SANDIA REPORT

SAND2020-12140
Printed September, 2020

Sandia National Laboratories

Noisy Intermediate-Scale Quantum Applications on a Pathfinder System

Jesse J. Lutz, Benjamin C.A. Morrison, Andrew D. Baczewski, Tzvetan S. Metodi

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

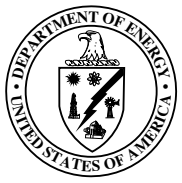
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods



ABSTRACT

Work performed under this one-year LDRD was concerned with estimating resource requirements for small quantum test beds that are expected to be available in the near future. This work represents a preliminary demonstration of our ability to leverage quantum hardware for solving small quantum simulation problems in areas of interest to the DOE. The algorithms enabling such studies are hybrid quantum-classical variational algorithms, in particular the widely-used variational quantum eigensolver (VQE). Employing this hybrid algorithm, in which the quantum computer complements the classical one, we implemented an end-to-end application-level toolchain that allows the user to specify a molecule of interest and compute the ground state energy using the VQE approach. We found significant limitations attributable to the classical portion of the hybrid system, including a greater than greater-than-quartic power scaling of the classical memory requirements with the system size. Current VQE approaches would require an exascale machine for solving any molecule with size greater than 150 nuclei. Our findings include several improvements that we implemented into the VQE toolchain, including a new classical optimizer that is decades old but hadn't been considered before in the VQE ecosystem. Our findings suggest limitations to variational hybrid approaches to simulation that further motivate the need for a gate-based fault-tolerant quantum processor that can implement larger problems using the fully digital quantum phase estimation algorithm.

Disclaimer: This report represents preliminary exploration on the limits of specific approaches to VQE. It does not represent broadly publishable, peer-reviewed, conclusive work. Mitigation approaches to both address some of the bottlenecks found in the hybrid quantum-classical VQE workflow and potential inconsistencies in our own analysis are being explored as follow-on efforts.

ACKNOWLEDGMENT

We gratefully acknowledge useful feedback from John Aidun, Gil Herrera, Andrew Landahl, and Rick Muller. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for DOE's National Nuclear Security Administration under contract DE-NA0003525. This report describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the report do not necessarily represent the views of the U.S. Department of Energy or the United States government.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Table 0-1. A list of acronyms used throughout the report

Abbreviation	Definition
AMO	Atomic, Molecular, and Optical
AO	Atomic Orbital
BCC	Brueckner Coupled Cluster
CBS	Complete Basis Set
CC	Coupled Cluster
CI	Configuraiton Interaction
COBYLA	Constrained Optimization by Linear Approximation
CSF	Configuration State Function
ERI	Electron-Repulsion Integral
FCI	Full Configuration Interaction
GPU	Graphical Processing Unit
GTO	Gaussian-Type Orbital
HF	Hartree-Fock
LDRD	Labratory Directed Research and Development
MBS	Minimum Basis Set
MPI	Message Passing Interface
NEWUOA	New Unconstrained Optimization Algorithm
NISQ	Near-term Intermediate Scale Quantum
oo	Orbital Optimized
PCC	Projection Coupled-Cluster
QNN	Quantum Neural Network
ORNL	Oak Ridge National Laboratory
PES	Potential Energy Surface
SCF	Self-Consistent Field
SPSA	Simultaneous perturbation stochastic approximation
STO	Slater-Type Orbital
VQE	Variational Quantum Algorithm
UCC	Unitary Coupled Cluster
UpCC	Unitary pair Coupled-Cluster

1. INTRODUCTION

Quantum computation is among the most highly anticipated technologies currently being pursued by the Department of Energy. The ultimate promise of quantum computers is the realization of an exponential speedup over classical computers, and, among the many proposed applications for quantum computing, high-accuracy simulation of quantum systems holds special potential for realizing this promise. The underlying reasoning is easily conveyed. Quantum simulation of atoms, molecules, and materials requires solution of the Schrödinger equation, which is infamously difficult due to the quantum nature of electron-electron interactions, or correlations. Algorithms for the description of electron correlation can be conveniently divided into two categories: there are exact methods, requiring classical computing resources that scale exponentially with the system size, and there are approximate methods, characterized by resource requirements that scale polynomially with system size. Exact algorithms provide solutions with arbitrarily high accuracy, but their steep classical resource requirements make them intractable for the vast majority of systems. A quantum computer is expected to enable near exact simulations at a cost which scales polynomially with the system size. At present, demonstration of a quantum simulation capability that outperforms existing classical computers remains an outstanding challenge.

Physical qubit counts in integrated systems[1] have risen to the point that it would take a sizeable fraction of the secondary-storage space of the world's largest supercomputer to store their state[2]. Assuming that qubit counts continue to rise while error rates continue to drop, due to some combination of physical hardware development and the eventual realization of quantum error correction, we will eventually arrive at a point where the limits to quantum simulation will be qualitatively very different. In fact, it is plausible that there will be a point in the development of quantum computing technologies for which the classical overhead associated with specifying problem instances becomes the limiting factor. There is a need to articulate what that might look like, specifically in the context of quantum simulations of electronic structure.

The variational quantum eigensolver (VQE), a hybrid quantum-classical algorithm for approximating the ground state energies and wave functions of Hamiltonians, has been identified as the leading computational approach for execution on noisy intermediate-scale quantum (NISQ) hardware*. While it is widely assumed that the VQE approach will allow useful problems to be implemented on NISQ-sized testbeds, it is unclear to what extent that is true when one considers both the qubit count needs and the classical resources required to support the classical calculations in the hybrid VQE approach. Various choices of algorithms and implementation methods for the interdependent functional elements (classical or quantum) have significant impact

*Here, 'noisy' suggests that control/environmental noise limits the achievable error rates, and 'intermediate-scale' refers to devices with a number of qubits ranging from 50 to a few hundred[3].

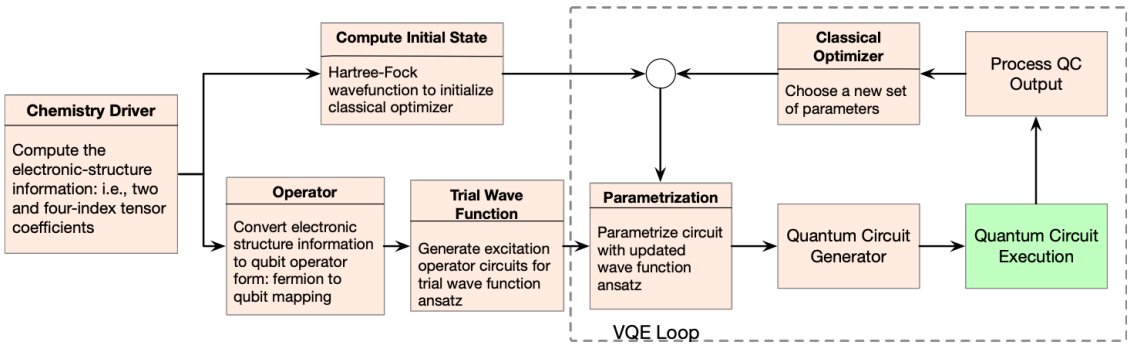


Figure 1-1. The basic workflow for the VQE software stack used in molecular electronic structure calculations. Preparatory classical steps are shown on the left and the hybrid quantum/classical VQE loop is shown within the dashed box on the right.

on the accuracy of the results, the capabilities required of different quantum computer test beds, and more generally, the limits of near-term quantum computers for applications beyond characterizing and improving quantum devices.

This report details our investigations of the classical and quantum resource requirements associated with employing VQE to generate ground-state energies of common small systems. The application focus was chosen to be molecular electronic structure, a popular and impactful field spanning a range of system sizes from analytically solvable diatomics (e.g., H_2^+, HeH^2+, etc.) to macromolecules constituting the building-blocks of life (e.g., lipids, proteins, and nucleic acids), and beyond. Figure 1-1 is an illustrative example of the interplay between quantum and classical elements for VQE-based electronic structure algorithms, where detailed descriptions of each step are provided in Sections 3.1–3.5. While outside the scope of this work, it is worth mentioning that similar VQE diagrams can be drawn for other quantum simulation applications; to name a few examples, within the last year VQE algorithms have been developed for molecular vibrational structure [4], solid-state electronic band structure [5], and many-particle nuclear structure [6].

Having introduced VQE as an alternative to fully classical electronic structure calculations, it is informative to compare the two by generating back-of-the-envelope estimates for the associated resource requirements. For this task around 20 small-to-medium-sized molecular species were chosen, forming a representative sample of molecules of interest to the DOE[†]. System sizes were considered at three basis set levels, [i.e., three discretization levels of the electronic structure Hamiltonian (see Sections 2.1 and 2.2 for further details)]. The largest basis set level was chosen to be one that facilitates extrapolation to the complete basis set (CBS) limit, where the basis-set error becomes acceptably small for quantitative applications of interest to chemists. Two smaller basis sets were also considered (6-31G$**$ and cc-pVTZ), but these can be recommended only for qualitative investigations. Basis sets at the smallest extreme are called minimum basis sets (MBS), but basis sets of this size usually fail to provide even qualitatively correct results.

Figure 1-2 collects quantum and classical resource requirements calculated by applying

[†]See Appendix A for the full list including the underlying data needed to produce Figure 1-2.

well-known scaling laws to the set of molecular systems described previously. Data is reported relative to the resource requirements for a MBS, where, for example, $1.0x10^2$ on the classical axis indicates that the simulation requires 100 times more RAM than the MBS case. VQE typically employs the unitary coupled-cluster theory with singles and doubles approach (see Section 2.4), requiring a gate count scaling as N^4, with N the system size. For the classical analogy, projection coupled-cluster with singles and doubles (see Section 2.3), the required RAM scales as N^6. Figure 1-2 is a composite of three graphs, with each one corresponding to a different basis-set level. The horizontal and vertical limits are aligned to produce a continuous effect.

At present, VQE calculations are feasible almost exclusively at the MBS level, so the quantum gate count values in Fig. 1-2 provide rough estimations of the advances in quantum hardware that are required to perform useful VQE simulations. Focusing on the relative costs for generating approximate CBS-limit energies on the right-hand side (RHS), gate-count requirements are between one- and ten-thousand times the capabilities of quantum hardware available today. Meanwhile, turning to classical limitations, RAM requirements on the RHS are shown to grow by a factor of about six thousand. Taking, for simplicity, the conservative estimate that performing a MBS calculation on taxol requires 1 gigabyte of RAM, then the desired large-basis calculation would require between 1 and 10 terabytes of RAM. Either limit far exceeds the RAM currently available on an individual node of ORNL's Summit supercomputer. On the quantum side, qubit counts are currently limited to around 50, both physically (on state-of-the-art hardware) and virtually (via qubit simulation on classical hardware), and this constraint guided our choice of chemical systems. We find that qubit counts exceed this limit for all molecules that we considered.

Since most systems in Figure 1-2 are too large for inclusion in a VQE scaling study, we chose to consider instead liquid deuterium, simulating explicit atomic models of increasing size. This is an exemplary problem for numerous reasons. Due to its minimal nuclear charge, freezing (or pseudization) of chemically inert orbitals is irrelevant. The number of electrons scales identically with the number of atoms. Further, recent work on highly accurate classical many-body calculations on this system nicely articulate the need for scalable many-body electronic structure simulation [7], while also outlining the relevance of liquid deuterium to DOE applications.

The first goal of the present work is to identify the resource requirements and the bottlenecks for both the quantum and the classical components across the entire VQE application stack. Rate-limiting bottlenecks could come in the form of any of the following metrics: wall times, RAM requirements, disk requirements, qubit count, qubit depth, or number of circuit repetitions. After careful examination of each element, opportunities for software engineering were identified and implemented to improve efficiency of the stack overall. While classical quantum simulation algorithms and implementations have been fine-tuned for generations, the VQE approach is only a few years old and is therefore fertile ground for development and optimization.

Presently the most evident factor limiting the realization of a quantum simulation capability that practically outperforms existing classical computers is the *error rates* that are achievable in those systems. Such errors can lead to incorrect measurements, and this may prevent the classical optimization step from converging to a solution. We also addressed the need to study the complexity of implementing a unified application framework that allows us to generate quantum circuits for a wide array of applications that can be compiled and run on different test beds and/or

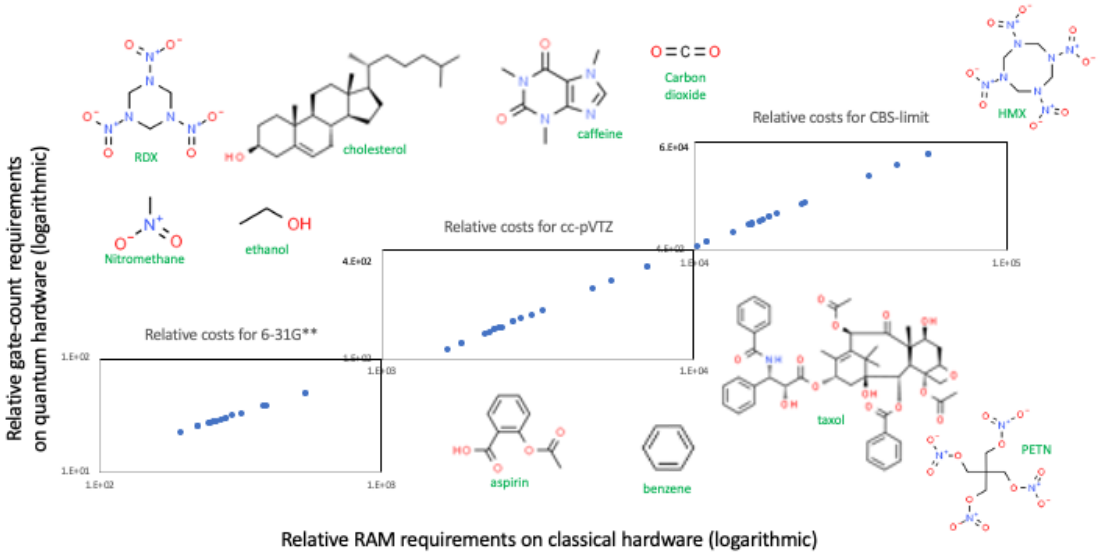


Figure 1-2. Resource requirements for the hybrid VQE and fully classical approaches for computing the correlation energy. All values are reported relative to a minimum basis set calculation for 20 chemical species of relevance to the Department of Energy. Gate-count (hybrid) and RAM (classical) requirements were calculated from known scaling laws associated with the lowest-order unitary (hybrid) and projective (classical) coupled-cluster methods, respectively.

simulators. In doing so, we combined a variety of existing tools to build a toolchain that serves as an end-to-end "flight simulator" for VQE applications implemented on quantum hardware. Results illustrating the use of this toolchain will be presented in follow-on work.

The report is structured as follows. In Section 2 the VQE protocol is described in detail, including sufficient background information to enable a technical interpretation of the following sections. Section 3 addresses our first goal, the identification of resource bottlenecks for the quantum and classical components of VQE, by singling out various components of the VQE workflow and performing detailed resource scaling studies. Finally, in Section 4 we recapitulate our findings and assess the long-term prospects of VQE for quantum simulation.

