



Quantum computing for solid mechanics and structural engineering – A demonstration with Variational Quantum Eigensolver

Yunya Liu^a, Jiakun Liu^b, Jordan R. Raney^b, Pai Wang^{a,*}

^a Department of Mechanical Engineering, University of Utah, Salt Lake City, UT, USA

^b Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, USA

ARTICLE INFO

Keywords:

Quantum computing
Finite element
Eigenvalue
Computational mechanics
Numerical methods
Natural frequency

ABSTRACT

Variational quantum algorithms exploit the features of superposition and entanglement to optimize a cost function efficiently by manipulating the quantum states. They are suitable for noisy intermediate-scale quantum (NISQ) computers that recently became accessible to the worldwide research community. Here, we implement and demonstrate the numerical processes on the 5-qubit and 7-qubit quantum processors on the IBM Qiskit Runtime platform. We combine the commercial finite-element-method (FEM) software ABAQUS with the implementation of *Variational Quantum Eigensolver* (VQE) to establish an integrated pipeline. Three examples are used to investigate the performance: a hexagonal truss, a Timoshenko beam, and a plane-strain continuum. We conduct parametric studies on the convergence of fundamental natural frequency estimation using this hybrid quantum-classical approach. Our findings can be extended to problems with many more degrees of freedom when quantum computers with hundreds of qubits become available in the near future.

1. Introduction

Variational quantum algorithms (VQAs) [1] can solve problems in optimization [2], machine learning [3,4], physics [5], chemistry [6], material sciences [7], and cryptography [8]. Due to the unique features of entanglement and superposition, quantum computers can leverage VQAs to tackle problems efficiently and accurately while bypassing the limitation of memory allocation and computational complexity. VQAs achieve the goal of finding the solution by integrating classical optimizers with a quantum circuit. The quantum part here is designed to prepare quantum states and their measurements thereafter while the classical part is used to tune the quantum circuit parameters. The potential advantages of VQAs lie in the scaling law with respect to degrees of freedom (DOFs) in the mathematical model of problems: A particular quantum state represented by N qubits can encode the information of 2^N DOFs. This translates to an exponential scaling that surpasses any possible classical computing process, thereby facilitating a significant speed-up as compared to traditional solvers.

Specialized algorithms within VQAs suitable for near-term noisy intermediate-scale quantum (NISQ) computers were recently demonstrated in analyzing electronic structures [9,10], molecular spectra [11], fluid flows [12,13], heat transfer [14,15], as well as general algebraic

[16,17] and differential systems [18–22].

While these recent studies exemplify the vast potential of quantum computing and its implementation, the research community has yet to showcase an integrated pipeline that unifies VQAs with the finite-element method (FEM), which is a robust and widely-used technique across many disciplines. Motivated by this gap, we aim to investigate strategies for deploying quantum solvers to solve eigenvalue problems that are ubiquitous in mechanical systems. Focusing specifically on vibration analyses, the precise identification of the fundamental natural frequency in structures is critical in engineering practice. The importance is underscored by the resonant phenomenon where a small oscillatory perturbation can provoke a disproportionately large response.

In this Letter, we combine the FEM capability of the commercial software package ABAQUS together with the *Variational Quantum Eigensolver* (VQE) [23–25] on Qiskit quantum computing platform to implement an integrative FEM-VQE pipeline aiming at finding the fundamental natural frequency of different structures. We demonstrate and analyze three example cases: (I) hexagonal truss, (II) Timoshenko beam, and (III) plane-strain continuum. We first test our hybrid quantum-classical algorithm on a simulator backend and then on quantum processing units (QPUs) with 3 ~ 7 qubits. Results from classical solvers are used as benchmarks to quantify the errors. We perform a

* Corresponding author.

E-mail address: pai.wang@utah.edu (P. Wang).

<https://doi.org/10.1016/j.eml.2023.102117>

Received 28 August 2023; Received in revised form 19 November 2023; Accepted 5 December 2023

Available online 14 December 2023

2352-4316/© 2023 Elsevier Ltd. All rights reserved.

series of parametric studies on the key factors in the implementation. In addition, we also discuss current limitations and potential future directions.

2. Numerical implementation

The field of quantum computing is burgeoning, giving rise to a suite of software toolsets that each offer a different method of implementation and computational capabilities [29]. These toolsets are pivotal in translating the intricate theories of quantum computing into a user-friendly interface. As the research community in this domain flourishes, an array of such toolsets are (in chronological order of release dates): Forest SDK/PyQuil [30,31], Microsoft Quantum Development Kit [32], ProjectQ [33], PennyLane [34], Amazon Braket [35], IBM Qiskit [36], Cirq/TensorFlow Quantum [37,38], and Bosonic Qiskit [39]. Here, we employ the IBM Qiskit platform, primarily due to its maturity and extensive community support.

Fig. 1 depicts the steps to realize the integrative quantum-classical pipeline. Our approach involves the following components:

- ABAQUS/CAE and ABAQUS/STANDARD
- Qiskit quantum hardware and simulators
- NumPy and Matplotlib

There are three major parts of this hybrid framework:

(A) **ABAQUS Pre-processing** is shown as the orange-colored steps in Fig. 1. This scripting approach indicates an automatic process with user-defined problem settings. By converting user inputs directly into a desired output format, ABAQUS Pre-processing enables a high level of customization and could potentially handle large-scale problems, depending on the robust computational capability of data processing.

(B) **VQE Implementation** is shown as blue-colored steps in Fig. 1. It consists of processes mapping the Hamiltonian operator to VQE settings, which include three key factors: classical optimizer, entanglement pattern, and entanglement depth. These settings are crucial to the performance of Iterative Convergence.

(C) **Iterative Convergence** includes purple-colored steps shown in Fig. 1. They span from quantum circuit design to the decision-making of cost function convergence, iterating until the minimal eigenvalue estimate is obtained. The central inset of Fig. 1 shows an example of a quantum circuit with entanglement pattern CX and entanglement depth = 1 applied on four qubits. In this part, efficiency is ensured by using a convergence mechanism and optimization strategy, while precision control influences the balance between computational cost and solution accuracy.

2.1. ABAQUS pre-processing

Within the ABAQUS/CAE environment, a scripting approach converts the input of user-defined problem settings to the .MTX file outputs. The stiffness and mass matrices are the outcome of data processing.

We develop pre-processing commands to generate a FEM model with a targeted number of free DOFs. The overall goal is to find a suitable combination of the total number of degrees of freedom $n_{\text{DOF}}^{\text{all}}$ and the prescribed (fixed) degrees of freedom $n_{\text{DOF}}^{\text{fixed}}$ in the model such that $2^N = n_{\text{DOF}}^{\text{all}} - n_{\text{DOF}}^{\text{fixed}}$ where N is an integer. The general procedures are summarized as follows:

- (1) Employ pre-processing Python script in the ABAQUS/CAE environment to create an initial model with corresponding geometry, material properties, and boundary conditions.
- (2) Mesh the model with appropriate element types: T2D2 for the hexagonal truss, B21 for the Timoshenko beam, and CPE3/CPE4 for the plane-strain continuum.

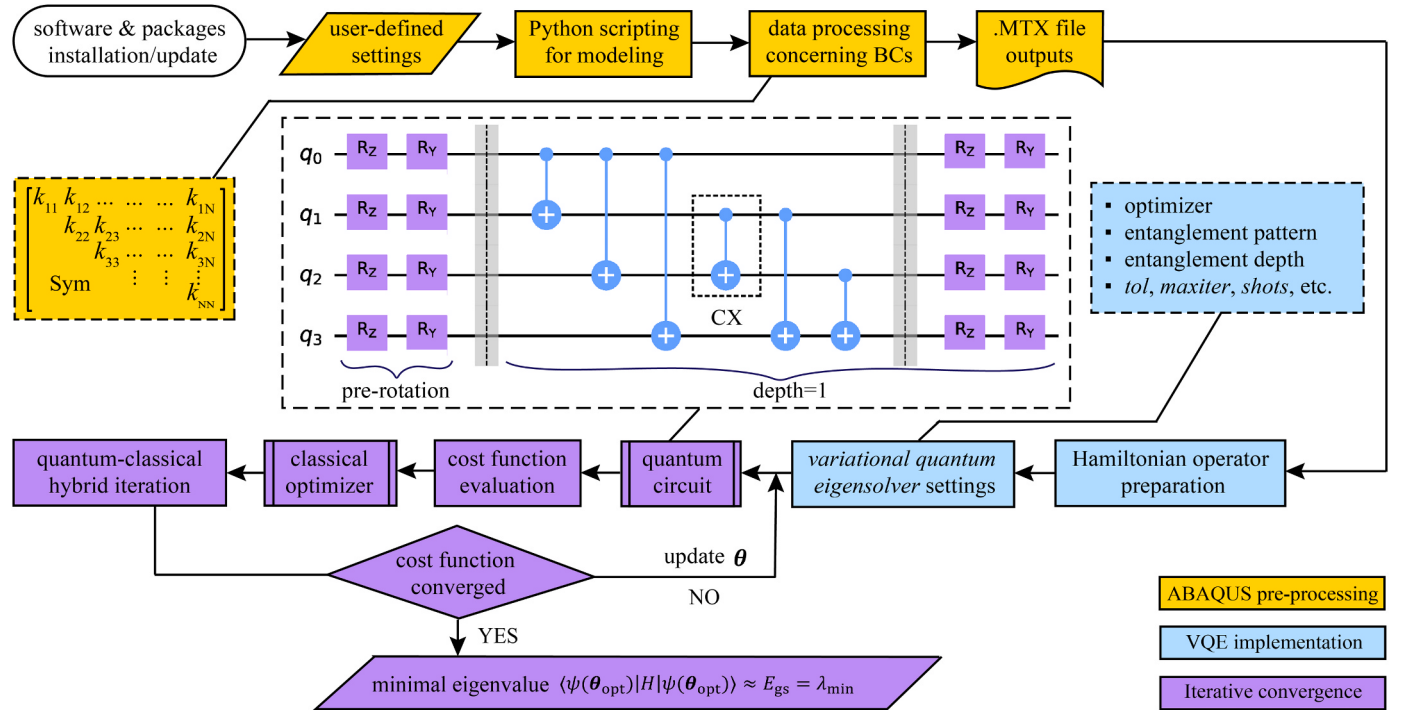


Fig. 1. Pipeline of Variational Quantum Eigensolver algorithm for mechanics problems. The orange-colored steps refer to ABAQUS Pre-processing, those in blue to VQE Implementation, and the remaining in purple to Iterative Convergence. An example stiffness matrix is shown in the left dashed-frame inset. A particular instance of the “hardware-efficient ansatz” [26,27] is depicted in the central dashed-frame inset, where the entanglement pattern with 6 CX gates and depth = 1 is shown as an example. Additional examples are presented in Supplemental Material [28]. The right dashed-frame inset lists the key user-defined parameters.

