangle\langle 1|$, $\sigma_- = |1\rangle\langle 0|$, $\sigma_+\sigma_- = |0\rangle\langle 0|$ and $\sigma_-\sigma_+ = |1\rangle\langle 1|$. Given a matrix $\tilde{\mathbf{A}}$, its LCU-type tensor product decomposition in terms of sigma basis takes a similar form as (33) but with $\tilde{\mathbf{A}}_i \in \mathcal{S}_n$ where

$$\mathcal{S}_n = \{\sigma_1 \otimes \sigma_2 \cdots \otimes \sigma_n : \sigma_i \in S\}.$$

Even though some of the operators in S are non-unitary, using the concept of unitary completion, one can still design efficient quantum circuits for computing the global/local VQLS cost functions, see [14] for details. We have used this alternate decomposition in the numerical study in Section 5.

3.3. Nonlinear BVQPCO

To solve the optimization problem (15)–(18) in a variational quantum framework, we extend the BVQPCO framework to nonlinear setting, where the outer optimization level iteratively selects \mathbf{p}^k using a classical black box optimizer based on the cost function evaluated using solution of CL truncated system obtained via VQLS. To do so, let $\mathbf{K}_0 \in \mathbb{R}^{n \times N_c(M+1)}$ and $\mathbf{K}_f \in \mathbb{R}^{n(M+1) \times N_c(M+1)}$ be matrices such that

$$\mathbf{u}_0 = \hat{\mathbf{w}}_1^0 = \mathbf{K}_0 \tilde{\mathbf{w}}, \quad (35)$$

and

$$\begin{pmatrix} \hat{\mathbf{w}}_1^0 \\ \hat{\mathbf{w}}_1^1 \\ \vdots \\ \hat{\mathbf{w}}_1^M \end{pmatrix} = \mathbf{K}_f \tilde{\mathbf{w}}, \quad (36)$$

where $\hat{\mathbf{w}}_1^k$ is the first n components extracted from the vectors $\hat{\mathbf{w}}^k$ for $k = 0, \dots, M$. Thus, the two terms in the cost function Eq. (14) can be approximated via the Riemann sum as

$$\langle \mathbf{u}, \mathbf{H}\mathbf{u} \rangle_T \approx \sum_{k=0}^M h \langle \mathbf{u}(kh), \mathbf{H}\mathbf{u}(kh) \rangle \approx h \langle \mathbf{K}_f \tilde{\mathbf{w}}, (\mathbf{I}_{M+1} \otimes \mathbf{H}) \mathbf{K}_f \tilde{\mathbf{w}} \rangle, \quad (37)$$

$$\langle \mathbf{u}, \mathbf{h} \rangle_T \approx h \langle \mathbf{1}_{M+1} \otimes \mathbf{h}, \mathbf{K}_f \tilde{\mathbf{w}} \rangle, \quad (38)$$

where $\mathbf{1}_{M+1} \in \mathbb{R}^{M+1}$ is a vector of ones. Furthermore

$$\begin{aligned} \langle \mathbf{K}_f \tilde{\mathbf{w}}, (\mathbf{I}_{M+1} \otimes \mathbf{H}) \mathbf{K}_f \tilde{\mathbf{w}} \rangle &= \|\mathbf{u}_0\|^2 \frac{\langle \mathbf{K}_f \tilde{\mathbf{w}}, (\mathbf{I}_{M+1} \otimes \mathbf{H}) \mathbf{K}_f \tilde{\mathbf{w}} \rangle}{\langle \mathbf{K}_0 \tilde{\mathbf{w}}, \mathbf{K}_0 \tilde{\mathbf{w}} \rangle}, \\ &= \|\mathbf{u}_0\|^2 \frac{\langle \tilde{\mathbf{w}} | \mathbf{K}_f^T (\mathbf{I}_{M+1} \otimes \mathbf{H}) \mathbf{K}_f | \tilde{\mathbf{w}} \rangle}{\langle \tilde{\mathbf{w}} | \mathbf{K}_0^T \mathbf{K}_0 | \tilde{\mathbf{w}} \rangle}, \end{aligned} \quad (39)$$

and similarly

$$\langle \mathbf{1}_{M+1} \otimes \mathbf{h}, \mathbf{K}_f \tilde{\mathbf{w}} \rangle = \|\mathbf{u}_0\| \frac{\langle \mathbf{1}_{M+1} \otimes \mathbf{h}, \mathbf{K}_f \tilde{\mathbf{w}} \rangle}{\sqrt{\langle \mathbf{K}_0 \tilde{\mathbf{w}}, \mathbf{K}_0 \tilde{\mathbf{w}} \rangle}} = \|\mathbf{u}_0\| \frac{\langle (\mathbf{1}_{M+1} \otimes \mathbf{h})^T \mathbf{K}_f | \tilde{\mathbf{w}} \rangle}{\sqrt{\langle \tilde{\mathbf{w}} | \mathbf{K}_0^T \mathbf{K}_0 | \tilde{\mathbf{w}} \rangle}}. \quad (40)$$

Thus, the optimization problem (15)–(18) can be expressed in a variational quantum form as

$$\min_{\mathbf{p}, \theta} f(g(\theta), \mathbf{p}), \quad (41)$$

$$\text{s.t. } \tilde{\mathbf{A}}(\mathbf{p}) \mathbf{V}(\theta) | \mathbf{0} \rangle = |\tilde{\mathbf{b}}(\mathbf{p}) \rangle, \quad (42)$$

$$g_i(\mathbf{p}) \leq 0, i = 1, \dots, n_c, \quad (43)$$

where

$$g(\theta) = \left(w_1 \|\mathbf{u}_0\|^2 \frac{\langle \tilde{\mathbf{w}} | \mathbf{K}_f^T (\mathbf{I}_{M+1} \otimes \mathbf{H}) \mathbf{K}_f | \tilde{\mathbf{w}} \rangle}{\langle \tilde{\mathbf{w}} | \mathbf{K}_0^T \mathbf{K}_0 | \tilde{\mathbf{w}} \rangle} + w_2 \|\mathbf{u}_0\| \frac{\langle (\mathbf{1}_{M+1} \otimes \mathbf{h})^T \mathbf{K}_f | \tilde{\mathbf{w}} \rangle}{\sqrt{\langle \tilde{\mathbf{w}} | \mathbf{K}_0^T \mathbf{K}_0 | \tilde{\mathbf{w}} \rangle}} \right)_{|\tilde{\mathbf{w}}\rangle = \mathbf{V}(\theta) | \mathbf{0} \rangle}, \quad (44)$$

$$\Phi_f = \mathbf{K}_f^T (\mathbf{I}_{M+1} \otimes \mathbf{H}) \mathbf{K}_f, \Phi_0 = \mathbf{K}_0^T \mathbf{K}_0, \phi = (\mathbf{1}_{M+1} \otimes \mathbf{h})^T \mathbf{K}_f. \quad (45)$$

Note that the above quantum formulation, unlike the original optimization problem (15)–(18), depends only on the normalized PDE solution. This is significant as VQLS generates only a normalized solution of the linear system as typically quantum algorithms do and would have otherwise required a substantial overhead to obtain the unnormalized solution via additional estimation of the norm of the solution.

Algorithm 1 nBVQPCO Algorithm.

Input: Semi-discretized ODE in the form (19), cost function (14), $\{g_i(\mathbf{p})\}_{i=1}^{n_c}$, CL truncation level N , $\mathbf{V}(\boldsymbol{\theta}), \gamma, n_{sh}$ and ϵ
Output: Optimal parameters: \mathbf{p}^*

- 1: Initialize $k = 0$ and $\mathbf{p}^0 = \mathbf{p}_{in}$.
- 2: Apply CL to (19) with truncation level N and determine $\tilde{\mathbf{A}}(\mathbf{p}), \tilde{\mathbf{b}}(\mathbf{p})$ in Eq. (30) and associated ϕ, Φ_f, Φ_0 as defined in Eq. (45).
- 3: Determine the unitary \mathbf{U}_ϕ to prepare $|\phi\rangle$, and LCU decomposition for $\Phi_f = \sum_{i=1}^{n_f} \alpha_{fi} \Phi_{fi}$ and $\Phi_0 = \sum_{i=1}^{n_0} \alpha_{0i} \Phi_{0i}$.
- 4: **while** stopping criteria not met **do**
- 5: Compute LCU $\{\alpha_i(\mathbf{p}^k), \tilde{\mathbf{A}}_i(\mathbf{p}^k)\}_{i=1}^{n_k}$ of $\tilde{\mathbf{A}}(\mathbf{p}^k)$, and determine a unitary \mathbf{U}_b^k such that $|\tilde{\mathbf{b}}(\mathbf{p}^k)\rangle = \mathbf{U}_b^k|\mathbf{0}\rangle$.
- 6: Compute $\boldsymbol{\theta}_*^k = VQLS(\{\alpha_i, \tilde{\mathbf{A}}_i\}, \mathbf{U}_b^k, \mathbf{V}(\boldsymbol{\theta}), \gamma, n_{sh})$.
- 7: Let $|\psi(\boldsymbol{\theta}_*^k)\rangle = \mathbf{V}(\boldsymbol{\theta}_*^k)|\mathbf{0}\rangle$. Compute $\langle\phi|\psi(\boldsymbol{\theta}_*^k)\rangle, \langle\psi(\boldsymbol{\theta}_*^k)|\Phi_f|\psi(\boldsymbol{\theta}_*^k)\rangle$ and $\langle\psi(\boldsymbol{\theta}_*^k)|\Phi_0|\psi(\boldsymbol{\theta}_*^k)\rangle$ using associated quantum circuits, and evaluate the cost function (41).
- 8: Use classical black box optimizer to select next \mathbf{p}^{k+1} subject to constraints $\{g_i(\mathbf{p})\}_{i=1}^{n_c}$.
- 9: Check the stopping criterion, e.g., whether $S(\mathbf{p}^k, \mathbf{p}^{k+1}) \leq \epsilon$.
- 10: $k \leftarrow k + 1$
- 11: **end while**
- 12: Return \mathbf{p}^k

The pseudo-code for nBVQPCO is summarized in Algorithm 1, and Figure 2 shows the overall flow diagram. Several remarks follow:

- Note that VQLS provides solution of the given linear system only up to a normalization constant. By reformulating the cost function expression as discussed above one can make the cost function evaluation independent of the normalization constant.
- The inputs for VQLS in line (6) include the linear combination of unitaries (LCU) decomposition of $\tilde{\mathbf{A}}(\mathbf{p})$, unitary \mathbf{U}_b^k which prepares $|\tilde{\mathbf{b}}(\mathbf{p}^k)\rangle$, $\mathbf{V}(\boldsymbol{\theta})$ the selected ansatz, γ the stopping threshold, and n_{sh} the number of shots used in VQLS cost function evaluation. See Section 5 for the details. From the relations (60), one can select γ such that, for instance, for $C_g \leq \gamma$, results in a VLQS approximation error $\epsilon \leq \kappa\sqrt{\gamma \log N}$, see Lemma 4.
- For LCU decomposition a popular approach is to use Pauli basis as discussed above. We propose to employ a more efficient tensor product decomposition [14], which exploits the underlying structure and sparsity of matrices such as arising from PDE discretizations, see Section 5.
- Depending on how $\tilde{\mathbf{A}}$ depends on the parameters \mathbf{p} , a parameter-dependent LCU decomposition $\{\alpha_i(\mathbf{p}), \mathbf{A}_i\}$ can be pre-computed once, thus saving computational effort, see Section 5.1 for an example.
- For computing $\langle\phi|\psi(\boldsymbol{\theta}_*^k)\rangle$ in line 7 one can use the unitary \mathbf{U}_ϕ as computed in line (3) and quantum circuit for the SWAP test. Similarly, given the LCU decompositions of Φ_f and Φ_0 as computed in line (3) one can express

$$\begin{aligned} \langle\psi(\boldsymbol{\theta}_*^k)|\Phi_f|\psi(\boldsymbol{\theta}_*^k)\rangle &= \sum_{i=1}^{n_f} \alpha_{fi} \langle\psi(\boldsymbol{\theta}_*^k)|\Phi_{fi}|\psi(\boldsymbol{\theta}_*^k)\rangle, \\ \langle\psi(\boldsymbol{\theta}_*^k)|\Phi_0|\psi(\boldsymbol{\theta}_*^k)\rangle &= \sum_{i=1}^{n_0} \alpha_{0i} \langle\psi(\boldsymbol{\theta}_*^k)|\Phi_{0i}|\psi(\boldsymbol{\theta}_*^k)\rangle, \end{aligned}$$

and compute each term in the above sums using quantum circuit associated with the Hadamard test.

- For the outer optimization one can utilize any global black box optimization method [1]. For instance, one can sample the optimization variables \mathbf{p} to generate a set of predetermined grid points, evaluate the cost function at those points and select the grid point with the minimum cost. Alternatively, one can use more adaptive techniques such as Bayesian optimization (BO) which is a sequential design strategy for global optimization of black-box functions and uses exploration/exploitation trade-off to find optimal solution with minimum number of function calls.