2. VARIATIONAL QUANTUM EIGENSOLVER

The VQE procedure coordinates execution between a quantum computer, which is periodically called upon to measure the expectation value of the Hamiltonian $\hat{H}$ for some parameterized wavefunction $|\Psi(\vec{\theta})\rangle$, and a classical computer, which analyzes measurement outcomes and suggests new energy-minimizing values for the $\vec{\theta}$ parameters [8, 9]. Here $|\Psi(\theta)\rangle$ is always normalized and $\{\theta\}$ is a set of real valued parameters ranging from 0 to 2π which we arrange into a vector, $\vec{\theta}$. The goal is the determination of the eigenvectors and eigenvalues. For the vast majority of chemical applications, only the lowest-energy eigenvectors and eigenvalues are of interest. This is because within the temperature range that supports chemical bonding, statistical sampling is dominated by low-lying states.

Given some parameterized form of the electronic wave function, $|\Psi(\vec{\theta})\rangle$, a quantum computer is first used to prepare the ansatz state. This step involves generation of an initial guess for the state, which is supplied by a classical quantum chemistry code known as the driver (Section 3.1). Next, fermionic operators associated with the ansatz state are encoded into qubits (Section 3.2) and an excitation operator circuit is created to prepare the trial wave function state (Section 3.3. The expectation value of the energy of that state, $E(\vec{\theta})$, is obtained by measuring the expectation values of each term of the Hamiltonian individually. These measurements are repeated a large number of times (called shots) to give high statistical confidence in the expected energy. The measured energy is fed into a classical "black-box" optimization algorithm, which chooses a new vector of parameters, $\vec{\theta}'$ (Section 3.5). These parameters are used to prepare a different candidate state, $|\Psi(\vec{\theta}')\rangle$ (Section 3.4). The process repeats until the optimizer converges to a state with minimum energy.

Appendix B provides a detailed description of all software used to perform VQE calculations and provides a link to a static code repository that includes all the compatible software versions needed to reproduce our results. To succinctly reiterate, the steps required for the VQE algorithm are as follows:

1. *State preparation*: The state $|\Psi(\vec{\theta})\rangle$ or $\rho(\vec{\theta})$ is prepared on the quantum hardware in terms of the angles $\vec{\theta}$, which can be any adjustable experimental or gate parameter.

2. *Measurement*: The expectation value of a Hamiltonian operator $\hat{H}$ is measured on the quantum hardware as $\langle\hat{H}\rangle(\vec{\theta}) = \langle\Psi(\vec{\theta})|\hat{H}|\Psi(\vec{\theta})\rangle$.

3. *Classical optimization*: A classical nonlinear optimizer is invoked to determine a new set of values $\vec{\theta}$ that decrease $\langle\hat{H}\rangle(\vec{\theta})$.

4. *Iterate until convergence*: Steps (1)–(3) are iterated until the procedure converges to a minimum value of $\langle \hat{H} \rangle (\vec{\theta})$, at which time the parameters $\vec{\theta}$ define the desired stationary state of $|\Psi(\vec{\theta})\rangle$.

The primary advantage of VQE in the NISQ era is its utilization of Hamiltonian averaging, meaning that it compensates for short coherence times by repeatedly sampling the quantum state to determine the mean value of an observable.

2.1. ELECTRONIC STRUCTURE HAMILTONIAN

2.1.1. First quantization

The objective of the electronic structure problem is to estimate the energy of electrons interacting in a fixed nuclear potential. The familiar non-relativistic, clamped-nucleus, electronic Hamiltonian describes the motion of N electrons with positions $\mathbf{r}_i$ in the Coulomb field of M nuclear point charges with atomic numbers Z_A and positions $\mathbf{R}_A$ as [10]

$$H = -\sum_{i}^{N} \frac{\nabla_{\mathbf{r}_i}^2}{2} - \sum_{i}^{N} \sum_{A}^{M} \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|} + \sum_{i>j}^{N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \tag{2.1}$$

where atomic units are assumed. Treatment of nuclei as stationary point-charges is known as the Born-Oppenheimer approximation, and it is common practice to vary the nuclear positions parametrically to construct Born-Oppenheimer potential energy surfaces (PESs).

2.1.2. Second quantization

For a fixed set of orthonormal single-particle spin orbitals, $\{\phi_i(\mathbf{r})\}_{i=1}^{N}$, the generic second-quantized form of the electronic structure Hamiltonian is

$$H = \sum_{pq} f_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} v_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \tag{2.2}$$

where $a_p^\dagger$ (a_q) are fermionic creation (annihilation) operators and the one-electron and two-electron integrals are defined as

$$f_{pq} = \int d\mathbf{r}\ \phi_p^*(\mathbf{r}) \left(-\frac{\nabla_{\mathbf{r}}^2}{2} - \sum_{A}^{M} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \right) \phi_q(\mathbf{r}) \tag{2.3a}$$

$$v_{pqrs} = \int d\mathbf{r} \int d\mathbf{r}'\ \frac{\phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) \phi_r^*(\mathbf{r}') \phi_s(\mathbf{r}'))}{|\mathbf{r} - \mathbf{r}'|}, \tag{2.3b}$$

At present the spin orbitals are extracted from the self-consistent solution to a mean-field (e.g., Hartree-Fock) theory executed on a classical computer and represented in terms of a set of

primitive basis functions (e.g., linear combinations of Gaussians or plane waves). In the future a set of mean-field spin orbitals may be efficiently obtained on quantum hardware, and this capability was recently demonstrated by Arute et al. [11], but, at least for now, it is standard to generate spin orbitals using existing classical chemistry drivers.

2.1.3. Potential energy surfaces

The scientific value of accurate PESs cannot be overstated. They are used by nuclear engineers to perform equilibrium molecular dynamics calculations that predict thermal conductivity of actinide oxides [12]. AMO physicists use them to perform fully quantum (coupled-channel) dynamics calculations to locate Feshbach resonances in ultracold atomic and molecular collisions [13], In chemistry and biochemistry, almost all phenomena are fundamentally governed by PESs. The corresponding free energy landscapes facilitate simulation of molecular solvation, molecular association, macromolecular stability, and enzyme catalysis [14], among countless other applications. It is worth stressing that high accuracy is paramount in all of these fields. As an example, free energy landscapes must be extremely accurate as chemical rates are exponentially sensitive to changes in free energy. In many applications, near-exact PESs are requisite for obtaining results having any value at all.

2.2. SINGLE-PARTICLE BASIS SETS

An ongoing research thrust in theoretical chemistry is the development of finite sets of basis functions for use in electronic structure calculations. For molecular calculations, the most popular type of basis set approximates nucleus-centered atomic orbitals (AOs). While the results found throughout this report are primarily concerned with accurately representing the properties of an extended system, which are typically described using plane-wave basis functions, we choose to focus on nucleus-centered Gaussian basis sets to be consistent with classical many-body benchmark calculations of this same system [7]. The purpose of this section is to provide background information essential for interpreting the VQE scaling studies.

2.2.1. Gaussian-type orbitals

Slater-type orbitals (STO) are the best physically-motivated basis set for nuclear-centered basis sets, primarily due to a qualitatively correct description of both the short-range nuclear cusp and the long-range exponential decay, but unfortunately no analytic solution exists for evaluation of their general four-index integral given by Equation 2.3. The expense associated with numerical integral evaluation forces use of an alternative function having analytic integrals. This is most easily achieved by modifying the e^{-r} form of STOs to the e^{-r^2} form characterizing a Gaussian

function. Gaussian-type orbitals (GTOs) take the following form,

$$\phi(x,y,z;\alpha,i,j,k) = \left(\frac{2\alpha}{\pi}\right)^{3/4} \left[\frac{(8\alpha)^{i+j+k} i!\,j!\,k!}{(2i)!(2j)!(2k)!}\right]^{1/2} \times x^i y^j z^k e^{-\alpha(x^2+y^2+z^2)} \tag{2.4}$$

where α controls the width of the GTO and i, j, k are non-negative integers dictating the Cartesian nature. Specifically for hydrogen, one is concerned with the case when all three of these indices are zero, which produces an s-type orbital with spherical symmetry. Gaussian basis set development has been undertaken for decades, and by now there are 500+ varieties available [15].

Gaussian functions have several undesirable features which are mitigated by forming linear combinations, or *contractions*, of uncontracted Gaussians, or *primitives*. A contraction of Gaussian functions takes the form

$$\phi(x,y,z;\alpha,i,j,k) = \sum_{a=1}^{M} c_a \phi(x,y,z;\alpha_a,i,j,k) \tag{2.5}$$

where M is the number of Gaussians used in the linear combination, and the coefficients c_a are chosen to optimally reproduce the shape of an STO while also ensuring normalization. In basis sets of contracted GTOs, each function is defined by a set of contraction coefficients and exponents assigned to each of its primitives.

2.2.2. Minimal basis sets: STO-3G and STO-6G

The STO-*M*G basis sets are the minimal representation, in terms of the number of Gaussians, which can capture the most basic physical features of AOs (e.g., their exponential tail, nodal structure, etc.). STO-3G and STO-6G basis sets mimic STOs by fitting contraction coefficients and exponents of three and six contracted GTOs, respectively. Pioneers observed that the optimal combination of speed and accuracy is achieved for $M = 3$, and it follows that STO-3G remains the most commonly used minimum basis set. While among the poorest performing GTOs in terms of accuracy, the STO-*M*G basis sets are still in use today. This is due to their extreme efficiency and exceptional availability, as they have been defined for all elements in the first five periods.

2.2.3. Pople basis sets (3-21G, 6-31G, and 6-31G**)

A minimal basis set can be given additional variational flexibility by strategically 'decontracting' its functions. For example, instead of constructing each basis function as a sum of three Gaussians, as in the STO-3G basis set, one could instead construct two basis functions for each AO, with the first being a contraction of the first two primitive Gaussians and the second being the normalized third primitive. This contraction scheme results in a 'double-ζ' basis set, which can be contrasted with 'single-ζ' sets such as STO-3G. Further decontraction results in a 'triple-*zeta*' set, and more functions can be added indefinitely to create higher multiple-ζ basis sets.

The 3-21G and 6-31G are two within a family colloquially referred to as 'Pople basis sets', a nod to their creator. In this case the nomenclature reveals their contraction scheme. The first number indicates how many primitives were used in the contracted core functions, while the numbers following the hyphen indicate the valence contraction scheme. Enumeration of the digits identifies the multiple-ζ character. The two digits in both 3-21G and 6-31G indicate a double-ζ contraction scheme. The digits themselves provide the number of primitives used to contract each valence function. The asterisks designate the inclusion of additional *polarization functions*, which provide anisotropic degrees of freedom required for orbital hybridization in molecular bonding environments.

2.2.4. Dunning basis sets (cc-pVDZ and cc-pVTZ)

In contrast to the previously discussed basis sets, where contraction coefficients and exponents were fit using data from independent-particle molecular calculations, the correlation consistent (cc) basis functions of Dunning were developed using improved data including a second-order perturbative correction for electron correlation. Periodic trends dictate the number of spin orbitals that should be used for a minimum description of each atom, and systematically improvable hierarchies, such as the Dunning basis sets, were developed to efficiently approach the complete basis set (CBS) limit through extrapolation. Here the nomenclature following the hyphen indicates that polarization functions are included and that the valence functions were constructed subject to a given contraction scheme [double-ζ (DZ), triple-ζ (TZ), etc.]. The systematically improvable hierarchy of Dunning basis sets is cc-pVXZ with X=D,T,Q,5,6,....

2.3. LIMITATIONS OF CLASSICAL ELECTRONIC STRUCTURE ANSÄTZE

This section describes the theoretical formulation and hardware limitations of relevant *classical* electronic structure ansätze. Since quantum computation has not yet delivered on its promise to revolutionize electronic structure, the computational complexity of classical methods is recapitulated here. With one exception, all electronic structure ansätze discussed in this report are Hermitian variational forms subject to the symmetric energy functional

$$E_0^A[\Psi_0^A] = \frac{\langle \Psi_0^A | H | \Psi_0^A \rangle}{\langle \Psi_0^A | \Psi_0^A \rangle}, \tag{2.6}$$

where E^A and Ψ^A are the energy and wave-function, respectively, for a given approximate method A. The variational principle given by Eq. 2.6 is, of course, equivalent to the exact Schrödinger equation when Ψ_0^A is the exact ground state wave-function Ψ_0.

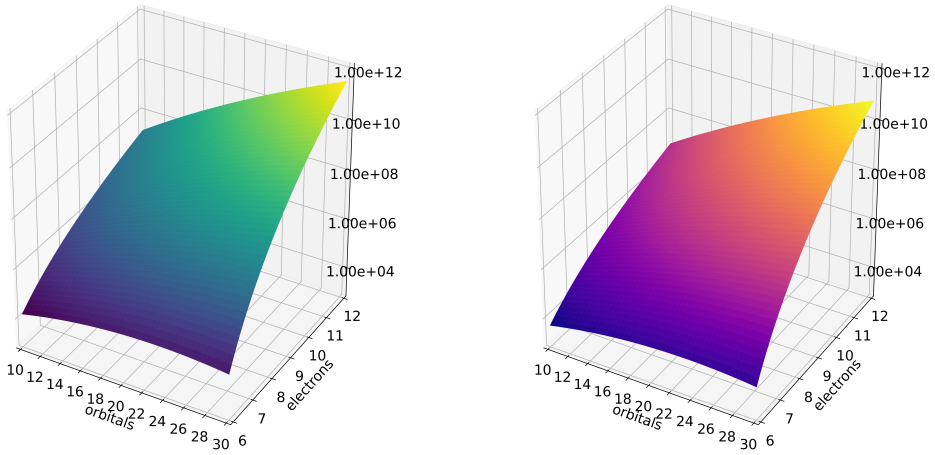


Figure 2-1. Total number of terms in the full configuration interaction expansion. The left plot is in terms of determinants (see Eq. 2.9), while the right plot is in terms of configuration state functions (see Eq. 2.10).

2.3.1. Configuration interaction

In theory, the exact many-electron wave-function is easily written down. To introduce general problem of electronic structure theory, this section develops configuration interaction (CI) theory as the most näive, brute-force approach to an exact solution. For a more modern perspective on CI approaches see, for example, Ref. [16].

