

# Closed-loop calculations of electronic structure on a quantum processor and a classical supercomputer at full scale

|                                                                                                                                                                   |                                                                                                                                                              |                                                                                                                                                                    |                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tomonori Shirakawa<br><i>Center for Computational Science</i><br><i>RIKEN</i><br>Kobe, Japan<br>0000-0001-7923-216X<br>t-shirakawa@riken.jp                       | Javier Robledo-Moreno<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>j.robledomoreno@ibm.com                | Toshinari Itoko<br><i>IBM Quantum</i><br><i>IBM Research – Tokyo</i><br>Tokyo, Japan<br>0009-0006-2611-2301<br>itoko@jp.ibm.com                                    | Vinay Tripathi<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>0000-0003-1418-9046<br>tripathi@ibm.com |
| Kento Ueda<br><i>IBM Quantum</i><br><i>IBM Research – Tokyo</i><br>Tokyo, Japan<br>Kento.Ueda@ibm.com                                                             | Yukio Kawashima<br><i>IBM Quantum</i><br><i>IBM Research – Tokyo</i><br>Tokyo, Japan<br>0000-0001-5918-1211<br>yukio.kawashima@ibm.com                       | Lukas Broers<br><i>Center for Computational Science</i><br><i>RIKEN</i><br>Kobe, Japan<br>lukas.broers@riken.jp                                                    | William Kirby<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>William.Kirby@ibm.com                    |
| Himadri Pathak<br><i>Interdisciplinary Theoretical and</i><br><i>Mathematical Sciences RIKEN</i><br>Wako, Japan<br>0000-0002-5919-8002<br>himadri.pathak@riken.jp | Hanhee Paik<br><i>IBM Quantum</i><br><i>IBM Research – Tokyo</i><br>Tokyo, Japan<br>hanhee.paik@us.ibm.com                                                   | Miwako Tsuji<br><i>Center for Computational Science</i><br><i>RIKEN</i><br>Kobe, Japan<br>miwako.tsuji@riken.jp                                                    |                                                                                                                                                        |
| Yuetsu Kodama<br><i>Center for Computational Science</i><br><i>RIKEN</i><br>Kobe, Japan<br>yuetsu.kodama@riken.jp                                                 | Mitsuhisa Sato<br><i>Center for Computational Science</i><br><i>RIKEN</i><br>Kobe, Japan<br>msato@riken.jp                                                   | Constantinos Evangelinos<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>0000-0003-1418-9046<br>cevange@us.ibm.com |                                                                                                                                                        |
| Seetharami Seelam<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>sseelam@us.ibm.com                              | Robert Walkup<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>walkup@us.ibm.com                              | Seiji Yunoki<br><i>Pioneering Research Institute</i><br><i>RIKEN</i><br>Wako, Japan<br>yunoki@riken.jp                                                             |                                                                                                                                                        |
| Mario Motta<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>0000-0003-1647-9864<br>mario.motta@ibm.com            | Petar Jurcevic<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>0000-0003-1234-6386<br>petar.jurcevic@ibm.com | Hiroshi Horii<br><i>IBM Quantum</i><br><i>IBM Research – Tokyo</i><br>Tokyo, Japan<br>horii@jp.ibm.com                                                             | Antonio Mezzacapo<br><i>IBM Quantum</i><br><i>IBM T.J. Watson Research Center</i><br>Yorktown Heights, NY, USA<br>mezzacapo@ibm.com                    |

**Abstract**—Quantum computers must operate in concert with classical computers to deliver on the promise of quantum advantage for practical problems. To achieve that, it is important to understand how quantum and classical computing can interact together, and how one can characterize the scalability and efficiency of hybrid quantum-classical workflows. So far, early

experiments with quantum-centric supercomputing workflows have been limited in scale and complexity. Here, we use a Heron quantum processor deployed on premises with the entire supercomputer Fugaku to perform the largest computation of electronic structure involving quantum and classical high-performance computing. We design a closed-loop workflow between the quantum processors and 152,064 classical nodes of Fugaku, to approximate the electronic structure of chemistry models beyond the reach of exact diagonalization, with accuracy comparable to some all-classical approximation methods. Our work pushes the limits of the integration of quantum and classical high-performance computing, showcasing computational resource orchestration at the largest scale possible for current classical supercomputers.

## I. INTRODUCTION AND BACKGROUND

The electronic structure problem amounts to solving the static Schrödinger equation  $\hat{H}|\Psi\rangle = E|\Psi\rangle$ . For molecular systems, it can be formulated on a discrete molecular basis set, where  $\hat{H}$  is the Born-Oppenheimer Hamiltonian

$$\hat{H} = \sum_{pr\sigma} h_{pr} \hat{a}_{p\sigma}^\dagger \hat{a}_{r\sigma} + \sum_{prqs \atop \sigma\tau} \frac{(pr|qs)}{2} \hat{a}_{p\sigma}^\dagger \hat{a}_{q\tau}^\dagger \hat{a}_{s\tau} \hat{a}_{r\sigma}. \quad (1)$$

Here we have defined the fermionic creation/annihilation operator  $\hat{a}_{p\sigma}^\dagger/\hat{a}_{p\sigma}$  associated to the  $p$ -th basis set element and the spin- $z$  polarization  $\sigma$ , while  $h_{pr}$  and  $(pr|qs)$  are the one- and two-body electronic integrals. The Hamiltonian is defined up to an additive constant.

For large systems that exhibit pronounced multireference character, solving the Hamiltonian Eq.(1) is a challenging task. While several classical approximation methods exist, like perturbative approaches, wavefunction ansatzes or Monte Carlo integration [1], [2], all of them have a specific domain of applicability, depending on the method considered.

On the other hand, finding the low-energy spectrum of  $\hat{H}$  is also a key application of quantum computing. The quantum phase estimation algorithm was initially proposed in the early days of quantum computing [3] to estimate the spectrum of many-body Hamiltonians such as Eq. (1). However, it cannot scale on pre-fault-tolerant quantum processors due to large circuit size and substantial depth requirements.

Instead of quantum phase estimation, many quantum algorithms have been proposed to enable the simulation of ground states in current pre-fault-tolerant quantum devices. Historically, the well-known Variational Quantum Eigensolver (VQE) [4] and its variants make use of the variational principle to construct an ansatz in the form of a parametric quantum circuit. However, for complex Hamiltonians such as the molecular problem, variational algorithms relying on the computation of expectation values on quantum devices face fundamental scaling limitations due to the high measurement overhead [5]. These issues have confined the scaling of quantum chemistry simulations to a few qubits [6]. Recent developments in quantum-centric algorithms provide a pathway to scaling to larger systems with quantum sample-based workflows, such as quantum-selected configuration interaction [7], which proposed to use quantum computers for selected-CI calculations, and the sample-based quantum diagonalization methods [8],

[9], which led to reliable electronic structure simulations with quantum data for up to 85 qubits.

Here, we target the ground-state properties of methyl-capped  $[\text{Fe}_2\text{S}_2(\text{SCH}_3)_4]^{2-}$  and  $[\text{Fe}_4\text{S}_4(\text{SCH}_3)_4]^{2-}$  iron-sulfur clusters – henceforth abbreviated as [2Fe-2S] and [4Fe-4S] and whose geometries are depicted in Fig. 1 – using active spaces of 50 electrons in 36 orbitals and 54 electrons in 36 orbitals, respectively. Iron-sulfur clusters are a universal biological motif. They are found in a variety of metalloproteins – collectively referred to as iron-sulfur proteins – which are characterized by the presence of one or more sulfide-linked di-, tri-, or tetra-iron centers in variable oxidation states, collectively forming a cluster. Iron-sulfur proteins include ferredoxins, hydrogenases, and nitrogenase, which participate in remarkable chemical reactions at ambient temperature and pressure [10], [11].

The electronic structure of iron-sulfur clusters – featuring multiple low-energy wavefunctions with diverse magnetic and electronic character – is key to their rich chemical activity. At the same time, it poses a formidable challenge for classical numerical methods. Practical simulations – Hartree-Fock and density functional theory – rely on a mean-field approximation that treats only classical-like quantum states without entanglement. This approximation is fundamentally inadequate for iron-sulfur clusters, because the iron  $3d$  shells are partially filled and near-degenerate on the Coulomb interaction scale, leading to strong electronic correlation in low-energy wavefunctions, and invalidating any independent-particle picture and the related concept of a mean-field electronic configuration. Broken-symmetry mean-field calculations – the earliest approach used for these clusters – only provide averaged properties across multiple electronic states and cannot resolve individual wavefunctions [12].

Accurate computation must instead involve entangled superpositions of multiple electronic configurations. In principle, the number of such configurations scales combinatorially with the number of iron and sulfur atoms in the cluster, and therefore quickly becomes intractable. However, by employing methods that impose structure on these entangled superpositions [13], [14], it is possible to elevate quantum simulations from the mean-field level to the level of correlated many-body quantum mechanics required to describe the cluster’s low-energy properties, while substantially mitigating the exponential complexity of a general quantum-mechanical formulation.

To investigate multireference ground-state wavefunctions in [2Fe-2S] and [4Fe-4S], we use active spaces of 50 electrons in 36 orbitals (of Fe[3d,4s,4p] and S[3p] character) and 54 electrons in 36 orbitals (of Fe[3d], S[3p], and ligand[ $\sigma$ ] character). The dimensions of the corresponding Hilbert spaces,  $\binom{36}{25} * \binom{36}{25} = 3.61 \cdot 10^{17}$  and  $\binom{36}{27} * \binom{36}{27} = 8.86 \cdot 10^{15}$ , are several orders of magnitude above the current scale of exact diagonalization,  $1.0 \cdot 10^{12}$  [15]. As a result, meaningful electronic structure calculations must rely on approximate methods. Furthermore, the breakdown of the mean-field approximation in iron-sulfur clusters implies that conventional methods for classical computers – restricted Hartree-Fock (RHF), configuration

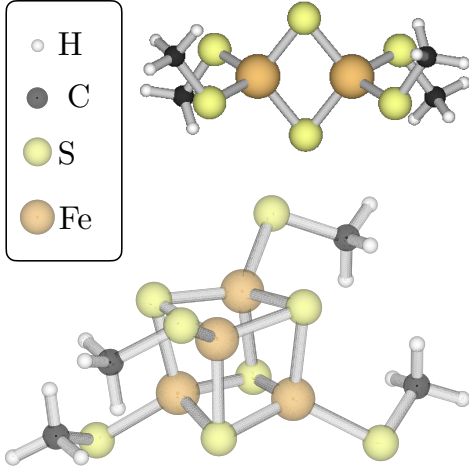


Fig. 1. Geometries of the [2Fe-2S] (top) and [4Fe-4S] (bottom) clusters considered in this study.

interaction singles and doubles (CISD), and coupled cluster singles and doubles (CCSD) – provide inaccurate results. Instead, the state of the art (SOTA) for classical electronic structure computations in iron-sulfur clusters is the density matrix renormalization group (DMRG) [12] yielding ground-state energy estimates of  $E_{\text{DMRG},[2\text{Fe}-2\text{S}]} = -5049.217 E_h$  and  $E_{\text{DMRG},[4\text{Fe}-4\text{S}]} = -327.239 E_h$ .