- For convergence, line 9 uses a step size tolerance, i.e., $S(\mathbf{p}^k, \mathbf{p}^{k+1}) = \|\mathbf{p}^{k+1} - \mathbf{p}^k\|$. However, other conditions can be employed, such as functional tolerance, i.e., the algorithm is terminated when the change in design cost $|C_d^{k+1} - C_d^k|$ is within a tolerance ϵ .
- In Section 4 we provide a detailed computational error analysis for CL+VQLS-based solution of polynomial ODEs. Since there are no theoretical results available for run time of VQLS, we employ empirically known results along with our rigorous CL+VQLS error analysis to assess the potential advantage of the nBVQPCO framework over classical methods.

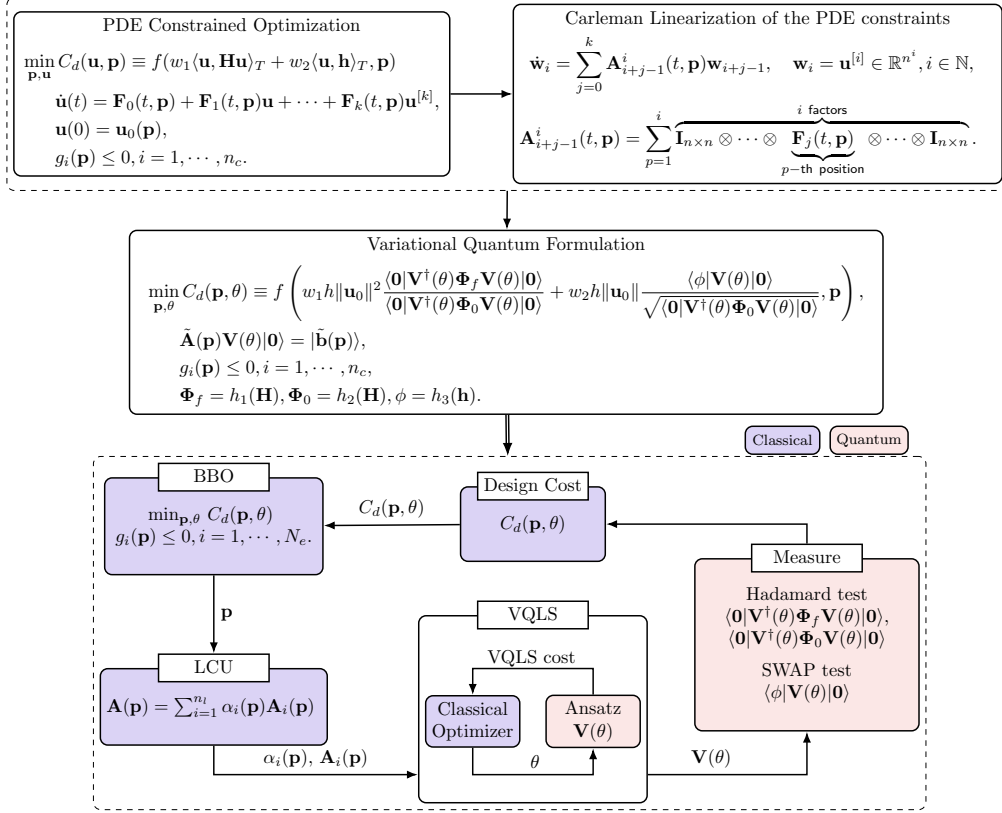


Figure 2. Top: Schematic showing transformation of a nonlinear PDE constrained optimization problem into a variational quantum form. Bottom: Flow diagram of the nBVQPCO framework. VQLS uses an inner level optimization to solve the linear system constraints, arising from the discretization of the underlying PDEs, for given design parameters, and evaluate the quantities related to design cost/objective function. A black box optimizer is used for the outer level optimization to select next set of parameters values based on the evaluated design cost.

4. Computational Complexity and Error Analysis

Consider a system of inhomogeneous quadratic polynomial ODEs (19)

$$\begin{aligned} \dot{\mathbf{u}} &= \mathbf{F}_0(t) + \mathbf{F}_1 \mathbf{u} + \mathbf{F}_2 \mathbf{u}^{[2]}, \\ \mathbf{u}(0) &= \mathbf{u}_0 \in \mathbb{R}^n, \end{aligned} \quad (46)$$

where we have dropped explicit dependence on the parameters \mathbf{p} and time t . Associated with above system, let $\mathbf{w}, \hat{\mathbf{w}}$, and $\tilde{\mathbf{w}}$ be as defined in the Eqs. (22), (25), and (27), and let

$$\mathbf{w}_c = \begin{pmatrix} \mathbf{w}(0) \\ \mathbf{w}(h) \\ \vdots \\ \mathbf{w}(Mh) \end{pmatrix}. \quad (47)$$

See also Figure 1, which illustrates the relationship between these different variables. We make following assumptions.

Assumption 3. For the system (46)

- $\mathbf{F}_1, \mathbf{F}_2$ be time independent matrices.
- Spectral norms $\|\mathbf{F}_1\|, \|\mathbf{F}_2\|, \|\mathbf{F}_0\| = \max_{t \in [0, T]} \|\mathbf{F}_0(t)\|$ are known, and $\max_{t \in [0, T]} \|\dot{\mathbf{F}}_0\| < \infty$
- \mathbf{F}_1 be diagonalizable with eigenvalues λ_j of \mathbf{F}_1 satisfying $\text{Re}(\lambda_n) \leq \dots \leq \text{Re}(\lambda_1) < 0$.
- Let

$$R_2 = \frac{1}{|\text{Re}(\lambda_1)|} (\|\mathbf{u}_0\| \|\mathbf{F}_2\| + \frac{\|\mathbf{F}_0\|}{\|\mathbf{u}_0\|}) < 1. \quad (48)$$

Condition (48) can be conservative and too restrictive to be satisfied in practical applications. As pointed out in the Introduction, for CFD-type applications, one can work with alternative form of conservation laws, e.g., LBM [10], for which the condition Eq. (48) can more readily be met in practical regimes of interest. Additionally, one can employ a rescaling technique proposed in [19] to further improve computational advantage when using the Carleman linearization framework.

Remark 3. Under the condition (48), system (46) can be rescaled $\mathbf{u} \rightarrow \frac{\bar{\mathbf{u}}}{\gamma}$ to (see Appendix in [8] for details)

$$\dot{\bar{\mathbf{u}}} = \bar{\mathbf{F}}_0 + \bar{\mathbf{F}}_1 \bar{\mathbf{u}} + \bar{\mathbf{F}}_2 \bar{\mathbf{u}}^{[2]}, \quad (49)$$

such that the following relations hold

$$\|\bar{\mathbf{F}}_2\| + \|\bar{\mathbf{F}}_0\| < |\text{Re}(\lambda_1)|, \quad (50)$$

$$\|\bar{\mathbf{u}}(0)\| < 1, \quad (51)$$

where $\bar{\mathbf{F}}_0 = \gamma \mathbf{F}_0$, $\bar{\mathbf{F}}_1 = \mathbf{F}_1$, $\bar{\mathbf{F}}_2 = \frac{1}{\gamma} \mathbf{F}_2$ and γ satisfies,

$$\gamma = \frac{1}{\sqrt{\|\mathbf{u}(0)\| r_+}}, \quad (52)$$

with

$$r_{\pm} = \frac{-\text{Re}(\lambda_1) \pm \sqrt{(\text{Re}(\lambda_1))^2 - 4\|\mathbf{F}_2\|\|\mathbf{F}_0\|}}{2\|\mathbf{F}_2\|}. \quad (53)$$

Note that under this rescaling, the value of R_2 in (48) remains unchanged.

Under Assumption 3, Lemmas 1–3 hold, see [8] for the details.

Lemma 1. The error $\boldsymbol{\eta}_j(t) = \mathbf{w}_j(t) - \hat{\mathbf{w}}_j(t)$ for $j \in \{1, 2, \dots, N\}$ is bounded as follows

$$\|\boldsymbol{\eta}_j(t)\| \leq \|\boldsymbol{\eta}(t)\| \leq tN\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1}, \quad (54)$$

where $\boldsymbol{\eta}(t) = (\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T, \dots, \boldsymbol{\eta}_N^T)^T$.

Lemma 2. Suppose that error $\boldsymbol{\eta}(t)$ as introduced in Lemma (1) satisfies

$$\|\boldsymbol{\eta}(t)\| \leq \frac{\|\mathbf{u}(T)\|}{4}, \quad (55)$$

then there exists a sufficiently small h such that

$$\begin{aligned} \|\hat{\mathbf{w}}_j^k - \hat{\mathbf{w}}_j(kh)\| &\leq \|\hat{\mathbf{w}}^k - \hat{\mathbf{w}}(kh)\| \\ &\leq 3N^{2.5}kh^2((\|\mathbf{F}_2\| + \|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\dot{\mathbf{F}}_0\|), \end{aligned} \quad (56)$$

for all $j \in \{1, \dots, N\}$ and $k \in \{0, 1, \dots, M-1\}$. Concretely, the choice of h should satisfy

$$h \leq \min \left\{ \frac{1}{N\|\mathbf{F}_1\|}, \frac{2(|\operatorname{Re}(\lambda_1)| - \|\mathbf{F}_2\| - \|\mathbf{F}_0\|)}{N(|\operatorname{Re}(\lambda_1)|^2 - (\|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\mathbf{F}_1\|^2)} \right\}. \quad (57)$$

Lemma 3. The stiffness κ of $\tilde{\mathbf{A}}$ in (Eq. (27)), recall arising from application of steps 1–3 to the ODE (46), is bounded as follows

$$\kappa \leq 3(M+1), \quad (58)$$

where $M = T/h$.

Lemma 4. Consider a linear system

$$\tilde{\mathbf{A}}|\psi\rangle = |\tilde{\mathbf{b}}\rangle, \quad (59)$$

where $\tilde{\mathbf{A}} \in R^{N \times N}$, $N = 2^n$, $n \in \mathbb{N}$ and $\psi, \mathbf{b} \in R^N$. Then following bounds hold for the different VQLS cost functions as defined in Eqs. (31) and (32)

$$C_{ug} \geq \frac{\epsilon^2}{\kappa^2}, \quad C_g \geq \frac{\epsilon^2}{\kappa^2 \|\tilde{\mathbf{A}}\|}, \quad C_{ul} \geq \frac{\epsilon^2}{n\kappa^2}, \quad C_l \geq \frac{\epsilon^2}{n\kappa^2 \|\tilde{\mathbf{A}}\|}, \quad (60)$$

where κ is the condition number of $\tilde{\mathbf{A}}$, and ϵ is the desired error tolerance

$$\epsilon = \rho(\psi, \psi(\boldsymbol{\theta}^*)),$$

with ρ being the trace norm, and $|\psi(\boldsymbol{\theta}_*)\rangle$ being the approximate VQLS solution.

The proof of above result can be found in [13].

Assumption 4. We assume that the run time of VQLS scales as

$$\mathcal{O}(\log^{8.5} N \kappa \log(1/\epsilon)),$$

where κ is the condition number, N is size of the matrix \mathbf{A} , and ϵ is the desired accuracy of the VQLS solution as described in Lemma 4.

Note that the above assumption is based on an empirical scaling study in [13] and no theoretical guarantees are available. While using Assumption 4 for a given application, one needs to be careful about how well the underlying empirical setup used to obtain this run time scaling of VQLS is met for that application.

Remark 4. Underlying Assumption 4, the following empirical setup was used (see Section 2.2 in [13]).

- The run time of VQLS solution is characterized in terms of time-to-solution, which refers to the number of exact cost function evaluations during the optimization needed to guarantee that ϵ is below a specified value. Since the true solution of the linear system is not known, the value of the cost computed during the VQLS, combined with the error bounds (60), is used to determine the worst case ϵ for the stopping criterion.
- Randomly sampled sparse matrices were used for which the number of LCU terms scales as $\mathcal{O}((\log N)^2)$.
- Normalized local VQLS cost function was used.
- A problem efficient ansatz was used and time-to-solution was averaged over several VQLS runs.
- VQLS was implemented with exact sampling i.e. no finite sampling or shots were used for the computation of the cost function. Furthermore, no hardware noise was considered.

Remark 5. As discussed in Remark 4, we will use the term run time to refer to time-to-solution throughout the rest of the paper unless otherwise stated.

Lemma 5. The trace norm and l_2 norm between $|\psi\rangle$ and $|\phi\rangle$ are related as follows

$$\left(1 - \frac{\|\psi\rangle - |\phi\rangle\|_2^2}{2}\right)^2 + \rho^2(\psi, \phi) = 1. \quad (61)$$

See Lemma 2 in [1] for the proof.

Theorem 1. Consider the ODE system (46). Then under Assumption 3, for any given $\epsilon > 0$, one can choose CL truncation level N (see Eq (A9)) and Euler discretization step size h (see Eq (A5)) such that

$$\| |\mathbf{w}_c\rangle - |\tilde{\mathbf{w}}\rangle \| \leq \epsilon, \quad (62)$$

where $|\tilde{\mathbf{w}}\rangle$ is the solution of the linear system (30) and \mathbf{w}_c is the solution of CL as defined in (47).

The proof of the above theorem is given in Appendix A.1.