2.3.1.1. Determinant-based full configuration interaction

Given an orthonormal set of p one-electron orbitals ϕ_p, one may construct an orthonormal basis of N-electron determinants, $|\Phi_{p_1,\ldots,p_N}\rangle$, and expand the exact N-electron wave-function as

$$|\Psi\rangle = \sum_{p_1<\ldots<p_N} c_{p_1,\ldots,p_N}|\Phi_{p_1,\ldots,p_N}\rangle = \sum_{\mathbf{n_i}} c_{\mathbf{n_i}}|\Phi_{\mathbf{n_i}}\rangle. \tag{2.7}$$

Substituting this vector into Equation 2.6, one obtains the matrix eigenvalue problem

$$\begin{pmatrix} \langle\Phi_{\mathbf{n_1}}|H|\Phi_{\mathbf{n_1}}\rangle & \langle\Phi_{\mathbf{n_1}}|H|\Phi_{\mathbf{n_2}}\rangle & \ldots \\ \langle\Phi_{\mathbf{n_2}}|H|\Phi_{\mathbf{n_1}}\rangle & \langle\Phi_{\mathbf{n_2}}|H|\Phi_{\mathbf{n_2}}\rangle & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} c_{\mathbf{n_1}} \\ c_{\mathbf{n_1}} \\ \vdots \end{pmatrix} = E \begin{pmatrix} c_{\mathbf{n_2}} \\ c_{\mathbf{n_2}} \\ \vdots \end{pmatrix}. \tag{2.8}$$

This approach is called full configuration interaction (FCI). The size of the FCI space is equivalent to the number of possible determinants,

$$N_d = \binom{N}{\frac{\eta}{2}+S}\binom{N}{\frac{\eta}{2}-S}, \tag{2.9}$$

where N is the total number of orbitals and η is the number of electrons, and $S(S+1)$ is the eigenvalue of the total spin operator S^2. From Figure 2-1 it is clear that even for small systems, such as the water molecule in a minimal basis (10 electrons and 20 orbitals), the FCI dimensionality is too large for diagonalization using standard `LAPACK` routines executed in serial on a modern desktop computer.

2.3.1.2. CSF-based full configuration interaction

The FCI dimensionality can be significantly reduced by employing a basis of configuration state functions (CSFs), where CSFs are linear combinations of the determinants $|\Phi_{\mathbf{n_i}}\rangle$ forming eigenfunctions of the total spin operator S^2. This change of basis is exact since the total spin is a constant of motion in the non-relativistic Hamiltonian (i.e., $[S,H]=0$). The total number of unique CSFs which can be formed is given by the Weyl formula as

$$N_c = \frac{2S+1}{N+1}\binom{N+1}{\frac{\eta}{2}-S}\binom{N+1}{\frac{\eta}{2}+S+1}. \tag{2.10}$$

The value N_c is collected for several combinations of N and η in Figure 2-1. Again using as an example the water molecule in a minimum basis (10 electrons and 20 orbitals), the determinant FCI dimension is seen to be a quarter of a billion, whereas the CSF FCI dimension is close to 50 million. While switching to a CSF basis reduces the size of the problem, it remains intractable on serial-execution classical hardware. The combinatorial growth of the FCI problem with system size, shown explicitly by the Weyl formula in Eq. 2.10, is sometimes referred to as "the curse of dimensionality". Approximation methods are thus essential for classical simulation of electronic structure.

2.3.1.3. Approximate configuration interaction and size extensivity

Approximations can be made within the CI framework by including only certain classes of determinants in the wave-function given by Eq. 2.7. This introduces formal problems, however, as approximate CI wave-functions break an important property called size-extensivity. The concept of size extensivity was introduced by Bartlett [17], and it is a mathematically rigorous characteristic referring to the correct (linear) scaling of a method with the number of electrons. Davidson introduced a perturbative correction for approximate restoration of size-extensivity[18, 19], but, for reasons discussed shortly, practical truncations of the CI wavefunction are typically abandoned in favor of size-extensive coupled-cluster (CC) theory analogs.

2.3.2. The Hartree-Fock method

Hartree-Fock (HF) theory provides an approximate solution to the electronic Schrödinger equation which is of fundamental importance in electronic structure theory. For an extensive review on the derivation of the HF self-consistent field (SCF) equations for both closed- and open-shell systems, see Ref. [20]. In the following a non-technical summary of HF theory is provided which enables a discussion of the current limitations of HF on classical hardware.

2.3.2.1. HF theory

Given a basis of single-particle atomic orbitals centered on nuceli with appropriate proximity, HF theory performs orbital rotations until reaching self-consistency, thereby producing a new set of orthonormal functions known as the molecular orbitals (MOs), $\{\phi_a\}$. The exact electronic correlation energy E^{corr} within a basis set is, by definition, the difference between the FCI and HF energies, or

$$E^{\text{corr}} = E^{\text{FCI}} - E^{\text{HF}}. \tag{2.11}$$

In general, the quality of a wave-function approximation is judged by the extent to which it recovers E^{corr}. The HF machinery produces the optimal single-Slater-determinant approximation of the many-body wave-function, and this serves as an especially good zeroth-order starting point for expanding the correlated many-body wave-function in post-HF methods. Post-HF methods often utilize the electron-electron repulsion integrals (ERIs) generated from converged HF orbitals, which are computed as indicated in Equation 2.3.

For an N-electron system, solving the HF equations is equivalent to minimization of the energy of the determinant $|\Phi_0\rangle = |\prod_{i=1}^{N_o} \phi_i\rangle$, with the RHS an antisymmtrized Hartree product of single-particle orbitals ϕ_i. This leads to an eigenvalue equation $\hat{f}|\phi_i\rangle = \varepsilon_i|\phi_i\rangle$ for the N_o occupied spin orbitals, with the Fock operator defined as

$$\hat{f} = \hat{H}^{\text{core}} + \sum_{j=1}^{N/2} \left[2\hat{J}_j - \hat{K}_j\right], \tag{2.12}$$

where $\hat{H}^{\text{core}}$, $\hat{J}_j$, and $\hat{K}_j$ are the one-electron core Hamiltonian, the Coulomb operator, and the exchange operator, respectively. For more information we direct the reader to Ref. [10].

After obtaining a set of converged SCF orbitals from HF theory, a variety of methods can be subsequently applied to recover, at least in part, the electron correlation energy (E^{corr} in Eq. 2.11). It is common practice in post-HF methods to partition the occupied HF molecular orbitals into valence and core sets, and, since the latter are inert in most chemical processes, core MOs can be excluded from consideration in post-HF methods. This is especially true when targeting relative energies. For the atoms lithium to neon typically the 1s atomic orbital is frozen, while for atoms sodium to argon the atomic orbitals $1s$, $2s$, $2p_{\text{x}}$, $2p_{\text{y}}$ and $2p_z$ are frozen. The frozen core approximation is so common, in fact, it is actually the default setting in many electronic structure software packages. Similarly, virtual orbitals can be omitted from post-HF calculations. However, there is not a universally accepted convention for choosing frozen virtual orbitals, so they are

seldom reported in the literature. For the systems of interest in this work (i.e., hydrogen clusters in small basis sets), all occupied and virtual orbitals participate to some degree in chemical bonding and therefore must remain unfrozen. Applying this approximation within VQE algorithms is also trivial, so we avoid repeating this discussion in Sect. 2.4.

2.3.2.2. Techniques for reduced-scaling HF calculations

The HF eigenvalue problem is quite straightforward to solve using iterative algorithms on classical hardware with memory requirements of $O(N^3) - O(N^4)$. This scaling is attributable to two bottlenecks. The first is the construction of the Fock matrix, which requires evaluation of as many as $O(N^4)$ ERIs. Negligibly small ERIs are eliminated using Cauchy-Schwarz screening [21, 22] and related techniques [23–27], such that, in the asymptotic limit of large systems, the number of evaluated ERIs is reduced to $O(N^2)$ or, for certain insulating systems, even $O(N)$ [28].

The second bottleneck is diagonalization of the $N \times N$ Fock matrix into its eigenvectors and eigenvalues. Eigensolvers applied to dense matrices run with a complexity of $O(N^3)$, but, when exploiting sparsity in large systems, it is again possible to achieve $O(N)$ scaling [29, 30]. Density fitting techniques [21, 31] can also reduce the number of required ERIs from $O(N^4)$ to $O(N^3)$, but it has recently been demonstrated that direct integral techniques are more efficient for large systems [32].

2.3.2.3. Limits on massively-parallel hardware

Specialized classical hardware can be used to accelerate HF computations. High-performance parallel algorithms for construction of the Fock matrix have been recently proposed [32, 33], and benchmark studies [32, 34] imply a single-node time requirement under 10 minutes for a system of 10,000 basis functions. Graphical processing units (GPUs) have been shown to accelerate HF calculations, both as an alternative [35–40], and as a complement to CPU-based SCF algorithms [41–44]. Mixed- and dynamic-precision approaches offer additional speedups [35, 36, 45]. Hybrid algorithms have also been implemented on Xeon Phi processors, and performance studies imply 50,000 basis functions can be treated on a single node with 200GB of memory in O(10 mins) [46, 47]. Thus, we regard 50,000 basis functions as the current limit for state-of-the-art canonical HF calculations, though obviously larger calculations will be possible on architectures built with more memory per node. For systems requiring more than 50k basis functions, non-canonical linear-scaling fragment-based SCF methods are available, but a detailed discussion of these is beyond the scope of this report (for further information, see, e.g., Refs. [48–50] for related reviews and Ref. [51] for a recent perspective article.)

2.3.3. Projection coupled-cluster methods

Single-reference methods based on the exponential wavefunction ansatz of CC theory are the most successful approximate electronic structure methods for the calculation of energies and

properties of ground-state atomic and molecular systems[52]. Contrary to the CI approach, which is characterized by a linear expansion of CSFs, the CC approach uses an exponential ansatz for the wave function, which inherently assures that truncated forms of CC theory remain size-extensive, and produces a much faster convergence to the full CI wave function. It can be shown using a perturbation theory analysis that the improved convergence of CC as compared with the same level of truncation in CI theory is due to higher-order excitations being folded in as products of lower-order excitations by the exponential form of the CC ansatz. At the same time, thanks to the use of diagram factorization techniques commonly employed in efficient computer implementations of CC methods, the computer costs of CC calculations are similar to those characterizing the CI approaches truncated at the same excitation levels. This is why CC methods can offer higher accuracy at relatively lower costs as compared with CI methods and even though the energies produced are not variational, they are typically considerably more accurate than those produced by CI at the same level of truncation.

2.3.3.1. Traditional coupled-cluster theory

Projection CC (PCC) utilizes an exponential ansatz, such that

$$|\Psi^{\mathrm{PCC}}\rangle = e^{T}\,|\Phi_0\rangle\,, \tag{2.13}$$

where $|\Phi\rangle$ is the reference determinant (usually HF) and T is the cluster operator, an N-body excitation operator of the form

$$T = \sum_{k} T_k. \tag{2.14}$$

with

$$T_k = \sum_{\substack{i_1<i_2<\ldots<i_k\in\mathrm{occ}\\ j_1<j_2<\ldots<j_k\in\mathrm{virt}}} t^{i_1,i_2,\ldots,i_k}_{j_1,j_2,\ldots,j_k} \left(\prod_{r=1}^{k} a^{\dagger}_{i_r}\right)\left(\prod_{s=1}^{k} a_{j_s}\right) \tag{2.15}$$

where $a^{\dagger}_{i_r}$ (a_{j_s}) are creation (annihilation) operators and $t^{i_1,i_2,\ldots,i_k}_{j_1,j_2,\ldots,j_k}$ are the cluster amplitudes.

In practice, the sum over k is truncated to some finite order. For example, in the most basic approximation*, PCC with single and double excitations (PCCSD), the cluster operator in Eq. 2.14 is approximated as $T \approx T_1 + T_2$. The PCCSD cluster amplitudes and the PCCSD energy E^{CCSD}, are obtained by solving the following system of equations:

$$\begin{aligned}
\langle|\Phi_0|e^{-(T_1+T_2)}He^{(T_1+T_2)}|\Phi_0\rangle &= E^{\mathrm{PCCSD}}\\
\langle|\Phi^{a}_{i}|e^{-(T_1+T_2)}He^{(T_1+T_2)}|\Phi_0\rangle &= 0\\
\langle|\Phi^{ab}_{ij}|e^{-(T_1+T_2)}He^{(T_1+T_2)}|\Phi_0\rangle &= 0,
\end{aligned} \tag{2.16}$$

where $|\Phi^{a}_{i}\rangle$ and $|\Phi^{ab}_{ij}\rangle$ are singly- and doubly-excited determinants out of the reference wave-function, Φ_0. Solving the PCCSD equations requires RAM scaling as $4N_o^2 \times N_u^2 + N_o \times N_u^3$

*PCC with only single excitations (PCCS) is never considered because Thouless's theorem states PCCS is equivalent to Hartree-Fock.

and serial execution time scaling as $N_o^2 N_u^4$, where N_o (N_u) is the number of occupied (unoccupied) correlated orbitals.

Following solution of the PCCSD equations, the CC wave function can be truncated at progressively higher orders (i.e., increasingly higher levels of excitation) until sufficient convergence to the exact solution is reached. Ascending the single-reference hierarchy is seldom done in practice, as CC with singles, doubles, and triples (PCCSDT) is already computationally intractable for the vast majority of systems due to its characteristic $O(N^8)$ scaling. As an alternative, one may use instead a multi-reference formulation of coupled-cluster theory, which can recover the effects of triples more efficiently than in the full PCCSDT method. Multi-reference CC methods are an active area of research (see Ref. [53] for a recent perspective article).

2.3.3.2. Brueckner coupled-cluster theory

Brueckner CC (BCC) performs orbital rotations such that the t_1 amplitudes from Eq. 2.16 become vanishingly small. The resulting orbital set, known as Brueckner orbitals [54], differs from the HF orbitals in that they are optimized self-consistently in the presence of electron correlation introduced by the $T_n = T_1 + \ldots + T_n$ operators (assuming $1 < n < N$). Within a given one-electron Hilbert space, Brueckner orbitals are defined sufficiently by satisfying one of the following two conditions: (a) the single-excitation contributions to the exact wave function vanish or (b) the reference determinant has the maximum overlap with the exact wave function. Employing Thouless's theorem, these two conditions can be shown to be equivalent by requiring stationarity of the overlap of the Brueckner determinant $|\Phi_B\rangle$ and the exact wave function Ψ with respect to the components of $\hat{T}_1$, or

$$0 = \frac{\partial}{\partial t_i^a} \langle \Psi | e^{\hat{T}_1} | 0 \rangle = \langle \Psi \hat{a}_a^\dagger \hat{a}_i e^{\hat{T}_1} | 0 \rangle = \langle \Psi | (\Phi_B)_i^a \rangle \tag{2.17}$$

An advantage of BCC calculations is that, with $\hat{T}_1 = 0$ many nonlinear terms (e.g., $\hat{T}_1 \hat{T}_3$, $\hat{T}_1 \hat{T}_4$, etc.) do not contribute to the total energy and thus do not limit the accuracy of truncated BCC methods, such as BCC doubles (BCCD)[†]. In classical computations, the Brueckner wave function is determined iteratively by repeatedly solving the PCCSD equations while searching numerically for the orbitals satisfying $\hat{T}_1 = 0$. This procedure adds a significant prefactor (typically > 10) to the expense. In the following section, a VQE-based BCC analog is discussed, which can be employed without consuming additional quantum resources.