In the context of quantum computing, active spaces of electrons in 36 spatial orbitals require 72 qubits using the standard Jordan-Wigner map for fermion-to-qubit degrees of freedom [16]. A previous study [8] investigated the electronic structure of iron-sulfur clusters with the Sample-based Quantum Diagonalization (SQD) method in chemically realistic active spaces. This is the current SOTA for quantum computations on pre-fault-tolerant quantum computers, both in terms of problem size and accuracy: for the [4Fe-4S] cluster, it obtained an estimate for the ground-state energy of  $-326.635 E_h$ , between RHF ( $E_{\text{RHF}} = -326.547 E_h$ ) and above CISD ( $E_{\text{CISD}} = -326.742 E_h$ ) – two methods that, as discussed above, essentially break down in this situation. Understanding scaling of integrated quantum-classical methods is a mandatory requirement to match classical state-of-the-art methods in these and other, equally and even more challenging, instances of the electronic structure problem.

Our approach to approximate the ground states of [2Fe-2S] and [4Fe-4S] builds on the SQD method [8], which uses quantum-centric supercomputing architectures [17]. The active-space Hamiltonian Eq. (1), for both [2Fe-2S] and [4Fe-4S], is mapped onto a 72-qubit operator using a Jordan-Wigner transformation. Then, a quantum circuit prepares a wavefunction ansatz  $|\Psi\rangle$ , which approximates the support of the exact ground state. Recent experimentation of the SQD method for chemistry relied on the Local Unitary Cluster Jastrow (LUCJ) circuit [18], with parameters derived from CCSD calculations.

A set of quantum measurements  $\mathcal{S}$  from the LUCJ quantum circuit first undergoes a configuration recovery step, and then

is processed by a classical computer to project and diagonalize  $\hat{H}$  on  $\mathcal{S}$

$$\hat{H}_{\mathcal{S}} = \hat{\mathcal{P}}_{\mathcal{S}} \hat{H} \hat{\mathcal{P}}_{\mathcal{S}}, \quad (2)$$

where the projector  $\hat{\mathcal{P}}_{\mathcal{S}}$  is  $\hat{\mathcal{P}}_{\mathcal{S}} = \sum_{\mathbf{x} \in \mathcal{S}} |\mathbf{x}\rangle\langle\mathbf{x}|$ . Under the Jordan-Wigner encoding, the sampled determinants are identified with the electronic configurations  $\mathbf{x} \in \mathcal{S}$ . The size and constituents of  $\mathcal{S}$  determine the accuracy of the approximate target eigenstate. The corresponding classical heuristics to search the set of relevant configurations to include in  $\mathcal{S}$  fall under the general umbrella of Selected Configuration Interaction (SCI) methods [19]–[22]. Recent studies suggest that a quantum circuit  $\Psi$  can produce accurate statistical models  $p(\mathbf{x}) = |\langle\mathbf{x}|\Psi\rangle|^2$  to produce better quality subspaces than some SCI heuristics [8].

Existing pre-fault-tolerant quantum computers are subject to quantum noise. In the present setting, the effect of the noise is to transform the ideal sample distribution  $P_{\Psi} = |\langle\mathbf{x}|\Psi\rangle|^2$  to some other  $\tilde{P}_{\Psi}$ , the distribution from which we sample actual measurement results yielding a noisy set of configurations  $\tilde{\mathcal{X}}$ . This effective distribution introduces unwanted configurations that do not contribute to low-energy states. We can think of this as a broadening of the distribution  $P_{\Psi}$ , and consequently only some fraction of the sampled configurations  $\tilde{\mathcal{X}}$  correspond to actual quantum signal. To mitigate this, a *self-consistent configuration recovery* technique was introduced in [8] to probabilistically recover noiseless samples from the noisy measured set  $\tilde{\mathcal{X}}$ , which is used to build  $\mathcal{S}$ .

The classical diagonalization step produces a wavefunction described by the expansion of a general many-body state in  $\mathcal{S}$  (of polynomial size in the number of electrons and orbitals) of the basis of single-particle electronic configurations:

$$|\psi_{\mathcal{S}}\rangle = \sum_{\mathbf{x} \in \mathcal{S}} \psi_{\mathbf{x}} |\mathbf{x}\rangle, \quad (3)$$

where,  $\mathbf{x} \in \{0,1\}^{2N_{\text{orb}}}$  are bitstrings whose length is twice the number of spatial orbitals  $N_{\text{orb}}$ . The first and second halves of each bitstring represent the occupancy of the spin-up and spin-down orbitals respectively:  $|\mathbf{x}\rangle$  represents a Slater determinant – or electronic configuration – of the form  $|\mathbf{x}\rangle = \prod_{p,\sigma} (\hat{a}_{p\sigma}^{\dagger})^{x_{p\sigma}} |0\rangle$ , where  $p = 1, 2, \dots, N_{\text{orb}}$ , and  $\sigma \in \{\uparrow, \downarrow\}$ . The coefficients of the expansion  $\psi_{\mathbf{x}}$  are stored classically in a lookup table, and the energy of  $|\psi_{\mathcal{S}}\rangle$  is used as a best approximation of the ground state energy.

## II. METHODS

Here we significantly improve on the SOTA quantum results of [8] with a combination of algorithmic innovations and optimal management of quantum and classical resources, improved quantum hardware, and large-scale deployment of classical computational resources.

We implement a closed-loop optimization between quantum and classical processors. The differential evolution optimizer that we employ allows for an efficient orchestration of the quantum and classical computational resources used. This is achieved through a synchronization of the classical component

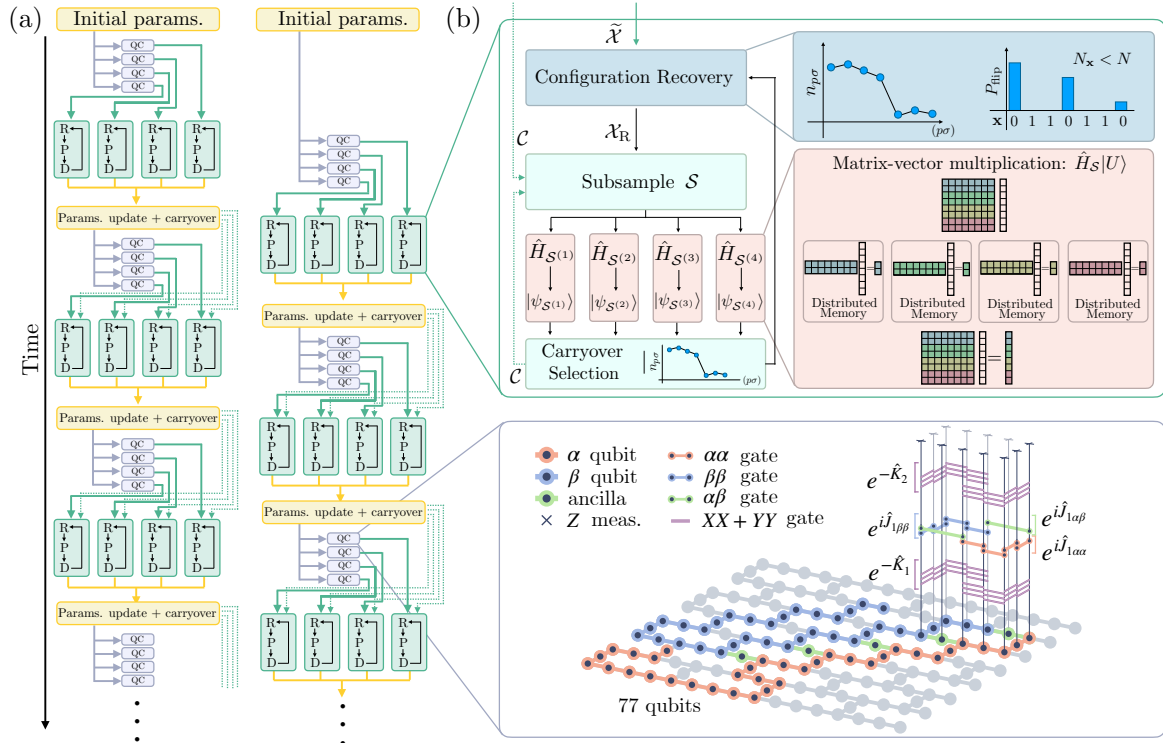


Fig. 2. **Depiction of the workflow used to optimize the quantum energies.** (a) High-level overview of the workflow to update the circuit parameters. The update of circuit parameters relies in a differential evolution algorithm with two populations of walkers, for a total of four walkers. The purple blocks labeled “QC” correspond to the execution of four quantum circuits associated to the walkers. The green blocks depict the SQD energy estimation for each of the walkers, which are executed in parallel in a HPC environment. Each of the green blocks executes three iterations of configuration recovery (R) with their corresponding projections (P) and diagonalizations (D). Purple arrows represent the loading of quantum circuits into the quantum processor. Green arrows represent the transfer of quantum measurement outcomes to the classical component. The dotted green lines denote the carryover of a fraction of configurations  $\mathbf{x}$  with the largest SQD wave function component. (b) **Top:** The classical component of the workflow includes the configuration recovery step and projection and diagonalization in the subspace  $\mathcal{S}^{(i)}$ . The diagonalization step, achieved with a Davidson eigensolver. The red inset shows the distribution of the matrix-vector multiplication required by the Davidson eigensolver, where the application of each row of the matrix is distributed across a number of nodes of the supercomputer *Fugaku*. The massive parallelization allows to perform diagonalizations on  $10^9$ -dimensional subspaces in under 10 mins. (b) **Bottom:** Depiction of the LUCJ quantum circuit considered in this study, executed on *Heron* quantum processors, and the corresponding circuit layout.

of one population with the quantum execution of the other population, thus optimizing the computational resource utilization.

