Incompressible Navier–Stokes solve on noisy quantum hardware via a hybrid quantum–classical scheme

Zhixin Song[1], Robert Deaton[1,2], Bryan Gard[2], Spencer H. Bryngelson[3,4,5]

[1]School of Physics, Georgia Institute of Technology, Atlanta, GA 30332, USA
[2]CIPHER, Georgia Tech Research Institute, Atlanta, GA 30332, USA
[3]School of Computational Science & Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[4]Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[5]George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract

Partial differential equation solvers are required to solve the Navier–Stokes equations for fluid flow. Recently, algorithms have been proposed to simulate fluid dynamics on quantum computers. Fault-tolerant quantum devices might enable exponential speedups over algorithms on classical computers. However, current and upcoming quantum hardware presents noise in the computations, requiring algorithms that make modest use of quantum resources: shallower circuit depths and fewer qubits. Variational algorithms are more appropriate and robust under resource restrictions. This work presents a hybrid quantum–classical algorithm for the incompressible Navier–Stokes equations. Classical devices perform nonlinear computations, and quantum ones use variational algorithms to solve the pressure Poisson equation. A lid-driven cavity problem benchmarks the method. We verify the algorithm via noise-free simulation and test it on noisy IBM superconducting quantum hardware. Results show that high-fidelity results can be achieved via this approach, even on current quantum devices. A multigrid preconditioning approach helps avoid local minima. HTree, a tomography technique with linear complexity in qubit count, reduces the quantum state readout time. We compare the quantum resources required for near-term and fault-tolerant solvers to determine quantum hardware requirements for fluid simulations with complexity improvements.

1 Introduction

With the rapid development of hardware platforms and algorithms [1–5], quantum computing has gained enormous attention due to its potential to solve many large and complex problems faster than classical methods. The ability of quantum computers to access an exponentially large Hilbert space and exploit unique quantum properties such as superposition and entanglement are at the heart of many current appealing quantum algorithms. Quantum computing algorithms already promise theoretical speed-up for integer factoring [6], unstructured database search [7], and many other problems of practical interest [8]. Numerous quantum PDE solvers have been proposed based on quantum linear system algorithms. The Harrow–Hassidim–Lloyd (HHL) algorithm [9] claims an exponential speed-up for solving linear systems compared to classical iterative methods such as conjugate gradient [10]. HHL and its later improvements [11–14] have been successfully adopted to solve various linear PDEs [15–20] and aid in the solution of nonlinear ones [21–24].

Quantum algorithms for solving computational fluid dynamics (CFD) problems have also gained much attention due to their nonlinear and non-Hermitian nature [25]. Some of these works focus on longer-term lattice-based methods, including the lattice Boltzmann method (LBM) and lattice gases [26, 26–34], including linearization techniques for handling advective terms [35–37] or otherwise

Email: shb@gatech.edu (Spencer H. Bryngelson)
Code available at https://github.com/comp-physics/NISQ-Quantum-CFD

treating nonlinearities [38]. Li et al. [24] claims a potential quantum advantage for fluid simulation when applying a PDE solver based on quantum linear system algorithms [17] to Carleman-linearized LBM. Although they are likewise long-term algorithms, methods based on oracles have also been proposed [39]. Perhaps closest to the work presented here is that of Lapworth [23], who proposed a hybrid quantum-classical algorithm for solving incompressible Navier–Stokes equations based on the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm [40] and an HHL linear solver.

The current noisy-intermediate scale quantum (NISQ) hardware capability [41] has qubits with meaningfully high error rates. NISQ methods for solving fluid dynamics problems are sparse, and we focus on them here. Most PDE solves suitable to the NISQ-era use variational strategies [5, 42], including the use of quantum algorithms for pressure Poisson solves in the spectral [43, 44] and physical domain [23, 45].

These variational quantum algorithms use shallow parametrized quantum circuits to prepare a general quantum state, combined with classical optimizers and a problem-tailored cost function to (iteratively) prepare the solution. Among them, the Variational Quantum Linear Solver (VQLS) [46] is a near-term solution, in some cases, for solving linear systems. Combined with linearization techniques, one can adopt VQLS to solve simple fluid dynamics problems [47–49]. To address general nonlinear problems using variational algorithms, Lubasch et al. [50] propose the quantum nonlinear processing unit to evaluate nonlinear cost functions given multiple trial states. Based on that, Jaksch et al. [51] develop the variational quantum computational fluid dynamics algorithm to solve the viscous Burgers' equation.

This work proposes a hybrid quantum–classical computing scheme for solving incompressible Navier–Stokes equations on NISQ hardware. One can combine classical pressure projection methods and variational quantum algorithms to solve the pressure Poisson equation. Unlike previous approaches focusing on fault-tolerant quantum computers, we perform the calculation on IBM's superconducting quantum processors to solve a 2D flow problem. Our results show that preconditioning for the linear system can improve the trainability and convergence speed of variational quantum algorithms. The hybrid algorithm requires a quantum state readout for each simulation time step. To partially address this well-known readout problem [52], we use an efficient quantum state tomography method for reading out the real-valued statevector in linear time complexity [53]. This tomography method simplifies the measurement setting compared to conventional quantum state tomography [54] and can be applied to other quantum applications beyond PDE solves.

We organize the remainder of this paper as follows. Section 2 briefly review the classical CFD solvers for incompressible Navier–Stokes equations, including Chorin's projection method [55] and the SIMPLE algorithm. Then, we introduce the hybrid quantum–classical scheme for incompressible fluid simulations on NISQ devices in section 3. Section 4 summarizes the simulation and hardware results for a 2D lid-driven cavity flow. We discuss what error mitigation and suppression methods can be adopted into our hybrid approach. We further conduct resource estimation to analyze the hardware requirement for solving the same problem with the HHL algorithm. Section 5 discusses the presented method's limitations and possible extensions.

2 Background

2.1 The incompressible Navier–Stokes equations

The incompressible Navier–Stokes equations are a set of nonlinear PDEs that describe the conservation of momentum and mass of a fluid flow. In dimensionless form, the momentum equation

is

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \boldsymbol{u}, \tag{1}$$

and the mass conservation equation is

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where $\boldsymbol{u} = (u, v)$ is the flow velocity vector, ρ is the density, p is the pressure, and Re is the Reynolds number or ratio of inertial to viscous effects.

2.2 Pressure projection methods for solving incompressible flow

The main difficulty of solving the incompressible Navier–Stokes equations is often the coupling of velocity and pressure under incompressibility constraint (2). To transform the governing equations and avoid the pressure-velocity coupling, derived quantities such as streamfunction and vorticity could be used instead of primitive variables (u, v, p). However, this approach poses difficulties in geometry modeling and setting the boundary conditions for practical 3D problems.

Two numerical schemes for solving viscous incompressible flow equations are pressure-based projection methods and artificial compressibility [56]. The former approach decouples the velocity and pressure calculation using a fractional step or pseudo-time-stepping. Chorin's projection method [55] is a well-known pressure projection method, which follows as

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} = -\frac{1}{\rho}\nabla p^{n+1} - (\boldsymbol{u}^n \cdot \nabla)\,\boldsymbol{u}^n + \mathrm{Re}\nabla^2 \boldsymbol{u}^n, \tag{3}$$

with explicit Euler time discretization and $\boldsymbol{u}^n$ the velocity at n^{th} time step. The time-step Δt is chosen following the CFL criterion: $u\Delta t/\Delta x < 1$. The first step is solving for an intermediate velocity $\boldsymbol{u}^*$ using the discretized momentum equation without the pressure gradient term ($\nabla p^{n+1} = 0$):

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -(\boldsymbol{u}^n \cdot \nabla)\,\boldsymbol{u}^n + \mathrm{Re}\nabla^2 \boldsymbol{u}^n. \tag{4}$$

This velocity field $\boldsymbol{u}^*$ is usually not divergence-free.