2.3.3.3. Limits on massively-parallel hardware

Leading MPI implementations of CCSD and CCSD(T) scale well in to the regime of thousands of cores, with representative packages including the MPQC[55, 56], ACESIII[57], Aquarius[58], FHI-aims [59], and NWChem [60, 61] program suites. Gyevi-Nagy et al. [62] recently compared the performance of their MPI/OpenMP CCSD and CCSD(T) implementations and that of several

[†]The 'S' is intentionally omitted since singles are eliminated.

leading packages. For systems with about 1000 orbitals CCSD required 4–7 hours on 112 cores. They went on to report performing one of the largest canonical CCSD(T) calculations ever, involving 1569 orbitals and requiring 224 cores running for 68 hours. For comparison, Fales et al. [63] recently reported a state-of-the-art GPU-accelerated CCSD implementation. Their code, when run on a single CPU node accelerated by 8 high-end GPUs, can tackle systems with 1000–1300 basis sets in less than 1 day. They claim that this timing is competitive with the performance of an unsophisticated MPI implementation of CCSD with distribution across 1024 CPU cores on 64 nodes. In contrast to the MPI/OpenMP implementations, the CPU/GPU implementation is memory-limited, which means advances in GPU hardware will be accompanied by rapid timing improvements. While progress in classical CC implementations continues to improve, we adopt a conservative estimate for the classical limit of roughly 1000 basis functions.

2.4. HYBRID ANSÄTZE FOR STATE PREPARATION

The choice of ansatz for state preparation strongly influences the performance of VQE. Currently there are two categories of ansatz which may be implemented on NISQ devices with a low-depth, high-fidelity circuit. The first class are the physically motivated ansätze, which are developed with insight into the physical nature of the simulation, while the second are the hardware heuristic ansätze, which are designed to exploit unique features of targeted quantum hardware platforms.

2.4.1. Physically motivated ansätze: The Unitary Coupled-Cluster Methods

In the development of physically-motivated ansätze, many classical electronic structure methodologies have been translated into quantum analogs. The unitary CC (UCC) theory is among the most successful of this type. The method has been known for many decades, but it remained unpopular due to implementation difficulties on classical hardware [64–71]. While the UCC method is not identical to variational CC [72], its energy is similarly based upon a symmetric expectation value, so it is guaranteed to obey the variational theorem.

The present study utilized a workflow that involves generation of non-relativistic HF integrals by the driver on classical hardware and these are subsequently used to generate the UCC ansatz quantum circuit. One might expect an advantage to implementing a quantum HF method rather than using classical quantum chemistry programs to provide integrals. A VQE HF solver was recently presented by the Google Artificial Intelligence Quantum team[11], where Thouless's theorem was used as a tool to solve the HF equations by applying Givens rotations. In response, Gulania and Whitfield pointed out that HF is NP-Hard and therefore it is not a strong candidate for exponential speedup[73]. Thus, the primary advancement brought about by a VQE HF solver is migrating more of the VQE workflow onto quantum hardware. Rather than surveying all hybrid VQE ansätze, we focus here on those approaches which are *enabled* by quantum hardware, meaning that they are virtually impossible to implement on classical hardware alone.

2.4.1.1. Unitary coupled-cluster theory

Unitary CC (UCC), which simply makes $e^{\hat{T}}$ unitary in Eq. 2.13, is a natural extension of projection coupled-cluster theory, and one that has undergone a renaissance since its consideration in the context of quantum computation. The UCC wave function is formally defined as

$$|\Psi^{\mathrm{UCC}}(\theta)\rangle = e^{\hat{T}-\hat{T}^\dagger}|\Phi_0\rangle = e^{\tau}|\Phi_0\rangle\,. \tag{2.18}$$

Variational minimization of the ground-state energy yields the desired wave function, with the unitary nature of the ansatz enabling variation to be carried out using quantum circuits. Similar to projection CC, the most efficient non-trivial approximation of the excitation operator truncates at doubles (i.e., UCCSD), or

$$\hat{\tau} \approx \hat{T}_1 + \hat{T}_2 - \hat{T}_1^\dagger - \hat{T}_2^\dagger, \tag{2.19}$$

where T_1 and T_2 are expressed as

$$\begin{aligned} T_1 &= \sum_{ia} \theta_i^a \hat{a}_a^\dagger \hat{a}_i \\ T_2 &= \sum_{ijab} \theta_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_i \hat{a}_j \end{aligned} \tag{2.20}$$

The quality of CC energies is, in general, ordered as CCSD $<$ UCCSD $<$ CCSD(T) $<\ldots$ Truncation schemes with explicit inclusion of excitation operators beyond singles and doubles have not yet been implemented. The number of parameters associated with the UCCSD ansatz grows as

$$\binom{N-\eta}{2}\binom{\eta}{2} + \binom{N-\eta}{2}\binom{\eta}{2} < O(N^2\eta^2) \tag{2.21}$$

with N the number of spin orbitals and η the number of electrons in the system[74].

2.4.1.2. Orbital-optimized unitary coupled-cluster

Mizukami et al. proposed the orbital-optimized (oo) VQE-UCC approach, in analogy to the Brueckner CC methods [75], where the coupled-cluster amplitudes and molecular orbital coefficients are simultaneously variationally optimized [76]. The key advantage of oo-VQE is that first analytical derivatives of the energy are obtainable without solving any additional equations. Derivatives of the VQE energy are given as

$$\frac{\mathrm{d}E(x,\theta,\kappa)}{\mathrm{d}x} = \frac{\partial E(x,\theta,\kappa)}{\partial x} + \frac{\partial E(x,\theta,\kappa)}{\partial x}\frac{\partial \theta}{\partial x} + \frac{\partial E(x,\theta,\kappa)}{\partial x}\frac{\partial \kappa}{\partial x}, \tag{2.22}$$

where θ and κ are the quantum circuit parameters and molecular orbital parameters, respectively. For a valid oo-VQE solution, the second and third terms in Eq. 2.22 go to zero and the first-order oo-VQE energy derivative can be determined by evaluating the expectation value of the derivative of the Hamiltonian. For VQE solutions without oo, clearly the third term in Eq. 2.22 is nonzero and this must be addressed if gradients are desired. Orbital-optimized methods have the same gate-count and circuit-depth requirements as their base methods, while the additional computational burden manifests as an increase on the parameter count in the classical optimizer. The rate of proliferation of classical parameters is investigated in Sect. 3.5.

Method	Resource requirements	
	Gate count	Circuit depth
VQE-HF	$O((N-\eta)\eta))$	$O(N)$
UCCSD	$O((N-\eta)^2\eta^2))$	$O((N-\eta)^2\eta))$
oo-UCCSD	$O((N-\eta)^2\eta^2))$	$O((N-\eta)^2\eta))$
UpCCD	$O((N-\eta)\eta))$	$O(N)$
oo-UpCCD	$O((N-\eta)\eta))$	$O(N)$
k-UpCCGSD	$O(kN^2)$	$O(kN)$
UGCCSD	$O(N^4)$	$O(N^3)$

Table 2-1. Scaling laws for the quantum resource requirements for various hybrid VQE methods, expressed in terms of system parameters including the number of electrons η, the total number of basis fucntions N, and k, a parameter specific to the k-UpCCGSD method.

2.4.1.3. The k-UpCCGSD ansätz

Among the most promising recent advances in UCC methodology is the k-UpCCGSD approach of Lee et al. [77]. This method applies k-times the UCC with generalized (i.e., summed over all orbitals without specifying occupation) single and pair double excitations to construct a wave function of the form

$$|\Psi^{k\text{-UpCCGSD}}\rangle = \prod_{\alpha=1}^{k} \left(e^{T^{(a)}-T^{(a)\dagger}}\right)|\Phi_0\rangle \tag{2.23}$$

where each $T^{(k)}$ contains an independent set of variational parameters contained in the generalized singles and pair-doubles operator

$$\hat{T} = \hat{T}_1 + \hat{T}_2 = \frac{1}{2}\sum_{pq} t_p^q \hat{a}_q^\dagger \hat{a}_p + \frac{1}{4}\sum_{ia} t_{i_\alpha i_\beta}^{\hat{a}_\alpha \hat{a}_\beta} \hat{a}_{a_\alpha}^\dagger \hat{a}_{a_\beta}^\dagger \hat{a}_{i_\beta} \hat{a}_{i_\alpha}, \tag{2.24}$$

where the pair-doubles T_2 operator promotes pairs of electrons from one spatial orbital to another. This approach has been shown to provide a systematic way to converge to FCI without introducing higher excitation rank operators. This significantly reduces the gate count and circuit depth requirements to $O(kN^2)$ and $O(kN)$, respectively. For convenience, Table 2-1 collects resource requirements for several UCC variants.

2.4.1.4. Trotterization

The Suzuki-Trotter decomposition, or simply Trotterization, is a general approach for decomposing the exponential of a sum of operators [78, 79], which is necessary for implementation of Eq. 2.18. Each θ_i must be mapped to a sum of commuting[74] Pauli strings. Trotter error can be systematically mitigated by choosing increasingly fine discretizations, but this

comes at the expense of drastic increases in circuit depth. For the sake of computational efficiency, often one considers only a single Trotter step,

$$|\Psi^{\mathrm{UCC}}\rangle \approx \prod_j e^{t_j(\hat{\tau}_j - \hat{\tau}_j^\dagger)},$$

where the t_j are the excitation amplitudes and the τ_j are the corresponding excitation operators. Trotter-Suzuki decomposition represents an uncontrollable error source [80–83], although a single-Trotter-step UCC ansatz was shown by Barkoutsos et al. to reproduce ground-state energies accurately [84] and O'Malley et al. observed that the variational flexibility actively compensates for severe Trotterization error [85].

Grimsley et al. recently noted that the non-uniqueness of Trotterization presents a crisis of reproducibility, and that the only solution is to explicitly report which among the combinatorial number of operator-ordering possibilities was employed for a given study [83]. Novel ansätze have also been proposed which either uniquely determine the operator ordering [86, 87] or do not require Trotteriztion [88–90]. Grimsley's argument is indisputable, yet a compact scheme for reporting Trotterization ordering has not been established. It remains an open question whether it is more convenient to abandon Trotterization or develop a reporting scheme.

2.4.1.5. Limits of VQE-UCC and comparison with classical analogs

Select VQE quantum resource requirements are known, but the associated power scalings can seem very abstract. Figure 1-2 provides explicit values for the quantum resource scaling of UCCSD and compares with the classical resource scaling of PCCSD, with three basis set levels and 20 species of chemical relevance considered. The data therein are reported relative to the expense of calculation of the system using a minimum (STO-3G) basis set, and, while they are approximate to the true resource requirements, they enable order-of-magnitude resource estimates. For the largest molecules of interest, PCCSD simulations performed in a large basis set (i.e., one which enables extrapolation to the CBS limit), is estimated to require up to a 1 million-fold increase in the classical resources (RAM) over an STO-3G calculation. For VQE-UCCSD, the same transition requires up to 1 hundred-thousand-fold increase in the gate count.

2.4.2. Hardware-efficient ansätze

Circuits based on the hardware-efficient ansätze involve alternating single-qubit rotations and entangling blocks. By design, the entangling blocks leverage quantum hardware capabilities specific to a given platform. This approach facilitated the original experimental demonstration of VQE for small molecules and quantum magnets[91]. While these approaches are easier to implement on NISQ devices than those based on physically motivated ansätze, it is unclear whether they are universally applicable for generating many-electron wave functions.

For hybrid quantum-classical algorithms suitable for NISQ devices, namely those involving a parametrized unitary circuit controlled by a classical optimization loop, success hinges upon

avoidance of certain topographic features in the classical parameter space. A pervasive limiting feature is the so-called "barren plateau", a region of parameter space where gradients in all directions are zero to within the sampling error of the estimator, rendering useless any optimizer that is not inherently derivative-free [9, 92]. McClean et al. estimated the variance of the energy gradient of a two-local Pauli term and discovered two important results: the variance decreases exponentially with increasing number of qubits and it converges toward its fixed lower value as a function of the circuit depth [93]. The implication for hardware-efficient ansätze[91] is that barren plateaus will occur with a frequency that grows exponentially with the number of qubits. Very recently vanishing gradients have also been demonstrated for *shallow* QNNs [94] and for ansätze requiring a one-depth circuit [95]. This represents a serious problem for algorithms based on 2-designs, and as such several interesting remediation strategies have been proposed, including refined parameter initialization schemes [96, 97], new classes of QNNs [98, 99], and even a variational quantum compiler [95], among others.

3. VQE RESOURCE ANALYSES

3.1. QUANTUM CHEMISTRY DRIVERS

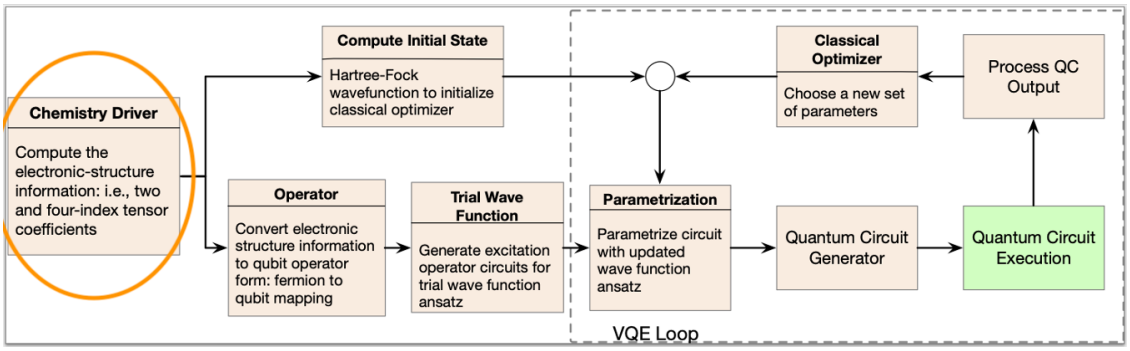


Figure 3-1. The purpose of the quantum chemistry driver is to supply HF ERIs to the VQE routine.

3.1.1. Overview of the Different Drivers

There are several drivers currently integrated with IBM's Qiskit or Microsoft's quantum development kit (QDK)[100].

3.1.1.1. NWChem

The NWChem software suite is a mature quantum-chemistry package written in C++ that is actively developed and maintained by the EMSL located at the Pacific Northwest National Laboratory in Richland, Washington [101]. The code is able to supply non-relativistic SCF integrals to Microsoft's open-source QDK [100], which includes Q#-language compilers and simulators. Within the QDK, a VQE-based sparse multi-configuration SCF method [102, 103], which is the base method required for constructing CSFs, has been implemented and has been made available to the community. Such wave-functions are generators of starting orbitals for multi-reference correlated methods, a topic which is beyond the scope of this report. NWChem is not currently integrated with Qiskit.

3.1.1.2. PyQuante

The PyQuante software suite, pronounced 'picante', is a pythonic open-source light-weight electronic structure software package created for ease-of-development by Richard Muller (and collaborators) at Sandia National Laboratories in Albuquerque, New Mexico [104]. Compared to the other suites on this list PyQuante has an incomplete set of classical features, yet its non-relativistic HF functionality was among the earliest to be adopted for quantum simulation. Qiskit has integrated PyQuante for integral generation.