Theorem 2. Consider the ODE system (46). Then under Assumption 3, for any given $0 < \epsilon < 1$, one can choose CL truncation level N , Euler discretization step size h and VQLS stopping threshold γ such that

$$\| |\mathbf{w}_c\rangle - |\bar{\mathbf{w}}\rangle \| \leq \epsilon, \quad (63)$$

where $|\bar{\mathbf{w}}\rangle = \mathbf{V}(\boldsymbol{\theta}^*)|\mathbf{0}\rangle$ is solution generated by VQLS for system (30) and \mathbf{w}_c is solution of CL as defined in (47). Furthermore, under the Assumption 4 the run time scales as

$$\mathcal{O}\left(\log^{8.5}\left(\frac{nT^4}{\|\mathbf{w}_c\|^2\epsilon^2}\right) \frac{T^4}{\|\mathbf{w}_c\|^2\epsilon^2} \log(1/\epsilon)\right). \quad (64)$$

Proof. From Theorem 1, choose N, h such that

$$\| |\mathbf{w}_c\rangle - |\tilde{\mathbf{w}}\rangle \| \leq \epsilon/2. \quad (65)$$

Let γ be

$$\gamma = \frac{1 - (1 - \frac{\epsilon^2}{8})^2}{9(M+1)^2 \ln(N_c(M+1))} < 1, \quad (66)$$

where note $M = T/h$ and N_c is the dimension of the truncated CL system (23). Let the VQLS algorithm be terminated under the condition

$$C_l \leq \gamma,$$

then from Lemmas 3 and 4 we obtain

$$\rho^2(\tilde{\mathbf{w}}, \bar{\mathbf{w}}) = (\epsilon')^2 \leq n\kappa^2\gamma \leq 1 - \left(1 - \frac{\epsilon^2}{8}\right)^2$$

where $n = \ln(N_c(M+1))$. It then follows from Lemma 5 that

$$1 - \left(1 - \frac{\|\tilde{\mathbf{w}}\rangle - |\bar{\mathbf{w}}\rangle\|_2^2}{2}\right)^2 \leq 1 - \left(1 - \frac{\epsilon^2}{8}\right)^2,$$

which implies

$$\| |\tilde{\mathbf{w}}\rangle - |\bar{\mathbf{w}}\rangle \|_2 \leq \epsilon/2. \quad (67)$$

Thus, using relations (65) and (67)

$$\| |\mathbf{w}_c\rangle - |\bar{\mathbf{w}}\rangle \| \leq \| |\mathbf{w}_c\rangle - |\tilde{\mathbf{w}}\rangle \| + \| |\tilde{\mathbf{w}}\rangle - |\bar{\mathbf{w}}\rangle \| \leq \epsilon. \quad (68)$$

Furthermore, by Assumption 4 the run time of VLQS behaves as $O((\log(N_c(M+1)))^{8.5} \kappa \log(1/\epsilon'))$ which using Lemma 3 can be simplified as follows

$$\begin{aligned} \mathcal{O}(\log^{8.5}(N_c(M+1))\kappa \log(1/\epsilon')) &= \mathcal{O}(\log^{8.5}(N_c M)3(M+1) \log(1/\epsilon)) \\ &= \mathcal{O}\left(\log^{8.5}\left(n^N \frac{T}{h}\right) \frac{T}{h} \log(1/\epsilon)\right) \\ &= \mathcal{O}\left(\log^{8.5}\left(\frac{nT^4}{\|\mathbf{w}_c\|^2 \epsilon^2}\right) \frac{T^4}{\|\mathbf{w}_c\|^2 \epsilon^2} \log(1/\epsilon)\right), \end{aligned} \quad (69)$$

where we have used the bounds Eq. (A5) and Eq. (A9), and note that

$$\|\mathbf{w}_c\|^2 = \sum_{k=0}^M \|\mathbf{w}(kh)\|^2 = \sum_{k=0}^M \sum_{i=1}^N \|\mathbf{u}(kh)\|^{2i}.$$

□

Remark 6. *Theorem 1 and Theorem 2 can also be applied to higher-order polynomial systems (13), by replacing Assumption 3 with Assumption 5.*

Assumption 5. *For the system (13)*

- Let $\mathbf{F}_i, i = 0, \dots, k$ be time independent matrices with bounded norms.
- Assume $\tilde{\mathbf{F}}_1$ is diagonalizable and its eigenvalues satisfy $\tilde{\lambda}_{n_k} \leq \dots \leq \tilde{\lambda}_1 < 0$, where

$$\tilde{\mathbf{F}}_1 = \begin{pmatrix} \mathbf{A}_1^1 & \mathbf{A}_2^1 & \mathbf{A}_3^1 & \dots & \mathbf{A}_{k-1}^1 \\ \mathbf{A}_1^2 & \mathbf{A}_2^2 & \mathbf{A}_3^2 & \dots & \mathbf{A}_{k-1}^2 \\ 0 & 0 & \dots & \dots & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_{k-2}^{k-1} & \mathbf{A}_{k-1}^{k-1} \end{pmatrix}, \quad (70)$$

and $\mathbf{A}_{i+j-1}^i \in \mathbb{R}^{n^i \times n^{i+j-1}}$ is given by

$$\mathbf{A}_{i+j-1}^i = \sum_{p=1}^i \overbrace{\mathbf{I}_{n \times n} \otimes \dots \otimes \mathbf{F}_j}^{i \text{ factors}} \otimes \dots \otimes \mathbf{I}_{n \times n}. \quad (71)$$

p -th position

which is a generalization of Eq. (21), see [9] for details.

- Let

$$R_k = \frac{1}{|\text{Re}(\tilde{\lambda}_1)|} \left((k-1) \sum_{j=2}^k \|\mathbf{F}_j\| \sqrt{\sum_{i=1}^{k-1} \|\mathbf{x}_0\|^{2i}} + \frac{\|\mathbf{F}_0\|}{\sqrt{\sum_{i=1}^{k-1} \|\mathbf{x}_0\|^{2i}}} \right) < 1. \quad (72)$$

To apply the nBVQPCO framework to the higher-order polynomial case, one can either transform (13) to the form (19) and then follow the procedure described in Section 3.2 or apply CL in Step 1 directly to (13) and then follow the subsequent steps. As shown in [9], the latter is equivalent to the former but leads to more compact CL representation.

Comparison with Classical Approach

Classically to solve the optimization problem (15)–(18) a variety of methods can be used. Consistent with the nBVQPCO framework we use forward Euler to solve the ODE (46) for a given parameter \mathbf{p} , compute the cost function and warp a black box optimizer, e.g., BO.

Applying forward Euler discretization scheme with step size $h = T/M$ to (46) results in a set of nonlinear finite difference equations

$$\hat{\mathbf{u}}^{k+1} = \hat{\mathbf{u}}^k + h(\mathbf{F}_0(kh) + \mathbf{F}_1 \hat{\mathbf{u}}^k + \mathbf{F}_2 (\hat{\mathbf{u}}^k)^{[2]}), \quad (73)$$

where $k = 0, \dots, M-1$, $\hat{\mathbf{u}}^k \approx \mathbf{u}(kh)$ and $\hat{\mathbf{u}}^0 = \mathbf{u}_0$. Let

$$\mathbf{u}_c = \begin{pmatrix} \mathbf{u}(0) \\ \mathbf{u}(h) \\ \vdots \\ \mathbf{u}(Mh) \end{pmatrix}, \quad \tilde{\mathbf{u}} = \begin{pmatrix} \hat{\mathbf{u}}^0 \\ \hat{\mathbf{u}}^1 \\ \vdots \\ \hat{\mathbf{u}}^M \end{pmatrix}. \quad (74)$$

The next theorem shows that one can always choose a step size h so that $\tilde{\mathbf{u}}$ approximates \mathbf{u}_c to within desired error.

Theorem 3. *Consider the ODE system (46). Then under Assumption 3, for any given $\epsilon > 0$ one can choose h (see Eq. (A23)) such that*

$$\|\mathbf{u}_c - \tilde{\mathbf{u}}\| \leq \epsilon. \quad (75)$$

For the proof see [Appendix A.2](#).

Let s be the maximum of sparsity of \mathbf{F}_1 and \mathbf{F}_2 . The computational complexity for solving (73) thus scales as

$$\mathcal{O}(nsM) = \mathcal{O}\left(sn\frac{T}{h}\right) = \mathcal{O}\left(snT\left(h_0 + \frac{C^2T}{\|\mathbf{u}_c\|^2\epsilon^2}\right)\right) = \mathcal{O}\left(\frac{snT^2}{\|\mathbf{u}_c\|^2\epsilon^2}\right),$$

where we have used the bound (A23) and

$$\|\mathbf{u}_c\|^2 = \sum_{k=0}^M \|\mathbf{u}(kh)\|^2.$$

On the other hand, based on [Theorem 2](#), under certain conditions the run time of VQLS based explicit Euler scheme scales polylogarithmically w.r.t n , see [Eq. \(64\)](#). Assuming that the number of outer loop iterations are similar for both this classical and the nBVQPCO framework, the run time of nBVQPCO will scale polylogarithmically with n , and thus could provide a significant computational advantage for simulation-based design problems.

Remark 7. *In the comparison above, note we have not accounted for computational effort for LCU decomposition required within the nBVQPCO framework (see line 5 in [Algorithm 1](#)). This however may not be a significant overhead. As pointed out earlier, depending on how $\tilde{\mathbf{A}}$ depends on the parameters \mathbf{p} , a parameter-dependent LCU decomposition $\{\alpha_i(\mathbf{p}), \mathbf{A}_i\}$ can be pre-computed once, thus saving computational effort: we show that via an example in [Section 5.1](#). Moreover, it may be possible to derive LCU decomposition analytically and implement efficiently, e.g., when using sigma basis, see [Section 5.2](#).*

Remark 8. *Finally, we would like to remind the reader that the conclusion nBVQPCO could be advantageous over equivalent classical methods is based on the strong Assumption 4 regarding VQLS scaling. Recall this scaling is determined empirically using random matrices whose LCU decomposition admits $\mathcal{O}((\log N)^2)$ terms. In the future, it will be worthwhile to characterize VQLS scaling behavior for sparse and structured matrices which typically arise from PDE discretization, and refine the comparison above.*

5. Numerical Study

5.1. Inverse Burgers' problem

Consider the 1D Burgers' equation, which models convective flow $u(x, t)$, on a spatial domain $[0, L]$ with Dirichlet boundary conditions

$$\partial_t u + u\partial_x u = \nu\partial_x^2 u + f(x, t), \quad (76)$$

$$u(x, 0) = u_0(x), u(0, t) = 0, u(L, t) = 0, \quad (77)$$

with $f(x, t)$ being the forcing function. We consider an inverse problem, where given $y(t) = u(x_p, t)$, $t \in [0, T]$ where $x_p \in [0, L]$ is some fixed point, find ν such that

$$\min_{\nu} \frac{\frac{1}{T} \int_0^T (y(t) - u(x_p, t))^2 dt}{u_0^2(x_p)}, \quad (78)$$

s.t. Eqs. (76) and (77) (79)

$$\nu_{min} \leq \nu \leq \nu_{max}. \quad (80)$$

Discretizing the PDE with n_x spatial grid points with grid size $\Delta x = L/(n_x + 1)$ using a central difference scheme, and letting $u_i(t) = u(x_i, t)$ with $x_i = i\Delta x, i = 1, \dots, n_x$, leads to

$$\dot{u}_i = -u_i \frac{u_{i+1} - u_{i-1}}{2\Delta x} + \nu \frac{u_{i+1} - 2u_i + u_{i-1}}{2(\Delta x)^2} + f(x_i, t), \quad (81)$$

for $i = 2, \dots, n_x - 1$. Using Dirichlet boundary conditions, we get at $i = 1$, and $i = n_x$

$$\dot{u}_1 = -u_1 \frac{u_2}{2\Delta x} + \nu \frac{u_2 - 2u_1}{2(\Delta x)^2} + f(x_1, t), \quad (82)$$

and

$$\dot{u}_{n_x} = -u_{n_x} \frac{-u_{n_x-1}}{2\Delta x} + \nu \frac{-2u_{n_x} + u_{n_x-1}}{2(\Delta x)^2} + f(x_{n_x}, t), \quad (83)$$

respectively. Expressing in vector form, the evolution of $\mathbf{u} = (u_1, \dots, u_{n_x})^T$ can be expressed as an ODE

$$\dot{\mathbf{u}} = \mathbf{F}_0(t) + \mathbf{F}_1\mathbf{u} + \mathbf{F}_2\mathbf{u}^{[2]}, \quad (84)$$

with an initial condition $\mathbf{u}(0) = \mathbf{u}_0 = (u_0(x_1), \dots, u_0(x_{n_x}))^T$,

$$\mathbf{F}_0(t) = \begin{pmatrix} f(x_1, t) \\ f(x_2, t) \\ \vdots \\ f(x_{n_x}, t) \end{pmatrix}, \quad \mathbf{F}_1 = \nu \tilde{\mathbf{F}}_1 = \nu \frac{1}{2\Delta x^2} \begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots \\ 0 & 1 & -2 & 1 & \dots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 & -2 \end{pmatrix}, \quad (85)$$

and

$$\mathbf{F}_2(i, j) = \frac{1}{2\Delta x} \begin{cases} -1 & \text{if } i = 1, j = 2 \\ 1 & \text{if } i = n_x, j = n_x - 1 \\ 1 & \text{if } j = (n_x + 1)(i - 1), 1 < i < n_x \\ -1 & \text{if } j = (n_x + 1)(i - 1) + 2, 1 < i < n_x \\ 0 & \text{otherwise} \end{cases}. \quad (86)$$