In the second step, the pressure Poisson equation is derived by imposing incompressibility constraint (2) on the velocity at the next time-step

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t}\nabla \cdot \boldsymbol{u}^*. \tag{5}$$

This projection step is achieved by solving the Poisson equation and is often more computationally expensive than other steps. Once we solve the Poisson problem, we correct the velocity from $\boldsymbol{u}^*$ to the next time step as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\rho}\nabla p^{n+1}. \tag{6}$$

We iterate the above steps until the residual r converges to a tolerance threshold, here chosen to be

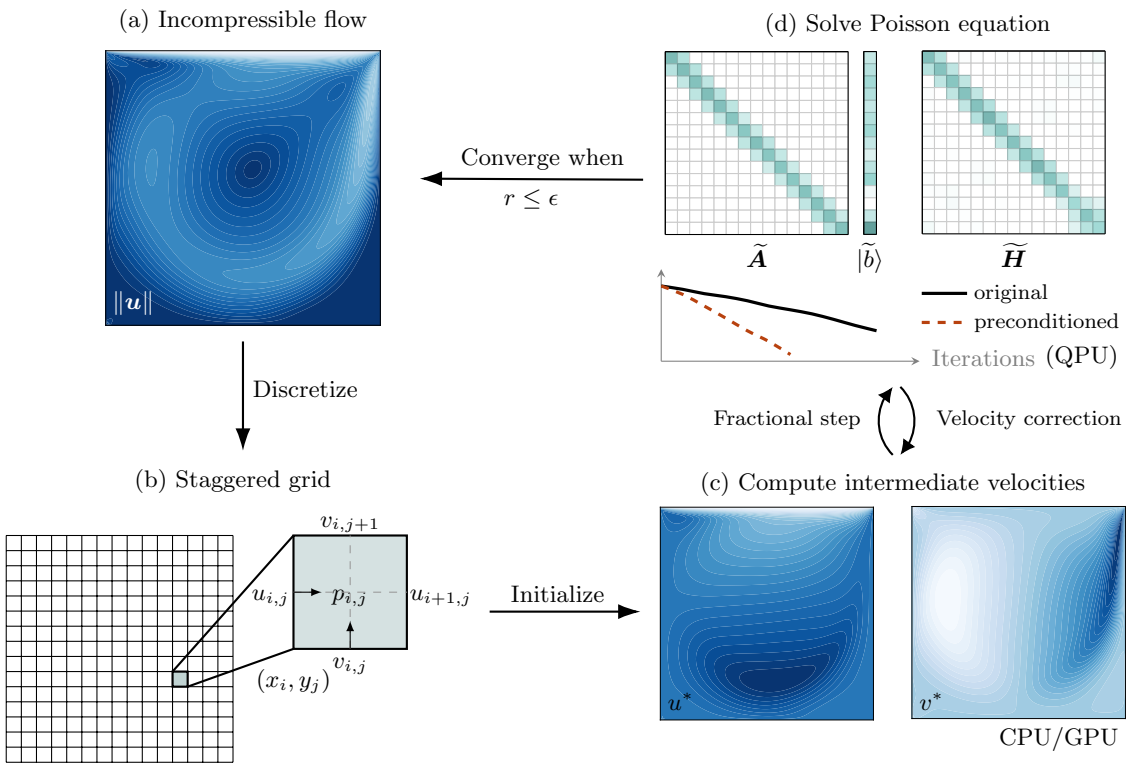

Figure 1: We show a schematic of the hybrid quantum–classical scheme for solving incompressible flow problems. (a) Velocity contour of the 2D lid-driven cavity flow at $\mathrm{Re} = 1000$. (b) A staggered grid allays the well-known checkerboard pressure problem. The pressure is stored at the cell center and the velocities are stored at the cell faces. (c) Calculate intermediate velocities and proceed to the classical computer's next fractional time step. (d) Solve the Poisson equation with proper preconditioning on a quantum computer and obtain velocity corrections. We iterate until the velocity residual converges to the threshold ϵ.

$\epsilon = 10^{-5}$, so our results are insensitive to this particular choice. The root mean square residual is

$$r_{\mathrm{RMS}} = \frac{\sqrt{\sum_{ij} |\phi_{i,j}^{n+1} - \phi_{i,j}^{n}|^2}}{\sqrt{\sum_{ij} |\phi_{i,j}^{n}|^2}}, \tag{7}$$

where the monitored physical quantity ϕ are the velocities (u, v) or the pressure p.

Different pressure projection schemes exist, such as the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm [40]. Compared to basic projection method, the SIMPLE algorithm uses a semi-implicit time-stepping and the velocity correction in (4) accounts for the pressure solution from the last step. Both methods can be integrated into our proposed hybrid quantum–classical scheme; see Lapworth [23] for further discussion.

Another common approach is artificial compressibility or pseudo-compressibility methods [57], where one adds a time-derivative of the pressure term to the continuity equation (2) as

$$\frac{\partial p}{\partial t} + \beta \nabla \cdot \boldsymbol{u} = 0, \tag{8}$$

where β is the artificial compressibility. Such a formulation relaxes the strict requirement to satisfy

mass conservation in each step, and the implicit schemes developed for compressible flows can be directly implemented. However, this method is computationally expensive for transient problems due to dual time-stepping since the pressure field has to go through one complete steady-state iteration cycle in each time step.

2.3 Quantum solutions for the pressure Poisson equation

We focus on pressure projection schemes in this study. We transform the pressure Poisson equation into a large-scale system of linear algebraic equations with numerical discretization

$$\boldsymbol{A}|p\rangle = |b\rangle, \tag{9}$$

where $\boldsymbol{A}$ is a sparse Laplacian matrix that only depends on boundary conditions and discrete formats, $|p\rangle$ is a vector composed of pressures on all discrete grids, and $|b\rangle$ is a vector containing intermediate velocity divergences and boundary conditions.

Among the family of variational quantum algorithms, the Variational Quantum Eigensolver (VQE) [58] and Variational Quantum Linear Solver (VQLS) [46] are promising candidates for solving linear systems on NISQ devices. These two approaches both solve the quantum linear system problem $\boldsymbol{A}|x\rangle = |b\rangle$ by encoding the solution as the ground-state $|\psi_g\rangle = |x\rangle$ of a problem Hamiltonian $\boldsymbol{H}$. VQE was created to find ground-state energy of chemistry molecules based on the Rayleigh–Ritz variational principle [59] and later adopted to solve Poisson equations [60, 61]. VQLS inherits the same variational principle except for the explicit N-qubit circuit construction V to load $|b\rangle = V|0\rangle^{\otimes N}$ and additional quantum subroutines such as the Hadamard test to estimate cost function $C(\vec{\boldsymbol{\theta}}) = 1 - |\langle\psi(\vec{\boldsymbol{\theta}})|x\rangle|^2$. Although it seems natural to use VQLS to solve (9), it requires meaningfully more quantum resources than VQE on NISQ devices (due to these extra components) for general engineering problems [48]. For example, the VQLS circuit runtime for our problem of interest can easily go beyond the coherence time of NISQ hardware. Section 4.4 compares VQE and VQLS.

3 Hybrid quantum–classical Scheme

3.1 Overview

We propose a hybrid quantum–classical scheme for solving incompressible Navier–Stokes problem by combining the Projection method and VQE as a quantum linear system algorithm. Figure 1 illustrates our workflow. We compute the intermediate velocity field $\boldsymbol{u}^*$ on a classical computer. We form the discretized Poisson equation and the solution is encoded as the ground state of an effective Hamiltonian as

$$\boldsymbol{H} = \boldsymbol{A}^\dagger(I - |\widehat{b}\rangle\langle\widehat{b}|)\boldsymbol{A}, \tag{10}$$

where $\boldsymbol{A}^\dagger$ is the Hermitian transpose of operator $\boldsymbol{A}$ and $|\widehat{b}\rangle = |b\rangle/\|b\|$. We use a parameterized quantum circuit $U(\vec{\boldsymbol{\theta}})$ to iteratively prepare the ground state of this Hamiltonian $|\psi(\vec{\boldsymbol{\theta}})\rangle = U(\vec{\boldsymbol{\theta}})|\psi_0\rangle$. By optimizing over the parameters $\vec{\boldsymbol{\theta}} = (\theta_1, \cdots, \theta_m)$ to minimize the cost function

$$\arg\min_{\vec{\boldsymbol{\theta}}} C(\vec{\boldsymbol{\theta}}) = \arg\min_{\vec{\boldsymbol{\theta}}} \langle\psi(\vec{\boldsymbol{\theta}})|\boldsymbol{H}|\psi(\vec{\boldsymbol{\theta}})\rangle. \tag{11}$$

we obtain an approximate solution $|\psi(\vec{\boldsymbol{\theta}}_{\text{opt.}})\rangle \approx |\psi_g\rangle$ when $C(\vec{\boldsymbol{\theta}}_{\text{opt.}}) \leq \gamma$.

To retrieve the pressure solution $|p\rangle$, we require a normalization factor $\mathcal{N}_p$ to scale the quantum solution such that $\mathcal{N}_p|\psi(\vec{\boldsymbol{\theta}}_{\text{opt.}})\rangle \approx |p\rangle$. The normalization is computed via the ratio of the largest element of $\boldsymbol{A}|\psi(\vec{\boldsymbol{\theta}}_{\text{opt.}})\rangle$ and $|\widehat{b}\rangle$. The cost function $C(\vec{\boldsymbol{\theta}})$ has an operational meaning of distance measure between the exact and prepared solutions. One can show the following lower bound holds in general [46],

$$C(\vec{\boldsymbol{\theta}}) \geq \frac{1}{4\kappa^2}\,\mathrm{Tr}^2\left[|\psi(\vec{\boldsymbol{\theta}}_{\text{opt.}})\rangle\langle\psi(\vec{\boldsymbol{\theta}}_{\text{opt.}})| - |\psi_g\rangle\langle\psi_g|\right], \tag{12}$$

where $\mathrm{Tr}[\,\cdot\,]$ is the trace distance and $\kappa(\boldsymbol{A}) := \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|$ is the matrix condition number. The parametrized circuit $U(\vec{\boldsymbol{\theta}})$ is decomposed as a chain of unitary operators

$$U(\vec{\boldsymbol{\theta}}) = \prod_{\ell=1}^{L} U_\ell(\theta_\ell) W_\ell, \tag{13}$$

where $U_\ell(\theta_\ell) = \exp(-i\theta_\ell/2P_\ell)$ with the Pauli product operator $P_\ell \in \{X, Y, Z\}$ and W_ℓ denotes a fixed (non-parameterized) operator such as two-qubit gates that provide entanglement.