3.1.1.3. Psi4

The Psi4 software suite is a mature electronic structure software suite written in C++ that is actively developed and maintained by The Psi4 Project team, who are based at Georgia Tech, at Virginia Tech, at Emory University, and, in the group of its genesis, at University of Georgia. It is the preferred quantum chemistry backend for the OpenFermion project, a pythonic open-source software library for simulation of quantum computation written by scientists at Google Research, Riggetti Computing, and their collaborators. Qiskit also accepts non-relativistic integrals generated by Psi4.

3.1.1.4. PySCF

The PySCF software suite is a pythonic open-source light-weight electronic structure software package actively developed and maintained by Qiming Sun of Garnet Chan's Caltech group in Pasadena, California, with contributions from other research groups [105]. It is the youngest package among the four, with the first official publication dated 2018 [106]. By now its classical feature list is relatively comprehensive, and it includes the capability to generate both non-relativistic and 4-component relativistic integrals. PySCF can supply non-relativistic integrals to both OpenFermion and Qiskit.

3.1.2. Results: Memory and Time Scaling

Generation of ERIs by the driver is one potential bottleneck on the classical side of Figure 3-1. Modern classical electronic structure calculations are usually either memory-limited or time-limited. Finite disk space is seldom a limiting factor in modern applications. HF algorithms are by now well established, so we do not expect that one of the above-mentioned drivers would significantly outperform another. As a result, PySCF was chosen for performing scaling studies because of its relative ease-of-use when interfacing with Qiskit.

Figure 3-2 tracks memory usage with increasing deuterium-cluster size as performed on two classical systems. The first system is a desktop with 8GB of RAM, which caps out at around 150 nuclei, while the second is the HPC system Uno with 64 of RAM, which can handle around 250 nuclei. Scaling with system size was found to be $n^{3.6}$ and $n^{4.0}$ for the desktop and HPC system,

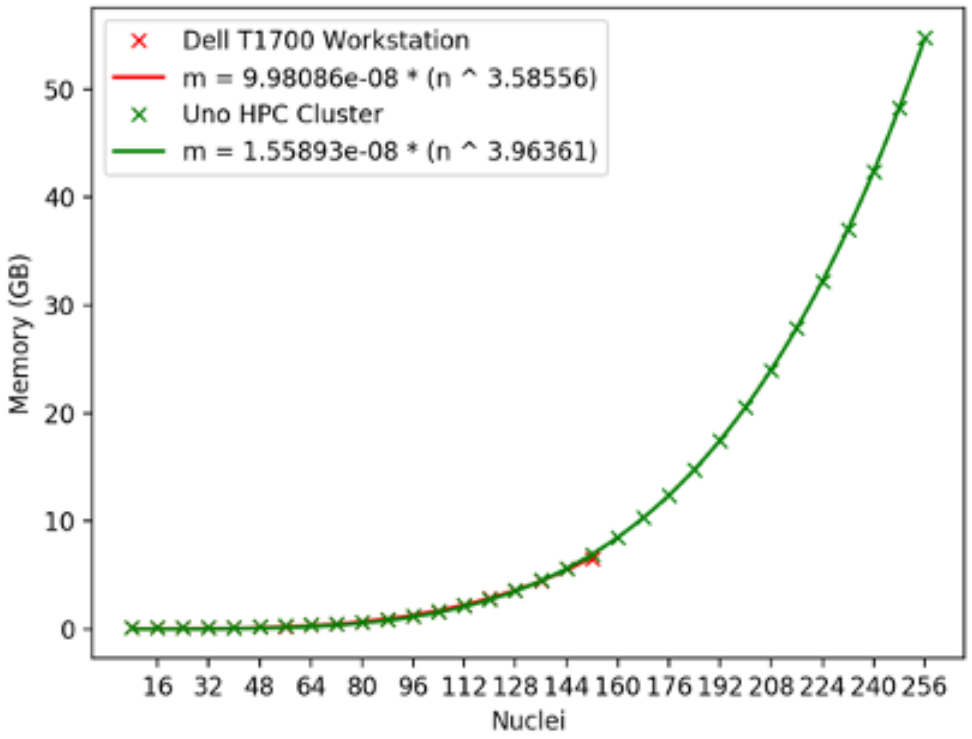


Figure 3-2. Scaling of the RAM requirement with increasing deuterium-cluster size on a desktop computer and a high-performance computing cluster housed at SNL.

respectively. These empirical values are consistent with the established $n^3 - n^4$ scaling discussed in Section 2.3.

Figure 3-16 collects CPU timings for various deuterium clusters as performed on a desktop, a laptop, and two HPC systems. Each of these platforms was executed in serial, and was thus temporally limited by similar processor clock speeds of 2.7–3.0 GHz. This resulted in nearly identical empirical scaling coefficients and polynomials, with the latter varying from $n^{3.0}$–$n^{3.1}$. At 220 nuclei, while approaching the memory limit of Uno, calculations take less than 5 minutes. Serial calculations are thus memory limited.

We have verified empirically that generation of ERIs requires time requirements with cubic scaling in the number of basis functions and quartic memory scaling. The associated scaling prefactors are between 1×10^{-7}–1×10^{-8}, but at this point we have no other data to compare to these values. In the following sections of this report, each of the remaining elements in the software stack are considered, leading to a final assessment which will identify the resource limiting step(s).

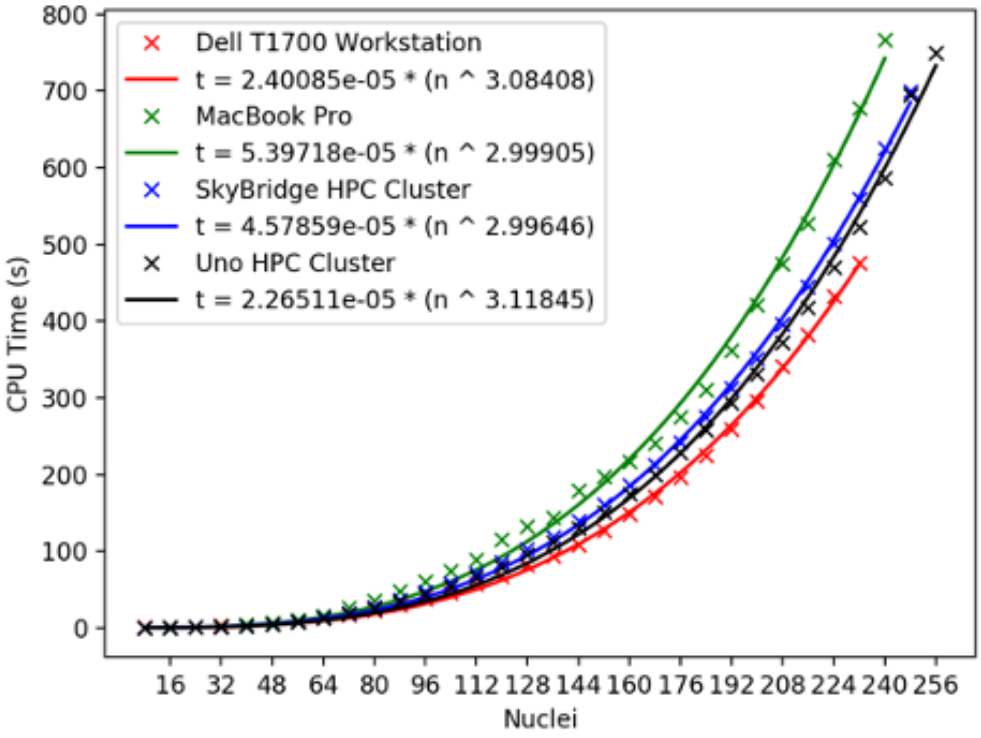


Figure 3-3. Scaling of the CPU time requirement with increasing deuterium-cluster size on a desktop workstation, a laptop, and two high-performance computing clusters housed at SNL.

Mitigation Strategies: Utilizing the SNL supercomputer Uno, we can already handle deuterium system sizes with atom counts in the hundreds. But what improvements should be expected in the next 10 years? Imagine a future commercially available classical-computing platform having similar specifications to ORNL's Summit supercomputer. Leveraging the full 600GB of RAM available on a single node, serial HF computations would still be limited to 440 nuclei - only twice the number treatable by Uno! Clearly we are approaching the limits of what is possible on serial classical hardware, but fortunately parallel HF routines are also available. As discussed in Section 2.3, our estimate for the maximum number of basis functions was 50,000 which equates to 25,000 deuterium atoms in our model. It is thus doubtful that the driver will be the bottleneck for VQE any time soon. The remainder of the tests employed the PySCF driver on the T1700 workstation or Uno supercomputer for generation of ERIs.

3.2. FERMION-TO-QUBIT MAPPING

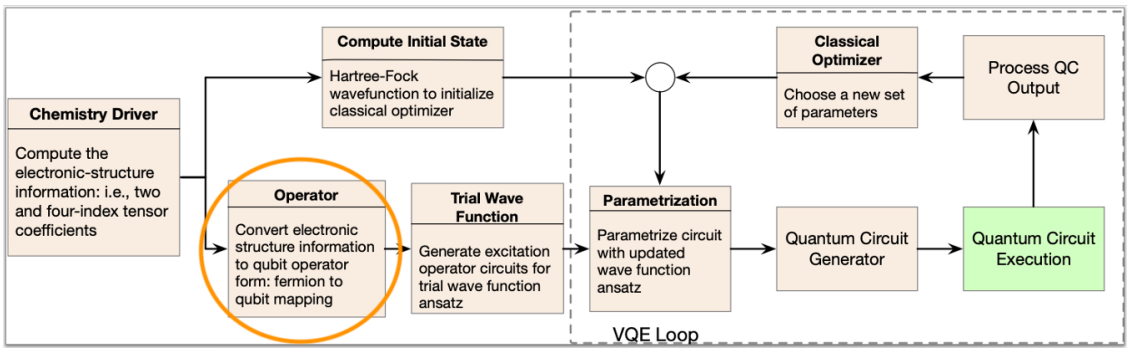


Figure 3-4. The purpose of the mapping is to convert fermionic operators to qubit operators.

3.2.1. Transformation Algorithms Overview

On most quantum computing architectures, the computational resources available are in the form of qubits. Thus, the fermionic operators appearing in the UCC ansätz, Eq. 2.18, must be encoded into qubit operators, consisting of products of Pauli operators with the correct commutation relations. Among the many standard ways of performing this mapping, we limit our consideration to the Jordan-Wigner transformation, the parity transformation, and the Bravyi-Kitaev transformation.

3.2.1.1. Jordan-Wigner

The Jordan-Wigner transformation maps N fermions on to N ordered qubits by assigning to the value of the j-th qubit the occupation of the j-th fermionic mode[107, 108]. Meanwhile, a Z-check on the neighboring qubits provides the parity information. The correspondence between

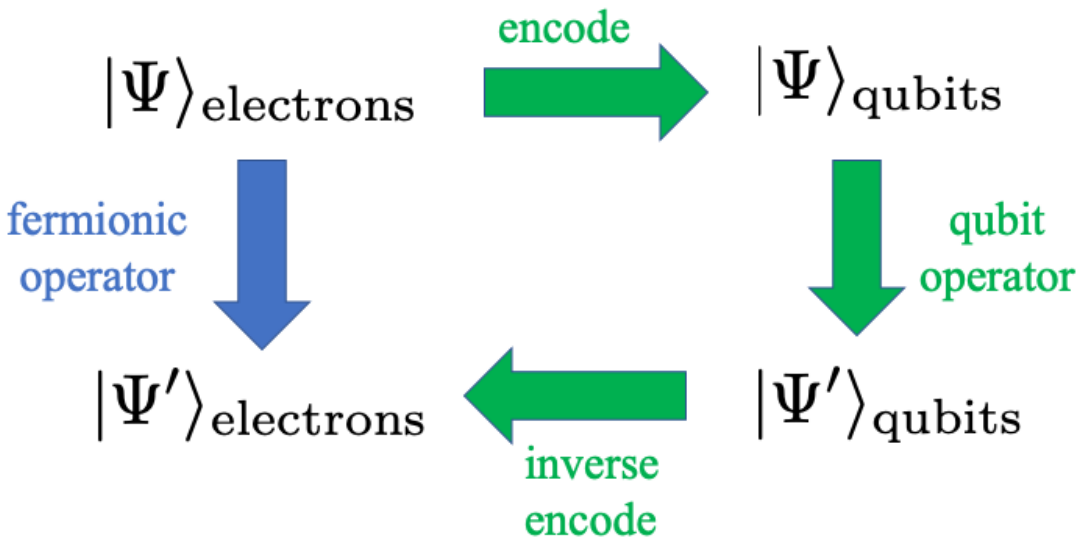


Figure 3-5. A quantum-computational simulation scheme (green path) first encodes fermionic states in qubits, then acts with the qubit operator representing the fermionic operator (obtained by the associated mapping), then inverts the encoding to obtain the resultant fermionic state. A successful simulation scheme reproduces the action of the classical fermionic operator, meaning that the terminals shared by the green path and the blue path are wholly equivalent.

fermionic creation/annihilation operators and the qubit operators is formally defined in the JW transformation as

$$\begin{aligned} a_j &\to \left(\prod_{i=1}^{j-1} \sigma_i^z\right) \sigma_j^+ \\ a_i^\dagger &\to \left(\prod_{i=1}^{j-1} \sigma_i^z\right) \sigma_j^-, \end{aligned} \tag{3.1}$$

assuming conventional definitions for $\sigma^+ = (\sigma^x + i\sigma^y)/2$ and $\sigma^- = (\sigma^x - i\sigma^y)/2$. The JW operators in Eq. 3.1 act to flip the i-th bit of a qubit state vector, while attributing a sign that is encoded into the count of 1-bits in the subset with index less than i. Although this keeps the occupation information entirely local, the parity information is widely distributed. Thus, the annihilation and creation operators are entirely nonlocal, acting on $\mathcal{O}(N)$ qubits.

3.2.1.2. Parity

The parity transformation encodes into the jth qubit the parity of the occupation numbers of the first j spin-orbitals[108, 109]. That is, if there are an odd number of electrons in spin-orbitals 1 through j, then the jth qubit will be in the state $|1\rangle$, and otherwise it will be in $|0\rangle$. In other words, the parity information is local and the occupation number information distributed, the opposite case from Jordan-Wigner transformation. The formal definition is

$$\begin{aligned} a_j &\to \frac{1}{2}\left[\sigma_{j-1}^z \sigma_j^x \prod_{i=j+1}^{M} \sigma_i^x + i\sigma_j^y \prod_{i=j+1}^{M} \sigma_i^x\right] \\ a_j^\dagger &\to \frac{1}{2}\left[\sigma_{j-1}^z \sigma_j^x \prod_{i=j+1}^{M} \sigma_i^x - i\sigma_j^y \prod_{i=j+1}^{M} \sigma_i^x\right]. \end{aligned} \tag{3.2}$$

However, the annihilation and creation operators still act on $\mathcal{O}(N)$ qubits each. One notable strength of the parity transformation, especially for small problems, is that two of the qubits store only redundant information. The last qubit stores the parity of the electron number of the system, which is known in advance. If the qubits are numbered correctly one of the other qubits will store the parity of the total spin, which is also known. Thus, those two qubits can be removed from the circuit entirely[110].