The system (84) is in the form (19). Applying CL with truncation level N , we can transform (84) to (23) with

$$\mathbf{A}_N(t) = \begin{pmatrix} \nu \tilde{\mathbf{A}}_1^1 & \mathbf{A}_2^1 & 0 & \dots & 0 \\ \mathbf{A}_1^2 & \nu \tilde{\mathbf{A}}_2^2 & \mathbf{A}_3^2 & \dots & 0 \\ 0 & \mathbf{A}_2^3 & \nu \tilde{\mathbf{A}}_3^3 & \mathbf{A}_4^3 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \mathbf{A}_{N-1}^N & \nu \tilde{\mathbf{A}}_N^N \end{pmatrix}, \quad (87)$$

where

$$\tilde{\mathbf{A}}_i^i = \sum_{p=1}^i \overbrace{\mathbf{I}_{n_x \times n_x} \otimes \dots \otimes \tilde{\mathbf{F}}_1}^{i \text{ factors}} \otimes \dots \otimes \mathbf{I}_{n_x \times n_x}. \quad (88)$$

p -th position

We further decompose \mathbf{A}_N as

$$\mathbf{A}_N = \mathbf{A}_{N,1} + \nu \mathbf{A}_{N,2},$$

with

$$\mathbf{A}_{N,1} = \begin{pmatrix} 0 & \mathbf{A}_2^1 & 0 & \cdots & 0 \\ \mathbf{A}_1^2 & 0 & \mathbf{A}_3^2 & \cdots & 0 \\ 0 & \mathbf{A}_2^3 & 0 & \mathbf{A}_4^3 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \mathbf{A}_{N-1}^N & 0 \end{pmatrix}, \quad \mathbf{A}_{N,2} = \begin{pmatrix} \tilde{\mathbf{A}}_1^1 & 0 & 0 & \cdots & 0 \\ 0 & \tilde{\mathbf{A}}_2^2 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{\mathbf{A}}_3^3 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{A}}_N^N \end{pmatrix}.$$

Using the backward Euler time stepping and taking $M = n_t - 1$ time steps with step size $h = T/M$, $\tilde{\mathbf{A}}$ in (29) can be expressed as,

$$\tilde{\mathbf{A}}(\nu) = \tilde{\mathbf{A}}_1 + \nu \tilde{\mathbf{A}}_2,$$

where

$$\tilde{\mathbf{A}}_1 = \begin{pmatrix} \mathbf{I} & 0 & 0 & \cdots \\ -\mathbf{I} & \mathbf{I} - \mathbf{A}_{N,1}(0)h & 0 & \cdots \\ 0 & 0 & \ddots & \ddots \\ 0 & 0 & -\mathbf{I} & \mathbf{I} - \mathbf{A}_{N,1}((M-1)h)h \end{pmatrix},$$

$$\tilde{\mathbf{A}}_2 = \begin{pmatrix} 0 & 0 & 0 & \cdots \\ 0 & -\mathbf{A}_{N,2}(0)h & 0 & \cdots \\ 0 & 0 & \ddots & \ddots \\ 0 & 0 & -\mathbf{A}_{N,2}((M-1)h)h & \end{pmatrix}.$$

Consequently, for VQLS one can apply LCU for $\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2$ once, and then generate a parameter-dependent LCU for $\tilde{\mathbf{A}}$ for any parameter ν for implementation of the nBVQPCO framework.

Furthermore, the optimization problem (78–80) can be approximated as

$$\min_{\nu} \frac{h}{T} \frac{\langle \mathbf{y} - \mathbf{H}\tilde{\mathbf{w}}, \mathbf{y} - \mathbf{H}\tilde{\mathbf{w}} \rangle}{|\langle \mathbf{h}, \tilde{\mathbf{w}} \rangle|^2}, \quad (89)$$

$$\text{s.t.} \quad \tilde{\mathbf{A}}(\nu)\tilde{\mathbf{w}} = \tilde{\mathbf{b}}, \quad (90)$$

$$\nu_{min} \leq \nu \leq \nu_{max}, \quad (91)$$

where $u_0(x_p) = \langle \mathbf{h}, \tilde{\mathbf{w}} \rangle$ for some appropriate vector \mathbf{h} , $\mathbf{y} = (y(0), \dots, y(kh), \dots, y(Mh))^T$ and \mathbf{H} is an appropriate matrix which picks sub-vector $(\hat{w}_{1p}^0, \dots, \hat{w}_{1p}^M)^T$ from $\tilde{\mathbf{w}}$, where \hat{w}_{1p}^k is p -th element taken from the first n_x components $\tilde{\mathbf{w}}_1^k$ of vector $\tilde{\mathbf{w}}^k$ for each $k = 0, \dots, M$. Note that

$$\begin{aligned} \frac{\langle \mathbf{y} - \mathbf{H}\tilde{\mathbf{w}}, \mathbf{y} - \mathbf{H}\tilde{\mathbf{w}} \rangle}{|\langle \mathbf{h}, \tilde{\mathbf{w}} \rangle|^2} &= \frac{\langle \mathbf{y}, \mathbf{y} \rangle}{u_0^2(x_p)} - \frac{2\langle \mathbf{y}, \mathbf{H}\tilde{\mathbf{w}} \rangle}{|u_0(x_p)|\langle \mathbf{h}, \tilde{\mathbf{w}} \rangle} + \frac{\langle \mathbf{H}\tilde{\mathbf{w}}, \mathbf{H}\tilde{\mathbf{w}} \rangle}{|\langle \mathbf{h}, \tilde{\mathbf{w}} \rangle|^2}, \\ &= \frac{\langle \mathbf{y}, \mathbf{y} \rangle}{u_0^2(x_p)} - \frac{2\langle \mathbf{y}, \mathbf{H}\tilde{\mathbf{w}} \rangle}{|u_0(x_p)|\langle \mathbf{h}, \tilde{\mathbf{w}} \rangle} + \frac{\langle \mathbf{H}\tilde{\mathbf{w}}, \mathbf{H}\tilde{\mathbf{w}} \rangle}{|\langle \mathbf{h}, \tilde{\mathbf{w}} \rangle|^2}, \\ &= \frac{\langle \mathbf{y}, \mathbf{y} \rangle}{u_0^2(x_p)} - \frac{2\|\mathbf{H}^T \mathbf{y}\|}{|u_0(x_p)|\|\mathbf{h}\|} \frac{\langle \mathbf{H}^T \mathbf{y} | \tilde{\mathbf{w}} \rangle}{\langle \mathbf{h} | \tilde{\mathbf{w}} \rangle} + \frac{1}{\|\mathbf{h}\|^2} \frac{\langle \tilde{\mathbf{w}} | \mathbf{H}^T \mathbf{H} | \tilde{\mathbf{w}} \rangle}{|\langle \mathbf{h} | \tilde{\mathbf{w}} \rangle|^2}. \\ &\equiv C_d(\nu, |\tilde{\mathbf{w}} \rangle). \end{aligned}$$

Thus, the optimization problem can be expressed in the form of (41–43) as

$$\min_{\nu, |\tilde{\mathbf{w}} \rangle} C_d(\nu, |\tilde{\mathbf{w}} \rangle) \quad (92)$$

$$\text{s.t.} \quad \tilde{\mathbf{A}}(\nu)|\tilde{\mathbf{w}} \rangle = |\tilde{\mathbf{b}} \rangle, \quad (93)$$

$$\nu_{min} \leq \nu \leq \nu_{max}. \quad (94)$$

Note that all the inner products can be computed using appropriate quantum circuits once the solution from VQLS is available as described in Section 3.3.

5.2. LCU Decomposition with Sigma Basis

We use a recursive strategy for tensor decomposition of $\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2$ under the sigma basis as described in Section 3.2. For simplicity, assume that the forcing function $f(x, t) \equiv 0$. Since, $\mathbf{F}_1, \mathbf{F}_2$ as defined in Eq. (85) and Eq. (86) are time independent, $\mathbf{A}_N(t) := \mathbf{A}_N$ is also time independent. Assume $n_x = 2^s, n_t = 2^t$. Note that $\mathbf{A}_N \in \mathbb{R}^{N_c \times N_c}$ where $N_c = \frac{n_x^{N+1} - n_x}{n_x - 1}$ is not a power of 2 and it is padded with zero block of size $N_0 = 2^{sN+1} - N_c$ and thus can be represented with a $sN + 1$ qubit system. Then, $\tilde{\mathbf{A}}_1$ corresponds to $sN + 1 + t$ qubits and can be written as

$$\tilde{\mathbf{A}}_1 := \tilde{\mathbf{A}}_1^{(sN+1+t)} = \hat{\mathbf{A}}_1^{(sN+1+t)} + \underbrace{\sigma_+ \sigma_- \otimes \cdots \otimes \sigma_+ \sigma_-}_{t \text{ times}} \otimes \mathbf{A}_{N,1} h,$$

where

$$\hat{\mathbf{A}}_1^{(sN+1+t)} = \begin{pmatrix} \hat{\mathbf{A}}_1^{(sN+t)} & 0 \\ \mathbf{D}_1^{(sN+t)} & \hat{\mathbf{A}}_1^{(sN+t)} \end{pmatrix},$$

$$\mathbf{D}_1^{(sN+t)} = \begin{pmatrix} 0 & \cdots & -\mathbf{I}_{sN+1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} = -\underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{t \text{ times}} \otimes \mathbf{I}_{sN+1}.$$

Then

$$\hat{\mathbf{A}}_1^{(sN+1+t)} = \mathbf{I}_2 \otimes \hat{\mathbf{A}}_1^{(sN+t)} + \sigma_- \otimes \mathbf{D}_1^{(sN+t)},$$

$$\hat{\mathbf{A}}_1^{(sN+2)} = \begin{pmatrix} \mathbf{I}_{sN+1} - \mathbf{A}_{N,1} h & 0 \\ -\mathbf{I}_{sN+1} & \mathbf{I}_{sN+1} - \mathbf{A}_{N,1} h \end{pmatrix} = \mathbf{I}_2 \otimes (\mathbf{I}_{sN+1} - \mathbf{A}_{N,1} h) - \sigma_- \otimes \mathbf{I}_{sN+1}.$$

Similarly, $\tilde{\mathbf{A}}_2$ can be written as

$$\tilde{\mathbf{A}}_2 = \mathbf{I}_{n_t} \otimes \mathbf{A}_{N,2} - \underbrace{\sigma_+ \sigma_- \otimes \cdots \otimes \sigma_+ \sigma_-}_{t \text{ times}} \otimes \mathbf{A}_{N,2}.$$

Next, we find the tensor decomposition of $\mathbf{A}_{N,1}, \mathbf{A}_{N,2}$. Each diagonal block of $\mathbf{A}_{N,2}$ is of the form

$$\tilde{\mathbf{A}}_i^i = \sum_{p=1}^i \overbrace{\mathbf{I}_{n \times n} \otimes \cdots \otimes \tilde{\mathbf{F}}_1 \otimes \cdots \otimes \mathbf{I}_{n \times n}}^{i \text{ factors}}_{p\text{-th position}}$$

where

$$\tilde{\mathbf{F}}_1 := \tilde{\mathbf{F}}_1^{(s)} = \mathbf{I}_2 \otimes \tilde{\mathbf{F}}_1^{(s-1)} + \sigma_- \otimes \tilde{\mathbf{D}}_1^{(s-1)} + \sigma_+ \otimes (\tilde{\mathbf{D}}_1^{(s-1)})^T,$$

$$\tilde{\mathbf{D}}_1^{(s-1)} = \underbrace{\sigma_+ \otimes \cdots \otimes \sigma_+}_{s-1 \text{ times}}, \quad \tilde{\mathbf{F}}_1^{(1)} = -2\mathbf{I}_2 + \sigma_- + \sigma_+.$$

With these relations, $\tilde{\mathbf{F}}_1^{(s)}$ can be written using $2 \log n_x + 1$ terms. As a result, $\tilde{\mathbf{A}}_i^i$ can be written using $i(2 \log n_x + 1)$ terms out of which the factor $\mathbf{I} \otimes \cdots \otimes \mathbf{I}$ will repeat $i - 1$ times and can be combined into a single term. Hence, $\tilde{\mathbf{A}}_i^i$ can be written using $2i \log n_x + 1$ terms. Finally $\mathbf{A}_{N,2}$ can be decomposed using $\sum_i (2i \log n_x + 1) = N + N(N + 1) \log n_x$ terms.