- (3) Adjust and modify the mesh of the model to reach the targeted number of free DOFs determined by N . For the beam system, a proper element size can be directly calculated for a given N , whereas for truss and continuum systems, we perform an iterative approach to adjust the assigned global element size (with subsequent re-meshing). More details are presented in the *Supplemental Material* [28].
- (4) Once the targeted free DOFs has been reached, export mass matrix M and stiffness matrix K from ABAQUS. These matrices, by default, will include all DOFs as a result of the internal processing logistics.
- (5) Execute another Python script to read the exported mass and stiffness matrices from .MTX files. Then, apply partitioning to get the matrices with only free DOFs by filtering out columns and rows associated with boundary conditions.

2.2. VQE implementation

In the nascent stages of quantum computing, NISQ devices demonstrate the practice of outpacing classical computers by leveraging exponentially fewer resources for certain computations. The implications of quantum computational abilities extend well beyond purely quantum mechanical realms. Due to the susceptibility of qubits to operational errors induced by quantum noise, algorithms suitable for NISQ systems are crafted to have a shallow circuit depth, enhancing their noise resilience. While constrained by such limitations, these noise-tolerant algorithms remain proficient in the physical interpretation of mechanical problems. The VQE algorithm stands at the forefront of NISQ applications, essential for calculating the ground state energy of a given Hamiltonian, which in turn reveals the fundamental natural frequency of a mechanical system [23,40,41]. In this work, we demonstrate two types of implementations: a noise-free simulator and quantum processors [42].

Generally, a $2^N \times 2^N$ square matrix needs N qubits to encode it as a quantum Hamiltonian. From the .MTX files exported from ABAQUS, we prepare and decompose our Hamiltonian as

$$H = M^{-1}K = \sum c_l P_l, \quad (1)$$

where $P_l \in \{I, X, Y, Z\}^{\otimes N}$ represents a multi-qubit (N -qubit) Pauli operator [43], and c_l 's are coefficients of decomposition.

The accuracy of the VQE depends on three key factors including classical optimizer [44,45], entanglement pattern [40,46], and entanglement depth [47,48]. In this study, we test all of the following.

- Classical optimizers:
 - Simultaneous Perturbation Stochastic Approximation (SPSA) - a derivative-free optimization algorithm utilizing stochastic approximation of the gradient to efficiently handle large-scale problems [49].
 - Constrained Optimization by Linear Approximation (COBYLA) - a derivative-free optimization method using linear approximations for cost function and constraints to handle nonlinear optimization [50].
 - Sequential Least Squares Programming (SLSQP) - a quasi-Newton method using a sequence of quadratic programming sub-problems to handle constrained nonlinear optimization problems [51].
 - Limited-memory Broyden-Fletcher-Goldfarb-Shanno Bound (L-BFGS-B) - a quasi-Newton method using a limited amount of memory to approximate the inverse Hessian matrix for bound-constrained optimization [52].
- Entanglement patterns:
 - Controlled Not (CNOT or CX) gate
 - Controlled Z (CZ) gate
 - Controlled Rotation X (CRX) gate

- Entanglement depths:
 - An integer (1 ~ 10) specifying the quantum circuit

Together, these components constitute a hardware-efficient ansatz (i.e., a parameterized quantum circuit) in the variational form of the “EfficientSU2” class [23]. In this class, the single-qubit gates manipulate individual quantum states, while entanglement patterns create correlations among each pair of qubits, resulting in entangled states that are inseparable into individual quantum states.

As a specific example, the central inset in Fig. 1 illustrates a particular instance of the ansatz comprising $N = 4$ qubits, utilizing single-qubit gates R_z and R_y , and adopting depth = 1 with entanglement pattern CX. In quantum state transformations, the single-qubit gates R_z and R_y are parameterized while the entanglement operators consisting of two-qubit CX gates are non-parameterized.

First, quantum states are prepared for the qubits, and their initial joint state can be written as [53].

$$|\psi\rangle^{\otimes N} = |\psi_{N-1}\rangle \otimes \cdots \otimes |\psi_1\rangle \otimes |\psi_0\rangle = \otimes_{j=0}^{N-1} |\psi_j\rangle, \quad (2)$$

where $\otimes$ denotes the tensor product between quantum states. Then, the initial joint state undergoes two sequential layers, or “slices”, of pre-rotation [43,54] denoted by the pre-operator U_{pre} , and this results in

$$\begin{aligned} & \left[\otimes_{j=0}^{N-1} R_y(\theta_{2,j}) \right] \left[\otimes_{j=0}^{N-1} R_z(\theta_{1,j}) \right] \left[\otimes_{j=0}^{N-1} |\psi_j\rangle \right] \\ &= \otimes_{j=0}^{N-1} [R_y(\theta_{2,j}) R_z(\theta_{1,j}) |\psi_j\rangle] = U_{\text{pre}} |\psi\rangle^{\otimes N}. \end{aligned} \quad (3)$$

Next, the states are fully entangled together by the entanglement operator CX gates,

$$U_{\text{ent}} = \prod_{j_1=0}^{N-2} \prod_{j_2=j_1+1}^{N-1} \text{CX}_{j_1 j_2}, \quad (4)$$

which is followed by another two “slices” of rotations,

$$U_{\text{rot}} = \left[\otimes_{j=0}^{N-1} R_y(\theta_{4,j}) \right] \left[\otimes_{j=0}^{N-1} R_z(\theta_{3,j}) \right], \quad (5)$$

where $\theta = \{\theta_{s,j}\}$ is a set of variational parameters that controls the qubit states. The angle θ in a single-qubit gate $R_y(\theta)$ represents the magnitude of rotation applied to the quantum state around the y-axis of the Bloch sphere [43]. Executing an $R_y(\theta_{s,j})$ gate effectively transforms the quantum state by an angle θ in a qubit's Hilbert space. Here, $s = 1, \dots, 4$ indicates the “slice” of the single-qubit gates, and $j, j_1, j_2 \in \{0, 1, \dots, N-1\}$ are indices of qubits. The combined effect of Eqs. (4) and (5) constitutes one “depth”, and the total operator of the parameterized variational ansatz is

$$U(\theta) = [U_{\text{rot}}][U_{\text{ent}}][U_{\text{pre}}]. \quad (6)$$

Alternative implementations of other entanglement patterns and expressions are presented in *Supplemental Material* [28].

2.3. Iterative convergence

Regarding the accuracy of iterative convergence, other tunable parameters are involved: 1) optimizer-dependent *tol*; 2) quantum processor-dependent *shots*, transpile optimization, statistical quantum state measurement, and error mitigation techniques; as well as 3) maximum number of iterations *maxiter* which depends on both the optimizer and quantum processor.

In each iteration, a new quantum state is generated as

$$|\psi(\theta)\rangle = U(\theta) |\psi\rangle^{\otimes N}, \quad (7)$$

where $|\psi\rangle^{\otimes N}$ denotes the initial quantum states defined in Eq. (2).

The new state $|\psi(\theta)\rangle$ is then used to evaluate the cost function defined in Eq. (8), where $|\psi(\theta)\rangle$ is the output quantum state determined

by both the parameters $\theta = \{\theta_{s,j}\}$ shown in Eqs. (3) and (5) and the entanglement shown in Eq. (4). The quantum-classical hybrid iteration process successively alters the quantum circuit parameters to minimize the cost function, which is defined as:

$$C(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle, \quad (8)$$

The hybrid algorithm iteratively updates the quantum-circuit parameters θ using protocols encoded in the classical optimizer until it reaches either the user-defined tolerance tol or the maximum number of iterations $maxiter$. Ideally, the convergence criterion, represented by the tolerance parameter, ensures convergence when the absolute difference between evaluations of two consecutive cost functions is smaller than the tolerance parameter, indicating proximity to the optimal solution of VQE Implementation. A list of user-defined parameters for the accuracy of Iterative Convergence:

- tol represents the convergence criterion, which is used against the absolute approximate error:

$$|E_j - E_{j-1}| < tol, \quad (9)$$

where E_j is the ground state energy estimate of the j -th iteration.

- $shots$ is the number of times the quantum circuit is executed for each evaluation of the cost function, determining the statistical accuracy of E_j .
- $maxiter$ denotes the maximum number of iterations that an optimizer is allowed to go through.

To minimize the effect of optimizer-dependent parameters on a noise-free simulator, as well as processor-dependent parameters on real quantum devices, we pre-define the following parameters with constant values to generate all data presented in this Letter. The noise-free simulator uses $shots = 10^5$ and $maxiter = 10^5$, while QPUs employ $shots = 2 \times 10^4$ and $maxiter = 100$, which are the largest possible on the quantum hardware. In all cases, we set $tol = 10^{-4}$.

When the threshold tol is reached by iterations, the optimization yields the final expectation value (i.e., minimized cost function) as

$$C(\theta_{opt}) = \langle \psi(\theta_{opt}) | H | \psi(\theta_{opt}) \rangle \approx E_{gs}, \quad (10)$$

where θ_{opt} and E_{gs} denote the optimal set of parameters and the quantum ground state energy, respectively.

Upon convergence, we obtain

$$\lambda_{min} \approx E_{gs} \quad \text{and} \quad |\psi_{min}\rangle \approx |\psi(\theta_{opt})\rangle, \quad (11)$$

where λ_{min} is the minimum eigenvalue, and $|\psi_{min}\rangle$ is the quantum state corresponding to λ_{min} .

Other parameters including transpile optimization, statistical quantum state measurement, and error mitigation techniques depend on the gate fidelity, qubit connectivity, coherence time, and noise level of specific quantum hardware [55–58].

3. Example problem cases

We investigate three different systems subjected to prescribed boundary conditions. In all cases, we use linear elastic, isotropic material properties that are similar to steel with density $\rho = 7850 \text{ kg/m}^3$, Young's modulus $E = 21 \times 10^4 \text{ GPa}$, and Poisson's ratio $\nu = 0.3$.

Case (I): The hexagonal truss, as illustrated in Fig. 2(a), consists of truss members with length $L = 1.5 \text{ mm}$ and circular cross-section radius $r = 0.5 \text{ mm}$. Boundary conditions are applied in the following manner: (1) For nodes located at the bottom, displacements in both horizontal and vertical directions are fixed ($u_1 = u_2 = 0$); (2) For nodes at the left, right, and top sides, u_1 is set to zero, whereas u_2 is free. In this formulation, a consistent mass matrix (rather than a lumped-mass matrix) is used.

Case (II): The Timoshenko beam is illustrated in Fig. 2(b) with length $L = 9 \text{ mm}$ and circular cross-section radius $r = 1 \text{ mm}$. For boundary conditions, the translational displacements u_1 and u_2 of both ends are set to zero. Here, we use the lumped-mass formulation for both mass and stiffness matrices.

Case (III): The plane-strain continuum model, as depicted in Fig. 2(c), consists of a quarter section of a square with a circular cutout in its geometric center. The entire square is $2 \text{ mm} \times 2 \text{ mm}$, and the open hole has radius $r = 0.5 \text{ mm}$. Boundary conditions are: x -symmetric constraint is prescribed on the left edge, while the y -symmetric constraint is prescribed on the bottom edge. The right edge is constrained by $u_1 = 0$. We also adopt the lumped-mass formulation in this case.