3.2 Preconditioning the linear system

Preconditioning techniques are widely used in iterative sparse linear system solvers. The common practice is finding a preconditioner $\boldsymbol{M}^{-1}$ and applying it to the original linear system as

$$\underbrace{\boldsymbol{M}^{-1}\boldsymbol{A}}_{\widetilde{\boldsymbol{A}}}|x\rangle = \underbrace{\boldsymbol{M}^{-1}|b\rangle}_{|\widetilde{b}\rangle}, \tag{14}$$

such that the matrix condition number of the newly assembled matrix is well behaved $\kappa(\widetilde{\boldsymbol{A}}) \ll \kappa(\boldsymbol{A})$. This technique reduces computational complexity since classical sparse linear solvers typically have a $\sqrt{\kappa}$ dependence, compared to the higher-power dependence of quantum algorithms for the same problem.

Clader et al. [62] introduced preconditioning to quantum linear system algorithms via a sparse approximate inverse (SPAI) method, where $\boldsymbol{M}^{-1}$ is constructed by solving a least-squares procedure for each row of the matrix $\boldsymbol{A}$ such that $\|\boldsymbol{M}^{-1}\boldsymbol{A} - I\|_F^2$ is minimized. For NISQ linear solvers, Hosaka et al. [63] demonstrated that incomplete LU factorization as a preconditioning technique could help VQLS prepare a higher fidelity solution with fewer iterations in the optimization loop. The preconditioning involves an LU factorization of $\boldsymbol{A}$ (lower and upper triangular): $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$. One can then drop certain elements in $\boldsymbol{L}$ and $\boldsymbol{U}$ based on the sparsity pattern of $\boldsymbol{A}$ to construct the preconditioner as $\boldsymbol{M} = \widetilde{\boldsymbol{L}}\widetilde{\boldsymbol{U}} \approx \boldsymbol{A}$.

The present hybrid quantum–classical solver observes a better improvement using the smoothed aggregation algebraic multigrid (AMG) preconditioner. An appropriate multigrid method accelerates the linear solver convergence so long as the operator is elliptic positive definite. So, we use AMG preconditioning, and the effective Hamiltonian is

$$\widetilde{\boldsymbol{H}} = \widetilde{\boldsymbol{A}}^\dagger(I - |\widetilde{b}\rangle\langle\widetilde{b}|)\widetilde{\boldsymbol{A}}. \tag{15}$$

Multigrid methods have been recently adopted to variational ansatz design to improve solution quality [64] or enhance trainability [65]. Here, we use AMG preconditioning to improve the trainability of variational algorithms by reducing local traps, as illustrated in fig. 2. The optimization

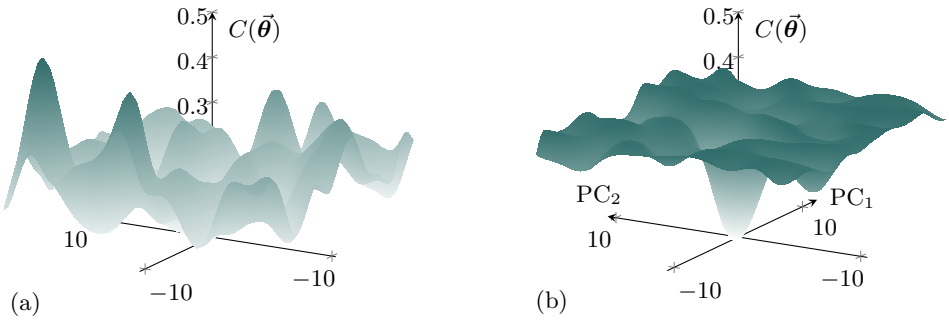

Figure 2: Comparing the original optimization landscape (a) and preconditioned landscape (b). The preconditioned landscape (b) has fewer local minima and thus exhibits better trainability. The global minimum are located at the center $(0, 0)$ in this illustration. PC_j is the j-th principal component.

Method	Qubits:	2	3	4	5	6
QST		3	10	93	310	939
HTree		11	12	13	14	17

Table 1: Runtime (seconds) comparison between two tomography methods on 27-qubit `ibmq_kolkata`.

landscape is visualized based on principal component analysis (PCA) of a successful optimization trajectory using the ORQVIZ package [66]. The scale of the landscapes is normalized by a factor of $\|\boldsymbol{H}\|$ and $\|\widetilde{\boldsymbol{H}}\|$ for visualization purposes, but not repeated for the actual simulations. Notably, we only need to compute the preconditioner once for Chorin's Projection method since the matrix $\boldsymbol{A}$ is the same in all the time steps. On the contrary, preconditioning could introduce overhead in an implicit time-stepping method such as the SIMPLE algorithm as the preconditioner $\boldsymbol{M}^{-1}$ has to be determined separately in each pseudo-time-step.

3.3 Efficient quantum state tomography

As mentioned in section 1, an efficient readout method is a key ingredient for our hybrid quantum–classical algorithm. We introduce the Hadamard tree method (HTree) [53] to address this bottleneck. Standard quantum state tomography (QST) reconstructs the density matrix ρ of arbitrary quantum state with complex amplitudes. So, QST requires collecting $\mathcal{O}(3^N)$ expectation values by measuring each qubit with the Pauli basis $\{X, Y, Z\}$, where N is the number of qubits involved. Reconstructing a physical quantum state satisfying $\mathrm{Tr}[\rho] = 1$ requires a maximum likelihood post-processing technique. HTree instead reconstructs real-valued quantum states $|\psi\rangle = \sum_j \psi_j |j\rangle, \psi_j \in \mathbb{R}$ by sampling the magnitude of state amplitude $|\psi_j|$ according to measurement distribution $|\psi_j|^2$. Hadamard gates on the $(N-k)$-th qubits determine the relative sign between amplitudes ψ_j and ψ_{j+2^k}. We compare the runtime- between these two readout methods on real quantum hardware and summarize the results in table 1. We use the Qiskit Experiments package to implement QST [67].

4 Results

4.1 CFD benchmark

Figure 3 shows the 2D lid-driven cavity problem. We consider a square domain with an edge length L, and the upper lid moves with velocity U in the horizontal (x) direction. The other three walls are solid and entail a Dirichlet boundary condition on velocity $\boldsymbol{u}\big|_{\partial\Omega} = 0$. At $t = 0$, the fluid is at rest, $u = v = 0$. We consider a viscous fluid with kinematic viscosity ν and seek solutions to the

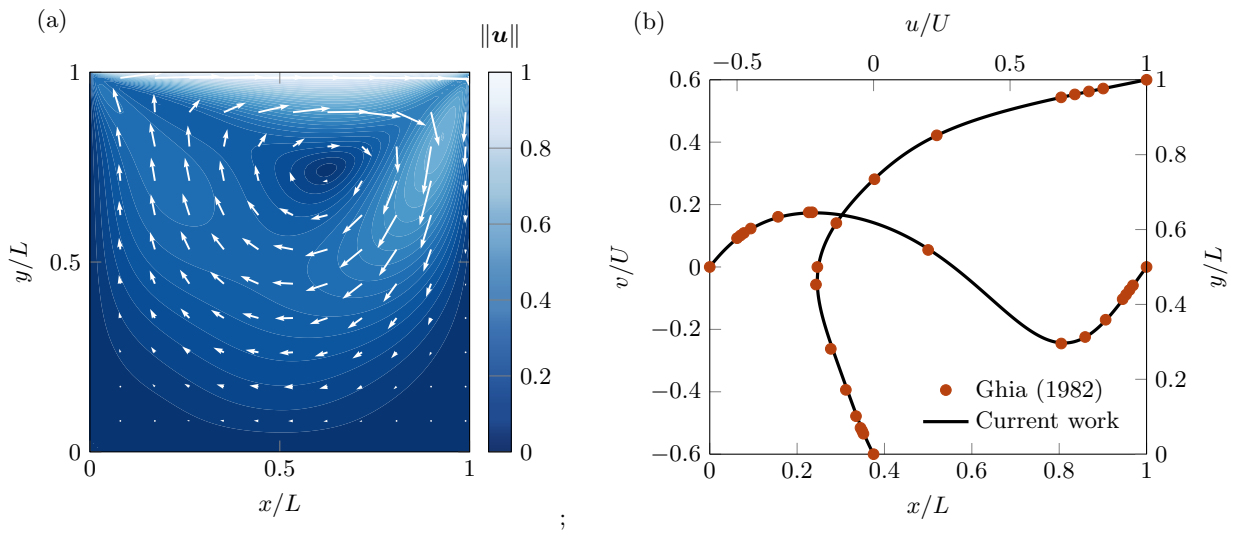

Figure 3: Lid-driven cavity flow at $\mathrm{Re} = 100$ over a 100×100 grid. (a) Velocity magnitude $\|\boldsymbol{u}\| = \sqrt{u^2 + v^2}$ at $t = 10$. (b) Benchmark results and comparison against a reference. The horizontal curve is y-direction velocity v along the horizontal line through the geometric center of the cavity at $y = 0.5$. The vertical curve compares the x-direction velocity u along the vertical line through the geometric center of the cavity at $x = 0.5$ with the reference values of Ghia et al. [68]. The convergence criterion is $r_{\mathrm{RMS}} \leq 10^{-5}$.

incompressible Navier–Stokes equations, eqs. (1) and (2). Solutions are then characterized by the Reynolds number $\mathrm{Re} = UL/\nu$.