Next, we focus on the $\mathbf{A}_{N,1}$ term. Since $f(x, t) = 0$, the sub-diagonal blocks \mathbf{A}_i^{i+1} are zero. Each of the super-diagonal block $\mathbf{A}_{i+1}^i \in \mathbb{R}^{n_x^i \times n_x^{i+1}}$. The first term $\mathbf{A}_2^1 = \mathbf{F}_2 \in \mathbb{R}^{n_x \times n_x^2}$ can be split into square blocks \mathbf{F}_2^i of size $n_x \times n_x$

$$\mathbf{A}_2^1 = \mathbf{F}_2 = \begin{pmatrix} \mathbf{F}_2^1 & \mathbf{F}_2^2 & \cdots & \mathbf{F}_2^{n_x} \end{pmatrix}.$$

Let's say that each block \mathbf{F}_2^i , $i = 1, \dots, n_x$ can be decomposed with at most C terms where C is a small constant. We will show below that $C \leq 2$ and there are no repeat factors in the decomposition of \mathbf{F}_2^p and \mathbf{F}_2^q for any $p \neq q$. Then, representing \mathbf{A}_2^1 in $\mathbf{A}_{N,1}$ requires $C n_x$ terms; one for each \mathbf{F}_2^i . Next, we can write

$$\mathbf{A}_3^2 = \mathbf{I} \otimes \mathbf{F}_2 + \mathbf{F}_2 \otimes \mathbf{I} = \begin{pmatrix} \mathbf{F}_2 & & & \\ & \mathbf{F}_2 & & \\ & & \ddots & \\ & & & \mathbf{F}_2 \end{pmatrix} + (\mathbf{F}_2^1 \otimes \mathbf{I} \quad \mathbf{F}_2^2 \otimes \mathbf{I} \quad \dots \quad \mathbf{F}_2^{n_x} \otimes \mathbf{I}).$$

Each block $\mathbf{F}_2^i \otimes \mathbf{I}$ is of size $n_x^2 \times n_x^2$ and can be written directly using the tensor decomposition of \mathbf{F}_2^i . Then, $\mathbf{F}_2 \otimes \mathbf{I}$ requires $C n_x$ terms. On the other hand, each of the diagonal blocks of $\mathbf{I} \otimes \mathbf{F}_2$ requires n_x terms. Thus, representing \mathbf{A}_3^2 in $\mathbf{A}_{N,1}$ requires $C(n_x + n_x^2)$ terms. In general, we can prove by induction that the number of terms needed are as follows:

$$\begin{aligned} \mathbf{A}_2^1 &: C n_x \text{ terms,} \\ \mathbf{A}_3^2 &: C(n_x + n_x^2) \text{ terms,} \\ \mathbf{A}_4^3 &: C(n_x + n_x^2 + n_x^3) \text{ terms,} \\ &\vdots \\ \mathbf{A}_N^{N-1} &: C \sum_{i=1}^{N-1} n_x^i \text{ terms.} \end{aligned}$$

Thus, $\mathbf{A}_{N,1}$ has a total of $\sum_{j=1}^{N-1} \sum_{i=1}^j C n_x^i = \frac{C n_x}{(n_x-1)^2} (n_x^N - 1 - N(n_x - 1))$. Putting it all together,

$$\begin{aligned} \tilde{\mathbf{A}}_1 &: \log n_t + 1 + 2 \left(\frac{C n_x}{(n_x-1)^2} (n_x^N - 1 - N(n_x - 1)) \right), \\ \tilde{\mathbf{A}}_2 &: 2(N + N(N+1) \log n_x), \end{aligned}$$

terms, respectively.

Finally, we look at the decomposition of the blocks \mathbf{F}_2^i and determine the constant C . The first and last block of \mathbf{F}_2 have one non-zero entry with the following structure

$$\begin{aligned} \mathbf{F}_2^1 &= \frac{1}{2\Delta x} \begin{pmatrix} 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ & \vdots & & \ddots & \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} = \frac{-1}{2\Delta x} \underbrace{\sigma_+ \sigma_- \otimes \dots \otimes \sigma_+ \sigma_-}_{s-1 \text{ times}} \otimes \sigma_+, \\ \mathbf{F}_2^{n_x} &= \frac{1}{2\Delta x} \begin{pmatrix} 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 \\ & \vdots & & \ddots & \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} = \frac{1}{2\Delta x} \underbrace{\sigma_- \sigma_+ \otimes \dots \otimes \sigma_- \sigma_+}_{s-1 \text{ times}} \otimes \sigma_-, \end{aligned}$$

and thus have a single decomposition term, i.e. $C = 1$. The remaining \mathbf{F}_2^k blocks have exactly two non-zero entries and can be decomposed into two terms (i.e., $C = 2$), which follows from the result below. In general, the tensor decomposition of a square matrix A_{rc} in sigma basis with a single non-zero entry at (r, c) position for $r, c \in \{0, 1, \dots, n_x - 1\}$ can be determined as given in [Theorem 4](#).

Theorem 4. Let A_s be a $2^s \times 2^s$ matrix with a single non-zero entry in position (r, c) , $r, c \in \{0, 1, \dots, n_x - 1\}$. Let the binary representation of $r = \sum_{p=0}^{s-1} 2^p q_r(p)$ and $c = \sum_{p=0}^{s-1} 2^p q_c(p)$. Then, the tensor decomposition of A in sigma basis is given by $A_s = A_s(r, c)(\otimes_p \sigma(q_r(p), q_c(p)))$, where

$$\sigma(q_r(p), q_c(p)) = \begin{cases} \sigma_+ \sigma_- & \text{if } q_r(p) = 0, q_c(p) = 0 \\ \sigma_+ & \text{if } q_r(p) = 0, q_c(p) = 1 \\ \sigma_- & \text{if } q_r(p) = 1, q_c(p) = 0 \\ \sigma_- \sigma_+ & \text{if } q_r(p) = 1, q_c(p) = 1 \end{cases}$$

Proof. This can be proved by induction. Trivially true for $s = 1$, by the definition of sigma basis. Assume, the statement holds for $s - 1$. Then

$$A_s = A_s(r, c) \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes A_{s-1} \right),$$

where $a, b, c, d \in \{0, 1\}$. Since, there is only one non-zero entry in A_s , only one of a, b, c or d can be non-zero. If $r // 2^{s-1} = c // 2^{s-1} = 0$, where $//$ corresponds to integer division, then $a = 1$. Equivalently, using the binary representation of r, c , if $q_r(s-1) = q_c(s-1) = 0$, then $a = 1$ and the corresponding basis is $\sigma_+ \sigma_-$. Similarly, $b = 1$ if $q_r(s-1) = 0, q_c(s-1) = 1$, $c = 1$ if $q_r(s-1) = 1, q_c(s-1) = 0$, and $d = 1$ if $q_r(s-1) = q_c(s-1) = 1$. Combining this with the induction hypothesis completes the proof. \square

To demonstrate the efficacy of using the sigma basis compared to the Pauli basis, we compute the number of terms needed in the LCU decomposition for different matrix sizes for \tilde{A}_1, \tilde{A}_2 . To vary the matrix size we consider different numbers of spatial/temporal discretization points n_x and n_t , respectively, and different values of CL truncation level N . For the Pauli basis we used the matrix slicing method proposed in [17] to determine the number of terms, while for sigma basis we used analytical expressions derived above. Figure 3 shows the comparison and highlights significantly better scaling for the sigma basis. For example, for $n_x = n_t = 4, N = 4$, LCU with Pauli basis generates 276,000 terms whereas sigma basis gives rise to only 111 terms. Furthermore, computation of the Pauli LCU decomposition was too time consuming as the problem size increases as indicated by the missing data points for $n_x = n_t = 8, 16$ and $N > 2$.

Finally, note that the sigma basis-based LCU decomposition analysis presented in this section, while developed in context of a specific example, is general and can be readily extended to Carleman matrices arising from the linearization of any other polynomial nonlinear ODE.

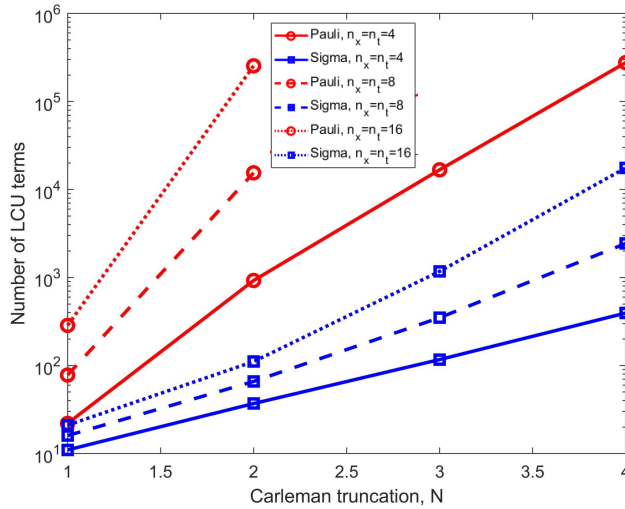


Figure 3. Comparison of the number of LCU terms using sigma basis (blue curve) and Pauli basis (red curves) for different values of CL truncation levels N and number of spatial/temporal discretization points n_x and n_t .

5.3. Implementation

In this section, we demonstrate the application of our nBVQPCO framework to solve the inverse problem described in Section 5.1. We take $L = 0.5$ and $T = 0.35$, and assume that the forcing function $f(x, t) \equiv 0$. The problem is discretized on a spatial grid with $n_x = 4$ grid points leading to $\Delta x = 0.1$. For time discretization we take $n_t = 8$ time steps resulting in $\Delta t = 0.05$. We consider an initial condition of the form $u_0(x) = c \sin(k(x - \Delta x))$, $k = \frac{2\pi}{L}$, which upon discretization takes the form

$$\mathbf{u}_0 = c \begin{pmatrix} \sin(0) \\ \sin(k\Delta x) \\ \vdots \\ \sin((n_x - 1)k\Delta x) \end{pmatrix}. \quad (95)$$

For simplicity we choose c such that $\|\mathbf{u}_0\| = 1$. To simulate the measurement data $y(t)$, $t \in [0, T]$, we take the measurement point to be $x_p = x_2$ where $x_i = i\Delta x$. We integrate the nonlinear ODEs (84), and generate the measurement data $y(t_i) = u_2(t_i)$, $t_i = ih$, $i = 0, \dots, n_t - 1$ using $\nu = 0.07$. In our numerical studies we investigate $N = 1$ and $N = 2$ truncation for the CL.

The nBVQPCO framework was implemented using the PennyLane software framework for quantum computing. PennyLane's `lightning.qubit` device was used as the simulator, `autograd` interface was used as the automatic differentiation library and the adjoint method was used for gradient computations. We do not consider any shot, device or measurement noise throughout this study.

5.3.1. VQLS Implementation and Results

In this section we provide details related to VQLS implementation including the state preparation circuit, and the selected ansatz, optimizer, and LCU decomposition approach.

State Preparation: Here we describe the circuit to construct $\tilde{\mathbf{b}}$ in (29). Given padding of \mathbf{A}_N as discussed in Section 5.2, we equivalently pad $\tilde{\mathbf{b}}$ which takes the form

$$\tilde{\mathbf{b}} = \begin{pmatrix} 0 \\ \hat{\mathbf{w}}(0) \\ 0 \\ h\mathbf{b}(0, \mathbf{p}) \\ \vdots \\ 0 \\ h\mathbf{b}((n_t - 2)h, \mathbf{p}) \end{pmatrix} = \begin{pmatrix} 0 \\ \hat{\mathbf{w}}(0) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad (96)$$

where $\hat{\mathbf{w}}(0) = (\mathbf{u}_0 \ (\mathbf{u}_0^{[2]})^T \dots \ (\mathbf{u}_0^{[N]})^T)^T$ with \mathbf{u}_0 defined in Eq. (95). Since $f(x, t) = 0$, implies $\mathbf{F}_0(t) = 0$ and thus $\mathbf{b}(t, \mathbf{p}) = 0$ in Eq. (25), and the second equality above follows. Since all other entries are zero, it is sufficient to focus on state preparation of the form $\frac{1}{\|\hat{\mathbf{w}}(0)\|} \begin{pmatrix} 0 \\ \hat{\mathbf{w}}(0) \end{pmatrix}$. We generate this state using a sequence of RY, (multi-) controlled RY gates and controlled applications of the circuit for generating \mathbf{u}_0 as follows.

State preparation for \mathbf{u}_0 can be accomplished using RY, CNOT, and PauliZ gates as shown in Figure 4 for different values of n_x . Next, starting from the state $|0\rangle^{n_x}$, we first create the state $|\hat{\mathbf{w}}^1(0)\rangle = \frac{1}{\sqrt{N}}(0, e_1^T, (e_1^{[2]})^T, \dots, (e_1^{[N]})^T)^T$ where $e_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^{n_x}$ using RY gates as shows in Figure 5. Finally, by adding the circuit to prepare \mathbf{u}_0 and its controlled application as illustrated in Figure 6, we sequentially create the states

$$\begin{aligned} |\hat{\mathbf{w}}^1(0)\rangle &\rightarrow \frac{1}{\sqrt{N}}(0^T, \mathbf{u}_0^T, \mathbf{u}_0^T \otimes e_1^T, \dots, \mathbf{u}_0^T \otimes (e_1^{[N-1]})^T)^T \\ &\rightarrow \frac{1}{\sqrt{N}}(0^T, \mathbf{u}_0^T, (\mathbf{u}_0^{[2]})^T, \dots, (\mathbf{u}_0^{[2]})^T \otimes (e_1^{[N-2]})^T)^T \\ &\rightarrow \frac{1}{\sqrt{N}}(0^T, \mathbf{u}_0^T, (\mathbf{u}_0^{[2]})^T, (\mathbf{u}_0^{[3]})^T, \dots, (\mathbf{u}_0^{[3]})^T \otimes (e_1^{[N-3]})^T)^T \\ &\vdots \\ &\rightarrow \frac{1}{\sqrt{N}}(0^T, \mathbf{u}_0^T, (\mathbf{u}_0^{[2]})^T, (\mathbf{u}_0^{[3]})^T, \dots, (\mathbf{u}_0^{[N]})^T)^T = \frac{1}{\|\hat{\mathbf{w}}(0)\|} \begin{pmatrix} 0 \\ \hat{\mathbf{w}}(0) \end{pmatrix}. \end{aligned}$$

Note that since $\|\mathbf{u}_0\| = 1$ and $\|\mathbf{u}_0^{[i]}\| = 1$ for $i = 1, 2, \dots, N$, the states generated above are normalized as required. Finally, since the forcing term $f(x, t) = 0$ in our example, simply adding empty $\log_2 n_t$ wires completes state preparation of the state $\tilde{\mathbf{b}}$. The technique described here can be used in general to prepare the vector $\hat{\mathbf{w}}$ arising due to Carleman linearization with the circuit depth scaling approximately as $\mathcal{O}(N^2)$ to $\mathcal{O}(N^{2.5})$. The depth also depends on the user-specified initial and boundary conditions.