3.2.1.3. Bravyi-Kitaev

The Bravyi-Kitaev transformation, also known as the binary tree mapping, represents a compromise between the parity and Jordan-Wigner methods[108, 109]. Neither the occupation information nor the parity information is stored entirely locally. Instead, each qubit stores the parity information of $\mathcal{O}(\log N)$ spin-orbitals. Even-indexed qubits store only the occupation number of the matching orbital, exactly as in the Jordan-Wigner case. However, a qubit with an odd index j stores the parity of the occupation numbers of k consecutive orbitals, ending with the

jth, where k is the least power of two in the binary expansion of $j+1$. For example, qubit 5 would store the parity of two orbitals (orbitals 4 and 5), since $5+1=6=4+2$; and qubit 7 would store the parity of eight orbitals (orbitals 0 through 7), since $7+1=8$. Just as in the parity transformation, two of the qubits store the electron number and spin information only, and could be removed[110].

$$a_j \to \prod_i \tag{3.3}$$

The primary benefit of this particular encoding is that it provides an asymptotic logarithmic reduction in the number of operations required to implement a fermionic operator. In the context of quantum chemistry simulations, even in the pre-asymptotic regime the Bravyi-Kitaev transformation has been shown to be at least as efficient as Jordan-Wigner with even further reductions in gate counts realized by limited circuit optimization [111].

3.2.2. Results: Memory Scaling

The mapping step is another potential bottleneck involving both classical and quantum hardware. Leveraging the parameter count given in Eq. 2.21, upper bounds can be estimated for the total number of operations required to prepare the UCCSD ansatz for a single iteration of the VQE algorithm. For the Bravi-Kitaev transformation the expected number of gates scales as $O(N^2\eta^2)$, while for the Jordan-Wigner transformation the scaling is $O(N^3\eta^2)$ [$O(N^2\eta^2)$ if non-local gates are available][74].

The first task is the identification of the limiting resource metric. While circuit depth and compilation time are also affected to a lesser extent[112], memory was found to be the biggest bottleneck for fermion-to-qubit mapping. The runtime should be roughly linear in the number of terms in the Hamiltonian. Unfortunately, for large chemistry Hamiltonians with O(n^4) terms, that's still expensive.

Figure 3-6 demonstrates that the T1700's 8 gigabytes of memory was entirely exhausted with a mere 20 atoms (requiring as few as 38 qubits). The scaling curves corresponding to the three transformation algorithms are visually indistinguishable on the scale of the plot. Recalling that the driver RAM requirements scaled as $1.0\times10^{-7}n^{3.5}$, here memory scaling prefactors were found to be much larger, of order 1.0×10^{-4}. According to this memory scaling, the 64 GB available on a single Uno node would restrict treatment to systems with fewer than 40 atoms. The 600 GB available on a single Summit node enables only up to 75 nuclei.

Strategies for mitigation: The steep memory scaling of the the above-mentioned mapping schemes can perhaps be mitigated by parallelization of the classical algorithm. Parallelization of the quantum component has already been accomplished[112]. Alternatively, there is a special routine that maps the operators directly to qubits without going through the fermion Hamiltonian as an intermediate[113]. Thus it is likely that the expense associated with mapping can be eliminated, or at least significantly reduced with sufficient software engineering.

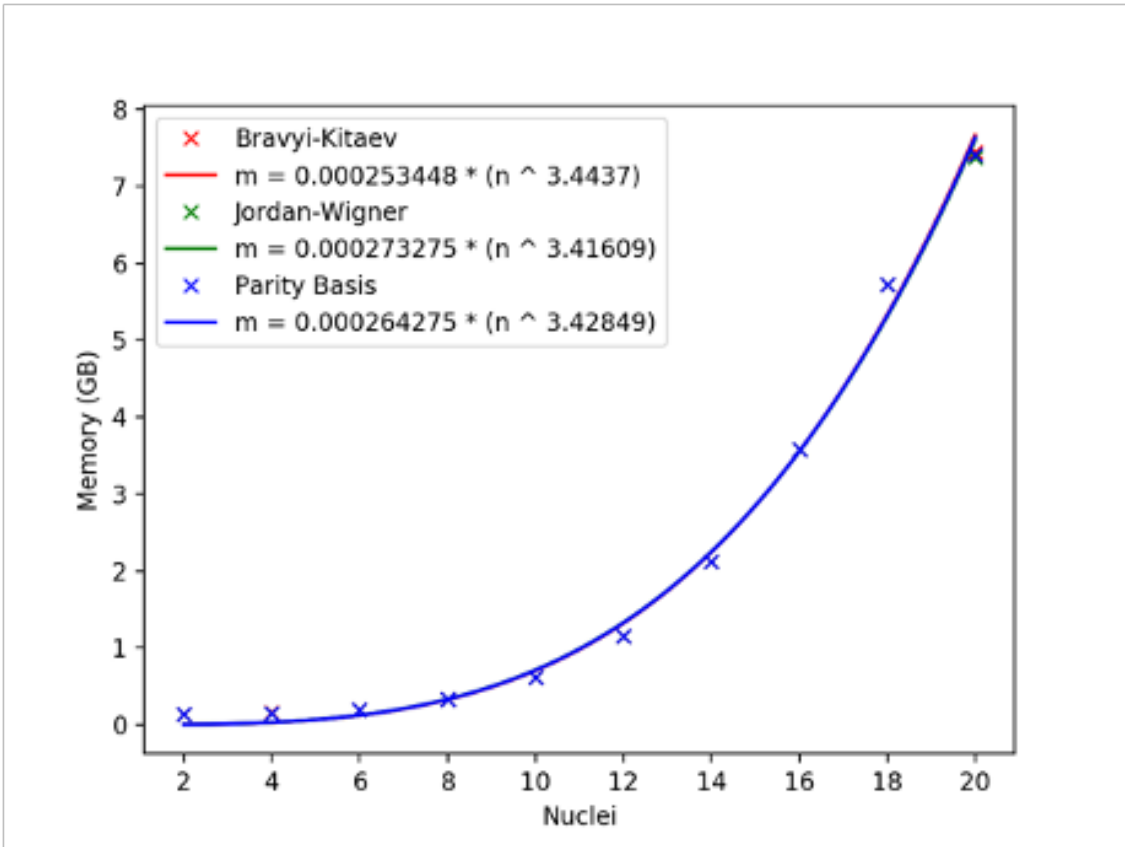


Figure 3-6. Scaling of the RAM requirement with increasing deuterium-cluster size while employing the three fermion-to-qubit transformation algorithms.

3.3. EXCITATION OPERATORS

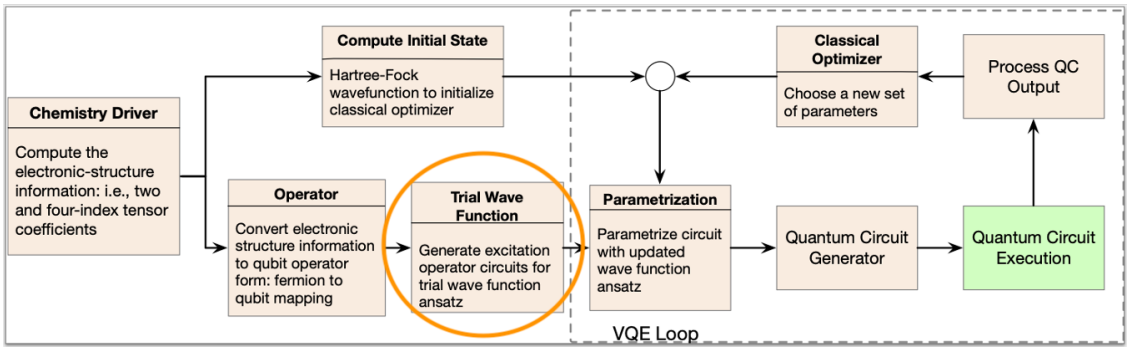


Figure 3-7. Excitation operator circuits corresponding to the wave function ansatz must be generated before execution of the VQE optimization loop.

3.3.1. Definitions

After the fermionic operators are mapped to qubit form, the excitation operator circuits must be prepared. Here we chose to employ the UCCSD ansatz approximated with a single Trotter step. As discussed in Section 2.4, the UCC method represents a heirarchy of systematically-improvable ansatze that can be prepared using quantum circuits with resource requirements scaling polynomially. For applications in quantum chemistry, the UCCSD approximation represents an good compromise of accuracy and expense, with an expected parameter count of $O(N^2\eta^2)$.

3.3.2. Results: Time scaling

For large circuits, it was determined that gereration of excitation operator circuits is strongly time-limited, while the associated memory requirements are entirely negligible. Initially it was unclear to what extent this process is dependant upon the choice of fermion-to-qubit mapping. Figure 3-8 considers the scaling of the computation time with increasing system size. It was found that, independent of the mapping, the time to build the circuits scales octically with the number of nuclei. Enabling qubit reduction in the parity basis offers a notable advantage, but the reduced scaling is still quite large, at $n^{7.2}$. Hence, this step requires over a CPU-week even for relatively modest problems (54 qubits).

Strategies for Mitigation: The native method for constructing excitation operator circuits in Qiskit is very slow. The associated n^7–n^8 time scaling with the the number of nuclei makes it the most time-consuming step of the VQE initialization (i.e., pre-loop) workflow. Fortunately we discovered that this classical cost can be reduced significantly by using sparse data structures and otherwise optimizing the circuit construction.

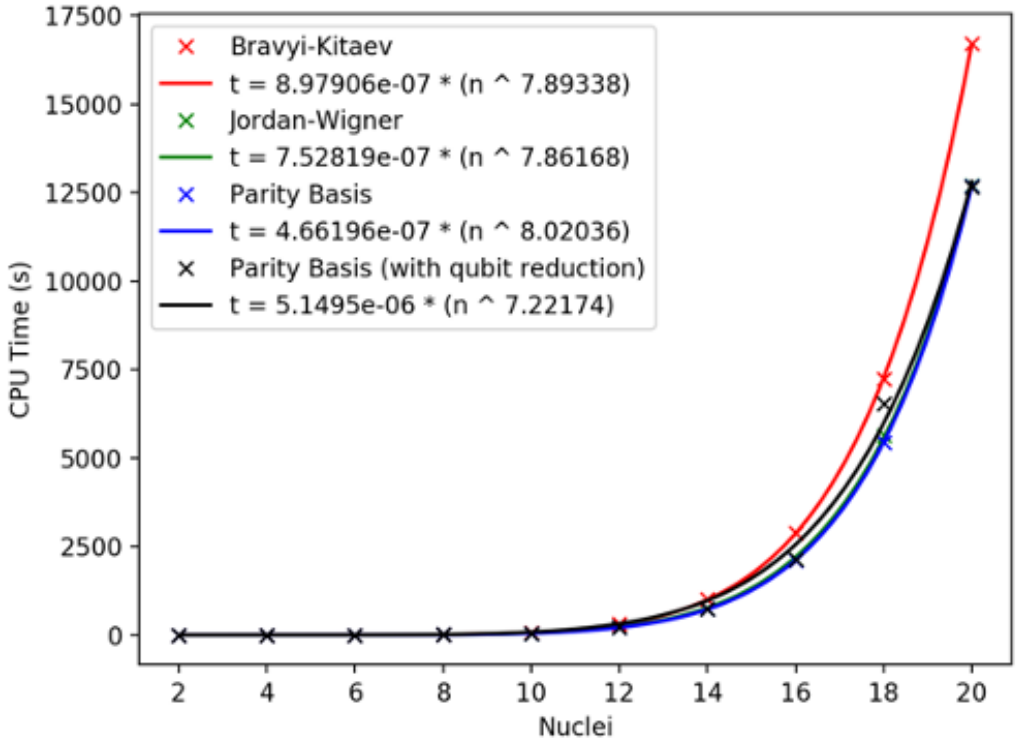


Figure 3-8. Temporal requirements for constructing excitation operator circuits using various fermion-to-qubit mappings.

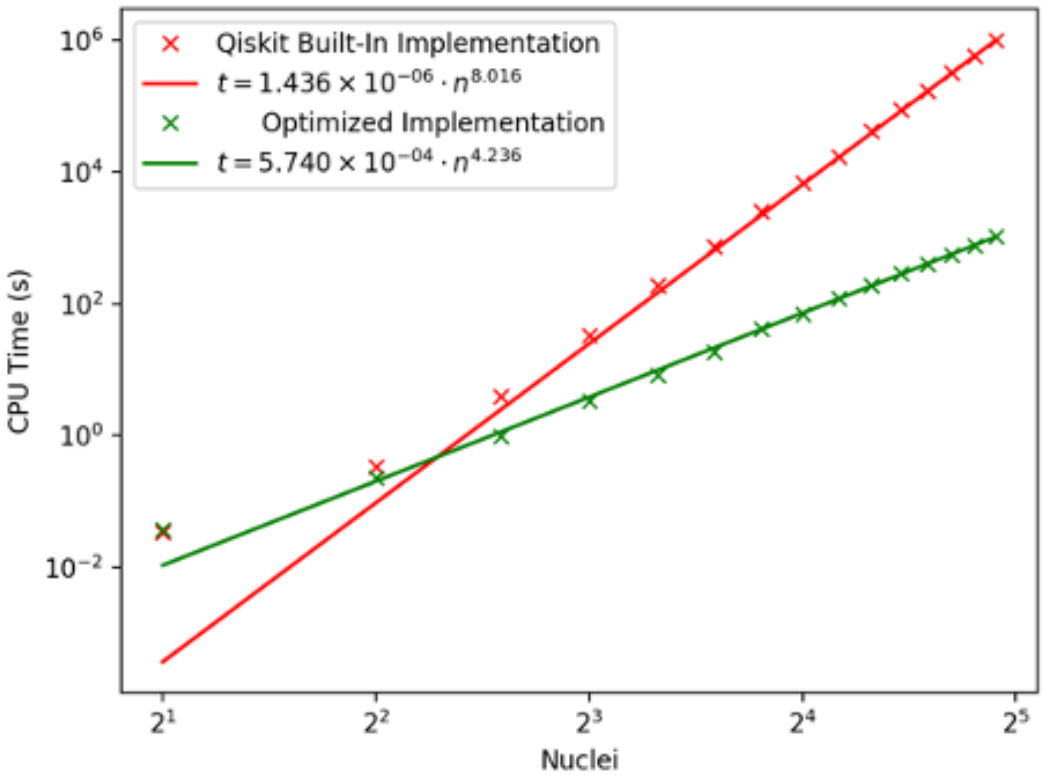


Figure 3-9. Temporal requirements for constructing excitation operator circuits using the native Qiskit implementation and our newly optimized implementation.