4. Results and discussion

4.1. Noise-free simulators

To test the performance, we first execute the FEM-VQE pipeline on a noise-free quantum simulator (e.g., *statevector_simulator* on Qiskit). We conduct comprehensive parametric studies to evaluate the algorithmic performance in the search for fundamental natural frequency on problems with a wide range of DOFs varying from 8 to more than 8000. We measure the accuracy of VQE from parametric studies using the percentage error,

$$\text{Error}(\%) = |\lambda_q - \lambda_c| / \lambda_c, \quad (12)$$

where λ_q and λ_c are eigenvalue estimates from VQE and conventional

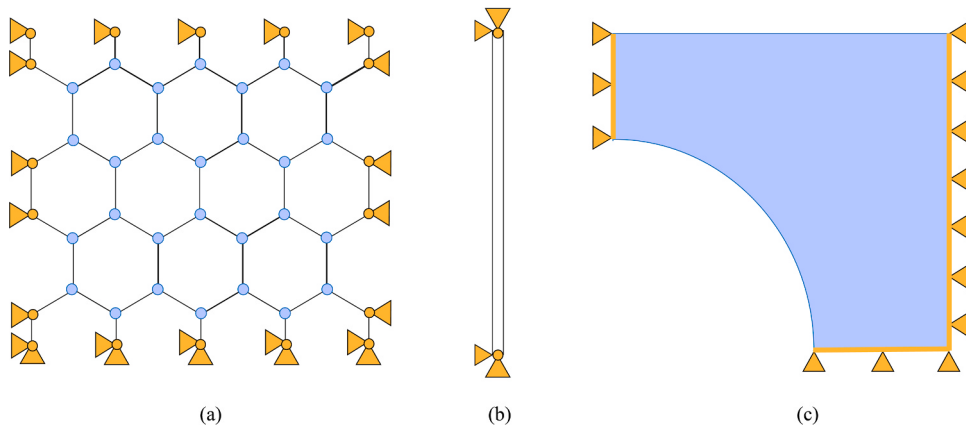


Fig. 2. Schematics of the three distinct case studies: (a) hexagonal truss, (b) Timoshenko beam, and (c) plane-strain continuum. Boundary conditions are prescribed at the locations denoted by yellow dots/lines.

classical solvers, respectively.

Focusing on the following three sets of simulator-based parametric studies by fixing two key factors and varying: (a) optimizer types, (b) entanglement patterns, or (c) entanglement depths, we plot the error values defined by Eq. (12) in Figs. 3, 4, and 5 for all cases. Additional data in terms of convergence rates are presented in the *Supplemental Material* [28]. Our findings reveal that there is no one-size-fits-all particular set of parameters ensuring fast and accurate computation for all three cases. We discuss problem-specific performance characteristics below. We note that all data presented here are reproducible due to the noise-free environment of the simulator. No statistical deviation may occur once all user-defined parameters have been set.

Case (I) - For the choice of optimizers, Fig. 3(a) indicates that, except SLSQP, the error remains under 7.5% for all. There is no discernable difference among the other three optimizers. Sparse Hamiltonians can lead to an optimization landscape characterized by noisy gradient evaluations. Both SPSA and COBYLA demonstrate resilience in such noisy environments, enabling them to circumvent these challenges and function effectively even when gradient information is uncertain. Moreover, COBYLA exhibits superior navigation through the feasible regions of parameter space in the context of sparse Hamiltonians. This is attributable to the fact that the decomposition of sparse Hamiltonians can introduce nonlinear constraints within the optimization space. Fig. 3(b) does not show any apparent correlation between errors and entanglement patterns either. Furthermore, Fig. 3(c) shows that the error is always below 5% for any entanglement depth up to 7, and the best performance with errors below 3.5% can be consistently obtained when depth = 4. However, the errors tend to rise above 10% when depth = 8,

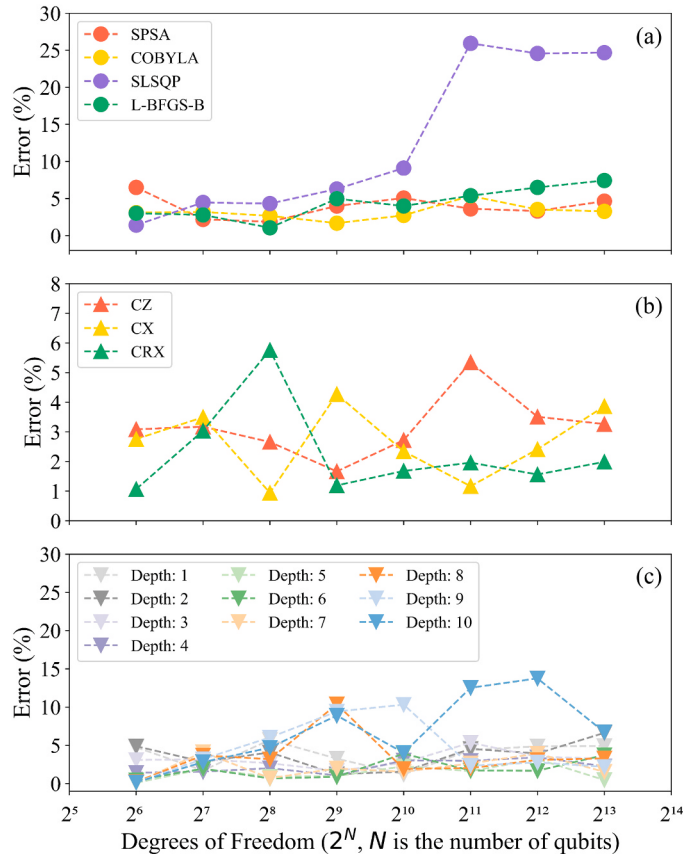


Fig. 3. Errors of VQE results, as defined by Eq. (12), for Case (I): (a) Different optimizer choices with entanglement pattern CZ and depth 3. (b) Different entanglement patterns with optimizer COBYLA and entanglement depth 3. (c) Different entanglement depths with optimizer COBYLA and entanglement pattern CZ.

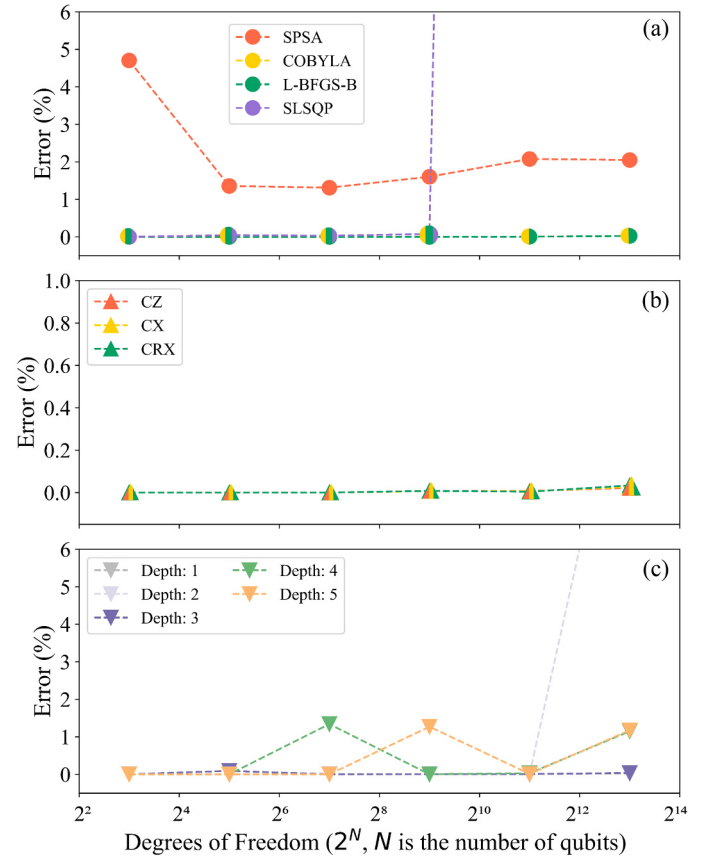


Fig. 4. Errors of VQE results, as defined by Eq. (12), for Case (II): (a) Different optimizer choices with entanglement pattern CZ and depth 1. Note that, at $N = 11$ and 13, the errors by SLSQP shoot up to the level of 130%, which is out of the range of the plot. (b) Different entanglement patterns with optimizer L-BFGS-B and entanglement depth 1. (c) Different entanglement depths with optimizer L-BFGS-B and entanglement pattern CZ. Note that, at $N = 13$, the error by depth = 2 shoots up to the level of 12%, which is out of the range of the plot.

9, or 10. Overall, we think an average error of 5% can be expected when applying the VQE algorithm to similar 2D truss problems.

Case (II) - As illustrated in Fig. 4(a), optimizers COBYLA and L-BFGS-B deliver error-free outcomes, while SPSA and SLSQP show inferior performance. In this case, we employ the Hamiltonian derived from the lumped-mass-matrix formulation. This simplifies constraints and facilitates smoother navigation of parameter space when using L-BFGS-B. The energy landscape is convex, conducive to rapid convergence using quasi-Newton methods. Gradient-based approaches are particularly adept at seeking minima within such well-conditioned landscapes. Additionally, L-BFGS-B leverages gradient information to construct a quadratic model of the cost function, enhancing the accuracy of its convergence. While both SLSQP and L-BFGS-B incorporate bound constraints, L-BFGS-B demonstrates greater robustness in managing parameter constraints. Fig. 4(b) also shows error-free outcomes for all three entanglement patterns, and the largest error observed is only at the level of 0.035%. We think this is primarily due to both the lumped-mass-matrix formulation and the high sparsity of the problem-specific Hamiltonian. Fig. 4(c) displays that, except for depth = 2, the errors remain under 2% for all. Overall, This case study shows that VQE tends to be much more accurate in dealing with similar quasi-1D beam problems.