We discretize the domain as a uniform staggered grid with n_k grid points in each spatial dimension k. The computational domain thus comprises n_k^2 equally spaced points with grid spacing $\Delta x_k = L/(n_k + 1)$. For a scale resolving simulation, the number of grid points n_k can be determined by resolving the Kolmogorov length scale η as $L/\eta \sim \mathcal{O}(\mathrm{Re}^{3/4})$. We conduct simulations at $\mathrm{Re} = 100$ and verify the solution against results from Ghia et al. [68].

4.2 Simulation results

In this section, we conduct numerical simulations that apply our hybrid quantum–classical CFD solver to simulate the lid-driven cavity flow. The parameterized quantum circuits (PQC) $U(\vec{\boldsymbol{\theta}})$, also known as the ansatz, used in this study is a specific type of hardware-efficient ansatz (HEA) [69], which consists of the gate set $\mathcal{S}_{\mathrm{gate}} \in \{R_y(\theta_\ell), \mathrm{CNOT}\}$. We provide a more detailed analysis to design PQC given hardware resource constraints in Appendix A. All simulations, including optimization loops, are conducted using Qiskit [70] as an open-source toolkit for quantum computing.

We first solve the pressure Poisson problem for $\mathrm{Re} = 100$ at $t = 6$ using the Qiskit statevector simulator. The preconditioning effect is analyzed by repeating the same experiment multiple times with random initial parameters $\theta_\ell \in [-\pi, \pi]$. We report the convergence history of VQE to prepare the ground state solution with fidelity $F \geq 99\%$ in fig. 4. The state fidelity F is a cosine distance measure between the true ground-state $|\psi_g\rangle$ and prepared state $|\psi(\vec{\boldsymbol{\theta}}_{\mathrm{opt.}})\rangle$

$$F\left(|\psi(\vec{\boldsymbol{\theta}}_{\mathrm{opt.}})\rangle, |\psi_g\rangle\right) = \left|\langle\psi(\vec{\boldsymbol{\theta}}_{\mathrm{opt.}})|\psi_g\rangle\right|^2. \tag{16}$$

Among the different choices of optimizers, we use the L-BFGS-B optimizer [71], which we find to consistently outperform other optimizers, like COBYLA and Adam, for statevector simulation.

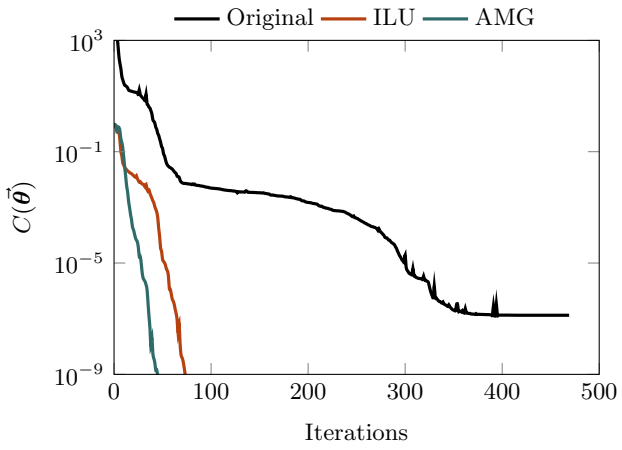

Figure 4: Preconditioning helps VQE converge faster to solve the linear system on a 5×5 mesh. The data are averaged from five VQE runs with random initialization. The PQC is chosen to be the real-amplitude ansatz with five repetitions, the gray line shows results without any preconditioning, ILU refers to incomplete LU factorization, and AMG represents algebraic multigrid. The number of iterations reported here could differ from other literature recordings of the number of function evaluations. For the L-BFGS-B optimizer, each iteration requires about 30 evaluations to approximate gradient information.

Mesh	N_{VQE}	Δt	$\kappa(\boldsymbol{A})$	$\kappa(\widetilde{\boldsymbol{A}}_{\text{ILU}})$	$\kappa(\widetilde{\boldsymbol{A}}_{\text{AMG}})$	$\Delta(\boldsymbol{H})$	$\Delta(\widetilde{\boldsymbol{H}}_{\text{ILU}})$	$\Delta(\widetilde{\boldsymbol{H}}_{\text{AMG}})$
5×5	4	0.2	4.43×10^2	8.01	**1.13**	11.35	14.89	**15.37**
9×9	6	0.1	3.98×10^3	51.51	**1.33**	10.32	13.83	**15.28**
17×17	8	0.05	3.26×10^4	2.94×10^2	**2.46**	9.97	12.17	**15.26**
33×33	10	0.01	2.62×10^5	1.53×10^3	**7.35**	9.12	10.75	**14.89**
65×65	12	0.005	2.10×10^6	7.56×10^3	**33.68**	7.70	9.89	**14.21**

Table 2: Scaling analysis of two preconditioners with different mesh sizes at $\text{Re} = 100$ and $t = 6$, where N_{VQE} is the number of qubits required for VQE to prepare the solution and Δt is the time step size.

Figure 4 show that a proper preconditioner can reduce VQE iterations. We observe the correlation between cost value and other standard distance measure metrics, which include the state fidelity F as $C(\vec{\boldsymbol{\theta}}) \approx 1 - F$ and the ℓ_2 norm as $C(\vec{\boldsymbol{\theta}}) \approx \| |\psi(\vec{\boldsymbol{\theta}}_{\text{opt.}})\rangle - |\psi_g\rangle \|_2^2$ when the AMG preconditioner is applied. The preconditioner aids in the preparation of a higher-fidelity solution by enlarging the spectral gap

$$\Delta(\boldsymbol{H}) = \log_{10} \frac{\left|\lambda_1(\boldsymbol{H})\right|}{\left|\lambda_0(\boldsymbol{H})\right|}, \tag{17}$$

where λ_0 is the smallest eigenvalue of $\boldsymbol{H}$ and λ_1 is the second smallest eigenvalue. A large spectral gap means more space for the classical optimizer to explore the solution once $C(\vec{\boldsymbol{\theta}}) \leq \lambda_1$. We further analyze the effect of preconditioners at different mesh sizes in table 2. The AMG preconditioner consistently creates a larger spectral gap and keeps the matrix condition number low.

Next, we consider more realistic cases, including finite sampling noise and quantum decoherence noise. We repeat the same experiment on the Qiskit QASM simulator and switch to the COBYLA optimizer due to its noise resilience performance. Figure 5 shows the convergence history. In the noise-free simulation, VQE can converge to an optimal cost $C(\vec{\boldsymbol{\theta}}_{\text{opt.}})$ of accuracy up to 10^{-4} and a state fidelity of 99.99% under 700 iterations. If we let the optimizer keep going until 1600