In order to achieve optimal resource utilization, the quantum and classical components must have comparable execution times. Since the quantum runtime is fixed by the repetition rate of the circuits (10  $\mu$ s), we require an efficient and parallelization-compatible projection and diagonalization implementation, whose runtime at fixed subspace dimension can be controlled by the number of parallel processes that are being executed. We achieve this by implementing a highly parallel Davidson diagonalization algorithm compatible with distributed memory environments, using up to 152,064 nodes of the supercomputer *Fugaku*.

### Energy optimization workflow

Empirically, the CCSD parameter initialization of the quantum wavefunction  $|\Psi\rangle$  has been shown to yield accurate ground state energies in a variety of molecular systems in the 20-orbital scale, and only reasonable results once the 30-orbital mark is crossed [8], [23]–[26]. It is for these larger systems, like the iron-sulfur clusters that we consider here, where there

is room for significant improvements. As the performance of SQD depends on parameters of the quantum circuit to sample, we optimize the parameters by a classical iterative algorithm, which determines the overall workflow. This algorithm must scale with parallel computing and also make it possible to simultaneously execute quantum and classical workloads, in order to utilize both computing resources as much as possible. We choose Differential Evolution (DE) [27], which is an evolutionary algorithm, a type of metaheuristic well studied in the classical combinatorial optimization community. DE has a set of individuals, or walkers, who move around the parameter space to find better solutions.

DE is a gradient-free optimizer which in our case requires cost function evaluations for the energy of  $|\psi_{\mathcal{S}}\rangle$ , see Eq. (3). At each iteration, the cost function is evaluated on a number of walkers in parameter space. We consider two populations, each consisting of four walkers. Each walker within a population collaborates to iteratively update their respective LUCJ parameters. We execute the cost function evaluations for the walkers in parallel, orchestrating the quantum and classical components of each population optimization in order

to minimize idle times of computational resources, as depicted in Fig. 2. For each walker, DE updates parameters by mixing with other walkers to create a new trial solution. There are several strategies for the update, among which we found the ‘best/2/bin’ strategy from [28] to work fairly well for our closed-loop optimization.

Furthermore, we retain a certain number of determinants found in a given wavefunction  $|\psi_S\rangle$  over to the next iteration of the optimizer, ranked and selected according to the magnitude of their coefficients  $\psi_x$ , upon reaching a total fraction of the previous wavefunction, named *carryover ratio*. We find that this carryover helps with stability of the optimization with respect to statistical fluctuations and improving the energy.

The variational expressivity of the wavefunction ansatz in Eq. (3), can be improved by dressing  $|\psi_S\rangle$  with parametric orbital rotations  $|\psi_S(\kappa)\rangle = U(\kappa)|\psi_S\rangle$ , with  $U(\kappa) = \exp\left(\sum_{pr;\sigma} \kappa_{pr} \hat{a}_{pr}^\dagger \hat{a}_{r\sigma}\right)$ , a fermionic Gaussian unitary. The unitary character of  $U(\kappa)$  is ensured by enforcing that the real-valued parameters that define the rotation are anti-Hermitian  $\kappa_{pr} = -\kappa_{rp}$ . The action of the Gaussian unitary on a determinant  $|\mathbf{x}\rangle$  is to transform it to another determinant [29]:

$$U(\kappa)|\mathbf{x}\rangle = |\mathbf{x}'(\kappa)\rangle = \prod_{p\sigma} \left(\hat{a}_{p'\sigma}^\dagger\right)^{x'_{p\sigma}} |0\rangle, \hat{a}_{p'\sigma}^\dagger = \sum_r \Phi_{rp} \hat{a}_{r\sigma}^\dagger, \quad (4)$$

with  $\Phi$  the  $N_{\text{orb}} \times N_{\text{orb}}$  matrix exponential of  $\kappa$ . The goal of the energy optimization workflow is to minimize the Rayleigh quotient:

$$E(\mathcal{S}, \psi, \kappa) = \frac{\langle \psi_S(\kappa) | \hat{H} | \psi_S(\kappa) \rangle}{\langle \psi_S(\kappa) | \psi_S(\kappa) \rangle} \quad (5)$$

with respect to the subspace choice  $\mathcal{S}$ , the wave function coefficients  $\psi_x$  for all  $\mathbf{x}$  in  $\mathcal{S}$ , and the parameters of the orbital rotation  $\kappa$ . For fixed  $\mathcal{S}$  and  $\kappa$ , the optimal value for  $\psi_x$  is found in closed form by the projection and diagonalization of  $\hat{H}'(\kappa) = U^\dagger(\kappa) \hat{H} U(\kappa)$ . Since  $U(\kappa)$  is a fermionic Gaussian unitary, its action on the Hamiltonian can be computed efficiently:

$$\hat{H}'(\kappa) = \sum_{pr} h'_{pr} \hat{a}_{pr}^\dagger \hat{a}_{r\sigma} + \sum_{\substack{prqs \\ \sigma\tau}} \frac{(pr|qs)'}{2} \hat{a}_{pr}^\dagger \hat{a}_{q\tau}^\dagger \hat{a}_{s\tau} \hat{a}_{r\sigma} \quad (6)$$

with

$$h'_{pr} = \sum_{ab} h_{ab} \Phi_{ap} \Phi_{br}, \quad (pr|qs)' = \sum_{abcd} (ac|bd) \Phi_{ap} \Phi_{cr} \Phi_{bq} \Phi_{ds}. \quad (7)$$

For fixed  $\mathcal{S}$  and  $\psi_x$  values, the parameters in  $\kappa$  are optimized by the *Limited-memory Broyden–Fletcher–Goldfarb–Shanno* (L-BFGS) algorithm [30]–[33]. Gradients with respect to the matrix elements of  $\kappa$  are computed via automatic differentiation as proposed in Ref. [34]. The maximum number of iterations to update  $\kappa$  is set to 1000, but we often observe convergence early in the iterative process. We remark that several first- and second-order methods [21], [35]–[40] exist to optimize the values of  $\kappa$  in the context of multiconfigurational,

complete active-space, and selected configuration-interaction self-consistent field calculations.

### Execution of quantum circuits on Heron quantum processors

Current pre-fault-tolerant quantum devices are prone to errors, which skew the ideal, noiseless sampling to a noisy distribution. As a direct consequence samples from quantum hardware contain determinants that are faulty and irrelevant to the diagonalization. We deploy various strategies to reduce the impact of noise on the quantum sampled distribution:

- In superconducting qubit architectures no two qubits, and hence gates between qubits, are identical due to uncertainties in the fabrication process, RTE and cryo hardware, and calibration procedures. A major task in reducing noise is to select the best available qubits given circuit graph and the connectivity constraints of the device. We initialize LUCJ circuits that are hardware efficient for a heavy-hex connectivity and optimize the circuit layout using a modified version of mapomatic [41]. Mapomatic uses the VF2++ search order heuristic to find all isomorphic graphs within the device connectivity map. Additionally, we update the backend noise model by measuring the two-qubit gate errors using layer-fidelity [42] and the state preparation and measurement (SPAM) errors through measurements of the all-zero and the all-one states across the device. These characterizations are performed as closely in time as possible to the actual quantum sampling experiments to ensure that the error model faithfully represents the device behavior. Single-qubit gate error rates are obtained from the backend-reported values measured via randomized benchmarking. These error rates are then used to build a cost-function for each isomorphic circuit graph:

$$\mathcal{F}_{\text{circuit}} = 1 - \prod_{q,k} \varepsilon_{q,k}^{\text{SQ}} \prod_{g,k} \varepsilon_{g,k}^{\text{TQ}} \prod_m \varepsilon_m^{\text{SPAM}} \quad (8)$$

where  $\varepsilon_{q,k}^{\text{SQ}}$ ,  $\varepsilon_{g,k}^{\text{TQ}}$  and  $\varepsilon_m^{\text{SPAM}}$  are the single-qubit errors, two-qubit errors and SPAM errors, respectively, as they appear for each operation within a given circuit. Finally, we map the LUCJ circuit onto the qubit subset that yields the highest estimated circuit fidelity,  $\mathcal{F}_{\text{circuit}}$ . For every subsequent LUCJ parameter update (walker/population), the qubit layout remains fixed.

- LUCJ circuits have significant idle periods due to the structure of the circuits and the real-time duration of gates. Qubits, despite sitting idle, still interact with noise channels from the environment and the qubit state decoheres adding to the reduction in the sampling fidelity. Dynamical decoupling (DD) [43] is a technique that can be used to suppress decoherence and some of the qubit-qubit crosstalk errors. We use a sequence of four pulses  $XX\bar{X}\bar{X}$  (super CPMG) in all the idling gaps to improve the fidelity of our sampling experiments, where  $X$  and  $\bar{X}$  rotate by  $\pi$  and  $-\pi$  (respectively) about the x-axis of the Bloch sphere. Note that we implement  $\bar{X}$  using

the symmetric compilation in which an  $X$  gate is applied between virtual  $Z$  gates of  $\pm\pi$  rotations [44].

- We use the configuration recovery routine as detailed in [8], with two improvements. First, by providing the initial electron occupancy distribution externally with MP2 perturbation theory, we perform configuration recovery even in the first iteration. Additionally, while subsampling quantum measurements, we set the batch number to 1 and instead perform diagonalization using a larger number of configurations.