Case (III) - Fig. 5(a) shows that, among all four optimizer types, only optimizer L-BFGS-B produces errors consistently below 5%. Moreover, in contrast to data of Cases (I) and (II), Fig. 5(b) seems to show an advantage for using the entanglement pattern CX, which keeps the error under 5%. This may be due to the lumped formulation in the setup,

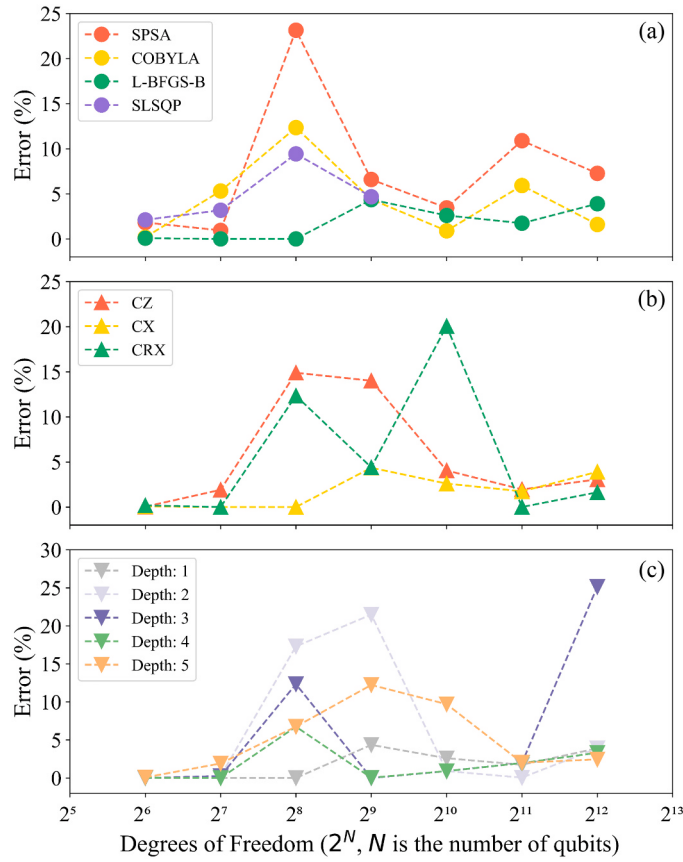


Fig. 5. Errors of VQE results, as defined by Eq. (12), for Case (III): (a) Different optimizer choices with entanglement pattern CX and depth 1. (b) Different entanglement patterns with optimizer L-BFGS-B and entanglement depth 1. (c) Different entanglement depths with optimizer L-BFGS-B and entanglement pattern CX.

which results in a Hamiltonian characterized by independent quantum states and can fully leverage the entanglement. Here, the CX gate enhances correlations by introducing an additional rotation around the X-axis of the Bloch sphere, complementing the single-qubit gates with rotations around the Y- and Z-axes in the ansatz. Although we also adopt the lumped formulation in Case (II), the results among all entanglement types are all negligible there, as shown in Fig. 4(b). Hence, no similar advantage of CX is detectable in data from Case (II). In addition, Fig. 5(c) supports a definite advantage of shallow circuits, in which the depth = 1 choice keeps the error consistently below 5% in Case (III). Overall, similar to the results in Case (I), we can expect that, through careful choices of the optimizer, entanglement pattern, and entanglement depth, we can expect an average error around 5% of VQE when it is used on plain-strain problems.

Furthermore, we also assess the mean of errors (ME) and the standard deviation of errors (SDE) across the different degree-of-freedom (DOF) data points shown in Figs. 3, 4, and 5. For instance, in Fig. 3 (a), employing different types of optimizers results in four sets of errors (data illustrated as four different colors), and we calculate the ME and SDE for each optimizer. The assessment of MEs and SDEs in all cases is presented in Supplemental Material [28].

4.2. Quantum processing units

IBM QPUs are hardware designed to execute quantum computations. Employing qubits as their fundamental units of information, they harness the principles of quantum mechanics, which allow them to perform complex calculations far beyond the capability of traditional,

binary-bit-based CPUs.

Leveraging free and open quantum hardware resources, we execute our FEM-VQE pipeline on 5-qubit (e.g. *ibmq_manila*) and 7-qubit (e.g. *ibmq_nairobi*) QPUs (e.g. Falcon r5.11H processors) via the IBM Qiskit Runtime platform. Guided by the error assessments presented in Figs. 3–5, as well as the allowable time and space limits on quantum devices, we choose optimizer COBYLA and entanglement depth 1 in all cases. Furthermore, we apply the entanglement pattern CZ for Case (I) and (II), and CX for Case (III).

Unlike noise-free simulators, QPUs may suffer statistical deviations since random noise could alter the outcome. After conducting multiple (roughly ~ 10) trials of each calculation on QPUs, we summarize the mean VQE estimate λ_q and ME of each case in Table 1. The data indicate that the algorithm accumulates much more error on QPUs than it does on the noise-free simulator. Even for Case (II), which is the best performer on the simulator, our results show the errors can jump up to more than 30% when $N > 5$. Here, the reliability of the FEM-VQE pipeline can be impeded by three principal factors:

First, the QPU capabilities are severely limited by the “quantum volume” (V_Q) metric [59–61] available on the IBM Qiskit Runtime platform. The quantum volume of N qubits is defined by the formula $V_Q = 2^{\min\{N, \text{path}(N)\}}$, where $\text{path}(N)$ denotes the longest path of gate operations from start to end in each iteration. This directly influences the QPU performance. In the cases presented in this paper, we have $\text{path}(N) = 10$ for the entanglement depth 1. This results in $V_Q = 2^N$ where $N = 3, 5, 6, 7$, i.e., $V_Q = 8, 32, 64, 128$. However, The current IBM QPUs have only the capability to handle up to $V_Q = 32$ with a high degree of quality [62]. This presents a challenge, as it results in greater errors in contrast to outcomes from a noise-free simulator, which can support up to 5000 qubits without quantum volume limitations.

Second, transpilation [63,64] poses additional challenges which inflate circuit metrics including gates, depths, V_Q , and error rates, further exacerbating the computational efficacy. This worsens the QPUs performance since the actual circuit depth can be increased by the transpilation, and deeper circuits produce more errors. In the end, the accumulative error may hinder the update of parameters in the cost function landscape.

Third, the thermal noise and electromagnetic interference on QPUs corrupt the preparation and measurement of quantum state $|\psi(\theta)\rangle$, resulting in low-quality of VQE estimates.

Lastly, error accumulation deriving from the variational form of the ansatz and the classical optimizer can impair the accuracy of eigenvalue prediction. The ideal condition would be achieving a fault-tolerant result from an aptly parameterized quantum circuit $U(\theta)$ without classical computers, due to the risk of a sub-optimal solution caused by a barren plateau problem [65,66]. These considerations are inherent and unavoidable aspects of the current technological landscape. Although this also occurs on simulators, the effects are more pronounced with QPUs.

4.3. Complexity

The space complexity of the proposed hybrid algorithm depends on the amount of quantum memory. For example, N qubits can store the same amount of information that requires 2^N conventional bits on

Table 1
Errors in QPU computations.

	λ_c	λ_q	ME (%)
Case (I) ($N = 6$)	0.0771	0.0855	10.895
Case (I) ($N = 7$)	0.0572	0.0688	20.280
Case (II) ($N = 3$)	0.0343	0.0343	0.000
Case (II) ($N = 5$)	0.0604	0.0606	0.331
Case (II) ($N = 7$)	0.0354	0.0483	36.441
Case (III) ($N = 6$)	0.0457	0.0535	17.068
Case (III) ($N = 7$)	0.0317	0.0380	19.874

classical computers. This results in $O(\log(n))$ space complexity on the quantum computer for a problem with $n \times n$ mass/stiffness matrices, even if they are non-sparse. In comparison, it requires the space complexity $O(n^2)$ in the classical computer to store a dense $n \times n$ matrix. In the case of dealing with sparse matrices, the scale of space-saving, as compared to the cases of dense matrices, is similar for both quantum and classical computers, thus maintaining the quantum advantage in terms of space complexity.

The time complexity of the hybrid algorithm depends on quantum circuit complexity (quantum), measurement complexity (quantum-to-classical), and optimization complexity (classical). Take Fig. 1 with four qubits, two layers of pre-rotations, and one depth of entanglement as an example.

First, the time complexity is $O((\log(n))^2)$ on a simulator. Considering the hardware with different qubit connectivity constraints and coherence times, the transpilation may increase the number of gates and lead to a higher time complexity than that on simulators.

Second, the measurement complexity, determined by the necessary number of quantum measurements for precise eigenvalue estimation, typically scales with $O(1/\epsilon^2)$, where ϵ denotes the tolerance in numerical precision. For the three examples in our work, we choose a constant number of quantum measurements (*shots*) on the simulator and quantum devices to minimize the effect of measurement. Assuming all measurements can be done concurrently as fully parallel processes in quantum hardware, the measurement complexity would be fixed since it no longer scales with the size of Hamiltonian.

Third, the optimization complexity is primarily determined by the type of optimizer and the maximum number of iterations an optimizer is allowed to go through (*maxiter*). In this work, the *maxiter* is capped at a constant value. Thus, for a fixed-depth ansatz, the optimization complexity depends on the complexity of each iteration regarding a specific optimizer [45,67–70]:

- SPSA: $O(1)$, as it requires two evaluations of cost function regardless of the number of parameters [71].
- COBYLA: $O(p^2)$, as it requires p linearization of cost function and constraints respectively [72].
- SLSQP: $O(p^3)$, as it requires an approximation of the inverse Hessian matrix [73].
- L-BFGS-B: $O(p)$, as above, but it uses a limited memory approach to approximate the inverse Hessian matrix [74].

where p is the number of parameters with the same polynomial order as $\log(n)$.

The theoretical bound of time complexity for each optimizer is established under ideal conditions. However, in practical scenarios, specific optimization and quantum hardware constraints present obstacles in accurately determining time complexity. For example, the optimization landscape has barren plateaus that the gradients vanish exponentially with the increasing number of qubits, making it hard to estimate reliable energy states [75]. Additional explanations are presented in *Supplementary Material* [28].

In theory, compared with $O(n^3)$ for purely classical algorithms, the time complexity of the hybrid VQE approach scales with $O((\log(n))^4)$ by using the ansatz depicted in Fig. 1 and the COBYLA optimizer. However, in the current implementation on readily accessible hardware devices, the time complexity of the VQE does not consistently offer an advantage over classical algorithms. Its efficiency is largely contingent upon specific factors such as the nature of the problem, the choice of optimizer, and the hardware used. This subject is currently a focal point of interest and remains an active area of research.

5. Conclusion

The successful implementation of a FEM-VQE pipeline presented in this Letter is the first step towards harnessing quantum computing to

solve problems in solid mechanics and structural engineering. This computational framework could be particularly useful to researchers who wish to take advantage of the noisy intermediate-scale quantum (NISQ) computing devices that are rapidly becoming available now. Our parametric studies on (I) 2D truss, (II) 1D beam, and (III) plane-strain continuum cases provide direct evidence supporting the validity of this quantum-classical hybrid algorithm. On a noise-free simulator, our data prescribe the following set of optimal parameters: (I) COBYLA with CZ and depth = 4; (II) L-BFGS-B with CZ and depth = 1; (III) L-BFGS-B with CX and depth = 1.

The VQE algorithm requires $O(\log(n))$ space complexity compared to $O(n^2)$ in classical computers to store a $n \times n$ dense matrix. Theoretically, the time complexity of this hybrid method, employing the ansatz from Fig. 1 and the COBYLA optimizer, scales at $O((\log(n))^4)$, compared to $O(n^3)$ for purely classical algorithms.

While the demonstration detailed in this letter does not manifest quantum supremacy over classical computers in terms of accuracy or efficiency, it does validate the integrative methodology that couples ABAQUS with Qiskit VQE Implementation. This methodology holds the potential to address problems encompassing significantly more degrees of freedom as quantum computers continue to become more capable and more widely available. The large error values in our results are not a reflection of the shortcomings of the algorithm. Rather, they are indicative of the current state of quantum computing technology [76–79]. The prospect of enhanced quantum processors, fortified with advanced error mitigation techniques and designed to operate at a utility-scale with improved quality, is an anticipated advancement that warrants keen attention [80,81]. For example, quantum hardware manufacturers are now making devices with up to 27, 65, 127, and 433 qubits available to the general research community [42]. This could potentially enable our FEM-VQE pipeline to solve complex mechanics problems with 10^8 , 10^9 , 10^{38} , and 10^{130} DOFs like dispersion bands and optimization-based inverse design of architected materials.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yunya Liu and Jiakun Liu report equipment, drugs, or supplies was provided by International Business Machines Corp. Jordan R. Raney and Jiakun Liu report financial support was provided by National Science Foundation.