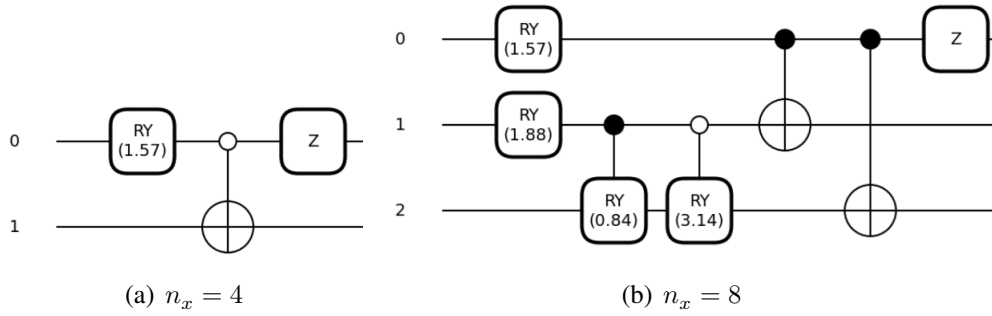


Figure 4. Quantum circuit to prepare the initial condition \mathbf{u}_0 Eq. (95) for $n_x = 4$ and $n_x = 8$ grid points.

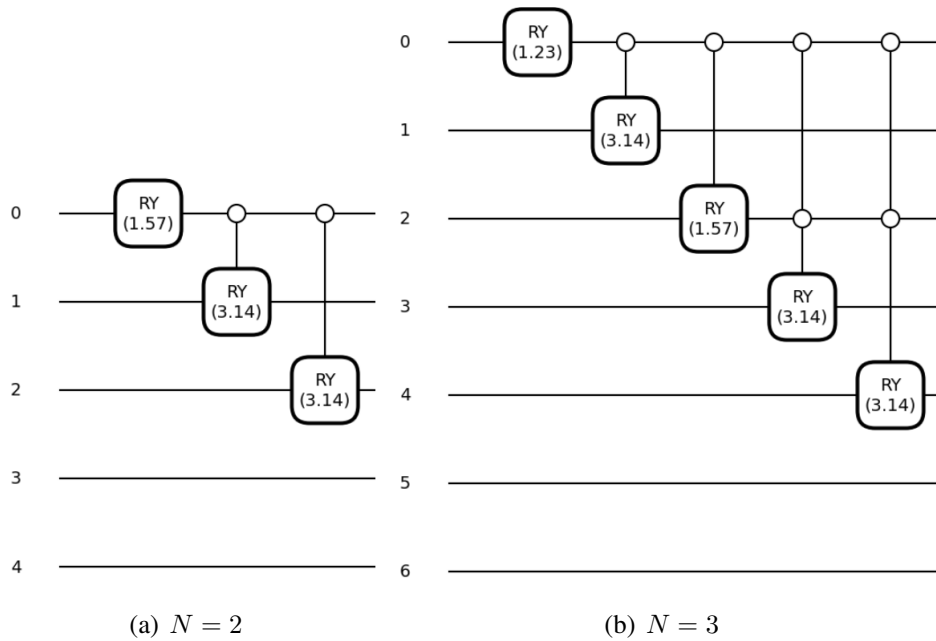


Figure 5. Quantum circuit to prepare the state $\frac{1}{\sqrt{N}}(0^T, e_1^T, (e_1^{[2]})^T, \dots, (e_1^{[N]})^T)^T$ for $n_x = 4$ with (a) $N = 2$ and (b) $N = 3$.

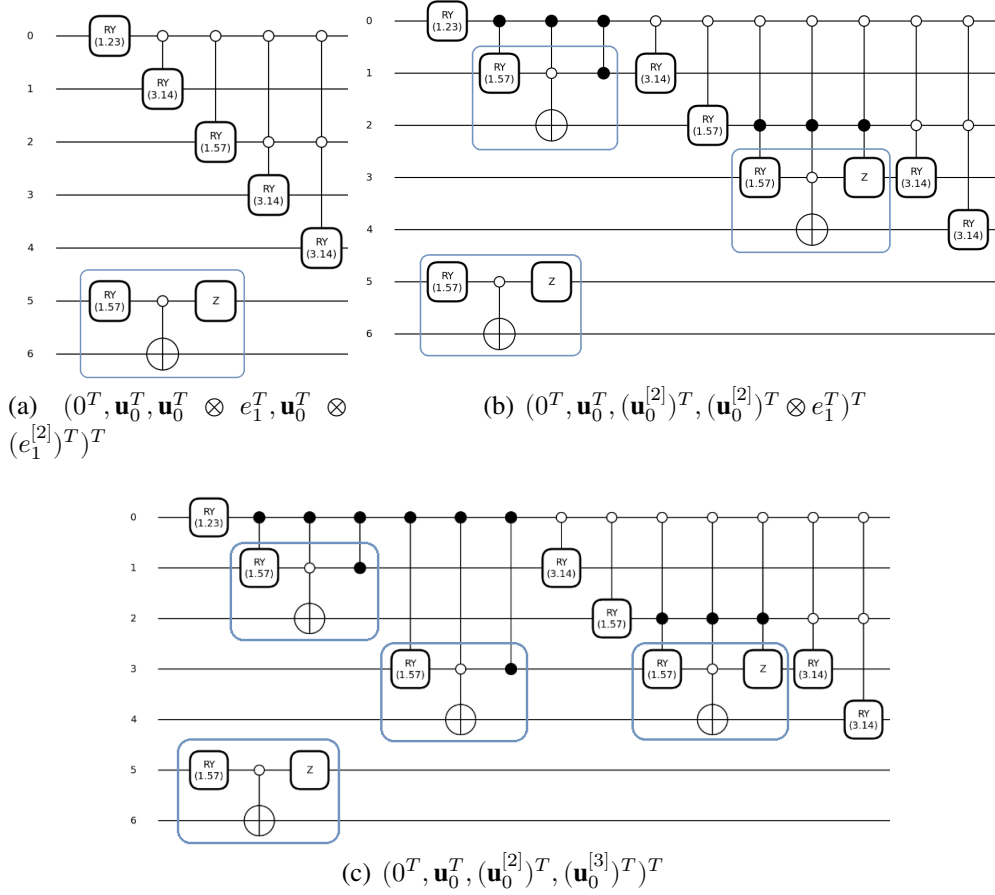


Figure 6. Sequential construction of the quantum circuit to prepare the desired state $(0, \mathbf{u}_0^T, (\mathbf{u}_0^{[2]})^T, (\mathbf{u}_0^{[3]})^T)^T$ with $N = 3$, $n_x = 4$ and \mathbf{u}_0 as defined in Eq. (95).

Ansatz: We use a modified version of the ansatz circuit 9 from [20] with 3 repetitions inspired by the work in [12]. This is shown in Figure 7.

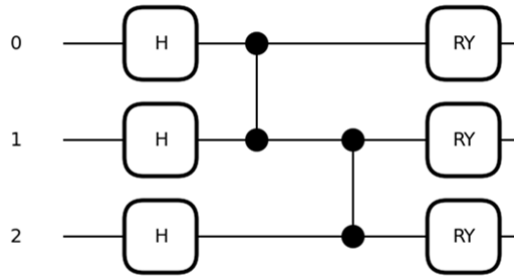


Figure 7. Modified real version of circuit 9 from [20] that is used as the VQLS ansatz.

LCU Decomposition: As discussed in Section 5.2, we use sigma basis-based tensor product decomposition due to its computationally efficiency compared to the Pauli basis for our application. Since sigma basis is non-unitary, we used circuits based on unitary completion [14] to implement the Hadamard test required for computing the VQLS cost functions.

Optimizer: There are several choices of optimizers that can be used with VQLS [21]. We used the Adagrad optimizer from PennyLane with a step size of 0.8. The convergence criteria for VQLS were set to a maximum of 200 iterations. The optimizer was initialized randomly with samples taken from the Beta distribution with shape parameters $\alpha = \beta = 0.5$.

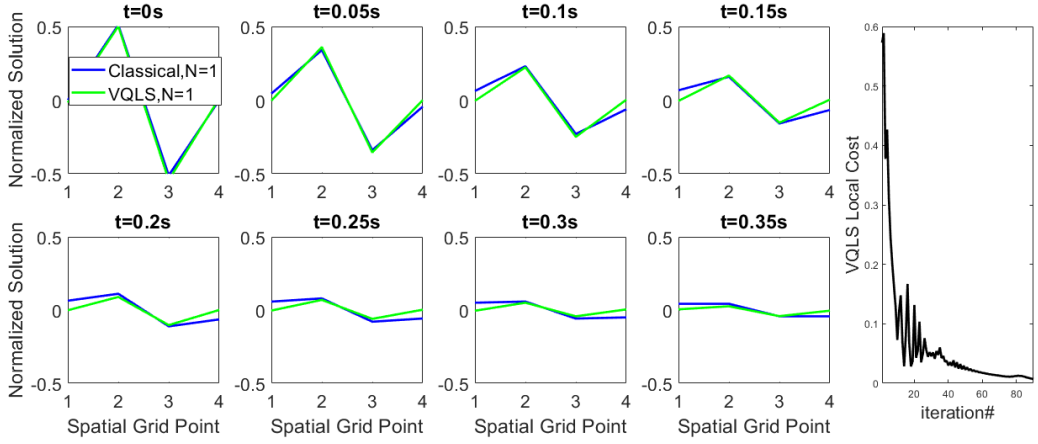
VQLS Results: We study the convergence of VQLS for two Carleman truncation levels $N = 1, 2$ using local VQLS cost function. The VQLS results and convergence of the VQLS cost functions are shown in Figure 8. The VQLS solution is compared with the solution obtained by classically solving the underlying linear system (29). Qualitative comparison of the two solutions indicates that VQLS produces reasonable solutions to the problem. To characterize VQLS solution quality we define aggregated solution error as time average of norm of normalized solution error at each time step, i.e.

$$\mathcal{E} = \frac{1}{n_t} \sum_{i=1}^{n_t} \|\tilde{\mathbf{w}}^i - \bar{\mathbf{w}}^i\|, \quad (97)$$

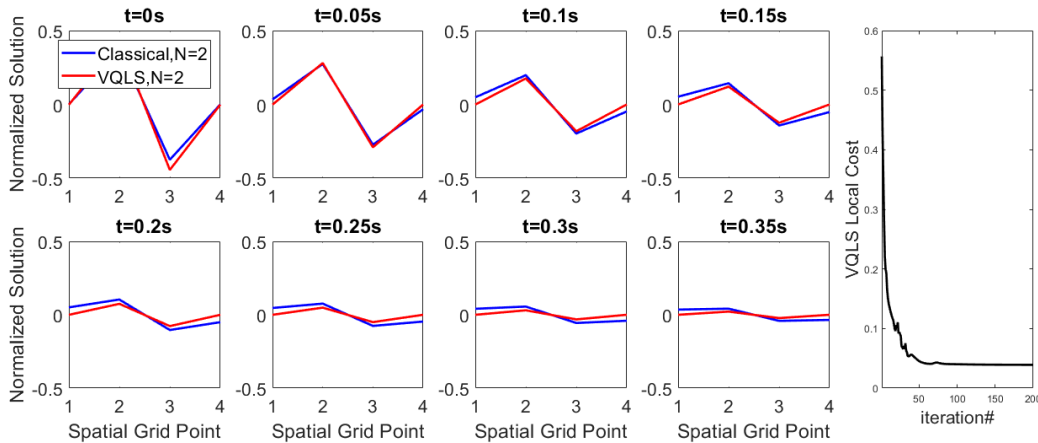
where $\tilde{\mathbf{w}}^i$ is the vector of components taken from $|\tilde{\mathbf{w}}\rangle$ corresponding to classical solution at the time step i , and similarly $\bar{\mathbf{w}}^i$ is the vector of components taken from $|\bar{\mathbf{w}}\rangle$ corresponding to VQLS solution at the i -th time step. Figure 9 shows the error \mathcal{E} for different values of ν and two CL truncation levels. The VQLS solution with $N = 2$ truncation level has lower error compared to the truncation level $N = 1$.

5.3.2. nBVQPCO Results

Next we study the performance of the nBVQPCO framework. The true optimum of the inverse problem is $\nu^* = 0.07$ as we used that value to generate the simulated measurement data $y(t)$. Since the optimization variable is a scalar, we use exhaustive search as the black box optimizer. In this approach we uniformly sample ν in the range $[\nu_{min}, \nu_{max}]$, use CL+VQLS to generate the normalized solution and compute the design cost (92).