#### *Classical diagonalization on the supercomputer Fugaku*

The most computationally demanding part of the classical computation is the computation of the lowest eigenpair of the Hamiltonian projected over the space of selected configurations. To achieve this goal we employ the Davidson’s method [45], wherein a set of vectors is produced iteratively: starting from a set of  $d$  vectors  $S = \{\mathbf{v}_k\}_{k=0}^{d-1}$ , one computes the lowest-energy eigenpair  $(E, |\mathbf{w}\rangle)$  in  $S$  and the residual vector  $|\mathbf{r}\rangle = H|\mathbf{w}\rangle - E|\mathbf{w}\rangle$ , applies a preconditioner operator to  $|\mathbf{r}\rangle$  (here, the standard diagonal preconditioner [46]) and uses it to expand  $S$ . The algorithm starts from  $\mathbf{v}_0$  defined as the sampled configuration with lowest energy, and terminates when  $\|\mathbf{r}\|$  is below a threshold (signaling that an eigenpair has been found) or a maximum number of iterations is reached. The most expensive part of the Davidson’s method is the application of  $\hat{H}$  to a vector in  $S$ , which is a sparse matrix-vector multiplication.

To ensure this is done as efficiently as possible and in sync with the quantum runtime, we have developed a code designed for use on massively parallel supercomputing platforms. The primary advantages of leveraging such large-scale parallel architectures are twofold: the ability to handle wavefunction vectors whose size exceeds the memory capacity of a single node, and the significant acceleration of computation through parallelized Hamiltonian operations on these vectors. We distribute a single wavefunction vector across multiple nodes and parallelize the Hamiltonian application over this distributed data structure. In the following, we describe the strategy used for the parallelization of the Hamiltonian operation.

Let  $\psi_b = \langle \mathbf{x}_b | \psi \rangle$  denote the wavefunction vector after the Hamiltonian application, and  $\phi_b = \langle \mathbf{x}_b | \phi \rangle$  denote the vector before the operation. The Hamiltonian operation can be expressed as

$$\psi_b = \sum_{b'} H_{bb'} \phi_{b'} \quad (9)$$

where  $H_{bb'} = \langle \mathbf{x}_b | \hat{H} | \mathbf{x}_{b'} \rangle$ . Since  $\psi_b$  is distributed across multiple nodes, evaluating the above summation requires the transfer of  $\phi_{b'}$  data from other nodes. By replicating  $\psi_b$  across different node groups, the computation of the summation over  $b'$  can be parallelized across these replicated domains. On the other hand, since  $\psi_b$  is already distributed, the computation across row indices  $b$  can naturally be parallelized, and the replication of  $\psi_b$  across node groups further increases the granularity of this parallelism. We do all these operations at double precision.

To enable scalable Hamiltonian operations on distributed wavefunction vectors, we employ a flexible three-dimensional communicator split method. Our implementation introduces three types of MPI communicators: (1) a basis (column index) communicator that manages the distribution of the wavefunction vector across a basis space of size  $B$ ; (2) a task communicator that coordinates  $T$  replicated processes to parallelize the summation over  $b'$ ; and (3) a row communicator that handles further division into  $M$  partitions along the row index  $b$ .

Our computational basis is defined as the product space spanned by the  $\alpha$ - and  $\beta$ -determinant configurations. Accordingly, in the implementation, the basis communicator  $B$  has a size corresponding to the product of the two determinant spaces, expressed as  $B = B_\alpha B_\beta$ , where  $B_\alpha$  and  $B_\beta$  represent the communicators associated with the  $\alpha$  and  $\beta$  determinants, respectively, and we have the option of partitioning each of those communicators separately.

This configuration allows the wavefunction vector to be distributed over  $B \times T \times M$  nodes. Neglecting communication overhead, each decomposition direction supports near-ideal scaling with respect to the number of nodes. Among these, only the basis communicator requires non-blocking neighborwise data shifts, which introduce relatively low overhead, while the replicated communicators (task and row) require synchronizing blocking global reductions on very large vectors via `MPI_Allreduce`. To improve load balancing, we also introduce a new task assignment method that dynamically allocates work based on the size of each task, ensuring more uniform workload distribution across all processes. This is essential to get high efficiency when executing workflows at full scale on *Fugaku*.

For our diagonalization task, which is shared with configuration interaction (CI) chemistry workflows, it is essential to efficiently evaluate only the non-zero elements of the Hamiltonian, whose sparse structure is determined by the Slater–Condon rules. The wavefunction is represented by configurations encoding the occupation of alpha- and beta-spin orbitals, and matrix elements are non-zero only when the corresponding determinants differ by one- or two-particle excitations [47]. While conventional selected CI codes accelerate this process by precomputing and storing allowed excitations in intermediate arrays [19], [22], our implementation differs by a key distinction: the CI space is defined as the tensor product of subspaces spanned by the alpha and beta components. As a result, all entries in the intermediate arrays become non-zero, enabling a simpler and more efficient implementation.

Since our optimization algorithm proceeds sequentially with the aforementioned wavefunction carryover, the previously obtained ground state is used as the initial seed for the next iteration step. However, because the subspace changes between iterations, only the components in the common subspace are used to construct the initial vector.

## Classical resources

In total, the classical component of the workflow used up to 152,064 nodes of the supercomputer *Fugaku*, each with 48 cores and 32 GiB of memory for a total of 7.3 million cores and 4.6PiB of memory. The SQD energy of four walkers of the population is evaluated simultaneously, with each walker performing a diagonalization in a distinct subspace. These four diagonalizations run in parallel, each making use of up to 38,016 nodes, with the only exception of calculations where the subspace dimension was  $10^{10}$  calculation, where we used 2 walkers and 76,032 nodes per walker. In this demonstration, we considered two populations of up to four walkers each for DE.

To update the hyperparameters of DE, we used a mutation scaling factor  $F = 0.25$  and crossover ratio  $CR = 0.7$ . For population 0, for one walker, we used the same initial LUCJ parameters from CCSD as proposed in [8] (CCSD initial parameters) and, for the other walkers, we used initial parameters computed from double excitation amplitudes with random perturbations of  $\pm 0.05$  from the amplitudes obtained by CCSD. For population 1, we used the CCSD initial parameters for all walkers.

For each iteration of the configuration recovery step, we used the occupancies from the previous step except for the first iteration where we used the initial occupancies. The initial occupancies was computed classically, using one-particle reduced density matrix from the second-order Møller–Plesset perturbation theory (MP2).

We enabled wavefunction carryover across all diagonalizations for each walker, between optimization iterations and between multiple diagonalizations for configuration recovery within a single SQD energy evaluation. After obtaining a noisy set of configurations  $\tilde{\mathcal{X}}$  via quantum sampling, we applied configuration recovery and randomly selected  $D$  configurations from the recovered samples.

Given the spin-symmetric nature of the [2Fe-2S] and [4Fe-4S] clusters, its configuration exhibits mirror symmetry between the first and second halves corresponding to  $\alpha$ - and  $\beta$ -spin orbitals, respectively. Accordingly, after subsampling, the  $D$  configurations were split into two halves, and unique half-configurations were selected. If the Hartree–Fock configuration  $|1^{\otimes 270 \otimes 9}\rangle$  was not among the half-configurations, it was manually added. The resulting  $D_h$  half-configurations were used in diagonalization. These were then used to form a  $D_h \times D_h$  subspace,  $[\mathbf{x}_b^h]_{b=1}^{D_h} \otimes [\mathbf{x}_{b'}^h]_{b'=1}^{D_h}$ , on which we performed diagonalization to obtain the ground-state wavefunction  $\psi_{bb'}$ . From this wavefunction, we selected  $cD_h$  half-configurations  $[\tilde{\mathbf{x}}_b^h]_{b=1}^{cD_h}$  for carryover with carryover ratio  $c$ , prioritizing those with the largest weights  $r_b = \sum_{b'} |\psi_{bb'}|^2$ . In our numerical experiments,  $D_h$  was set to 3,200; 12,000; 32,000; and 10,000, corresponding to subspace dimensions of  $1.0 \times 10^7$ ,  $1.4 \times 10^8$ ,  $1.0 \times 10^9$ , and  $1.0 \times 10^{10}$  respectively.

For the diagonalization part, the maximum number of Davidson iterations was limited to 10 steps. The iteration process was terminated when either of the following two

TABLE I  
SUMMARY OF CLASSICAL COMPUTING SETUPS USED IN EACH SUBSPACE DIAGONALIZATION. THE TABLE LISTS THE SUBSPACE DIMENSION, COMMUNICATOR PARAMETERS ( $B$ ,  $T$ ,  $M$ ), NUMBER OF WALKERS, TOTAL NODES USED, AND THE ACTUAL TIME REQUIRED FOR THE DIAGONALIZATION.

| Subspace  | ( $B, T, M$ ) | Walkers | Total Nodes | Time [min]    |
|-----------|---------------|---------|-------------|---------------|
| $10^7$    | (16,2,3)      | 4       | 384         | $\sim 2$      |
| $10^8$    | (36,6,6)      | 4       | 5,184       | $\sim 5$      |
| $10^9$    | (64,8,32)     | 4       | 65,536      | $\sim 10$     |
| $10^{10}$ | (576,12,22)   | 4       | 152,064     | $\sim 15$ -20 |
| $10^{10}$ | (576,12,22)   | 2       | 76,032      | $\sim 15$ -20 |

conditions was met: (1) the desired accuracy was reached within 10 steps, or (2) the predefined maximum wall-clock time was exceeded. The convergence criterion was based on the norm of the residual vector, with the iteration stopping when it dropped below  $1.0 \times 10^{-3}$ . While this threshold typically corresponds to an energy error of about  $1.0 \times 10^{-6}$ , our actual target is a looser energy accuracy on the order of  $1.0 \times 10^{-3}$ . Therefore, this residual norm threshold is in fact rather stringent for our purposes. In practice, even when the iteration was terminated due to the wall-clock time limit, the energy error at termination was still around  $1.0 \times 10^{-6}$ , confirming that the solution had already reached sufficient accuracy. Furthermore, since the ground state obtained in the previous iteration is used as the initial guess for the subsequent iteration, convergence is usually achieved in fewer than 10 steps.