Acknowledgment

YL and PW are supported by both the Research Incentive Seed Grant Program and the start-up research funds of the Department of Mechanical Engineering at the University of Utah. JR and JL gratefully acknowledge support via National Science Foundation (NSF) FM 2036881. JL was also partially supported by the NSF MRSEC program under award DMR-1720530. The support and resources from the Center for High-Performance Computing at the University of Utah are gratefully acknowledged.

We acknowledge the use of IBM Quantum services 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. In the process of developing the code necessary for this project, we also immensely benefited from the resources and community support provided by the IBM Qiskit channel. The coding techniques and strategies derived from this resource were instrumental in overcoming computational challenges and refining the efficiency of our algorithms. We wish to express our gratitude for valuable assistance and guidance from the Qiskit community.

The authors declare no business or financial connections with either ABAQUS Inc. or IBM Corp.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at [doi:10.1016/j.eml.2023.102117](https://doi.org/10.1016/j.eml.2023.102117).

References

- [1] 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, *Nat. Rev. Phys.* 3 (2021) 625.
- [2] L. Zhu, H.L. Tang, G.S. Barron, F. Calderon-Vargas, N.J. Mayhall, E. Barnes, S. E. Economou, Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer, *Phys. Rev. Res.* 4 (2022) 033029.
- [3] M. Sajjan, J. Li, R. Selvarajan, S.H. Sureshbabu, S.S. Kale, R. Gupta, V. Singh, S. Kais, Quantum machine learning for chemistry and physics, *Chem. Soc. Rev.* (2022).
- [4] R. Divya, J.D. Peter, Quantum machine learning: a comprehensive review on optimization of machine learning algorithms, in: Proceedings of the Fourth International Conference on Microelectronics, Signals & Systems (ICMSS) IEEE, 2021, pp. 1–6.
- [5] H. Weimer, A. Kshetrimayum, R. Orús, Simulation methods for open quantum many-body systems, *Rev. Mod. Phys.* 93 (2021) 015008.
- [6] S. McArdle, S. Endo, A. Aspuru-Guzik, S.C. Benjamin, X. Yuan, Quantum computational chemistry, *Rev. Mod. Phys.* 92 (2020) 015003.
- [7] B. Bauer, S. Bravyi, M. Motta, G.K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, *Chem. Rev.* 120 (2020) 12685.
- [8] C.H. Bennett, G. Brassard, Quantum Cryptography: Public Key Distribution and Coin Tossing, arXiv preprint arXiv:2003.06557, 2020.
- [9] B. Huang, M. Govoni, G. Galli, Simulating the electronic structure of spin defects on quantum computers, *PRX Quantum* 3 (2022) 010339.
- [10] T. Ohgoe, H. Iwakiri, M. Kohda, K. Ichikawa, Y.O. Nakagawa, H.O. Valencia, S. Koh, Demonstrating Quantum Computation for Quasiparticle Band Structures, arXiv preprint arXiv:2307.14607, 2023.
- [11] V. Barone, S. Alessandrini, M. Biczysko, J.R. Cheeseman, D.C. Clary, A.B. McCoy, R.J. DiRisio, F. Neese, M. Melosso, C. Puzzarini, Computational molecular spectroscopy, *Nat. Rev. Methods Prim.* 1 (2021) 1.
- [12] S.S. Bharadwaj, K.R. Sreenivasan, Quantum Computation of Fluid Dynamics, arXiv preprint arXiv:2007.09147, 2020.
- [13] R. Demirdjjan, D. Gunlycke, C.A. Reynolds, J.D. Doyle, S. Tafur, Variational quantum solutions to the advection-diffusion equation for applications in fluid dynamics, *Quantum Inf. Process.* 21 (2022) 322.
- [14] Y. Liu, Z. Chen, C. Shu, S.-C. Chew, B.C. Khoo, X. Zhao, Y. Cui, Application of a variational hybrid quantum-classical algorithm to heat conduction equation and analysis of time complexity, *Phys. Fluids* 34 (2022).
- [15] C. Lu, Z. Hu, B. Xie, N. Zhang, Quantum CFD simulations for heat transfer applications, ASME International Mechanical Engineering Congress and Exposition, Vol. 84584, American Society of Mechanical Engineers, 2020. V010T10A050.
- [16] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, P.J. Coles, Variational Quantum Linear Solver, arXiv preprint arXiv:1909.05820, 2019.
- [17] X. Xu, J. Sun, S. Endo, Y. Li, S.C. Benjamin, X. Yuan, Variational algorithms for linear algebra, *Sci. Bull.* 66 (2021) 2181.
- [18] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, D. Jaksch, Variational quantum algorithms for nonlinear problems, *Phys. Rev. A* 101 (2020) 010301.
- [19] D. Fang, L. Lin, Y. Tong, Time-marching based quantum solvers for time-dependent linear differential equations, *Quantum* 7 (2023) 955.
- [20] H. Krovi, Improved quantum algorithms for linear and nonlinear differential equations, *Quantum* 7 (2023) 913.
- [21] Y. Lee, K. Kanno, Modal Analysis on Quantum Computers Via Qubitization, arXiv preprint arXiv:2307.07478, 2023.
- [22] R. Babbush, D.W. Berry, R. Kothari, R.D. Somma, N. Wiebe, Exponential Quantum Speedup in Simulating Coupled Classical Oscillators, arXiv preprint arXiv:2303.13012, 2023.
- [23] 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, *Nat. Commun.* 5 (2014) 1.
- [24] R.R. Fergusson, L. Dellantonio, A. AlBalushi, K. Jansen, W. Dür, C.A. Muschik, Measurement-based variational quantum eigensolver, *Phys. Rev. Lett.* 126 (2021) 220501.
- [25] 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, *Phys. Rep.* 986 (2022) 1.
- [26] 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.
- [27] J.-B. You, D.E. Koh, J.F. Kong, W.-J. Ding, C.E. Png, L. Wu, Exploring Variational Quantum Eigensolver Ansatzes for the Long-range xy Model, arXiv preprint arXiv:2109.00288, 2021.
- [28] See Supplemental Information at Url for Additional Results and Further Derivations, 2023.
- [29] M. Oskin, F.T. Chong, I.L. Chuang, A practical architecture for reliable quantum computers, *Computer* 35 (2002) 79.
- [30] R.S. Smith, M.J. Curtis, W.J. Zeng, A Practical Quantum Instruction Set Architecture, arXiv preprint arXiv:1608.03355, 2016.
- [31] P.J. Karalekas, N.A. Tezak, E.C. Peterson, C.A. Ryan, M.P. daSilva, R.S. Smith, A quantum-classical cloud platform optimized for variational hybrid algorithms, *Quantum Sci. Technol.* 5 (2020) 024003.
- [32] K. Svore, A. Geller, M. Troyer, J. Azariah, C. Granade, B. Heim, V. Kliuchnikov, M. Mykhailova, A. Paz, M. Roetteler, Q# enabling scalable quantum computing and development with a high-level dsl, in: Proceedings of the real World Domain Specific Languages Workshop 2018, 2018, pp. 1–10.
- [33] D.S. Steiger, T. Häner, M. Troyer, Projectq: an open source software framework for quantum computing, *Quantum* 2 (2018) 49.
- [34] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M.S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, et al., PennyLane: Automatic Differentiation of Hybrid Quantum-classical Computations, arXiv preprint arXiv:1811.04968, 2018.
- [35] Amazon Web Services, Amazon Braket, 2020. (<https://aws.amazon.com/>).
- [36] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F.J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, et al., Qiskit: An Open-source Framework for Quantum Computing, 2019 (Accessed on: Mar 16).
- [37] V. Omole, A. Tyagi, C. Carey, A. Hanus, A. Hancock, A. Garcia, J. Shedenhelm, Cirq: A Python Framework for Creating, Editing, and Invoking Quantum Circuits, 2020.
- [38] M. Broughton, T. Verdon, T. McCourt, A.J. Martinez, J.H. Yoo, S.V. Isakov, P. Massey, R. Halavati, M.Y. Niu, A. Zlokapa, et al., Tensorflow Quantum: A Software Framework for Quantum Machine Learning, arXiv preprint arXiv:2003.02989, 2020.
- [39] T.J. Stavenger, E. Crane, K. Smith, C.T. Kang, S.M. Girvin, N. Wiebe, Bosonic qiskit, arXiv preprint arXiv:2209.11153, 2022.
- [40] J.-G. Liu, Y.-H. Zhang, Y. Wan, L. Wang, Variational quantum eigensolver with fewer qubits, *Phys. Rev. Res.* 1 (2019) 023025.
- [41] Y.S. Yordanov, V. Armaos, C.H. Barnes, D.R. Arvidsson-Shukur, Qubit-excitation-based adaptive variational quantum eigensolver, *Commun. Phys.* 4 (2021) 228.
- [42] N.C. Hugh Collins, IBM Unveils 400 Qubit-Plus Quantum Processor and Next-Generation IBM Quantum System Two Kernel Description, 2022. (<https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two>).
- [43] T.G. Wong, Introduction to Classical and Quantum Computing, 2022.
- [44] N. Moll, P. Barkoutsos, L.S. Bishop, J.M. Chow, A. Cross, D.J. Egger, S. Filipp, A. Fuhrer, J.M. Gambetta, M. Ganzhorn, et al., Quantum optimization using variational algorithms on near-term quantum devices, *Quantum Sci. Technol.* 3 (2018) 030503.
- [45] X. Bonet-Monroig, H. Wang, D. Vermetten, B. Senjean, C. Moussa, T. Bäck, V. Dunjko, T.E. O'Brien, Performance comparison of optimization methods on variational quantum algorithms, *Phys. Rev. A* 107 (2023) 032407.
- [46] X.-M. Zhang, T. Li, X. Yuan, Quantum state preparation with optimal circuit depth: implementations and applications, *Phys. Rev. Lett.* 129 (2022) 230504.
- [47] E. Malveti, R. Iten, R. Colbeck, Quantum circuits for sparse isometries, *Quantum* 5 (2021) 412.
- [48] N. Gleinig, T. Hoefler, An efficient algorithm for sparse quantum state preparation, in: Proceedings of the 58th ACM/IEEE Design Automation Conference (DAC), IEEE, 2021, pp. 433–438.
- [49] Q.D. Team, SPSA kernel description, 2023.
- [50] Q.D. Team, COBYLA Kernel Description, 2023.
- [51] Q.D. Team, SLSQP Kernel Description, 2023.
- [52] Q.D. Team, LBFGSB Kernel Description, 2023.
- [53] John Wright, Quantum Computation, lecture2:quantummathbasics, 2015.
- [54] Qiskit Contributors, Qiskit: An Open-source Framework for Quantum Computing, 2023.
- [55] J. Allcock, P. Yuan, S. Zhang, Does Qubit Connectivity Impact Quantum Circuit Complexity? arXiv preprint arXiv:2211.05413, 2022.
- [56] P. World, Putting-quantum-noise-to-work Kernel Description, 2018.
- [57] X.-D. Yu, J. Shang, O. Gühne, Statistical methods for quantum state verification and fidelity estimation, *Adv. Quantum Technol.* 5 (2022) 2100126.
- [58] A. Holmes, S. Johri, G.G. Guerreschi, J.S. Clarke, A.Y. Matsuura, Impact of qubit connectivity on quantum algorithm performance, *Quantum Sci. Technol.* 5 (2020) 025009.
- [59] A.W. Cross, L.S. Bishop, S. Sheldon, P.D. Nation, J.M. Gambetta, Validating quantum computers using randomized model circuits, *Phys. Rev. A* 100 (2019) 032328.
- [60] K. Miller, C. Broomfield, A. Cox, J. Kinast, B. Rodenburg, An Improved Volumetric Metric for Quantum Computers Via More Representative Quantum Circuit Shapes, arXiv preprint arXiv:2207.02315, 2022.
- [61] K. Kechedzhi, S. Isakov, S. Mandrà, B. Villalonga, X. Mi, S. Boixo, V. Smelyanskiy, Effective Quantum Volume, Fidelity and Computational Cost of Noisy Quantum Processing Experiments, arXiv preprint arXiv:2306.15970, 2023.
- [62] J.G., et al., Driving Quantum Performance: More Qubits, Higher Quantum Volume, and Now A Proper Measure of Speed, 2021. (<https://research.ibm.com/blog/circuit-layer-operations-per-second>).
- [63] N. Earnest, C. Tornow, D.J. Egger, Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware, *Phys. Rev. Res.* 3 (2021) 043088.
- [64] E. Younis, C. Iancu, Quantum circuit optimization and transpilation via parameterized circuit instantiation, in: Proceedings of the IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE, 2022, pp. 465–475.
- [65] S. Endo, Z. Cai, S.C. Benjamin, X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, *J. Phys. Soc. Jpn.* 90 (2021) 032001.
- [66] M. Salm, J. Barzen, U. Breitenbücher, F. Leymann, B. Weder, K. Wild, The nlsq analyzer: automating the selection of quantum computers for quantum algorithms,