Figure 3-9 compares the time scaling with system size for Qiskit's native implementation and our newly optimized implementation. While our implementation is shown to have a significantly larger prefactor (by $\sim$ 200%), the associated polynomial scaling is dramatically lower, at $n^{4.2}$. For the right-most data points shown, corresponding to H_{32}, the acceleration was so great that the wall-clock time to develop and run the optimized version was shorter than the time to run the VQE initialization steps in the native Qiskit version.

3.4. CIRCUIT PARAMETERIZATION

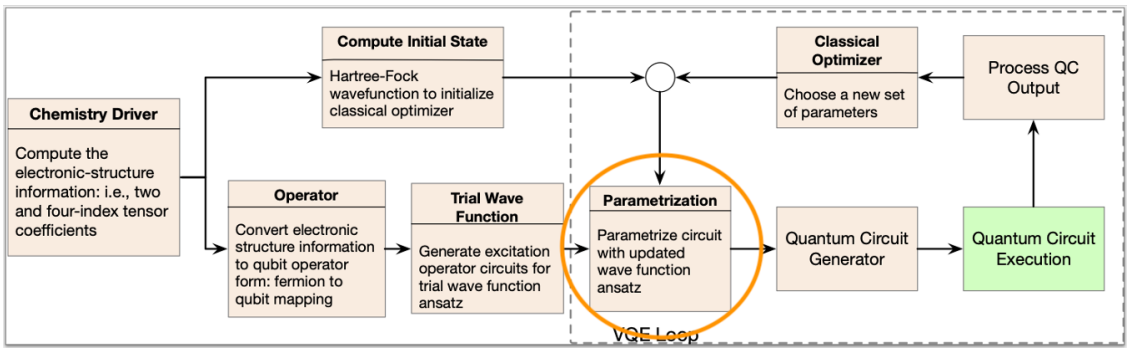


Figure 3-10. The excitation operator circuit parameters must be updated as part of the VQE optimization loop.

3.4.1. Definitions

At each iteration of the VQE loop, a new vector of excitation amplitude parameters will determine the angles of the single-qubit rotation gates constituting the exponentiated Pauli strings. These parameters are tuned variationally by a classical optimization routine based on energy measurements of the states those parameter values define. The number of parameters thus directly determines the size of the space our optimizer must explore, and so indirectly the number of circuits that must be run before convergence.

Considering both the scaling of the number of UCCSD parameters given in Eq. 2.21 and the upper bounds on the number of operations required to implement a single parameter, it is possible to determine an upper bound for the total number of operations involved in preparing the UCCSD ansatz for single iteration of the VQE algorithm[74]. The Bravi-Kitaev transformation has a gate count scaling as $\tilde{O}(N^2\eta^2)$, while Jordan-Wigner scales as $O(N^3\eta^2)$, or $O(N^2\eta^2)$ when non-local gates are employed.

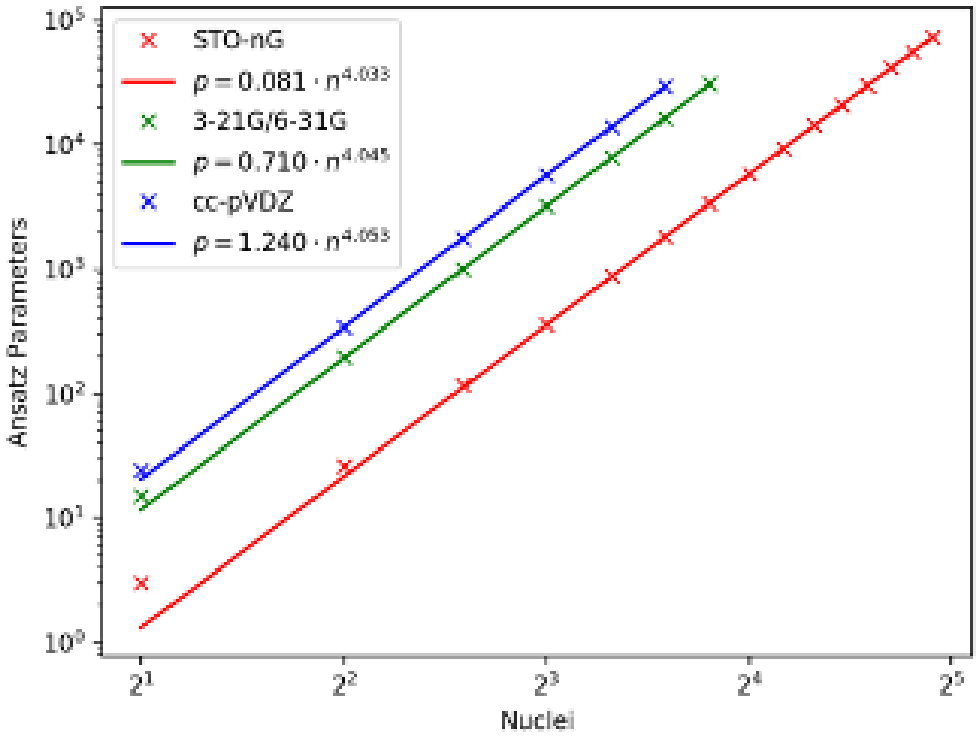


Figure 3-11. Growth of the parameter count with system size when utilizing a variety of small-to-medium basis sets.

3.4.2. Results

3.4.2.1. Parameter-count scaling

Figure 3-11 summarizes our study of the growth of the parameter count with the system size. UCCSD was used in conjunction with basis sets from the MBS, Pople, and Dunning families. In each case, the empirically derived power scaling with parameter count is n^4, which is fully consistent with our estimated scaling of $O(N^2\eta^2)$. Only the prefactors were found to grow significantly with increasing basis size. The maximum possible system size was limited to ~ 10 nuclei while using the cc-pVDZ basis set. We attempted to extend the study to include the cc-pVTZ basis set, but the expense became intractable.

3.4.2.2. Circuit-depth scaling

With increasing parameter counts, circuit depths also grow with system size. Minimum circuit depth requirements were investigated as a function of system size, and the data including fits to power curves are plotted in Figure 3-12. In a comparison between the various mappings, Bravyi-Kitaev produced a significantly better scaling than Jordan-Wigner or the parity basis

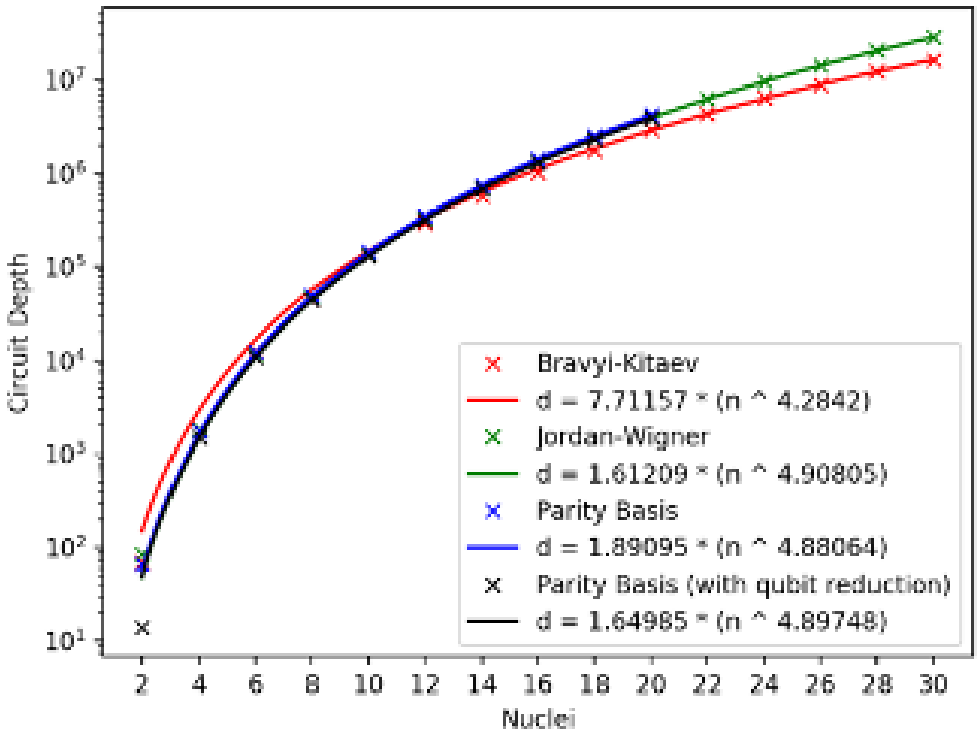


Figure 3-12. Growth of the minimum circuit depth with the system size for parameterization when using a variety of fermion-to-qubit mappings.

methods, but all had circuit depths that scaled with system size in the range $n^{4.3}$–$n^{4.9}$. These represent the characteristic scaling with circuit depth for the VQE process on the whole.

3.4.2.3. Time scaling

Next we investigated the time scaling of the parameterization step with increasing system size, as shown in Figure 3-13. Basis sets from the MBS, Pople, and Dunning families were used in tests, all of which yielded power scalings in the range $n^{4.4}$–$n^{4.6}$. Thanks to our newly implemented code for constructing excitation operator circuits in time scaling as $n^{4.2}$ rather than $n^{8.0}$, parameterization represents the overall rate-limiting step for the VQE protocol.

The memory scaling of the parameterization with increasing system size was the last resource requirement left to check. Figure 3-14 Again, several basis sets were employed to determine that the power scaling of the memory required goes as $n^{4.1}$–$n^{4.3}$. The largest cc-pVDZ basis set yielded a significantly smaller power ($n^{3.8}$), but we disregard this result as too few data points were generated to produce a good fit. With a memory scaling power greater than 4, the parameterization step is also the memory-limiting step.

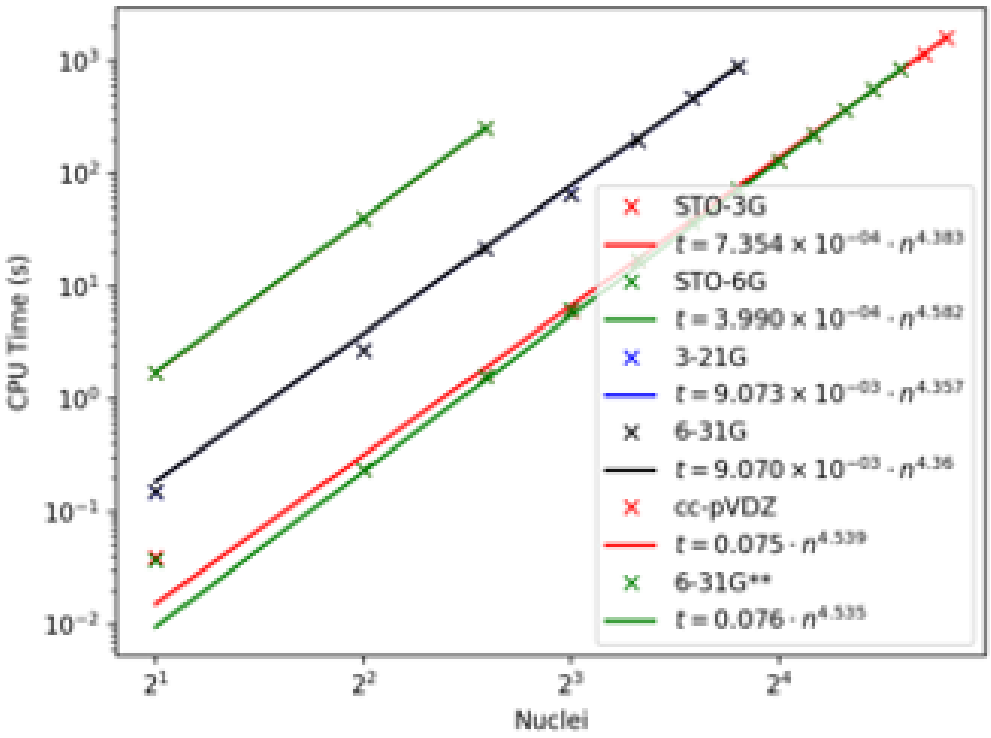


Figure 3-13. Growth of the parameterization time requirements with the system size when using a variety of basis sets.

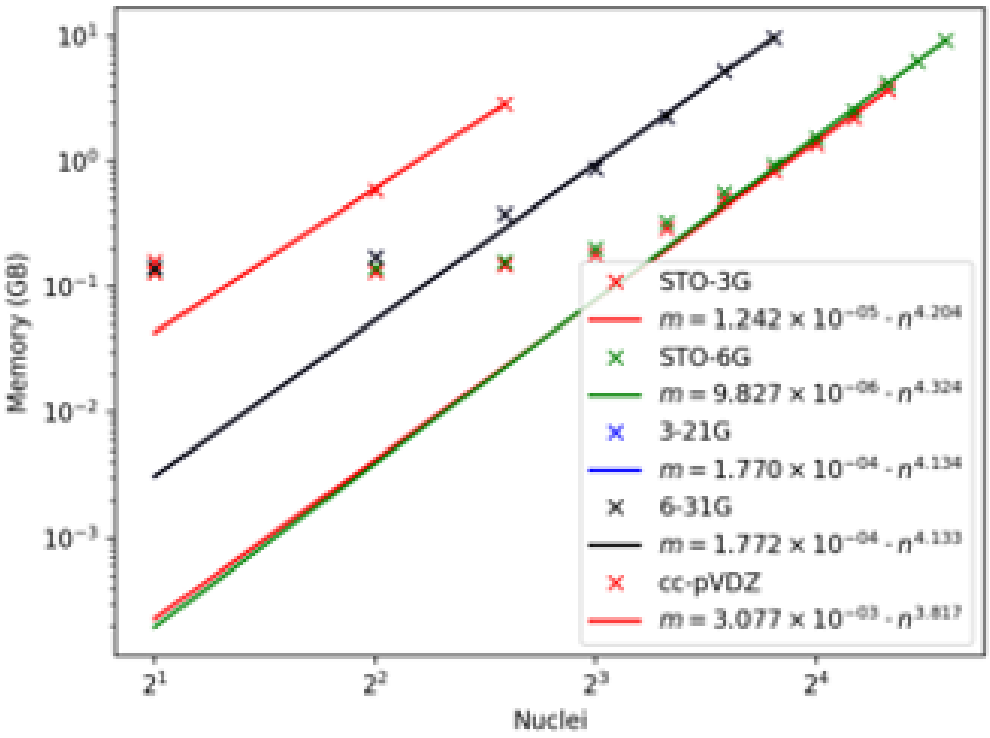


Figure 3-14. Growth of the memory requirements with the system size for parameterization when using a variety of basis sets.