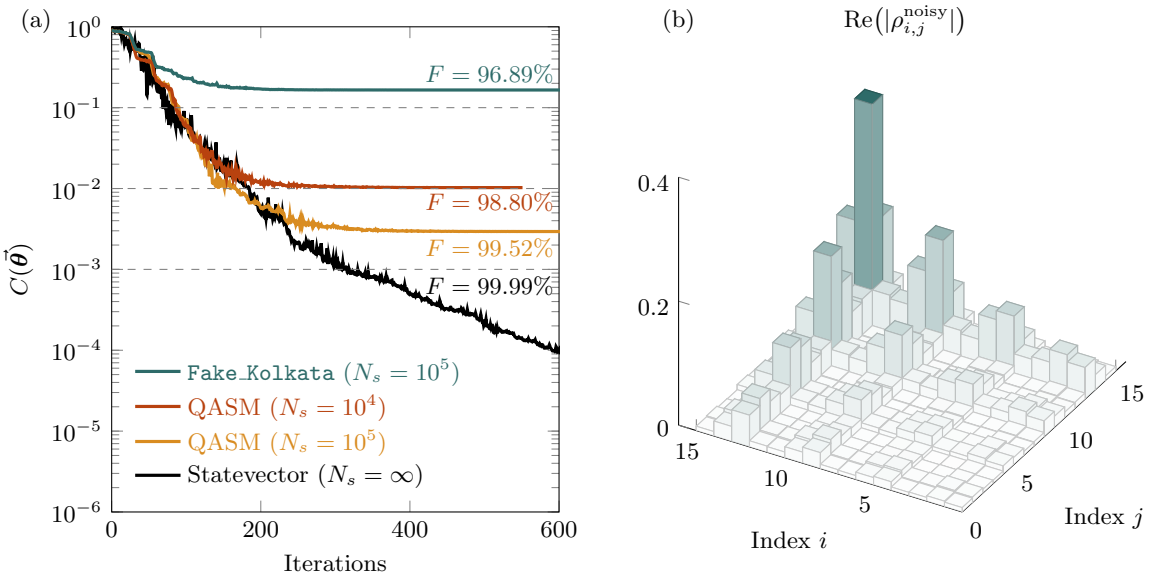

Figure 5: (a) Effect of quantum/classical noise on solving the 4-qubit Poisson problem at $t = 6$ using VQE. (b) Tomography result of the state $\rho_{\mathrm{noisy}} = \mathcal{M}_{\mathrm{HTree}}(|\psi(\vec{\boldsymbol{\theta}}_{\mathrm{opt.}})\rangle)$ prepared by noisy training on `Fake_Kolkata`, where $\mathcal{M}_{\mathrm{HTree}}(\cdot)$ represent the efficient readout method discussed in section 3.3.

iterations, it will converge to $C(\vec{\boldsymbol{\theta}}_{\mathrm{opt.}}) = 4 \times 10^{-8}$ with state infidelity $1 - F = 4.14 \times 10^{-8}$. When the state fidelity is higher than $F \geq 99.99\%$, we will report the state infidelity $1 - F$ instead. The sampling-based simulation limits this accuracy to $C(\vec{\boldsymbol{\theta}}_{\mathrm{opt.}}) \geq 1/\sqrt{N_s}$, where s is the shot number. This limitation is common for variational algorithms due to the accuracy required for the prepared ground state, though one can mitigate this via more samples. We last consider quantum noise from a real quantum device, the 27-qubit `ibmq_kolkata` with past calibration data. This experiment results in a solution with state fidelity 96.89% compared to the true ground state solution.

4.3 Quantum hardware results

We numerically study the proposed hybrid quantum–classical scheme on IBM's superconducting quantum hardware. Considering the noisy nature of quantum hardware, we relax the convergence criterion of the lid-driven cavity flow problem to $r_{\max} \leq 10^{-4}$, where

$$r_{\max} = \max_{\phi \in \{u,v\}} \left| \phi_{i,j}^{n+1} - \phi_{i,j}^{n} \right|. \tag{18}$$

This maximal residual is strictly larger than the root mean square residual of (7). We are interested in two questions: (i) What is the minimum tolerable noise to reach the steady-state solution $r_{\max} \leq 10^{-4}$ for the lid-driven cavity benchmark using the proposed hybrid scheme, and (ii) whether current NISQ hardware meets the requirement. To address (i), we run VQE on a statevector simulator to obtain optimal parameters $\vec{\boldsymbol{\theta}}_{\mathrm{opt.}}$ and readout the state $|\psi_{\mathrm{noisy}}\rangle$ from real hardware or noisy simulators using HTree-based tomography. One could, in principle, conduct end-to-end hardware experiments by including on-device training of VQE. However, this will introduce extra error sources as the hardware noise behavior drifts with time, and on-device training requires prohibitively long simulation times.

We present the results with a 4-qubit test case in fig. 6. The calibration data of IBM's quantum hardware is recorded in table 3. The statevector simulation and QASM simulation with a large

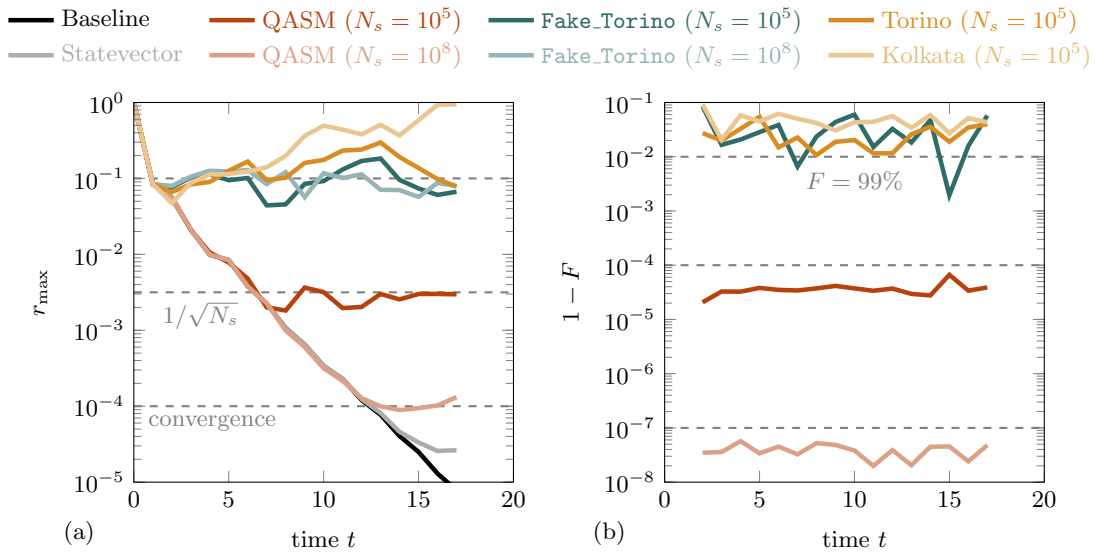

Figure 6: (a) Convergence history of the hybrid algorithm on simulators and hardware. (b) State infidelity as a quality measure of the hybrid solver in each time step.

shot number $N_s = 10^8$ meet the loose convergence criterion $r_{\max} \leq 10^{-4}$. The QASM simulation with $N_s = 10^5$ shots shows an accuracy limit near $1/\sqrt{N_s}$. This conclusion holds for fault-tolerant quantum linear system algorithms, including HHL. Without quantum noise, the solver still needs many shots to converge. As a reference value, 10^5 shots take about 43 seconds on the 133-qubit IBM Heron processor `ibm_torino` and 19 seconds on the 27-qubit Falcon processor `ibmq_kolkata`. The difference is caused by the measurement time t_{mea} in two generations of IBM quantum processors as listed in table 3.

To address the second question (ii) raised at the beginning of this section, we test the hybrid algorithm on a noisy simulator and hardware with a moderate shot number $N_s = 10^5$. The top two curves in fig. 6 (a) are real hardware results. `ibmq_kolkata` is an older processor with higher two-qubit gate error rates, so we observe an accumulation of error in the maximal residual $r_{\max}$. While on the latest 133-qubit processor `ibm_torino`, we observe a lower error rate but still do not reach convergence. The noisy simulator results on `Fake_Torino` assembles the behavior of real hardware. The maximal residual consistently stays on the level $r_{\max} \approx 10^{-1}$. In the presence of quantum noise, increasing the shot number from $N_s = 10^5$ to $N_s = 10^8$ does not help the hybrid solver converge. From the perspective of quantum state tomography, the HTree method achieves good fidelity $95\% - 99\%$ on real hardware as shown on fig. 6 (b). We use dynamical decoupling sequences and measurement error mitigation to suppress the effect of quantum noise. However, the QASM ($N_s = 10^8$) simulation result shows that a converging CFD solution requires at least $1 - F \leq 10^{-7}$. The above results indicate that the current NISQ hardware is still too noisy to implement explicit time-stepping numerical schemes for solving time-dependent PDE problems.

4.4 Resource estimation

This section compares the quantum resources required for three quantum linear systems: VQE, VQLS, and HHL. We determine the most suitable one for NISQ hardware based on qubits required, 2-qubit gate counts, circuit depth, and estimated runtime on the 127-qubit IBM Eagle processor `ibm_sherbrooke`. Table 4 summarizes the results. We manually transpile the quantum circuits

Backend	Qubits	T1 (µs)	T2 (µs)	Error_{1q}	Error_{2q}	t_{2q} (ns)	t_{mea} (ns)
`ibmq_kolkata`	27	103	57	2.55×10^{-4}	8.93×10^{-3}	452	640
`ibm_sherbrooke`	127	263	184	2.17×10^{-4}	7.61×10^{-3}	533	1244
`ibm_torino`	133	168	132	3.08×10^{-4}	3.40×10^{-3}	124	1560

Table 3: Calibration data of the two IBM quantum computers used in this study. The data are averaged among all the qubits.