- in: Symposium and Summer School on Service-Oriented Computing, Springer, 2020, pp. 66–85.
- [67] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 1999.
 - [68] J. Haferkamp, P. Faist, N.B. Kothakonda, J. Eisert, N. YungerHalpern, Linear growth of quantum circuit complexity, *Nat. Phys.* 18 (2022) 528.
 - [69] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, M.D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* 10 (2020) 021067.
 - [70] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, F. Petruccione, A comparison of various classical optimizers for a variational quantum linear solver, *Quantum Inf. Process.* 20 (2021) 202.
 - [71] J.C. Spall, An overview of the simultaneous perturbation method for efficient optimization, *Johns. Hopkins Apl. Tech. Dig.* 19 (1998) 482.
 - [72] M.J. Powell, A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation, Springer, 1994.
 - [73] D. Kraft, A Software Package for Sequential Quadratic Programming, *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
 - [74] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization, *ACM Trans. Math. Softw. (TOMS)* 23 (1997) 550.
 - [75] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, P.J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nat. Commun.* 12 (2021) 1791.
 - [76] R. Mandelbaum, A Useful Application for 127-qubit Quantum Processors with Error Mitigation, 2023. (<https://research.ibm.com/blog/utility-toward-useful-quantum>).
 - [77] IBM, The IBM Quantum Development Roadmap, 2023. (<https://www.ibm.com/quantum/roadmap>).
 - [78] A. Cho, IBM Promises 1000-qubit Quantum Computer—A Milestone—by 2023, 2020. (<https://www.science.org/content/article/ibm-promises-1000-qubit-quantum-computer-milestone-2023>).
 - [79] P. Smith-Goodson, IBM Rolls Out A Game-Changing 127-Qubit Quantum Computer That Redefines Scale, Quality, And Speed, 2021. (<https://www.forbes.com/sites/monitors/2021/12/02/ibm-rolls-out-a-game-changing-127-qubit-quantum-computer-that-redefines-scale-quality-and-speed/?sh=5d21d9353f40>).
 - [80] 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.
 - [81] E. Van Den Berg, Z.K. Mineev, A. Kandala, K. Temme, Probabilistic error cancellation with sparse pauli-lindblad models on noisy quantum processors, *Nat. Phys.* 1 (2023).

Supplemental Material
to
*Quantum Computing for Solid Mechanics and Structural
Engineering*
– *a Demonstration with Variational Quantum Eigensolver*

Yunya Liu,¹ Jiakun Liu,² Jordan R. Raney,² and Pai Wang¹

¹*Department of Mechanical Engineering,*

University of Utah, Salt Lake City, UT, USA

²*Department of Mechanical Engineering and Applied Mechanics,*

University of Pennsylvania, Philadelphia, PA, USA

CONTENTS

Alternative Entanglement Depths	3
Validation of Convergence	5
Error Evaluations	8
Mesh adjustment for targeted Degrees of Freedom	11

ALTERNATIVE ENTANGLEMENT DEPTHS

In the main text, we mention the VQE entanglement pattern can be either CZ, CX, or CRX and provide a depth = 1 with entanglement pattern CX as an example. Here we supplement all the entanglement patterns in Fig.S1 with depth = 3 as a complete illustration of a parameterized quantum circuit in standard VQE.

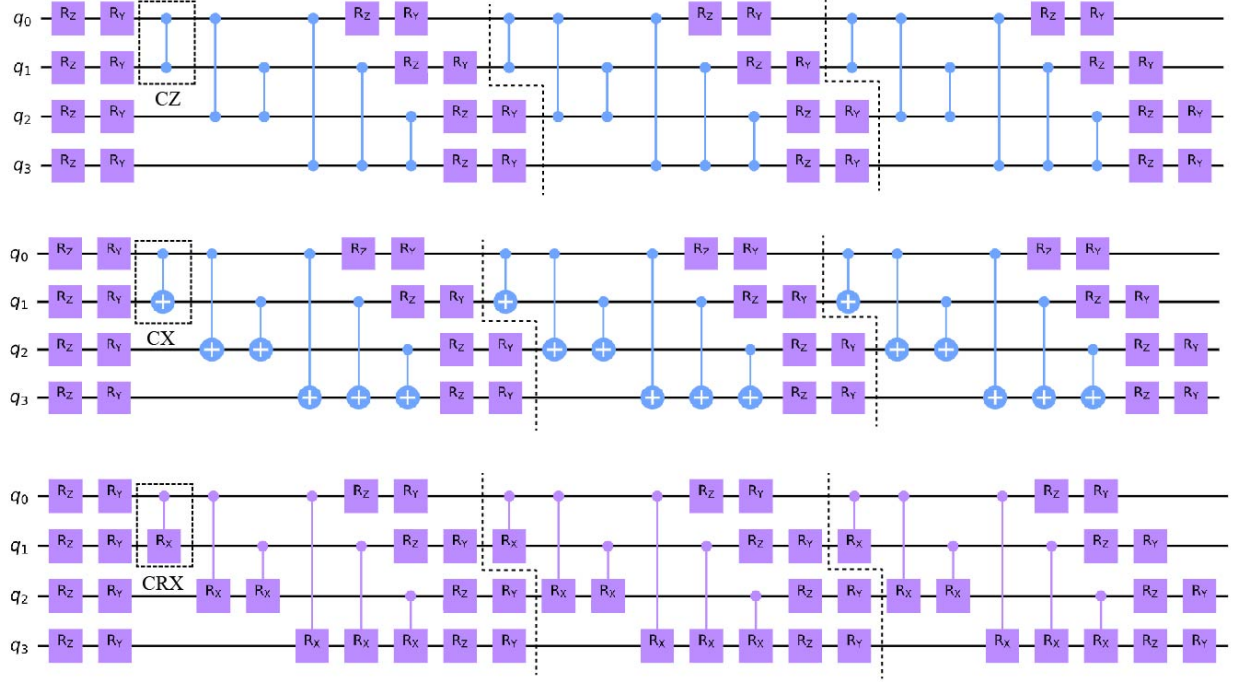


Figure S1. The entanglement patterns CZ, CX, and CRX in depth = 3.

The complete parameterized variational ansatz

$$U(\boldsymbol{\theta}) = [U_{\text{rot},3}][U_{\text{ent},3}][U_{\text{rot},2}][U_{\text{ent},2}][U_{\text{rot},1}][U_{\text{ent},1}][U_{\text{pre}}]. \quad (\text{S1})$$

Changing entanglement pattern and depth has no effect on U_{pre} , but U_{ent} and U_{rot} need new definitions. Assume entanglement pattern CZ is applied, the states are fully entangled together by the entanglement operator composing two-qubit CZ gates, and each particular operator from Eq.(S1) is

$$U_{\text{pre}} = \left[\bigotimes_{j=0}^{N-1} R_y(\theta_{2,j}) \right] \left[\bigotimes_{j=0}^{N-1} R_z(\theta_{1,j}) \right] \quad (\text{S2})$$

$$U_{\text{ent}} = U_{\text{ent},1} = U_{\text{ent},2} = U_{\text{ent},3} = \prod_{j_1=0}^{N-2} \prod_{j_2=j_1+1}^{N-1} \text{CZ}_{j_1,j_2}, \quad (\text{S3})$$

$$U_{\text{rot},1} = \left[\bigotimes_{j=0}^{N-1} R_y(\theta_{4,j}) \right] \left[\bigotimes_{j=0}^{N-1} R_z(\theta_{3,j}) \right] \quad (\text{S4})$$

$$U_{\text{rot},2} = \left[\bigotimes_{j=0}^{N-1} R_y(\theta_{6,j}) \right] \left[\bigotimes_{j=0}^{N-1} R_z(\theta_{5,j}) \right] \quad (\text{S5})$$

$$U_{\text{rot},3} = \left[\bigotimes_{j=0}^{N-1} R_y(\theta_{8,j}) \right] \left[\bigotimes_{j=0}^{N-1} R_z(\theta_{7,j}) \right] \quad (\text{S6})$$

where $\boldsymbol{\theta} = \{\theta_{s,j}\}$ is a set of variational parameters, which control all the single-qubit gates, $s = 1, \dots, 8$ indicates the “slice” of the single-qubit gates, and $j, j_1, j_2 \in \{0, 1, \dots, N-1\}$ are indices of qubits. The combined effects of Eqs.(S3) and (S4) constitutes one “depth”. Together with another two “depth” of $[U_{\text{rot},2}][U_{\text{ent},2}]$ and $[U_{\text{rot},3}][U_{\text{ent},3}]$, the total parameterized variational ansatz is complete.