The performance of the diagonalization also depends on the configuration of MPI communicators. In the experiment involving a subspace of size  $10^9$  using 16,384 nodes, four MPI processes were assigned to each node, and the wavefunction vector was distributed over 64 nodes, so  $B=256$ . To accelerate the computation, parallelization over the row and column indices was employed, with the number of partitions set to  $T=8$  and  $M=32$ , respectively. Consequently, for each of the four walkers, the diagonalization was carried out using 16,384 MPI processes across 4,096 nodes. In the experiment with a larger subspace of size  $10^{10}$ , we used 152,064 nodes with two MPI processes per node and four walkers ( $B = 24 \times 24$ ,  $T = 12$ ,  $M = 11$ ). For comparison, a subset of the runs was performed using 76,032 nodes with two walkers under the same communicator configuration. A summary of the parameters used in all experiments, including those described above, is provided in Table I for reference. In our numerical experiments, we observed that increasing the subspace dimension by a factor of 10 tends to result in approximately a 20–25 $\times$  increase in classical computational cost. The actual computation times for each setup, reflecting this scaling behavior, are summarized in the table.

## Quantum resources

The quantum experiments were executed on the *ibm\_marrakesh* and *ibm\_kobe*, two *Heron* quantum processors with 156 superconducting qubits laid out in a heavy-hex connectivity. Fig. 3 provides a comprehensive overview of

the *ibm\_kobe* architecture. The top panel illustrates the device connectivity. From the available 156 qubits, we use 77 qubits of which  $2 \times 36 = 72$  are used to encode the LUCJ circuit and 5 qubits are used as auxiliary qubits to mediate the density-density interactions between the two spin populations. 72 primary qubits are marked in green and 5 ancillary qubits in orange, while the qubits that do not take part in the circuit are depicted in gray.

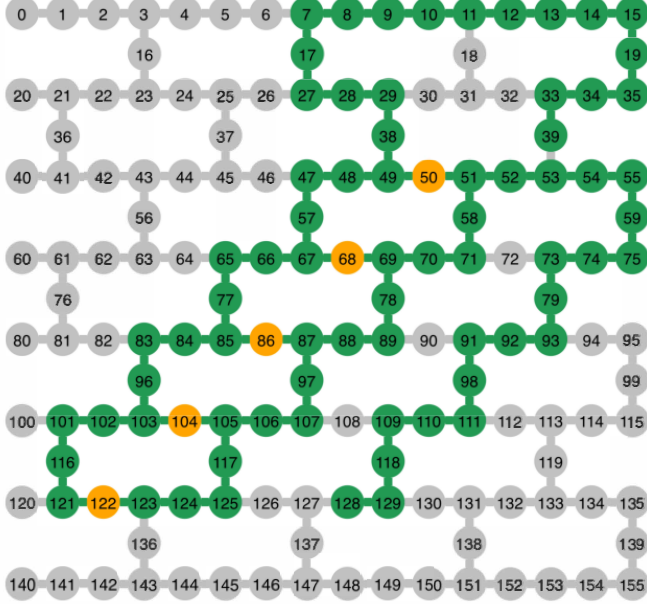


Fig. 3. **Schematic of the Kobe quantum processor.** Out of the full chip, 77 qubits are employed in the experiment, with 72 primary qubits highlighted in green and 5 ancillary qubits in orange; the remaining qubits are shown in gray.

### Implementation of the workflow

Our approach maximizes the utilization of both quantum and classical computing resources while maintaining a simple workflow by overlapping their execution as shown in Algorithm 1. The process involves two primary functions: `RUNQUANTUM` and `RUNCLASSICAL`.

`RUNQUANTUM` is invoked asynchronously for each population. It generates a LUCJ circuit based on parameters determined by a differential evolution algorithm and subsequently executes this circuit on a quantum processor.

Concurrently, while `RUNQUANTUM` is processing a population on a quantum processor, `RUNCLASSICAL` performs several classical tasks for a different population. These tasks include: recovering configurations from previous quantum computations, selecting relevant configurations, and performing a projected diagonalization. Crucially, `RUNCLASSICAL` also drives the differential evolution process by generating new parameters that will be used by `RUNQUANTUM` in its subsequent calls for the next iteration.

Our implementation (see our code [48]) employs a hybrid programming approach to optimize performance and

---

### Algorithm 1 Synchronous Quantum and Classical computations for Two Populations in Differential Evolution

---

```

1: async RUNQUANTUM(iteration=0, population=0)
2: async RUNQUANTUM(iteration=0, population=1)
3: for itr  $\leftarrow$  0 to MaxItr - 1 do
4:   wait RUNQUANTUM(iteration=itr, population=0)
5:   RUNCLASSICAL(iteration=itr, population=0)
6:   if itr + 1 < MaxItr then
7:     RUNQUANTUM(iteration=itr + 1, population=0)
8:   end if
9:   wait RUNQUANTUM(iteration=itr, population=1)
10:  RUNCLASSICAL(iteration=itr, population=1)
11:  if itr + 1 < MaxItr then
12:    RUNQUANTUM(iteration=itr + 1, population=1)
13:  end if
14: end for

```

---

resource utilization. The quantum computation component, `RUNQUANTUM`, is implemented in Python, while the classical processing component, `RUNCLASSICAL`, is implemented in C++ for performance-critical operations. Table II lists the exact software versions used in these experiments. Within `RUNQUANTUM` (Python), we utilize Qiskit 1.4.2 [49] for constructing LUCJ circuits, transpiling these circuits for execution on specific quantum hardware, and retrieving the resulting measurements after quantum computation. We employ `qiskit-ibm-runtime` 0.37.0 to send quantum circuits and retrieve the output measurements. Additionally, we use `ffsim` 0.0.52 for the initial construction of the LUCJ circuits based on the provided parameters. For the classical computation component, `RUNCLASSICAL` (C++), we have rewritten performance-critical parts of the configuration recovery and measurement subsampling functionality, originally found in the Qiskit add-on of `SQD` 0.10.0, in C++ to enhance execution speed. The entire differential evolution workflow, including parameter generation for the LUCJ circuits used by `RUNQUANTUM`, is also implemented in C++. The Python-based `RUNQUANTUM` is integrated into the C++-based differential evolution workflow by invoking it through system calls, enabling the asynchronous execution of quantum computations managed by the classical C++ framework. This architecture allows us to benefit from the ease of use and rich quantum circuit manipulation capabilities of Qiskit and `ffsim` in Python. At the same time, we maximize the performance of computationally intensive classical tasks like configuration recovery, subsampling, diagonalization, and the core differential evolution algorithm through our custom C++ implementation.

### Hybrid time performance measurement

To effectively evaluate the utilization of both quantum and classical resources, we track the timing of each step within `RUNQUANTUM` and `RUNCLASSICAL`. Leveraging the metadata provided in the results of our experiments with

Qiskit, we accurately determine the start and end times of quantum circuit execution on the hardware.

Within our analysis, we categorize the key steps of RUN-CLASSICAL as follows: **throw** encompasses the processes of constructing, optimizing (transpiling), and sending the quantum circuit to the quantum computer; **retrieve** involves serializing and receiving the results (measurements) from the quantum computer; **pre-processing** serializes the obtained bit strings, subsamples them, and recovers configurations for subsequent diagonalization; and **diagonalization** calculates the energy of the molecule based on the processed measurements. By timing these distinct stages in both the quantum (RUNQUANTUM) and classical (RUNCLASSICAL) computation, we gain insights into the efficiency of our hybrid approach and identify potential bottlenecks in resource utilization.

### III. RESULTS

Figure 4(a) and (b) present the results obtained by executing the closed-loop workflow for a series of iteration steps. Figure 4(a) corresponds to the [2Fe–2S] cluster, while Fig. 4(b) shows the results for the [4Fe–4S] cluster. For reference, Fig. 4(b) also includes the previous state-of-the-art (SOTA) results obtained with quantum data [8], and both panels (a) and (b) show the energies computed with Hartree–Fock (HF), configuration interaction with singles and doubles (CISD), and coupled cluster with singles and doubles (CCSD). In both systems, the walker energies in each population decrease monotonically with iterations, eventually achieving lower energies than those obtained by CCSD. This indicates that the quantum data can capture correlations beyond single and double excitations from the reference HF state.

For the [2Fe–2S] cluster, the lowest energy of  $-5048.932 E_h$  was obtained after 15 iterations using a subspace dimension of  $10^9$ , while for the [4Fe–4S] cluster, the lowest energy of  $-326.912 E_h$  was achieved after 10 iterations using a subspace dimension of  $10^{10}$ .

In the experiments on the [4Fe–4S] cluster using  $10^8$  and  $10^9$  determinants, we used four walkers for each of the two populations (0 and 1) and plotted the energy trajectories of each walker in population 1. For these runs, we repeated three configuration recovery and diagonalization steps within a single SQD iteration, and set the carryover ratio to 0.75. In contrast, for the  $10^{10}$  determinants run, each SQD iteration included only one configuration recovery and diagonalization step, and the carryover ratio was set to 0.5. Using *ibm\_marrakesh*, we acquired 750,000 measurement samples for each LUCJ circuit in the  $10^8$  and  $10^9$  runs. For the  $10^{10}$  run, we used *ibm\_kobe* and acquired 500,000 samples per circuit. The  $10^{10}$  determinants experiment began with a warm start using the LUCJ parameters and carryover bitstrings obtained at the final step of the  $10^9$  run. We also incorporated the orbital optimization procedure described in Section II. In the first five iterations, the algorithm was executed independently for each population. In the latter five iterations, we introduced a cooperative scheme: at the end of each

iteration, carryover bitstrings from one population were shared with the other. Within each population, the better-performing walker—selected from the two walkers—provided the energy and carryover bitstrings for the next iteration. This cooperative strategy effectively doubled the number of diagonalizations and orbital optimizations per iteration. As a result, a noticeable difference in energy descent behavior was observed between the first and second halves of the run, as shown in the blue plots in Fig. 4(b), which represent energies for population 1. Note that one walker per population was used during the first five iterations, whereas two walkers per population were used in the latter five iterations. The best point at  $10^{10}$  determinants improves on the best existing results [8] by  $278 mE_h$ .

For the experiment on the [2Fe–2S] cluster, we employed the same algorithm setup as in the latter five iterations of the [4Fe–4S] cluster experiment with  $10^{10}$  determinants, except that the number of measurement samples per LUCJ circuit was reduced to 200,000. In Fig. 4(a), we plotted the energy trajectories of two walkers from population 1. For this active space of [Fe2-S2], we also confirm that quantum data has a signal sufficient to obtain energies better than classical CCSD methods for a subspace of  $10^9$  determinants.