Solver	Regime	Qubits	2q gates	Circuit depth	Est. runtime (µs)
VQE	NISQ	4	15	59	6.3
VQLS	NISQ	5	1.9×10^3	8.7×10^3	1.1×10^3
HHL	FT	12	3.7×10^5	1.7×10^6	2.4×10^5

Table 4: Resource estimation for the 4-qubit case. The coherence time for `ibm_sherbrooke` is T2 $\approx$ 132 µs.

using the highest optimization level in Qiskit. As mentioned in section 2.3, VQLS would require more quantum resources than VQE on NISQ hardware. It requires one ancillary qubit to perform the Hadamard test and compute the global cost function $C(\vec{\boldsymbol{\theta}})$. This quantum subroutine introduces a deeper circuit mainly due to the controlled application of the unitary component of the problem matrix $\boldsymbol{A}$, which composes 73% of the estimated runtime on `ibm_sherbrooke` and 1016 (1393) 2-qubit gates before (after) transpilation. The results indicate VQLS is only suitable for NISQ hardware when $\boldsymbol{A}$ has an efficient Pauli decomposition such as the Ising-inspired linear system problems introduced in the original VQLS paper [46]. Whether this condition holds for a general engineering problem varies case by case. Meanwhile, HHL is even more resource-intensive as a fault-tolerant algorithm than VQLS. It requires 8 ancillary qubits, and most of the 2-qubit gate counts come from implementing the $\exp(\boldsymbol{A}t)$ operator. The physical runtime required for VQLS and HHL exceeds the NISQ hardware limitation. Although obtaining better gate counts using more advanced transpilation methods for fault-tolerant algorithms like HHL is possible, the conclusions here still hold.

5 Conclusion

We propose and test a hybrid quantum–classical algorithm for solving the incompressible Navier–Stokes equations on NISQ hardware. The algorithm combines the classical pressure projection method and quantum linear system algorithms. We use an efficient tomography method to address the well-known quantum readout problem, largely unexplored in previous art. Preconditioning techniques can improve the trainability of variational quantum linear system algorithms. Although VQLS is widely considered a promising candidate for solving linear systems on NISQ hardware, our resource estimation indicates it would require non-affordable quantum resources unless the matrix $\boldsymbol{A}$ has a simple Pauli decomposition. Our results indicate that current quantum hardware is too noisy to solve time-dependent PDE problems with explicit time-stepping. Besides quantum linear system algorithms, another promising approach for solving PDEs on a quantum computer is via Hamiltonian simulation [72]. With recent hardware demonstrations for linear PDEs [73, 74], Hamiltonian simulation could, eventually, provide a different approach for solving the Navier–Stokes equations.

Declaration of competing interests

The authors declare no conflicts of interest.

Data availability

All code associated with this work is available under the MIT license at https://github.com/comp-physics/NISQ-Quantum-CFD.

Acknowledgements

SHB and BG gratefully acknowledge the support of DARPA under grant no. HR0011-23-3-0006. 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. The authors thank Nicolas Renaud for introducing the HTree method, Siyuan Niu for fruitful discussions on dynamical decoupling sequence designs, and Tianyi Hao for optimization landscape visualization. SHB acknowledges the resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. IBM Quantum services were used for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

References

[1] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, et al., Evidence for the utility of quantum computing before fault tolerance, Nature **618** (2023) 500–505.

[2] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614** (2023) 676–681.

[3] S. Moses, C. Baldwin, M. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, et al., A race track trapped-ion quantum processor, Physical Review X **13** (2023) 041052.

[4] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, et al., Logical quantum processor based on reconfigurable atom arrays, Nature **626** (2024) 58–65.

[5] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al., Noisy intermediate-scale quantum algorithms, Reviews of Modern Physics **94** (2022) 015004.

[6] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Review **41** (1999) 303–332.

[7] L. K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, 1996, pp. 212–219.

[8] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, et al., Quantum algorithms: A survey of applications and end-to-end complexities, arXiv preprint arXiv:2310.03011 (2023).

[9] A. W. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for linear systems of equations, Physical Review Letters **103** (2009) 150502.

[10] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards **49** (1952).

[11] A. M. Childs, R. Kothari, R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM Journal on Computing **46** (2017) 1920–1950.

[12] L. Wossnig, Z. Zhao, A. Prakash, Quantum linear system algorithm for dense matrices, Physical Review Letters **120** (2018) 050502.

[13] Y. Subaşı, R. D. Somma, D. Orsucci, Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing, Physical Review Letters **122** (2019) 060504.

[14] D. An, L. Lin, Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, ACM Transactions on Quantum Computing **3** (2022) 1–28.

[15] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, S. Kais, Quantum algorithm and circuit design solving the Poisson equation, New Journal of Physics **15** (2013) 013021.

[16] A. Montanaro, S. Pallister, Quantum algorithms and the finite element method, Physical Review A **93** (2016) 032324.

[17] D. W. Berry, A. M. Childs, A. Ostrander, G. Wang, Quantum algorithm for linear differential equations with exponentially improved dependence on precision, Communications in Mathematical Physics **356** (2017) 1057–1081.

[18] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei, Y. Gu, Quantum fast Poisson solver: The algorithm and complete and modular circuit design, Quantum Information Processing **19** (2020) 1–25.

[19] A. M. Childs, J.-P. Liu, A. Ostrander, High-precision quantum algorithms for partial differential equations, Quantum **5** (2021) 574.

[20] N. Linden, A. Montanaro, C. Shao, Quantum vs. classical algorithms for solving the heat equation, Communications in Mathematical Physics **395** (2022) 601–641.

[21] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, A. M. Childs, Efficient quantum algorithm for dissipative nonlinear differential equations, Proceedings of the National Academy of Sciences **118** (2021) e2026805118.

[22] S. Jin, N. Liu, Quantum algorithms for computing observables of nonlinear partial differential equations, arXiv preprint arXiv:2202.07834 (2022).

[23] L. Lapworth, A hybrid quantum-classical CFD methodology with benchmark HHL solutions, arXiv preprint arXiv:2206.00419 (2022).

[24] X. Li, X. Yin, N. Wiebe, J. Chun, G. K. Schenter, M. S. Cheung, J. Mülmenstädt, Potential quantum advantage for simulation of fluid dynamics, arXiv preprint arXiv:2303.16550 (2023).

[25] S. Succi, W. Itani, K. Sreenivasan, R. Steijl, Quantum computing for fluids: Where do we stand?, Europhysics Letters **144** (2023) 10001.

[26] B. Ljubomir, Quantum algorithm for the Navier–Stokes equations by using the streamfunction-vorticity formulation and the lattice Boltzmann method, International Journal of Quantum Information **20** (2022) 2150039.

[27] L. Budinski, Quantum algorithm for the advection-diffusion equation simulated with the lattice Boltzmann method, Quantum Information Processing **20** (2021) 57.

[28] J. Yepez, Type-II quantum computers, International Journal of Modern Physics C **12** (2001) 1273–1284.

[29] J. Yepez, Quantum lattice-gas model for the Burgers equation, Journal of Statistical Physics **107** (2002) 203–224.

[30] M. M. Micci, J. Yepez, Measurement-based quantum lattice gas model of fluid dynamics in 2+1 dimensions, Physical Review E **92** (2015) 033302.

[31] S. Kocherla, Z. Song, F. E. Chrit, B. Gard, E. F. Dumitrescu, A. Alexeev, S. H. Bryngelson, Fully quantum algorithm for lattice Boltzmann methods with application to partial differential equations, arXiv preprint arXiv:2305.07148 (2023).

[32] S. Kocherla, A. Adams, Z. Song, A. Alexeev, S. H. Bryngelson, A two-circuit approach to reducing quantum resources for the quantum lattice Boltzmann method, arXiv preprint arXiv:2401.12248 (2024).

[33] B. N. Todorova, R. Steijl, Quantum algorithm for the collisionless Boltzmann equation, Journal of Computational Physics **409** (2020) 109347.

[34] W. Itani, K. R. Sreenivasan, S. Succi, Quantum algorithm for lattice Boltzmann (QALB) simulation of incompressible fluids with a nonlinear collision term, Physics of Fluids **36** (2024).

[35] W. Itani, S. Succi, Analysis of Carleman linearization of lattice Boltzmann, Fluids **7** (2022) 24.

[36] C. Sanavio, S. Succi, Lattice Boltzmann–Carleman quantum algorithm and circuit for fluid flows at moderate Reynolds number, AVS Quantum Science **6** (2024).

[37] C. Sanavio, R. Scatamacchia, C. de Falco, S. Succi, Three Carleman routes to the quantum simulation of classical fluids, Physics of Fluids **36** (2024).

[38] R. Steijl, Quantum algorithms for nonlinear equations in fluid mechanics, Quantum Computing and Communications (2020).

[39] F. Gaitan, Finding flows of a Navier–Stokes fluid through quantum computing, npj Quantum Information **6** (2020) 61.

[40] S. Pantankar, D. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer **15** (1972) 1787–1806.

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

[42] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al., Variational quantum algorithms, Nature Reviews Physics **3** (2021) 625–644.

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

[44] K. Griffin, S. Jain, T. Flint, W. Chan, Investigations of quantum algorithms for direct numerical simulation of the Navier–Stokes equations, Center for Turbulence Research Annual Research Briefs **347** (2019).