As for the entanglement pattern CRX with depth 3, the Eq.(S3) of U_{ent} is to be replaced by

$$\begin{aligned} U_{\text{ent},1} &= \prod_{j_1=0}^{N-2} \prod_{j_2=j_1+1}^{N-1} \text{CRX}(\theta_{1,k})_{j_1,j_2} \\ U_{\text{ent},2} &= \prod_{j_1=0}^{N-2} \prod_{j_2=j_1+1}^{N-1} \text{CRX}(\theta_{2,k})_{j_1,j_2} \\ U_{\text{ent},3} &= \prod_{j_1=0}^{N-2} \prod_{j_2=j_1+1}^{N-1} \text{CRX}(\theta_{3,k})_{j_1,j_2} \end{aligned} \quad (\text{S7})$$

where $k = \{1, 2, \dots, \frac{N!}{2(N-2)!}\}$, $\boldsymbol{\theta} = \{\theta_{d,k}\}$ is another set of variational parameters, which control all the two-qubit gates (i.e., CRX). Here, $d = 1, 2, 3$ indicates the “slice” of the two-qubit gates.

VALIDATION OF CONVERGENCE

Here we present the convergence of fundamental quantum eigenvalue estimates for Case (I) hexagonal truss system with the number of free DOFs: 2^N ($N = 6, 7, 8, 9, 10, 11, 12, 13$) in Fig. S2 and S3. There's no determinant choice of a specific *variational quantum eigensolver* setting across all Hamiltonians.

In Fig. S2 and S3, the top subplots apply different optimizer choices with entanglement pattern CZ and depth 3, the middle subplots apply different entanglement patterns with optimizer COBYLA and entanglement depth 3, and the bottom subplots apply different entanglement depths with optimizer COBYLA and entanglement pattern CZ. The black dashed line denotes the classical numerical eigenvalue for the same Hamiltonian. The black vertical dashed line shows the iteration time of reaching the minimum eigenvalue estimate. Based on the control variable of different optimizer choices, entanglement patterns, and entanglement depths, the convergence rates vary. Take Fig. S2(b) for instance, the top subplot contains the convergence behavior by applying optimizers SPSA (blue solid line) and COBYLA (orange solid line), the former reaches optimal solution at around iteration time 10^5 , whereas the latter reach the same optimal solution at 1/10 time of the former. In addition, adding depths in a parameterized quantum circuit won't guarantee a more reliable optimal solution due to the conflict of expressivity and accuracy, as well as the barren plateau phenomenon. This is evidenced by Fig. S2(c), the bottom subplot covers entanglement depth from 1 to 10, in which depth = 9 delivers the least optimal solution than all other depths less than it.

Further, in Fig. S2(c)(d), and Fig. S3(f), the middle subplots show the convergence accuracy is not prioritized by entanglement pattern CZ across all problem-specific Hamiltonians, though CZ preserves the Z-rotation and benefits the capturing of local minima during tuning parameters in optimization. However, the entanglement patterns CX and CRX involve X-rotation, aside from the single-qubit rotations about Y and Z-axes, and potentially interfere with all qubits more thoroughly.

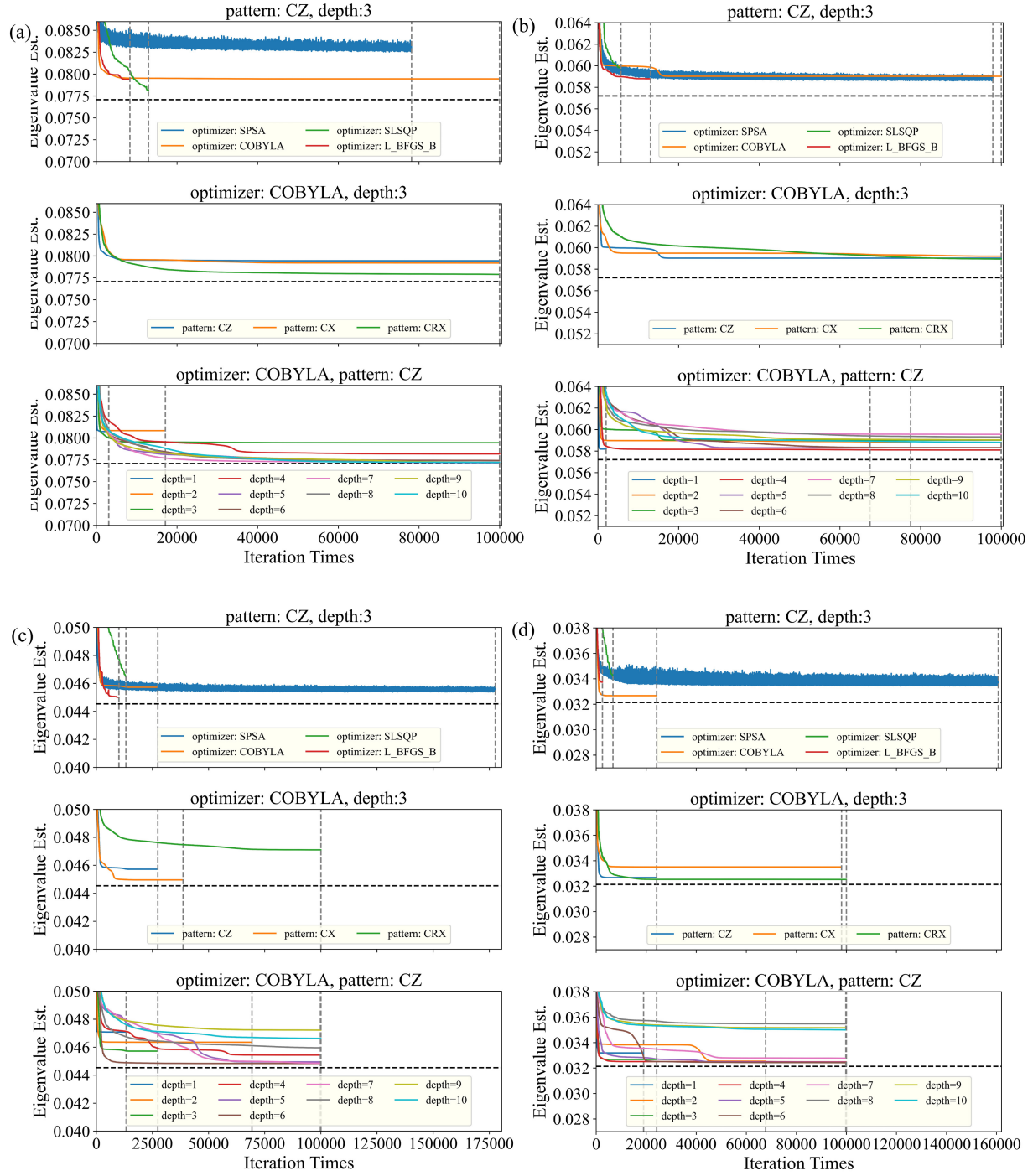


Figure S2. Convergence plots for hexagonal truss system with degrees of freedom of 2^N (a) $N=6$, (b) $N=7$, (c) $N=8$, (d) $N=9$

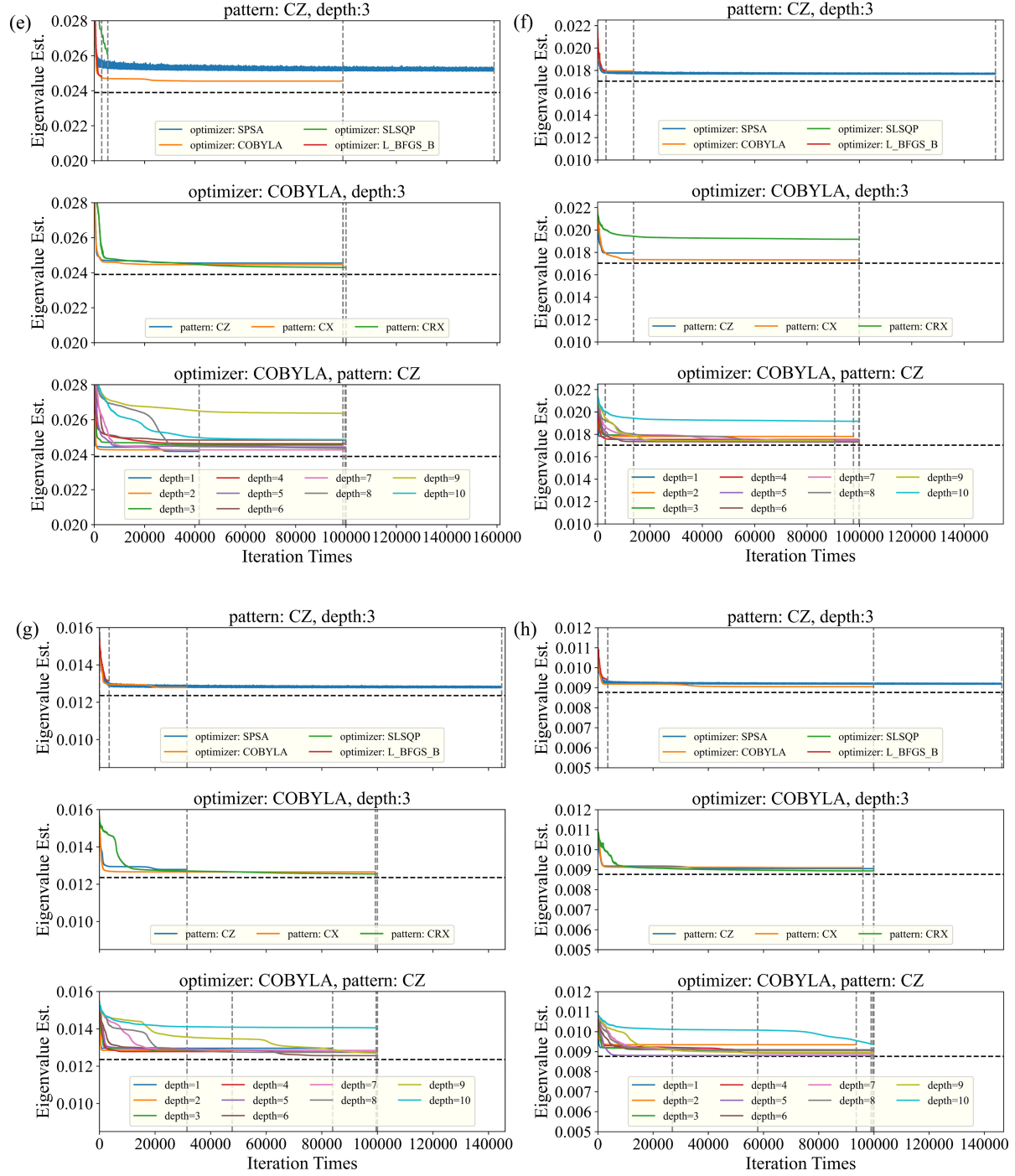


Figure S3. Convergence plots for hexagonal truss system with degrees of freedom of 2^N (e) $N=10$, (f) $N=11$, (g) $N=12$, (h) $N=13$

ERROR EVALUATIONS

In the main text, we display the relative error of VQE estimates compared to a classical solver. The comparison of ME and SDE is shown in Table I to III for Case (I), Table IV to VI for Case (II), and Table VII to IX for Case (III).