Figure 4 (c) shows, as a function of the elapsed wall time, the quantum and classical resource utilization by the walkers of both populations. As the iteration number increases, so does the classical wall time required to diagonalize the Hamiltonian projected into the subspaces proposed by the quantum circuits. Since the subspace dimension is kept fixed, this is a consequence of the increased number of off diagonal elements added to  $\hat{H}_S$ , which indirectly signals the increased quality of the explored subspaces, resulting in better energies. After the first 5 iterations the idle time of quantum and classical resources is minimized, showcasing a nearly optimal synchronization in the scheduling of both quantum and classical resources. This is further showcased in Figure 5. We report the utilization of both quantum and classical resources by walkers in a typical iteration step, as a function of the number of classical nodes used and the dimension of the subspace. Since the quantum runtime is mostly fixed - except one execution where we observed a delay in the cloud access of the device - by setting fixed number of quantum measurements per walker, independent of the subspace size, one can increase the number of classical nodes to keep the diagonalization time manageable even as the subspace dimension increases.

As mentioned in Section II, we introduce a flexible three-dimensional communicator decomposition and a load-balancing algorithm for task assignments over column indices in order to improve the parallel performance of the diagonalization step. To evaluate the impact of this strategy, we compared the wall-clock time required to reach a solution with and without the load-balancing algorithm under fixed node and MPI process configurations, as shown in Fig. 6 (a). Note that the total number of processes is fixed. Increasing  $T$  (the communicator size along the column index) sometimes causes performance degradation in the absence of load balancing. On the other hand, by introducing the load-balancing algorithm,

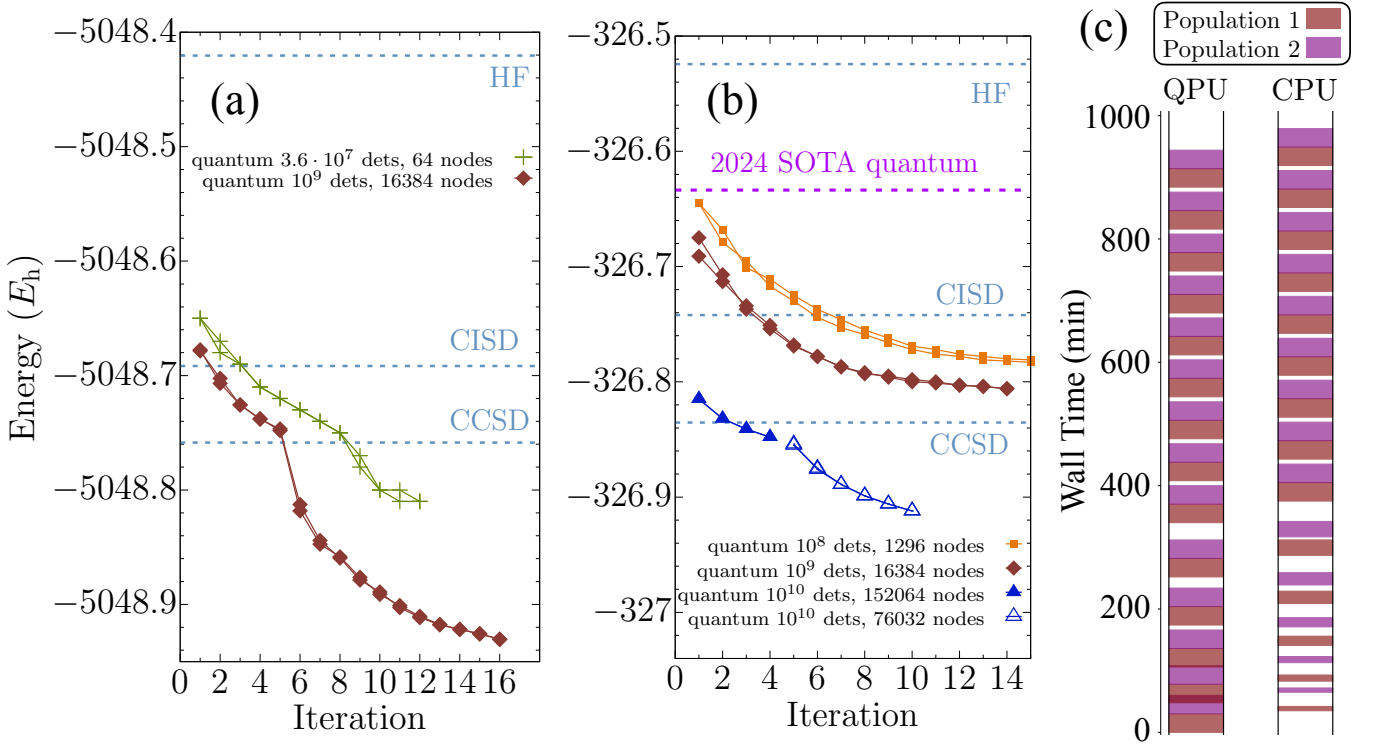


Fig. 4. (a) **Electronic structure calculations using Heron quantum processors and the supercomputer Fugaku.** Quantum energies (markers) as a function of the optimization step (a) for the [2Fe-2S] cluster, and (b) for the [4Fe-4S] cluster. The calculations for the [4Fe-4S] cluster with  $10^8$  and  $10^9$  determinants were performed on IBM Marrakesh, whereas the  $10^{10}$  case and all [2Fe-2S] calculations were carried out on IBM Kobe. For reference, we also show the previous state-of-the-art (SOTA) quantum energy [8], as well as the energies obtained from Hartree-Fock (HF), configuration interaction with singles and doubles (CISD), and coupled cluster with singles and doubles (CCSD). (c) Quantum (Heron processor) and classical (Fugaku supercomputer) resource usage as a function of elapsed wall time. The orchestration of quantum and classical resources aims at minimizing idle (white) time. The data is from the  $10^8$  determinants subspace from (b).

we observed a reduction in execution time of up to 16 processes for the task-parallel communicators. Based on these results, in our actual experiments we chose the diagonalization parameters such that the communicator size  $T$  was as close as possible to  $\sqrt{B}$ , where the best performance was observed. Although the available number of nodes sometimes limited us from realizing the exact configuration, we selected parameters that approximated this optimal condition as closely as possible.

To evaluate the parallel performance of the diagonalization, we also measure the wall-clock time required to obtain the solution for problems with subspace dimensions of  $10^8$  and  $10^9$ , constructed based on measurements obtained from the quantum processor. The results are shown in Fig. 6 (b). As the number of nodes increases, the time required to reach a solution decreases accordingly. For example, when using 512 nodes, the wall time is 65.899 seconds for the  $10^8$ -dimensional case, while it is 656.574 seconds for the  $10^9$ -dimensional case.

To assess parallelization efficiency, we show strong scaling performance, based on wall-clock time, in Fig. 6 (c). In practical computations, the performance achieved at larger node counts was approximately 60–70% of that at the baseline (smaller) node count. It is worth noting, however, that in quantum-classical hybrid computations, the quantum part is typically executed within a fixed time frame of several tens

of minutes (e.g., 30 minutes). Therefore, the diagonalization process must be designed to complete within the timeframe dictated by the quantum execution. This temporal constraint, arising from the need to synchronize with the quantum execution schedule, inherently limits the granularity of parallel task decomposition. Nevertheless, it reflects a deliberate design choice aimed at efficient coordination between quantum and classical resources. The observed scaling performance should thus be understood not as a limitation of the implementation itself, but rather as a consequence of balancing architectural synchronization with practical execution requirements.

Figure 7 shows the relationship between the energy variance, defined as  $\Delta E = \langle \psi | \hat{H}^2 | \psi \rangle - \langle \psi | \hat{H} | \psi \rangle^2$ , and the total energy obtained after the optimization loop for the [2Fe-2S] and [4Fe-4S] clusters. As the dimension of the subspace increases, both the magnitude of the energy variance decreases and the corresponding energy improves. Similar to the results in [8], we observe linear arrangements of points in the energy-variance plane, which allow for extrapolations to zero variance. The extrapolated value toward zero energy variance is approximately  $-327.118 \pm 0.02 E_h$  for [4Fe-4S], which deviates by about  $0.12 E_h$  from the DMRG reference energy of  $-327.239 E_h$ . Similarly, for the [2Fe-2S] cluster, the extrapolated energy is  $-5049.127 \pm 0.02 E_h$ , with a

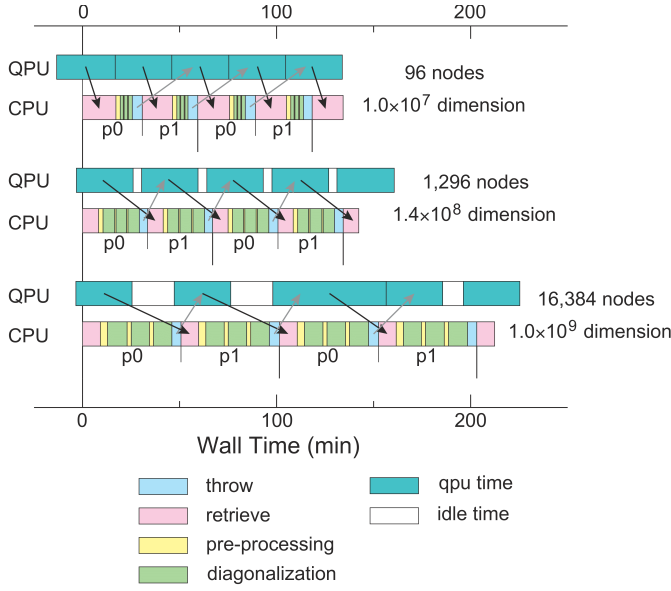


Fig. 5. **Wall time profile of QPU and CPU usage.** Computations for population 0 (p0) and population 1 (p1) are alternately executed on the classical processor. Here **throw** is the processing to prepare and send instructions to the quantum computer, **retrieve** receives output from the quantum computer; **pre-processing** prepares the list of configurations for diagonalization, and **diagonalization** performs projection and diagonalization to obtain molecular energies.

deviation of only about  $0.08 E_h$  from the DMRG reference value of  $-5049.217 E_h$ . These extrapolated energies, near to the DMRG values, indicate that the improved closed-loop workflow presented here, coupled with massive deployment of classical capabilities, can extract meaningful signal from current generation of quantum devices. This signal can capture a significant amount of quantum correlation in challenging chemistry problems.