[45] S. S. Bharadwaj, K. R. Sreenivasan, Hybrid quantum algorithms for flow problems, Proceedings of the National Academy of Sciences **120** (2023) e2311014120.

[46] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, P. J. Coles, Variational quantum linear solver, Quantum **7** (2023) 1188.

[47] R. Demirdjian, D. Gunlycke, C. A. Reynolds, J. D. Doyle, S. Tafur, Variational quantum solutions to the advection–diffusion equation for applications in fluid dynamics, Quantum Information Processing **21** (2022) 322.

[48] M. Ali, M. Kabel, Performance study of variational quantum algorithms for solving the Poisson equation on a quantum computer, Physical Review Applied **20** (2023) 014054.

[49] Y. Liu, Z. Chen, C. Shu, P. Rebentrost, Y. Liu, S. Chew, B. Khoo, Y. Cui, A variational quantum algorithm-based numerical method for solving potential and Stokes flows, Ocean Engineering **292** (2024) 116494.

[50] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, D. Jaksch, Variational quantum algorithms for nonlinear problems, Physical Review A **101** (2020) 010301.

[51] D. Jaksch, P. Givi, A. J. Daley, T. Rung, Variational quantum algorithms for computational fluid dynamics, AIAA Journal **61** (2023) 1885–1894.

[52] S. Aaronson, Read the fine print, Nature Physics **11** (2015) 291–293.

[53] N. Renaud, HTree, https://github.com/QuantumApplicationLab/qiskit_tomography_toolbox, 2024. Accessed: 2024-03-09.

[54] J. A. Smolin, J. M. Gambetta, G. Smith, Efficient method for computing the maximum-likelihood quantum state from measurements with additive Gaussian noise, Physical Review Letters **108** (2012) 070502.

[55] A. J. Chorin, Numerical solution of the Navier–Stokes equations, Mathematics of Computation **22** (1968) 745–762.

[56] D. Kwak, C. Kiris, C. S. Kim, Computational challenges of viscous incompressible flows, Computers & Fluids **34** (2005) 283–299.

[57] A. J. Chorin, A numerical method for solving incompressible viscous flow problems, Journal of Computational Physics **135** (1997) 118–125.

[58] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, et al., The variational quantum eigensolver: A review of methods and best practices, Physics Reports **986** (2022) 1–128.

[59] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nature Communications **5** (2014) 4213.

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

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

[62] B. D. Clader, B. C. Jacobs, C. R. Sprouse, Preconditioned quantum linear system algorithm, Physical Review Letters **110** (2013) 250504.

[63] A. Hosaka, K. Yanagisawa, S. Koshikawa, I. Kudo, X. Alifu, T. Yoshida, Preconditioning for a variational quantum linear solver, arXiv preprint arXiv:2312.15657 (2023).

[64] C. M. Keller, S. Eidenbenz, A. Bärtschi, D. O'Malley, J. Golden, S. Misra, Hierarchical multigrid ansatz for variational quantum algorithms, arXiv preprint arXiv:2312.15048 (2023).

[65] A. J. Pool, A. D. Somoza, C. M. Keever, M. Lubasch, B. Horstmann, Nonlinear dynamics as a ground-state solution on quantum computers, arXiv preprint arXiv:2403.16791 (2024).

[66] M. S. Rudolph, S. Sim, A. Raza, M. Stechly, J. R. McClean, E. R. Anschuetz, L. Serrano, A. Perdomo-Ortiz, ORQVIZ: Visualizing high-dimensional landscapes in variational quantum algorithms, arXiv preprint arXiv:2111.04695 (2021).

[67] N. Kanazawa, D. J. Egger, Y. Ben-Haim, H. Zhang, W. E. Shanks, G. Aleksandrowicz, C. J. Wood, Qiskit experiments: A Python package to characterize and calibrate quantum computers, Journal of Open Source Software **8** (2023) 5329.

[68] U. Ghia, K. N. Ghia, C. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, Journal of Computational Physics **48** (1982) 387–411.

[69] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549** (2017) 242–246.

[70] Qiskit contributors, Qiskit: An open-source framework for quantum computing, 2023.

[71] R. H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM Journal on Scientific Computing **16** (1995) 1190–1208.

[72] S. Lloyd, Universal quantum simulators, Science **273** (1996) 1073–1078.

[73] Y. Sato, R. Kondo, I. Hamamura, T. Onodera, N. Yamamoto, Hamiltonian simulation for time-evolving partial differential equation by scalable quantum circuits, arXiv preprint arXiv:2402.18398 (2024).

[74] L. Wright, C. M. Keever, J. T. First, R. Johnston, J. Tillay, S. Chaney, M. Rosenkranz, M. Lubasch, Noisy intermediate-scale quantum simulation of the one-dimensional wave equation, arXiv preprint arXiv:2402.19247 (2024).

[75] S. Sim, P. D. Johnson, A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum–classical algorithms, Advanced Quantum Technologies **2** (2019) 1900070.

[76] K. Nakaji, N. Yamamoto, Expressibility of the alternating layered ansatz for quantum computation, Quantum **5** (2021) 434.

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

[78] J. Y. Araz, M. Spannowsky, Classical versus quantum: Comparing tensor-network-based quantum circuits on large hadron collider data, Physical Review A **106** (2022) 062423.

[79] G. Li, Y. Ding, Y. Xie, Tackling the qubit mapping problem for NISQ-era quantum devices, in: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019, pp. 1001–1014.

[80] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics **22** (1951) 79–86.

[81] D. A. Meyer, N. R. Wallach, Global entanglement in multiparticle systems, Journal of Mathematical Physics **43** (2002) 4273–4278.

[82] V. Tripathi, H. Chen, M. Khezri, K.-W. Yip, E. Levenson-Falk, D. A. Lidar, Suppression of crosstalk in superconducting qubits using dynamical decoupling, Physical Review Applied **18** (2022) 024068.

[83] P. Das, S. Tannu, S. Dangwal, M. Qureshi, ADAPT: Mitigating idling errors in qubits via adaptive dynamical decoupling, in: MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture, 2021, pp. 950–962.

[84] S. Niu, A. Todri-Sanial, Effects of dynamical decoupling and pulse-level optimizations on ibm quantum computers, IEEE Transactions on Quantum Engineering **3** (2022) 1–10.

[85] N. Ezzell, B. Pokharel, L. Tewala, G. Quiroz, D. A. Lidar, Dynamical decoupling for superconducting qubits: A performance survey, Physical Review Applied **20** (2023) 064027.

[86] H. R. Grimsley, S. E. Economou, E. Barnes, N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, Nature Communications **10** (2019) 3007.

[87] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, S. Han, QuantumNAS: Noise-adaptive search for robust quantum circuits, in: 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), IEEE, 2022, pp. 692–708.

[88] Z. Liang, J. Cheng, H. Ren, H. Wang, F. Hua, Z. Song, Y. Ding, F. T. Chong, S. Han, X. Qian, et al., NAPA: Intermediate-level variational native-pulse ansatz for variational quantum algorithms, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2024).

Appendix A Quantum circuit and hardware co-design

We use different parameterized quantum circuit (PQC) designs of $U(\vec{\boldsymbol{\theta}})$ to prepare the ground-state solution for pressure Poisson equations on quantum hardware. We consider four design candidates (illustrated in fig. A.7) widely used in the literature of variational quantum algorithms [69, 75–78] and evaluate them in terms of simulation performance and hardware resource cost.

The Hardware-Efficient Ansatz (HEA) [69] was originally proposed for quantum hardware with nearest-neighbor connectivity (such as that of IBM's and Google's superconducting quantum processors) where 2-qubit gates are only applied on adjacent qubits to avoid additional SWAP gates introduced in transpilation [79]. The Real-Amplitude Ansatz (RAA) is a variant of HEA where the general single qubit rotation layer is limited to $R_y(\theta)$ gate and only generates real-valued states. Alternating Layered Ansatz (ALT) is composed of blocks forming local 2-designs and proven to be trainable for circuit depth $D \in \mathcal{O}(\log(n))$ when combined with a local cost function $C_l(\vec{\boldsymbol{\theta}})$ [77]. Finally, the Tensor Network Ansatz (TEN) is a family of circuits inspired by classical Tensor Network architectures such as Matrix Product States (MPS), Tree Tensor Networks (TTN) and Multi-scale Entanglement Renormalisation Ansatz (MERA)[78]. For simplicity, we consider a tensor product ansatz here to probe if entanglement is necessary to find the ground state.