(a) CL truncation level $N = 1$.



(b) CL truncation level $N = 2$.

Figure 8. Comparison of normalized solution, obtained classically and via VQLS, of the implicit linear system (29) for the inverse Burgers problem with $\nu = 0.07$ and for two CL truncation levels.

As the curves in Fig. 9b indicate, our algorithm returns the minimum solution at $\nu = 0.06$ for both CL truncation levels $N = 1$ and $N = 2$, which is close to the true optimal value of $\nu^* = 0.07$. The value of R_2 in Eq. (48) ranges from 2.46 to 18 over the range of ν values considered here. Despite being well outside the bound in Eq. (48), CL works well in this practical use case.

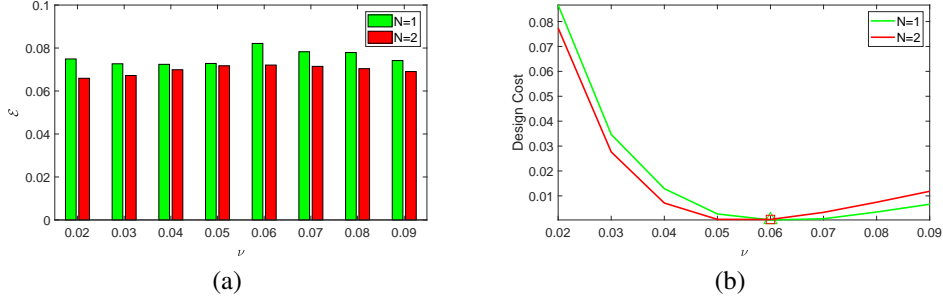


Figure 9. (a) CL+VQLS solution error \mathcal{E} (97) for the inverse Burgers problem for different values of ν and two CL truncation levels $N = 1$ and $N = 2$. (b) Design cost (Eq. (92)) as a function of ν obtained via our nBVQPCO framework for CL truncation levels $N = 1$ and $N = 2$. Also, marked are ν values where the cost takes the smallest value, which is $\nu = 0.06$ for both the truncation levels. The true optima is known to be $\nu = 0.07$.

6. Discussion

In this section we discuss various pros/cons of the nBVQPCO framework and a path toward fully fault-tolerant quantum computing implementation. While the nBVQPCO framework is well suited for NISQ implementation, like other variational quantum algorithms it is difficult to subject it to a fully theoretical complexity analysis. As pointed out in Remark 8, it will be worthwhile to analyze VQLS empirical scaling behavior for sparse and structured matrices. Furthermore, the error and computational complexity results we presented depend on the condition in Eq. (48), which can be conservative and too restrictive to be satisfied in practical applications. As we pointed out earlier, for CFD type applications, one can work with alternative forms of conservation laws, e.g., LBM [10], for which the condition in Eq. (48) can more readily be met in practical regimes of interest. Additionally, one can employ a rescaling technique and higher-order discretization schemes proposed in [19] to further improve computational advantage when using the Carleman linearization framework.

One of the bottlenecks in the nBVQPCO framework is the computation of VQLS cost function which scales polynomially with the number of LCU terms. By replacing conventionally employed Pauli basis with sigma basis, we showed in our application that the number of LCU terms scales more favorably. Ideally, one would like this scaling to have a polylogarithmic dependence on the matrix size, and further enhancing the sigma basis type approaches would be necessary to make the proposed framework scalable. Other variational quantum algorithms such as variational quantum eigensolver (VQE), and quantum approximate optimization algorithm (QAOA) can also significantly benefit from such efficient LCU decompositions.

In the nBVQPCO framework one can replace VQLS with the quantum linear systems algorithm (QLSA) [22–24]. This could be beneficial since QLSA has provable exponential advantage over classical linear system solvers and thus comes with rigorous guarantees unlike variational methods. However, QLSA-based implementation is expected to have high quantum resource needs. For instance, in [11], the authors obtained a detailed quantum resource estimate for implementing the CL-based LBM simulation of incompressible flow fields and concluded that the QLSA-based framework can only be feasible in a fault-tolerant quantum computing setting. This study also highlighted that dominant quantum resource needs arise due to the block encoding step which is required to implement the oracle in QLSA. Block encoding in fault-tolerant implementation can be thought of as an analogous step to LCU decomposition in NISQ implementation. Techniques like sigma basis which exploit sparsity and structure for more efficient LCU decomposition can also be potentially leveraged to make block encoding schemes more efficient.

7. Conclusions

In this paper we presented nBVQPCO, a novel variational quantum framework for nonlinear PDE constrained optimization problems. The proposed framework utilizes Carleman linearization, VQLS algorithm, and a black box optimizer nested in bi-level optimization structure. We presented a detailed computational error and complexity analysis to establish potential benefits of our framework over classical techniques under certain assumptions. We demonstrated the framework on an inverse problem and presented simulation results. The analysis and results demonstrate the correctness of our framework for solving nonlinear simulation-based design optimization problems.

Future work will involve studying and mitigating the effect of device/ measurement noise and making the nBVQPCO framework robust. It will also be important to study the scalability of the framework by applying it to larger problem sizes and implementing it on quantum hardware. It will also be beneficial to explore the application of our framework with other PDE discretization approaches such as finite volume and finite element methods, and extend the error and complexity analysis for higher-order temporal discretization schemes. Finally, exploring and refining the framework to better exploit sparsity/structure and extension to fault-tolerant setting as discussed above are also avenues for future research.

Author Contributions

AG and AS contributed equally to this study including conceptualization, experimental design and execution, theoretical analysis, and writing. HZ contributed to performing the experiments. All authors have read and agreed to the published version of the manuscript.

Funding

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. government.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

No new data was created in this study.

Appendix A

Appendix A.1. Proof of Theorem 1

Proof. Let $\hat{\mathbf{w}}_d = (\hat{\mathbf{w}}^T(0), \dots, \hat{\mathbf{w}}^T(Mh))^T$. Using Lemma 2

$$\|\hat{\mathbf{w}}_d - \tilde{\mathbf{w}}\|^2 = \sum_{k=0}^M \|\hat{\mathbf{w}}(kh) - \tilde{\mathbf{w}}^k\|^2 \leq \left(3N^{2.5}h^2 \left((\|\mathbf{F}_2\| + \|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\dot{\mathbf{F}}_0\| \right)\right)^2 \sum_{k=0}^M k^2,$$

which implies

$$\begin{aligned} \|\hat{\mathbf{w}}_d - \tilde{\mathbf{w}}\| &\leq 3N^{2.5}h^2 \left((\|\mathbf{F}_2\| + \|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\dot{\mathbf{F}}_0\| \right) M^{3/2} \\ &\leq 3N^{2.5}h^{1/2} \left((\|\mathbf{F}_2\| + \|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\dot{\mathbf{F}}_0\| \right) T^{3/2}. \end{aligned}$$

Similarly, using Lemma 1, we get

$$\|\mathbf{w}_c - \hat{\mathbf{w}}_d\| \leq N\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1}h \sqrt{\sum_{k=0}^M k^2} \leq N\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1}T^{3/2}h^{-1/2}.$$

We next show that for given $\epsilon' > 0$, one can choose N and h such that they satisfy a system of inequalities

$$3N^{2.5}h^{1/2} \left((\|\mathbf{F}_2\| + \|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\dot{\mathbf{F}}_0\| \right) T^{3/2} \leq \epsilon'/2, \quad (\text{A1})$$

$$N\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1}T^{3/2}h^{-1/2} \leq \epsilon'/2, \quad (\text{A2})$$

$$\|\boldsymbol{\eta}(t)\| \leq tN\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1} \leq TN\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1} \leq \frac{\|\mathbf{u}(T)\|}{4}, \quad (\text{A3})$$

$$h \leq \min \left\{ \frac{1}{N\|\mathbf{F}_1\|}, \frac{2(|\text{Re}(\lambda_1)| - \|\mathbf{F}_2\| - \|\mathbf{F}_0\|)}{N(|\text{Re}(\lambda_1)|^2 - (\|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\mathbf{F}_1\|^2)} \right\}, \quad (\text{A4})$$

where the last two inequalities comes from Assumption (55) and the requirement (57) in Lemma 2. Combining (A1) and (A4) implies

$$h \leq \left\{ \frac{1}{N\|\mathbf{F}_1\|}, \frac{2(|\operatorname{Re}(\lambda_1)| - \|\mathbf{F}_2\| - \|\mathbf{F}_0\|)}{N(|\operatorname{Re}(\lambda_1)|^2 - (\|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\mathbf{F}_1\|^2)}, \frac{(\epsilon')^2}{36N^5T^3(2(\|\mathbf{F}_2\| + \|\mathbf{F}_1\| + \|\mathbf{F}_0\|)^2 + \|\dot{\mathbf{F}}_0\|^2)} \right\}. \quad (\text{A5})$$

Consequently, we can take h to be of the form $h = K/N^\alpha$, where $\alpha > 1$ and $K > 0$ is an appropriate constant. Thus, (A2) implies

$$N\|\mathbf{F}_2\|\|\mathbf{u}_0\|^{N+1}T^{3/2} \leq \frac{1}{2}\epsilon'\sqrt{h} \leq \frac{\epsilon'\sqrt{K}}{2N^{\alpha/2}},$$

and N needs to satisfy an equation of the form

$$N^{1+\alpha/2}\|\mathbf{u}_0\|^N \leq \delta_1 = \frac{\epsilon'\sqrt{K}}{2T^{3/2}\|\mathbf{F}_2\|\|\mathbf{u}_0\|}, \quad (\text{A6})$$

and similarly from (A3)

$$N\|\mathbf{u}_0\|^N \leq \delta_2 = \frac{\|\mathbf{u}(T)\|}{4T\|\mathbf{F}_2\|\|\mathbf{u}_0\|}. \quad (\text{A7})$$

Note that if we choose $N > 1$ such that

$$N^{1+\alpha/2}\|\mathbf{u}_0\|^N \leq \delta = \min\{\delta_1, \delta_2\}, \quad (\text{A8})$$

then the inequality (A7) is satisfied. Recall, from the Remark 3 we can always assume $\|\mathbf{u}_0\| < 1$. Consider a function of the form

$$f(x) = x^\beta a^x,$$

where $0 < a < q, \beta \geq 1$ and $x \geq 0$. Then it is easy to show that $f(x)$ is monotonically decreasing for $x > \frac{\beta}{\log \frac{1}{a}}$ and $\lim_{x \rightarrow \infty} f(x) = 0$. Thus, one can always find an integer

$$N > \frac{1 + \alpha/2}{\log \frac{1}{\|\mathbf{u}_0\|}}, \quad (\text{A9})$$

such that (A8) is satisfied.

Thus, we conclude that for given ϵ' one can always choose N and h , such that

$$\|\mathbf{w}_c - \hat{\mathbf{w}}_d\| \leq \epsilon'/2, \quad \|\hat{\mathbf{w}}_d - \tilde{\mathbf{w}}\| \leq \epsilon'/2,$$

and so

$$\|\mathbf{w}_c - \tilde{\mathbf{w}}\| \leq \|\mathbf{w}_c - \hat{\mathbf{w}}_d\| + \|\hat{\mathbf{w}}_d - \tilde{\mathbf{w}}\| \leq \epsilon'. \quad (\text{A10})$$

Finally, note that

$$\begin{aligned} \left\| \frac{\mathbf{w}_c}{\|\mathbf{w}_c\|} - \frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|} \right\| &\leq \left\| \frac{\mathbf{w}_c}{\|\mathbf{w}_c\|} - \frac{\tilde{\mathbf{w}}}{\|\mathbf{w}_c\|} + \frac{\tilde{\mathbf{w}}}{\|\mathbf{w}_c\|} - \frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|} \right\| \\ &\leq \frac{\|\mathbf{w}_c - \tilde{\mathbf{w}}\|}{\|\mathbf{w}_c\|} + \frac{\|\|\mathbf{w}_c\| - \|\tilde{\mathbf{w}}\|\|}{\|\mathbf{w}_c\|} \leq \frac{2\|\mathbf{w}_c - \tilde{\mathbf{w}}\|}{\|\mathbf{w}_c\|} \leq \frac{2\epsilon'}{\|\mathbf{w}_c\|}, \end{aligned} \quad (\text{A11})$$

where we have use the fact

$$\frac{\|\|\mathbf{w}_c\| - \|\tilde{\mathbf{w}}\|\|}{\|\mathbf{w}_c\|} \leq \frac{\|\mathbf{w}_c - \tilde{\mathbf{w}}\|}{\|\mathbf{w}_c\|}.$$

Thus, for given ϵ , by choosing $\epsilon' = \|\mathbf{w}_c\|\epsilon/2$ and determining N and h as described above, we can conclude,

$$\left\| \frac{\mathbf{w}_c}{\|\mathbf{w}_c\|} - \frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|} \right\| \leq \epsilon.$$

as required. \square

Appendix A.2. Proof of Theorem 3

Proof. Consider the R.H.S. of (46)

$$\mathbf{F}(t, \mathbf{u}) = \mathbf{F}_0(t) + \mathbf{F}_1\mathbf{u} + \mathbf{F}_2\mathbf{u}^{[2]}. \quad (\text{A12})$$