#### IV. CONCLUSIONS

Quantum computers are poised to revolutionize the way we approach a few selected computational tasks: the simulation of quantum matter, specifically the computation of electronic structures for molecular systems, is one of them. In order to extract value from quantum computers, the quantum technology must interact with existing classical HPC facilities, within quantum-centric supercomputing environments [17]. For real-world computational tasks that can leverage quantum computing, the challenge ahead is to design workflows that exploit both quantum and classical processors in an optimal way.

The closed-loop workflow presented here achieves this by using a pre-fault-tolerant *Heron* quantum processor, coupled to up to 152,064 nodes of the supercomputer *Fugaku*. We have shown the capability of these two computing accelerators by addressing electronic structure problems which are beyond the reach of classical exact diagonalization methods: two 36-orbital active spaces for the iron sulfide clusters [2Fe-2S] and [4Fe-4S]. We have improved on the state-of-the-art calculations from quantum data for [4Fe-4S] [8] by  $278 mE_h$ ,

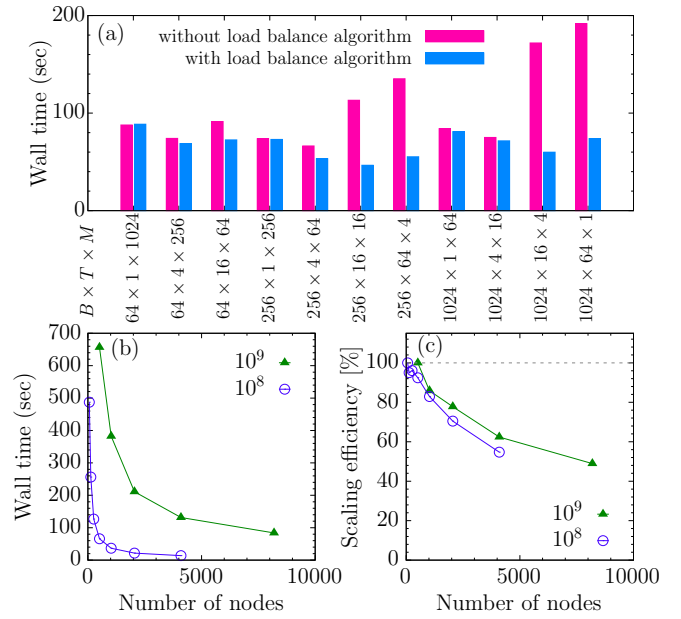


Fig. 6. **Classical parallelization performance.** (a) Comparison of wall time-to-solution with and without a load balancing algorithm applied to the task communicator under different node assignments. The benchmark is based on a  $10^{10}$ -dimensional subspace derived from measurements obtained from a quantum computer. The total number of nodes is fixed at 16,384 (4 times larger than at  $10^9$  for a single walker), with four MPI processes assigned per node. The communicator sizes  $B$ ,  $T$ , and  $M$  respectively correspond to the sizes of basis communicator, the task communicator for partitioning over column indices, and the row communicator for partitioning over row indices. We use this test to determine the parameters used in the experiments in Table I. (b) Wall-clock time to solution and (c) strong scaling performance of the diagonalization as functions of the number of nodes, for problems with subspace dimensions of  $1.0 \times 10^9$  and  $1.0 \times 10^8$ , constructed based on measurements generated by a quantum computer.

and have shown energy-variance extrapolated energies for both [2Fe-2S] and [4Fe-4S] with an energy error of about  $0.1 E_h$  from classical state-of-the-art DMRG calculations. Although showcased for the iron sulfides, the methods presented are widely applicable across electronic structure problems in quantum chemistry.

We have used an improved workflow, with a quantum-classical feedback loop and the carryover of configurations between optimization iterations. We have orchestrated quantum and classical resources to minimize idle times and maximize performance, adopting a strategy that varied the number of nodes of *Fugaku* used as a function of the quantum runtime, the classical subspace size and the quality of the solution. Although the diagonalization code [48] was specifically designed for *Fugaku*, we have found that it is well suited for current cloud computing environments and other high-performance computing clusters such as IBM Vela and BlueVela systems [50].

Our results show what can be achieved with quantum data deploying some of the largest HPC resources available today, using the current generation of pre-fault-tolerant *Heron* quantum computers. This is important both to understand how much value can be extracted from noisy quantum signals,

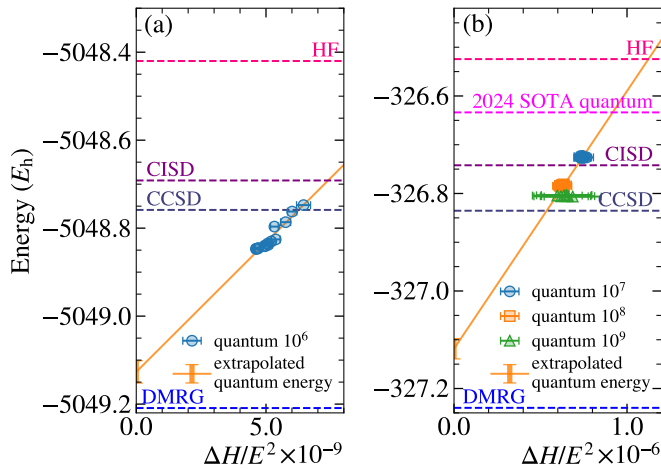


Fig. 7. **Extrapolated quantum energies** Quantum SQD energies and corresponding energy variances  $\Delta H$  obtained during the last iterations for each subspace dimension for (a) [2Fe-2S] and (b) [4Fe-4S]. The orange solid line represents a linear fit to the zero-variance quantum energy. The horizontal dashed lines denote, from top to bottom, the Hartree-Fock (HF), configuration interaction with single and double excitations (CISD), coupled-cluster with single and double excitations (CCSD), and density matrix renormalization group (DMRG) energies. In panel (b), the dashed line corresponding to the 2024 SOTA quantum [8] indicates the best computational quantum result achieved at that time.

and how efficiently quantum processors can interact with classical HPC infrastructures. Looking forward, it is important to observe that quality of the quantum signal improves exponentially with better quantum error rates, while the overall amount of classical resources decreases correspondingly, at fixed accuracy. This in turn implies that better solutions can be achieved using less classical resources than what we used in this work, offering a path forward to leverage the quantum technology to reach quantum advantage for realistic problems in computational chemistry.

#### ACKNOWLEDGMENTS

Part of this work is funded by project JPNP20017, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work used computational resources of Fugaku provided by RIKEN Center for Computational Science (Project ID: ra010014), and through the HPCI System Research Project (Project ID: hp240496, hp240552, hp240553, hp240554). This study is supported by JSPS KAKENHI Grants No. JP21H04446 and No. JP22K03479. We further acknowledge funding from JST COI-NEXT (Grant No. JPMJPF2221) and the Program for Promoting Research of the Supercomputer Fugaku (Grant No. MXP1020230411) from MEXT, Japan. Additionally, we appreciate the support provided by the UTokyo Quantum Initiative and the RIKEN TRIP initiative (RIKEN Quantum).

#### REFERENCES

[1] J. P. LeBlanc, A. E. Antipov, F. Becca, I. W. Bulik, G. K.-L. Chan, C.-M. Chung, Y. Deng, M. Ferrero, T. M. Henderson, C. A. Jiménez-Hoyos *et al.*, “Solutions of the two-dimensional Hubbard model: benchmarks and results from a wide range of numerical

algorithms,” vol. 5, no. 4, p. 041041, 2015. [Online]. Available: <https://journals.aps.org/prx/abstract/10.1103/PhysRevX.5.041041>

[2] M. Motta, D. M. Ceperley, G. K.-L. Chan, J. A. Gomez, E. Gull, S. Guo, C. A. Jiménez-Hoyos, T. N. Lan, J. Li, F. Ma *et al.*, “Towards the solution of the many-electron problem in real materials: equation of state of the hydrogen chain with state-of-the-art many-body methods,” vol. 7, no. 3, p. 031059, 2017. [Online]. Available: <https://journals.aps.org/prx/abstract/10.1103/PhysRevX.7.031059>

[3] A. Y. Kitaev, “Quantum measurements and the abelian stabilizer problem,” *arXiv preprint quant-ph/9511026*, 1995.

[4] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature communications*, vol. 5, no. 1, p. 4213, 2014.

[5] D. Wecker, M. B. Hastings, and M. Troyer, “Progress towards practical quantum variational algorithms,” vol. 92, no. 4, p. 042303, 2015. [Online]. Available: <https://journals.aps.org/prx/abstract/10.1103/PhysRevA.92.042303>

[6] L. Zhao, J. Goings, K. Shin, W. Kyoung, J. I. Fuks, J.-K. Kevin Rhee, Y. M. Rhee, K. Wright, J. Nguyen, J. Kim *et al.*, “Orbital-optimized pair-correlated electron simulations on trapped-ion quantum computers,” vol. 9, no. 1, p. 60, 2023. [Online]. Available: <https://www.nature.com/articles/s41534-023-00730-8>

[7] K. Kanno, M. Kohda, R. Imai, S. Koh, K. Mitarai, W. Mizukami, and Y. O. Nakagawa, “Quantum-selected configuration interaction: classical diagonalization of hamiltonians in subspaces selected by quantum computers,” *arXiv:2302.11320*, 2023. [Online]. Available: <https://arxiv.org/abs/2302.11320>

[8] J. Robledo-Moreno, M. Motta, H. Haas, A. Javadi-Abhari, P. Jurcevic, W. Kirby, S. Martiel, K. Sharma, S. Sharma, T. Shirakawa, I. Sitdikov, R.-Y. Sun, K. J. Sung, M. Takita, M. C. Tran, S. Yunoki, and A. Mezzacapo, “Chemistry beyond the scale of exact diagonalization on a quantum-centric supercomputer,” *Science Advances*, vol. 11, no. 25, p. eadu9991, 2025. [Online]. Available: <https://www.science.org/doi/abs/10.1126/sciadv.adu9991>

[9] J. Yu, J. R. Moreno, J. T. Iosue, L. Bertels, D. Claudino, B. Fuller, P. Groszkowski, T. S. Humble, P. Jurcevic, W. Kirby *et al.*, “Quantum-centric algorithm for sample-based krylov diagonalization,” *arXiv preprint arXiv:2501.09702*, 2025.