Case (I) - Table.I-III unveils that, the MEs consistently lie within a narrow 1% to 3% range for depths ranging from 3 to 7. Given the trade-off between the expressivity of the quantum circuit and accuracy, the preferred set of parameters is optimizer COBYLA with entanglement pattern CZ and depth = 4, yielding a ME of 2.220% and an SDE of 0.802%.

Case (II) - Referencing Table.IV-VI, the employment of optimizer L-BFGS-B, entanglement pattern CZ, and depth = 1 produces a ME of 0.005% and an SDE of 0.008%.

Case (III) - As evidenced by Table.VII-IX, utilizing optimizer L-BFGS-B with entanglement pattern CX and depth = 1 results in a ME of 1.812% and an SDE of 1.735%.

For Figs. (4) and (5) in the main text, the outlier data points, which rocket far beyond errors of 6% and 25%, respectively, are excluded to emphasize the reliable data points. But we display the MEs and SDEs for overall results from Table.IV to IX.

The optimal set of parameters for Case (I) is not the most reliable choice for Case (III), as shown in Table X to XII, and can be further reduced to data shown from Table VII to IX, which suggests that there is still room for improvement in the VQE estimates.

Here, we analyze the impact of main factors among cases:

When comparing the MEs and SDEs for different choices of optimizers in Table.I, IV, and VII, it suggests that the sparse Hamiltonians from Case (I) paired with optimizer COBYLA offer superior predictions of VQE estimates. Conversely, lumped Hamiltonians with reduced complexity from Case (II) and (III) feature lower MEs and SDEs when coupled with optimizer L-BFGS-B.

The variance in MEs and SDEs due to entanglement patterns, as depicted in Tables.II, V, and VIII, is attributed to the structural feature of the one-dimensional Case (II) used for illustrative simplicity, where each node contains a concentrated lumped mass, and henceforth is not directly interacting with their nearby nodes. In contrast, the two-dimensional Case (III), consisting of shell elements is more structurally integrated and responsive to the

interaction among nodal displacements due to a relatively more prominent coupling effect between nodes. Regarding the effect of entanglement pattern CX in the same Tables, it demonstrates reliable VQE estimates that remain sufficiently low MEs and SDEs.

Finally, the impact of entanglement depths on the prediction accuracy of the VQE estimates requires additional investigation, since the calculation is intractable during the process of optimizing parameters θ . Typically, a deeper entanglement is more expressive which amplifies predictive accuracy. Yet, for Case (II) and (III), the MEs and SDEs in Table.III, IV, and IX are intensified and the barren plateau phenomenon is exacerbated, possibly due to reaching the criterion *maxiter* instead of threshold *tol*. In contrast, entanglement depths from 3 to 7 with MEs below 3.177% and SDEs below 1.323% tend to be more advantageous for sparse Hamiltonians in Case (I).

Table I. *CZ* pattern and three depths

Error evaluation		
	mean	std
SPSA	3.891	1.422
COBYLA	3.177	0.974
L.BFGS.B	4.383	1.961
SLSQP	12.592	9.870

Table II. COBYLA optimizer and three depths

Error evaluation		
	mean	std
<i>CZ</i>	3.177	0.974
<i>CX</i>	2.655	1.123
<i>CRX</i>	2.281	1.432

Table III. COBYLA optimizer and *CZ* pattern

Error evaluation					
	mean	std		mean	std
1	3.851	1.526	6	1.871	1.230
2	3.713	1.670	7	2.069	1.323
3	3.177	0.974	8	3.472	2.808
4	2.220	0.802	9	4.553	3.424
5	1.433	0.955	10	6.696	4.438

Table IV. *CZ* pattern and one depth

Error evaluation		
	mean	std
SPSA	2.185	1.166
COBYLA	0.006	0.011
L.BFGS.B	0.006	0.010
SLSQP	42.737	60.384

Table V. L.BFGS.B optimizer and one depth

Error evaluation		
	mean	std
<i>CZ</i>	0.006	0.008
<i>CX</i>	0.007	0.009
<i>CRX</i>	0.008	0.012

Table VI. L.BFGS.B optimizer and *CZ* pattern

Error evaluation		
	mean	std
1	0.005	0.008
2	2.008	4.486
3	0.023	0.033
4	0.420	0.585
5	0.409	0.575

Table VII. *CX* pattern and one depth

Error evaluation		
	mean	std
SPSA	7.732	7.058
COBYLA	4.378	3.857
L.BFGS.B	1.812	1.735
SLSQP	22.036	22.631

Table VIII. L.BFGS.B optimizer and one depth

Error evaluation		
	mean	std
<i>CZ</i>	5.706	5.652
<i>CX</i>	1.812	1.735
<i>CRX</i>	5.519	7.210

Table IX. L.BFGS.B optimizer and *CX* pattern

Error evaluation		
	mean	std
1	1.812	1.735
2	6.273	8.464
3	5.806	8.891
4	1.858	2.300
5	5.020	4.253

Table X. *CZ* pattern and three depths

Error evaluation		
	mean	std
SPSA	25.126	33.775
COBYLA	7.607	8.361
SLSQP	267.655	591.337
L.BFGS.B	52.330	132.800

Table XI. COBYLA optimizer and three depths

Error evaluation		
	mean	std
<i>CZ</i>	7.607	8.361
<i>CX</i>	5.047	4.177
<i>CRX</i>	8.019	7.181

Table XII. COBYLA optimizer and *CZ* pattern

Error evaluation		
	mean	std
1	15.328	25.324
2	77.806	162.811
3	7.607	8.361
4	5.692	4.293
5	9.914	15.942

MESH ADJUSTMENT FOR TARGETED DEGREES OF FREEDOM

Here we explain in more detail the developed pre-processing commands for mesh adjustment and modification to reach a targeted number of free DOFs. The overall idea is to iteratively refine the assigned element size and find a suitable value leading to a DOFs close to the targeted value, and then insert a certain number of nodes by means of element edge splitting.

Take the Case (II) plane-strain continuum as an example, assuming that the target free DOFs is $2^9 = 512$, the following steps are applied. First, as shown in Fig.S4(a), the processing command starts by creating a relatively coarse mesh by assigning a large element size such that the total number of DOFs ($n_{\text{DOF}}^{\text{all}}$) is smaller than the target value. Second, the assigned element size is gradually reduced (i.e., the number of nodes and DOFs increase) until a 'suitable' element size that would result in a free DOFs ($n_{\text{DOF}}^{\text{all}} - n_{\text{DOF}}^{\text{fixed}}$) that is close to, but smaller than the target value, has been obtained. Such a 'suitable' element size also needs to result in an even value of difference towards the target value. As shown in Fig.S4(b), the corresponding 'suitable' mesh contains 434 free DOFs, and there are still 78 extra DOFs needed to reach the target value. Then based on this difference, pick exactly $78/2 = 39$ element edges not involved in essential boundary conditions and then 'split' each of these edges by inserting a new node in its midpoint, resulting in 78 new DOFs. As shown in a magnified view in Fig.S4(d), each inserted node in a free edge would result in 2 new free DOFs (horizontal and vertical displacements), and the inserted nodes will be connected to nearby element vertices to form new elements, resulting in 'partitioning' of some elements. A simple way of picking edges as applied here is to find the longest 39 free element edges and split them, and the final resultant mesh, as shown in Fig.S4(c), has exactly 512 free DOFs.

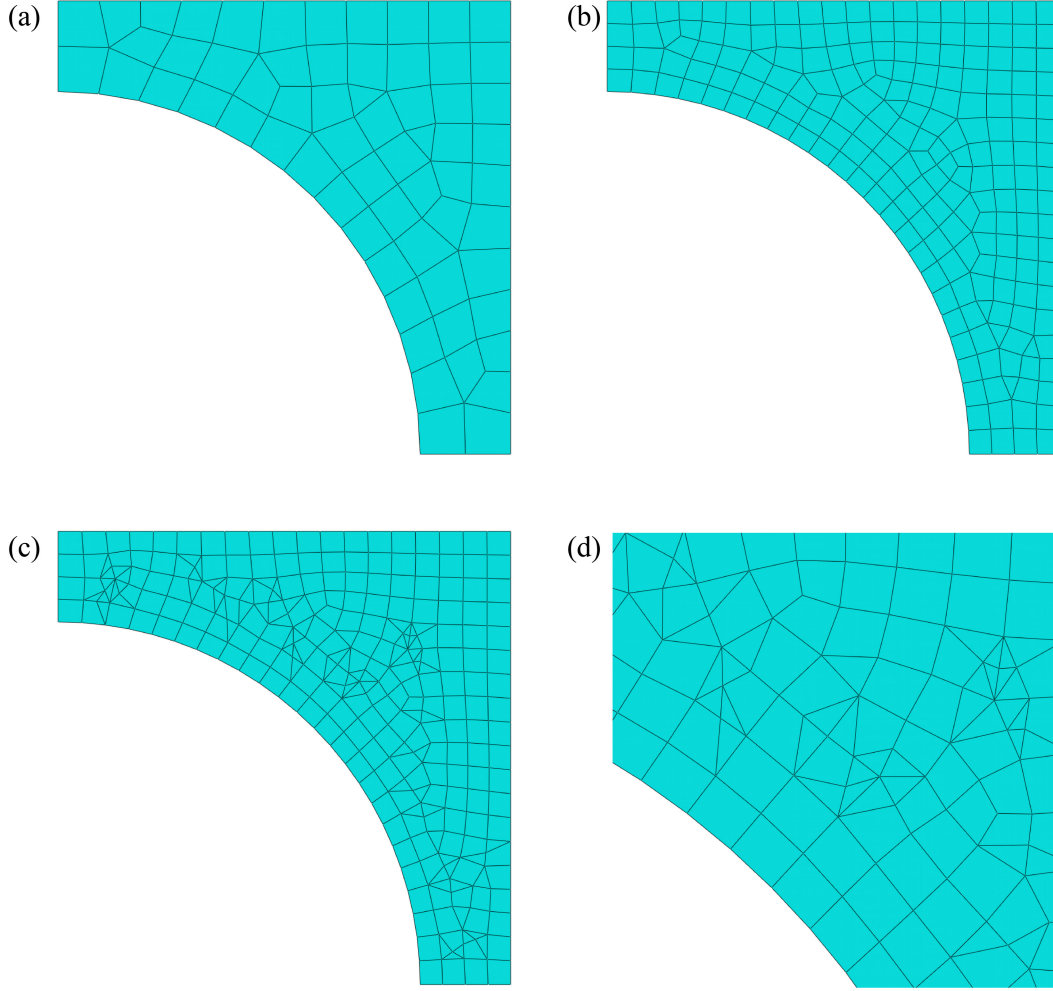


Figure S4. Example of iterative mesh adjustments to reach a targeted number of free DOFs. (a) Initial coarse mesh with 148 free DOFs. (b) A suitably fine mesh having 434 free DOFs. (c) The final mesh contains $2^9 = 512$ free DOFs after splitting 38 edges. (d) A magnified view of the final mesh in (c) to illustrate the splitting of element edges and resultant element partitioning.