Let $r > 0$, then for any $\mathbf{u}_1, \mathbf{u}_2 \in \{\mathbf{u} : \|\mathbf{u}\| \leq r\}$

$$\begin{aligned} \|\mathbf{F}(t, \mathbf{u}_1) - \mathbf{F}(t, \mathbf{u}_2)\| &\leq \|\mathbf{F}_1\|\|\mathbf{u}_1 - \mathbf{u}_2\| + \|\mathbf{F}_2\|\|\mathbf{u}_1^{[2]} - \mathbf{u}_2^{[2]}\| \\ &= \|\mathbf{F}_1\|\|\mathbf{u}_1 - \mathbf{u}_2\| \\ &\quad + \|\mathbf{F}_2\|\|\mathbf{u}_1 \otimes \mathbf{u}_1 - \mathbf{u}_1 \otimes \mathbf{u}_2 + \mathbf{u}_1 \otimes \mathbf{u}_2 - \mathbf{u}_2 \otimes \mathbf{u}_2\| \\ &\leq \|\mathbf{F}_1\|\|\mathbf{u}_1 - \mathbf{u}_2\| + \|\mathbf{F}_2\|\|\mathbf{u}_1\|\|\mathbf{u}_1 - \mathbf{u}_2\| \\ &\quad + \|\mathbf{F}_2\|\|\mathbf{u}_2\|\|\mathbf{u}_1 - \mathbf{u}_2\| \\ &\leq \|\mathbf{F}_1\|\|\mathbf{u}_1 - \mathbf{u}_2\| + (\|\mathbf{F}_2\|\|\mathbf{u}_1\| + \|\mathbf{u}_2\|)\|\mathbf{u}_1 - \mathbf{u}_2\|. \end{aligned}$$

Thus

$$\|\mathbf{F}(t, \mathbf{u}_1) - \mathbf{F}(t, \mathbf{u}_2)\| \leq L\|\mathbf{u}_1 - \mathbf{u}_2\| \quad (\text{A13})$$

where

$$L = \|\mathbf{F}_1\| + 2r\|\mathbf{F}_2\|. \quad (\text{A14})$$

As shown in Appendix A.1 in [8], under Assumption 3 the solution of (46) is bounded, i.e.

$$\|\mathbf{u}(t)\| \leq \|\mathbf{u}_0\|, \forall t \geq 0. \quad (\text{A15})$$

Hence, taking the time derivative of (46) and since $\|\dot{\mathbf{F}}_0\|$ is bounded as per Assumption 3, we conclude that $\|\ddot{\mathbf{u}}(t)\|$ is bounded. Then by Taylor expansion

$$\mathbf{u}((k+1)h) = \mathbf{u}(kh) + h\mathbf{F}(kh, \mathbf{u}(kh)) + \tau_k, \quad (\text{A16})$$

where τ_k is the local truncation error which can be bounded as

$$\|\tau_k\| \leq \frac{M_k}{2}h^2, \quad (\text{A17})$$

with $M_k = \max_{t \in [kh, (k+1)h]} \|\ddot{\mathbf{u}}(t)\| < \infty$. Let $M = \max_k M_k$.

Similar to (A15) we next show that for sufficiently small h , iterates of Eq. (73) also satisfy

$$\|\mathbf{u}^k\| \leq \|\mathbf{u}^0\|, \forall k \geq 0, \quad (\text{A18})$$

and thus the Euler scheme is stable. To show this, from Eq. (73) we get

$$\begin{aligned} \frac{(\hat{\mathbf{u}}^{k+1})^T \hat{\mathbf{u}}^{k+1} - (\hat{\mathbf{u}}^k)^T \hat{\mathbf{u}}^k}{h} &= (\hat{\mathbf{u}}^k)^T \mathbf{F}_0(kh) + \mathbf{F}_0(kh)^T \hat{\mathbf{u}}^k + (\hat{\mathbf{u}}^k)^T (\mathbf{F}_1 + \mathbf{F}_1^T) \hat{\mathbf{u}}^k \\ &\quad + (\hat{\mathbf{u}}^k)^T \mathbf{F}_2 (\hat{\mathbf{u}}^k)^{[2]} + ((\hat{\mathbf{u}}^k)^{[2]})^T \mathbf{F}_2^T \hat{\mathbf{u}}^k + h\|\mathbf{F}(kh, \hat{\mathbf{u}}^k)\|^2, \end{aligned}$$

which implies

$$\frac{\|\hat{\mathbf{u}}^{k+1}\|^2 - \|\hat{\mathbf{u}}^k\|^2}{h} \leq 2\|\mathbf{F}_0\|\|\hat{\mathbf{u}}^k\| + 2\text{Re}(\lambda_1)\|\hat{\mathbf{u}}^k\|^2 + 2\|\mathbf{F}_2\|\|\hat{\mathbf{u}}^k\|^3 + h\|\mathbf{F}(kh, \hat{\mathbf{u}}^k)\|^2.$$

Under the condition (48), for $k = 0$, since

$$\|\mathbf{F}_0\| + \text{Re}(\lambda_1)\|\hat{\mathbf{u}}^0\| + \|\mathbf{F}_2\|\|\hat{\mathbf{u}}^0\|^2 < 0,$$

one can choose h

$$h \leq h_0 = -\frac{\|\mathbf{F}_0\| + \text{Re}(\lambda_1)\|\hat{\mathbf{u}}^0\| + \|\mathbf{F}_2\|\|\hat{\mathbf{u}}^0\|^2}{\|\mathbf{F}(kh, \hat{\mathbf{u}}^0)\|^2} \|\hat{\mathbf{u}}^0\|, \quad (\text{A19})$$

so that

$$\|\hat{\mathbf{u}}^1\| \leq \|\hat{\mathbf{u}}^0\|.$$

Following similar argument one can show that $\|\hat{\mathbf{u}}^2\| \leq \|\hat{\mathbf{u}}^1\|$ and so on, concluding (A18).

Next, subtracting (73) from (A16) leads to

$$\mathbf{u}((k+1)h) - \hat{\mathbf{u}}^{k+1} = \mathbf{u}(kh) + h\mathbf{F}(kh, \mathbf{u}(kh)) + \boldsymbol{\tau}_k - (\hat{\mathbf{u}}^k + h\mathbf{F}(kh, \hat{\mathbf{u}}^k)),$$

which implies that the norm of the error $e_k = \|\mathbf{u}(kh) - \hat{\mathbf{u}}^k\|$ satisfies

$$e_{k+1} \leq (1 + hL)e_k + \|\boldsymbol{\tau}_k\|, \quad (\text{A20})$$

where we have used the facts (A15), (A18), and (A13) with $L = \|\mathbf{F}_1\| + 2\|\mathbf{F}_2\|\|\mathbf{u}_0\|$. Iterating on k we get

$$\begin{aligned} e_{k+1} &\leq (1 + hL)^{k+1}e_0 + (1 + hL)^k\|\boldsymbol{\tau}_0\| + (1 + hL)^{k-2}\|\boldsymbol{\tau}_2\| + \dots + \|\boldsymbol{\tau}_k\|, \\ e_{k+1} &\leq (1 + hL)^{k+1}e_0 + \max\|\boldsymbol{\tau}_k\| \frac{(1 + hL)^k + 1}{hL}. \end{aligned} \quad (\text{A21})$$

Since $e_0 = 0$ for Euler method, using (A17) and $1 + Lh \leq e^{hL}$ we conclude that

$$\|\mathbf{u}(kh) - \hat{\mathbf{u}}^k\| = e_k \leq Ch,$$

where $C = \frac{M}{2L}(e^{LT} - 1)$. It then follows that

$$\|\mathbf{u}_c - \tilde{\mathbf{u}}\|^2 = \sum_{k=0}^M \|\mathbf{u}(kh) - \hat{\mathbf{u}}^k\|^2 \leq (Ch)^2 M,$$

which implies

$$\|\mathbf{u}_c - \tilde{\mathbf{u}}\| \leq ChM^{1/2} \leq Ch^{1/2}T^{1/2}. \quad (\text{A22})$$

Thus, for any given ϵ' one can choose

$$h \leq \min \left\{ \frac{(\epsilon')^2}{C^2T}, h_0 \right\}, \quad (\text{A23})$$

where h_0 is as defined in (A19), so that

$$\|\mathbf{u}_c - \tilde{\mathbf{u}}\| \leq \epsilon'.$$

Finally, for given ϵ , by choosing $\epsilon' = \|\mathbf{u}_c\|\epsilon/2$, selecting h as described above and following similar step (A11) as discussed in the proof of Theorem 1, we conclude

$$\left\| \frac{\mathbf{u}_c}{\|\mathbf{u}_c\|} - \frac{\tilde{\mathbf{u}}}{\|\tilde{\mathbf{u}}\|} \right\| \leq \epsilon.$$

□

References

1. A. Surana and A. Gnanasekaran (2024). ‘‘Variational quantum framework for partial differential equation constrained optimization’’. *arXiv preprint arXiv:2405.16651*.
2. H. Antil, D.P. Kouri, M.-D. Lacasse and D. Ridzal (2018). *Frontiers in PDE-constrained Optimization*, volume 163. Springer.
3. L.T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van BloemenWaanders (2003) ‘‘Large-scale pde-constrained optimization: an introduction’’, in *Large-scale PDE-Constrained Optimization*. Springer, pp. 3–13.
4. M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich (2008). *Optimization with PDE Constraints*, volume 23. Springer Science & Business Media.
5. K. Kowalski and W.-H. Steeb (1991). *Nonlinear Dynamical Systems and Carleman Linearization*. Singapore: World Scientific.

6. A. Amini, C. Zheng, Q. Sun, and N. Motee (2025). “Carleman linearization of nonlinear systems and its finite-section approximations”. *Discrete and Continuous Dynamical Systems - B*, 30: 2. 577–603.
7. M. Forets and A. Pouly (2017). “Explicit error bounds for Carleman linearization.” *arXiv preprint arXiv:1711.02552*.
8. J.-P. Liu, H. Ø. Kolden, H.K. Krovi, N.F. Loureiro, K. Trivisa, and A.M. Childs. (2021). “Efficient quantum algorithm for dissipative nonlinear differential equations”. *Proceedings of the National Academy of Sciences*, 118: 35, e2026805118.
9. A. Surana, A. Gnanasekaran, and T. Sahai (2024). “An efficient quantum algorithm for simulating polynomial dynamical systems”. *Quantum Information Processing*, 23: 3, 1–22.
10. X. Li, X. Yin, N. Wiebe, J. Chun, G. K. Schenter, M. S. Cheung, and J. M’ulmenst’adt (2025). “Potential quantum advantage for simulation of fluid dynamics”. *Physical Review Research* 7, 013036.
11. J. Penuel, A. Katarawa, P.D Johnson, C. Farquhar, Y. Cao, and M.C. Garrett (2024). “Feasibility of accelerating incompressible computational fluid dynamics simulations with fault-tolerant quantum computers”. *arXiv preprint arXiv:2406.06323*.
12. R. Demirdjian, D. Gunlycke, C.A. Reynolds, J.D. Doyle, and S. Tafur. (2022). “Variational quantum solutions to the advection–diffusion equation for applications in fluid dynamics”. *Quantum Information Processing*, 21: 9, 322.
13. C. B.-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles. (2023). “Variational quantum linear solver.” *Quantum*, 7, 1188.
14. A. Gnanasekaran and A. Surana. (2024). “Efficient variational quantum linear solver for structured sparse matrices.” In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1. IEEE, pp. 199–210.
15. M. A. Nielsen and I. L. Chuang (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.
16. J.C. D. Reyes (2015). *Numerical PDE-Constrained Optimization*. Springer.
17. L. Hantzko, L. Binkowski, and S. Gupta. (2024). “Tensorized pauli decomposition algorithm.” *Physica Scripta*, 99: 8, 085128.
18. H.-L. Liu, Y.-S. Wu, L.-C. Wan, S.-J. Pan, Su-Juan Qin, Fei Gao, and Qiao-Yan Wen (2021). “Variational quantum algorithm for the poisson equation.” *Physical Review A*, 104: 2, 022418.
19. P. Costa, P. Schleich, M.ES Morales, and D.W Berry (2023). “Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling.” *arXiv preprint arXiv:2312.09518*.
20. S. Sim, P.D. Johnson, and A. Aspuru-Guzik (2019). “Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms.” *Advanced Quantum Technologies*, 2: 12, 1900070.
21. A. Pellow-Jarman, I. Sinayskiy, A. Pillay, and F. Petruccione (2021). “A comparison of various classical optimizers for a variational quantum linear solver.” *Quantum Information Processing*, 20: 6, 202.
22. P.C.S. Costa, D. An, Y.I.R. Sanders, Y. Su, R. Babbush, and D.W. Berry (2022). “Optimal scaling quantum linear-systems solver via discrete adiabatic theorem.” *PRX quantum*, 3: 4, 040303.
23. D. Dervovic, M. Herbster, P. Mountney, S. Severini, N. Usher, and L. Wossnig (2018). “Quantum linear systems algorithms: a primer.” *arXiv preprint arXiv:1802.08227*.
24. A.W. Harrow, A. Hassidim, and S. Lloyd (2009). “Quantum algorithm for linear systems of equations.” *Physical Review Letters*, 103: 15, 150502.