[10] H. Beinert, R. H. Holm, and E. Munck, “Iron-sulfur clusters: nature’s modular, multipurpose structures,” *Science*, vol. 277, no. 5326, pp. 653–659, 1997.

[11] D. C. Johnson, D. R. Dean, A. D. Smith, and M. K. Johnson, “Structure, function, and formation of biological iron-sulfur clusters,” *Annu. Rev. Biochem.*, vol. 74, no. 1, pp. 247–281, 2005.

[12] S. Sharma, K. Sivalingam, F. Neese, and G. K.-L. Chan, “Low-energy spectrum of iron-sulfur clusters directly from many-particle quantum mechanics,” *Nature chemistry*, vol. 6, no. 10, pp. 927–933, 2014.

[13] J. Miralles, O. Castell, R. Caballol, and J.-P. Malrieu, “Specific calculation of energy differences: Transition energies and bond energies,” *Chemical physics*, vol. 172, no. 1, pp. 33–43, 1993.

[14] G. K.-L. Chan and S. Sharma, “The density matrix renormalization group in quantum chemistry,” *Annual review of physical chemistry*, vol. 62, no. 1, pp. 465–481, 2011.

[15] H. Gao, S. Imamura, A. Kasagi, and E. Yoshida, “Distributed implementation of full configuration interaction for one trillion determinants,” *J. Chem. Theory Comput.*, 2024. [Online]. Available: <https://doi.org/10.1021/acs.jctc.3c01190>

[16] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, “Simulating physical phenomena by quantum networks,” *Physical Review A*, vol. 65, no. 4, p. 042323, 2002.

[17] Y. Alexeev, M. Amsler, P. Baity, M. A. Barroca, S. Bassini, T. Battelle, D. Camps, D. Casanova, F. T. Chong, C. Chung *et al.*, “Quantum-centric supercomputing for materials science: A perspective on challenges and future directions,” *arXiv preprint arXiv:2312.09733*, 2023.

[18] M. Motta, K. J. Sung, K. B. Whaley, M. Head-Gordon, and J. Shee, “Bridging physical intuition and hardware efficiency for correlated electronic states: the local unitary cluster jastrow ansatz for electronic structure,” vol. 14, pp. 11213–11227, 2023. [Online]. Available: <https://pubs.rsc.org/en/Content/ArticleLanding/2023/SC/D3SC02516K>

[19] A. A. Holmes, N. M. Tubman, and C. Umrigar, “Heat-bath configuration interaction: An efficient selected configuration interaction algorithm inspired by heat-bath sampling,” *Journal of chemical theory and computation*, vol. 12, no. 8, pp. 3674–3680, 2016.

TABLE II  
SOFTWARE VERSIONS USED IN THE WORKFLOW.

| Software            | Version |
|---------------------|---------|
| Python              | 3.11.6  |
| Qiskit              | 1.4.2   |
| qiskit-ibm-runtime  | 0.37.0  |
| ffsim               | 0.0.52  |
| Qiskit addon of SQD | 0.10.0  |

- [20] A. A. Holmes, H. J. Changlani, and C. Umrigar, "Efficient heat-bath sampling in fock space," *Journal of Chemical Theory and Computation*, vol. 12, no. 4, pp. 1561–1571, 2016.
- [21] J. E. Smith, B. Mussard, A. A. Holmes, and S. Sharma, "Cheap and near exact casscf with large active spaces," *Journal of chemical theory and computation*, vol. 13, no. 11, pp. 5468–5478, 2017.
- [22] S. Sharma, A. A. Holmes, G. Jeanmairet, A. Alavi, and C. J. Umrigar, "Semistochastic heat-bath configuration interaction method: Selected configuration interaction with semistochastic perturbation theory," *Journal of chemical theory and computation*, vol. 13, no. 4, pp. 1595–1604, 2017.
- [23] S. Barison, J. R. Moreno, and M. Motta, "Quantum-centric computation of molecular excited states with extended sample-based quantum diagonalization," *Quantum Science and Technology*, vol. 10, no. 2, p. 025034, 2025.
- [24] D. Kaliakin, A. Shajan, J. R. Moreno, Z. Li, A. Mitra, M. Motta, C. Johnson, A. A. Saki, S. Das, I. Sitdikov *et al.*, "Accurate quantum-centric simulations of supramolecular interactions," *arXiv preprint arXiv:2410.09209*, 2024.
- [25] I. Liepuoniute, K. D. Doney, J. Robledo-Moreno, J. A. Job, W. S. Friend, and G. O. Jones, "Quantum-centric study of methylene singlet and triplet states," *arXiv preprint arXiv:2411.04827*, 2024.
- [26] A. Shajan, D. Kaliakin, A. Mitra, J. R. Moreno, Z. Li, M. Motta, C. Johnson, A. A. Saki, S. Das, I. Sitdikov *et al.*, "Towards quantum-centric simulations of extended molecules: sample-based quantum diagonalization enhanced with density matrix embedding theory," *arXiv preprint arXiv:2411.09861*, 2024.
- [27] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341–359, 1997.
- [28] T. H. Bäck, A. V. Kononova, B. van Stein, H. Wang, K. A. Antonov, R. T. Kalkreuth, J. de Nobel, D. Vermetten, R. de Winter, and F. Ye, "Evolutionary algorithms for parameter optimization—thirty years later," *Evolutionary Computation*, vol. 31, no. 2, pp. 81–122, 2023.
- [29] D. Thouless, "Stability conditions and nuclear rotations in the hartree-fock theory," *Nuc. Phys.*, vol. 21, pp. 225–232, 1960. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0029558260900481>
- [30] C. G. BROYDEN, "The convergence of a class of double-rank minimization algorithms 1. general considerations," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 03 1970. [Online]. Available: <https://doi.org/10.1093/imamat/6.1.76>
- [31] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Math. Comp.*, vol. 24, pp. 23–26, 1970. [Online]. Available: <https://doi.org/10.1090/S0025-5718-1970-0258249-6>
- [32] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, pp. 317–322, 01 1970. [Online]. Available: <https://doi.org/10.1093/comjnl/13.3.317>
- [33] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Math. Comp.*, vol. 24, pp. 647–656, 1970. [Online]. Available: <https://doi.org/10.1090/S0025-5718-1970-0274029-X>
- [34] J. R. Moreno, J. Cohn, D. Sels, and M. Motta, "Enhancing the expressivity of variational neural, and hardware-efficient quantum states through orbital rotations," 2 2023. [Online]. Available: <http://arxiv.org/abs/2302.11588>
- [35] M. Head-Gordon and J. A. Pople, "Optimization of wave function and geometry in the finite basis hartree-fock method," *The Journal of Physical Chemistry*, vol. 92, no. 11, pp. 3063–3069, 1988.
- [36] B. O. Roos, P. R. Taylor, and P. E. Sigbahn, "A complete active space scf method (casscf) using a density matrix formulated super-ci approach," *Chemical Physics*, vol. 48, no. 2, pp. 157–173, 1980.
- [37] H.-J. Werner and P. J. Knowles, "A second order multiconfiguration scf procedure with optimum convergence," *The Journal of chemical physics*, vol. 82, no. 11, pp. 5053–5063, 1985.
- [38] D. A. Kreplin, P. J. Knowles, and H.-J. Werner, "Second-order mcsf optimization revisited. i. improved algorithms for fast and robust second-order casscf convergence," *The Journal of chemical physics*, vol. 150, no. 19, 2019.
- [39] —, "Mcsf optimization revisited. ii. combined first-and second-order orbital optimization for large molecules," *The Journal of Chemical Physics*, vol. 152, no. 7, 2020.
- [40] Q. Sun, J. Yang, and G. K.-L. Chan, "A general second order complete active space self-consistent-field solver for large-scale systems," *Chemical Physics Letters*, vol. 683, pp. 291–299, 2017.
- [41] P. D. Nation and M. Treinish, "Suppressing quantum circuit errors due to system variability," *PRX Quantum*, vol. 4, p. 010327, Mar 2023. [Online]. Available: <https://link.aps.org/doi/10.1103/PRXQuantum.4.010327>
- [42] D. C. McKay, I. Hincks, E. J. Pritchett, M. Carroll, L. C. G. Govia, and S. T. Merkel, "Benchmarking quantum processor performance at scale," 2023. [Online]. Available: <https://arxiv.org/abs/2311.05933>
- [43] L. Viola, E. Knill, and S. Lloyd, "Dynamical decoupling of open quantum systems," *Phys. Rev. Lett.*, vol. 82, pp. 2417–2421, Mar 1999. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.82.2417>
- [44] A. Vezvae, V. Tripathi, D. Kowsari, E. Levenson-Falk, and D. A. Lidar, "Virtual z gates and symmetric gate compilation," 2025. [Online]. Available: <https://arxiv.org/abs/2407.14782>
- [45] E. R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *Journal of Computational Physics*, vol. 17, no. 1, pp. 87–94, 1975. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0021999175900650>
- [46] R. B. Morgan and D. S. Scott, "Generalizations of davidson's method for computing eigenvalues of sparse symmetric matrices," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 817–825, 1986.
- [47] P. J. Knowles and N. C. Handy, "A new determinant-based full configuration interaction method," *Chemical physics letters*, vol. 111, no. 4–5, pp. 315–321, 1984.
- [48] T. Shirakawa, "Libary for selected basis diagonalization," <https://github.com/r-ccs-cms/sbd>, 2025, accessed: 2025-10-06.
- [49] G. Aleksandrowicz *et al.*, "Qiskit: An open-source framework for quantum computing," 2019.
- [50] G. Gershon, S. Seelam, B. Belgodere *et al.*, "The infrastructure powering ibm's gen ai model development," 2025. [Online]. Available: <https://arxiv.org/abs/2407.05467>