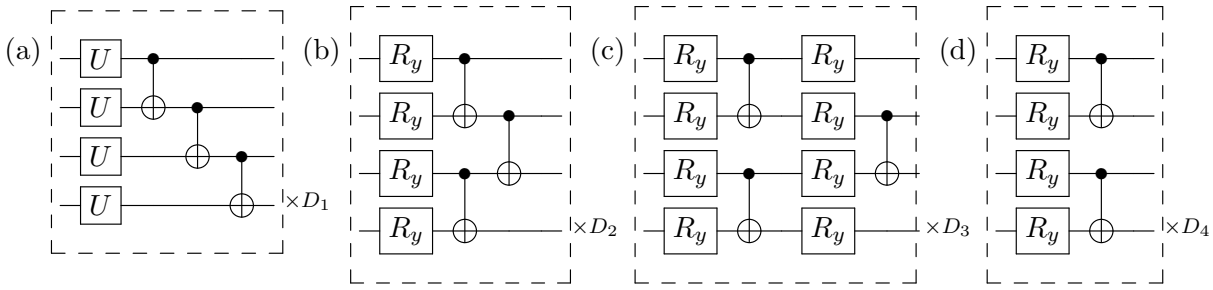

Figure A.7: The ansatze considered in this study. After repeating D_j times of each block, a final rotation layer enhances expressibility, though it is not illustrated further here. (a) HEA uses $U = R_z(\theta_j)R_y(\phi_j)R_z(\varphi_j)$ as a general rotation on the Bloch sphere along with non-parameterized adjacent CNOT gates providing entanglement to prepare the solution state; RAA uses $U = R_y(\theta_j)$ in the rotation layer to reduce the trainable parameters. (b) RAA with a pairwise entanglement layer to reduce the circuit duration of one CNOT gate per block. (c) Alternating Layered Ansatz (ALT). (d) Tensor Network Ansatz (TEN) prepares tensor product states and is expected to have a weaker entangling capability.

Ansatze	Avg. $\bar{F}$	Expr. ↓	Ent. ↑	n_{2q}	$D_{\text{trans.}}$ (DD)	$\tau_{\text{circ.}}$ (µs)	Coherence
HEA	99.99%	**0.0003**	**0.8149**	15	47 (68)	5.76	10.08%
RAA	99.99%	0.1899	0.7428	15	35 (56)	5.76	10.08%
RAA-p	99.27%	0.1895	0.7469	15	34 (54)	5.40	**9.45**%
ALT	83.82%	0.2349	0.5843	8	29 (34)	2.88	5.05%
TEN	53.04%	0.6350	0.3314	10	29 (30)	2.72	4.85%

Table A.5: Summary of a 4-qubit circuit design case study where each PQC candidate contains 24 trainable parameters except for HEA $N_{\text{param}} = 72$. The expressibility and entanglement capability are sampled from 10^3 random samples. The average entanglement for the true ground state $|\psi_g\rangle$ across different simulation times is 0.46. The transpiled circuit depth with dynamical decoupling (DD) sequence is included. Digital DD sequence generally increases transpiled circuit depth but not actual hardware runtime $\tau_{\text{circ.}}$. Coherence is calculated as $\tau_{\text{circ.}}/T_2$, where the average qubit lift-time $T_2 \approx 57.15$ µs for `ibmq_kolkata`.

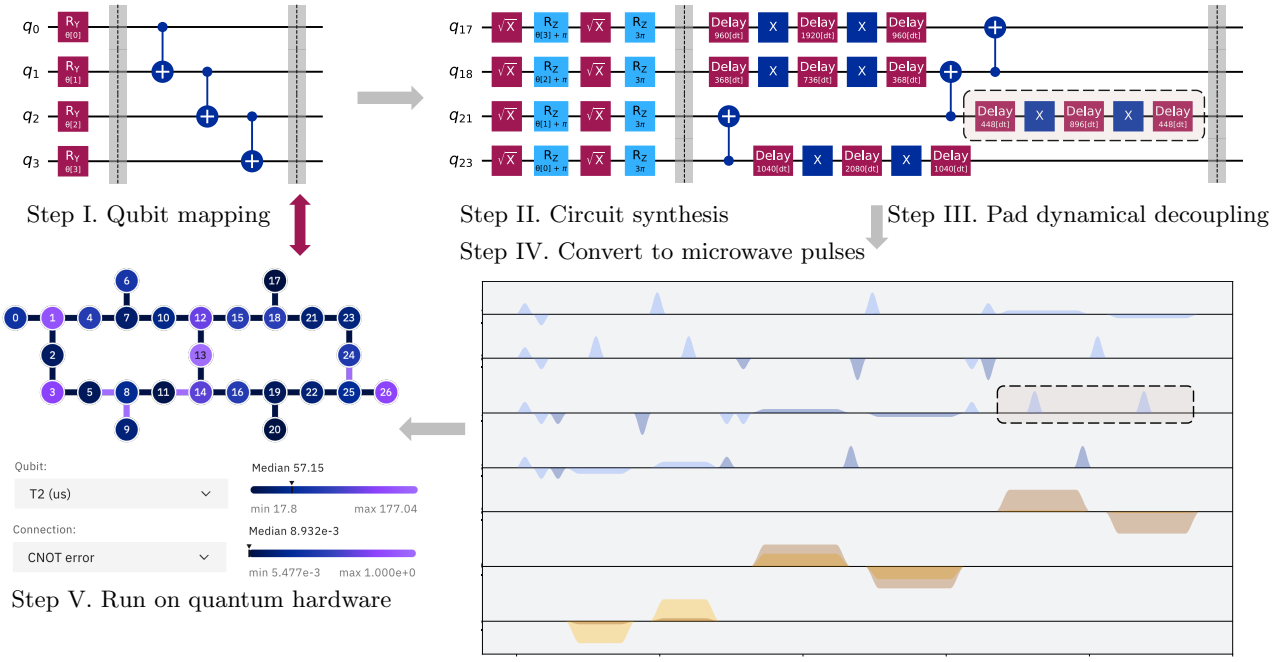

Figure A.8: Visualization of the transpilation process of 4-qubit RAA ($D = 1$) on 27-qubit `ibmq_kolkata`.

To evaluate each circuit candidate performance, we randomly sample $K = 10$ pre-conditioned Hamiltonian $\{H_{\text{pre.}}(t)\}_{i=1}^{K}$ at different times of the simulation with $\Delta t = 1$. We then randomly initiate the parameters of VQE and solve for ground-state preparation with the L-BFGS-B optimizer [71] on the Qiskit statevector simulator.

We calculate the expressibility and the entanglement capability [75] as auxiliary measures for each ansatz candidate. The ansatz expressibility describes its capability to sample uniformly from the Hilbert space and approximate arbitrary states. It can be defined as

$$\text{Expr} = \mathcal{D}_{\text{KL}}\left(\widehat{P}_{\text{PQC}}(F_{\vec{\boldsymbol{\theta}}}) \| P_{\text{Haar}}(F)\right), \tag{A.1}$$

where the Kullback–Leibler (KL) divergence [80] measures the distance between $\widehat{P}_{\text{PQC}}(F_{\vec{\boldsymbol{\theta}}})$ distribution of state fidelities between two randomly sampled parametrized states, and $P_{\text{Haar}}(F)$ is the state fidelity distribution for the ensemble of Haar random states. Expr closer to zero indicates more expressivity for an ansatz. Following the original definition [75], we calculate the entanglement capability as the average Meyer–Wallach Q-measure [81] from an ensemble of randomly sampled states

$$\text{Ent} = \frac{1}{|S|}\sum_{\theta_j \in S} Q\left(\left|\psi(\theta_j)\right\rangle\right) = \frac{2}{|S|}\sum_{\theta_j \in S}\left(1 - \frac{1}{n}\sum_{k=1}^{n}\text{Tr}\left[\rho_k^2(\theta_j)\right]\right), \tag{A.2}$$

where ρ_k is the reduced density matrix of the k-th qubit and $|S|$ represents the size of samples. Notice $Q = 0$ for any product state and $Q = 1$ for the GHZ state $|\psi\rangle_{\text{GHZ}} = (|0\rangle^{\otimes n} + |1\rangle^{\otimes n})/\sqrt{2}$. We further record the number of 2-qubit gates involved n_{2q}, transpiled circuit depth $D_{\text{trans.}}$ and duration t_{hardware} on 27-qubit `ibmq_kolkata`. The circuits are transpiled using the highest optimization level and adopt staggered XX dynamical decoupling sequence [82–85] to suppress decoherence and crosstalk during the idling time of the qubits, as illustrated in fig. A.8.

We report a $N = 4$ case study in table A.5. The random seeds for parameter generation, simulator, and optimizer are fixed to ensure a fair comparison of averaged state fidelity $\bar{F}$. One can conclude that the real-amplitude ansatz (RAA) outperforms all the other candidates in terms of fidelity and resource burden. In this case, switching from the linear entanglement structure via RAA pairwise only reduces 0.6% of the coherence time. The difference can be as high as 5% (≈ 2.8 µs) at $N = 10$ with $D_2 = 11$. HEA has higher expressibility and entanglement capability but achieves the same average fidelity with 3 times more parameters. The results from TEN emphasize that quantum entanglement is a necessary resource in general state preparation. To further enhance performance, one can consider strategies such as ADAPT-VQE [86] to design problem-tailored ansatze, noise-adaptive circuit design [87] to cope with hardware noise, and pulse-level ansatze [88] to suppress decoherence due to much shorter duration on hardware.