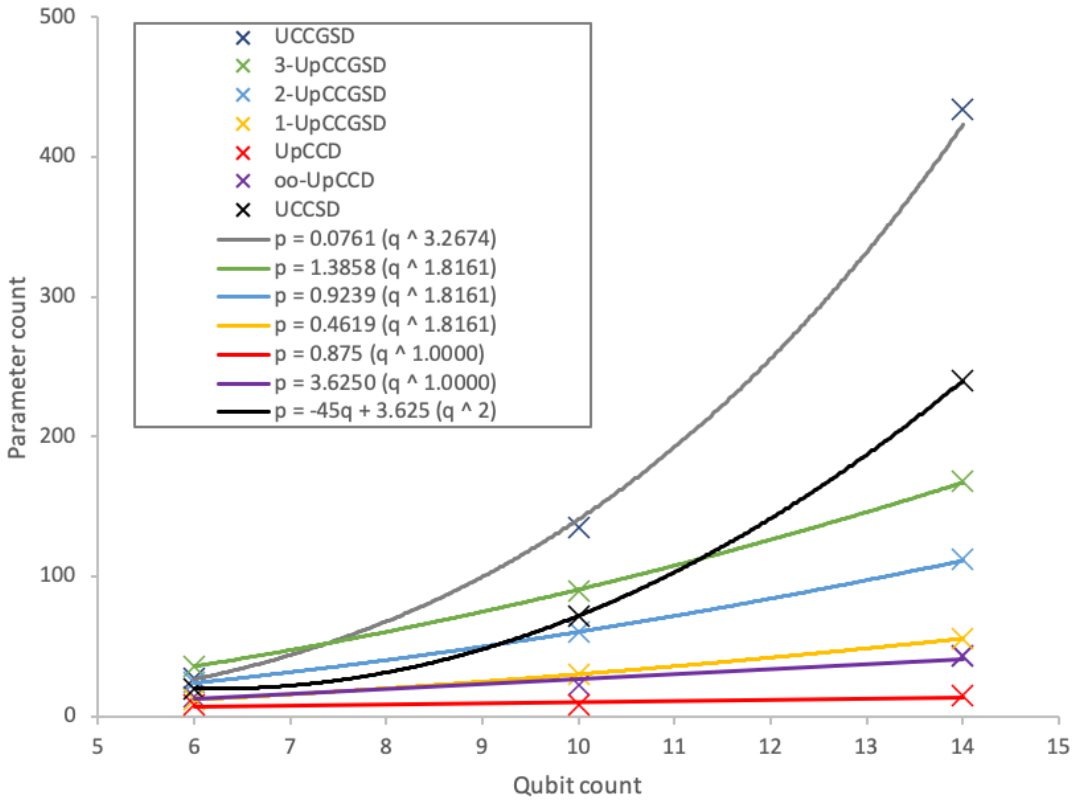


Figure 3-15. Growth of the parameter count with the qubit count when using a variety of electronic structure ansätze. The three qubit count levels correspond to the H_4, H_2O, and N_2 systems, with each described by a MBS.

3.4.2.4. Strategies for mitigation

One way to reduce the parameter count, thereby reducing the required circuit depth, time, and memory, is to adopt a different wave-function ansätz (see Section 2.4). Novel VQE-UCC methods such as the k-UpCCGSD method of Lee et al.[77] look at first very promising because they reduces the gate count and circuit depth of UCCSD from $\sim n^4$ and $\sim n^3$ to $\sim n^2$ and $\sim n$, respectively. However, the parameterization memory requirements also need to be considered. While Table 2-1 described UCC resource requirements for gate count and circuit depth, the scaling of the parameter count is not well established.

Figure 3-15 documents the relationship between the parameter count and qubit count for a variety of ansätze. Unfortunately UCCSD is the only UCC method currently implemented in Qiskit, so the underlying data was taken from recent studies by Lee et al.[77], Hickman et al.[114], and Mizukami et al.[76] Qubit counts of 6, 10, and 14 correspond to the systems H_4, H_2O, and N_2 modeled with a STO-3G basis set.

For the standard UCCSD approach, the number of parameters was observed to scale with qubit count as a quadratic polynomial. The UCCGSD method was shown to be comparatively

expensive, fitting to a $q^{3.3}$ power function, whereas the k-UpCCGSD approximations required slightly fewer parameters than UCCSD, scaling with power $q^{1.8}$. For the UpCCD and oo-UpCCD methods parameter counts scaled linearly with parameter count. It is worth noting that for oo-UpCCD the additional oo step only adds a factor of 4–5 to the linear scaling. Moving toward the treatment of larger systems, the alternative UpCCD and oo-UpCCD methods should be considered. For all other methods considered here, the parameter count proliferates rapidly such that the resulting parameter counts cause the parameterization step to become both time-limiting and memory-limiting.

3.5. CLASSICAL OPTIMIZATION

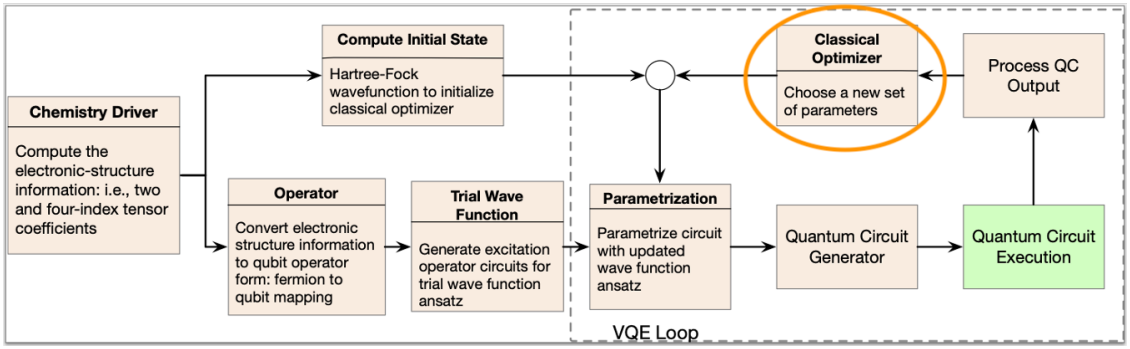


Figure 3-16. The classical optimization determines a new set of parameters to re-parameterize excitation operator circuits during the VQE optimization loop.

The classical optimization step follows measurement of the quantum circuit, where the parameters $\vec{\theta}$ of the quantum state are updated using a classical nonlinear optimization routine. The choice of classical optimizer determines how many different circuits must be executed to achieve a desired accuracy. As the system size increases, the number of adjustable classical parameters grows quickly. Thus, the optimizer of choice must perform well in high-dimensional parameter spaces. Quality of the optimizer is critical for obtaining high-accuracy results and rapid convergence.

The first VQE paper used a derivative-free optimization algorithm known as Nelder-Mead. McClean et al.[8] compared four numerical algorithms side-by-side in a VQE study of the H_2 system. Among GLCCLUSTER, LGO, MULTMIN, and Nelder-Mead, they recommended LGO. Romero et al.[74] compared four more numerical algorithms applied to the H_4 system, including COBYLA, L-BFGS-B, Nelder-Mead, and Powell. For the compatively challenging case of H_4, the gradient algorithms COBYLA and L-BFGS-B had significantly fewer convergence problems as compared with derivative-free methods. Romero et al. went at step further to consider analytic gradient methods and concluded that they often converged more rapidly. Numerical methods were found to perform poorly in the presence of noise, often requiring orders-of-magnitude more accuracy to achieve the same accuracy. Since both analytical and numerical gradients require a number of circuits proportional to the dimension of the space at every time step[74], gradient-free algorithms are favored unless their convergence is vastly slower.

3.5.1. Algorithm definitions

We consider the performance of three numerical gradient optimization methods, namely conjugate gradient descent, SPSA, and COBYLA, and one numerical derivative-free method called NEWUOA. NEWUOA was not considered in previous studies, so its inclusion is a novel aspect of this work.

The conjugate gradient algorithm takes single steps in a sequence of conjugate directions, which together form a complete basis for the parameter space[115][116]. The direction of each step is chosen by taking the gradient numerically (requiring a number of evaluations proportional to the dimension of the space), then using conjugate Gram-Schmidt to find the component of the gradient conjugate to the previous search directions. The length of each step is computed by a line search along the chosen direction[117]. After each step, the process repeats, finding another numerical gradient, performing conjugate Gram-Schmidt, and performing another line search. The number of iterations required, generally, is equal to the dimension of the space. We use the implementation built into the SciPy package[118], which performs the Gram-Schmidt conjugation by the Polak-Ribière method[119].

Simultaneous perturbation stochastic approximation (SPSA) is a variant of the Kiefer-Wolfowitz stochastic approximation (SA) algorithm. SA functions almost identically to simple gradient descent, but uses a finite-difference-like numerical gradient that performs better in the presence of noisy measurements. While SA requires taking numerical gradients by finite differences, SPSA avoids that cost, allowing it to perform better in high-dimensional spaces. Specifically, it uses a different noise-resilient estimator for the gradient, known as the simultaneous perturbation estimate. The simultaneous perturbation estimate chooses a random direction in which to take a partial derivative at each step, rather than taking derivatives along every basis vector. A step is then taken, either in the random direction chosen or the opposite direction depending on the sign of the partial derivative, and a new random direction is chosen and partial derivative is taken. We use the implementation built into the Qiskit package[120], which is based on that used by Kandala et al.[91]

Constrained optimization by linear approximation (COBYLA) approximates the energy as a linear function of the parameters, interpolating between points at the corners of a simplex in parameter space[121]. It uses this linear approximation to choose a point in the simplex to replace with a new point, running a circuit with the new parameters and updating the linear approximation accordingly. The implementation we use is built into the SciPy package[118].

New unconstrained optimization algorithm (NEWUOA) is one among a variety of optimization schemes invented by M. J. D. Powell, the inventor of COBYLA [122]. NEWUOA uses quadratic approximations instead of linear ones, and typically performs much better in high-dimensional spaces[123]. Unfortunately, unlike the other algorithms we considered, it lacks

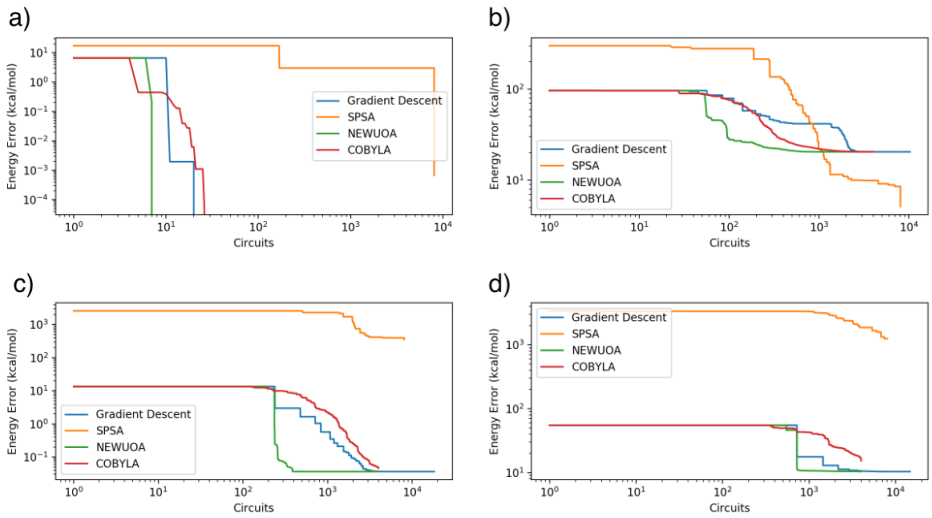


Figure 3-17. Performance of various optimization algorithms for the determination of the ground-state wave functions for H_2 (a), H_4 (b), H_6 (c), and H_8 (d). Wave functions were parameterized according to the UCCSD ansätz with the STO-3G basis set.

a convenient integration with SciPy or Qiskit. We ported Powell's original Fortran implementation to Python/NumPy.

3.5.2. Results

Tests of the four optimizers were performed on the H_2, H_4, H_6, and H_8 systems. We are not aware of prior optimizer comparisons studying deuterium clusters larger than H_4. As shown in Figure 3-17, the gradient descent, NEWUOA, and COBYLA methods each demonstrated similar convergence patterns, with NEWUOA performing best for all four clusters. Using H_8 as a specific example, NEWUOA "catches" at fewer iterations than the other methods, matching the exact UCCSD energy to four significant figures after only 722 iterations. After 1463 iterations the optimization converged, matching the exact UCCSD energy to five significant figures (a discrepancy of only 0.002%). For every system SPSA converged to a different state than the other three optimizers.

4. DISCUSSION AND CONCLUSIONS

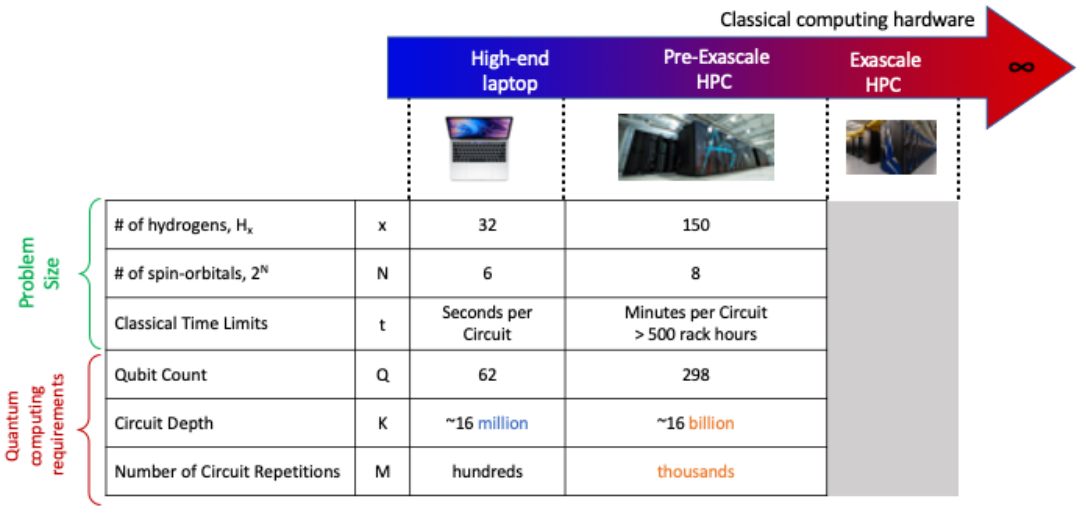


Figure 4-1. Ballpark estimations of hardware limitations for electronic structure calculations using a minimum basis set and hybrid VQE simulation approaches

VQE is viewed by the community as a promising application on near-term quantum devices thanks to the relatively low qubit-count requirement. Our study, which unpacked the VQE workflow and considered individual resource requirements for each step, found circuit parameterization to be the among the most expensive steps, in terms of memory, wall time, and circuit depth. In Figure 4-1 we collect our best estimates of the quantum resource requirements associated with VQE. In the following we consider each VQE resource requirement limitation in turn and assess the implications.

Circuit depth In terms of *quantum* resources, the quartic to quintic scaling of the circuit depth during parameterization means that for all but the smallest problems, error mitigation, detection, or correction will be necessary. At present, the thought of circuit depths in the billions is incredibly daunting, but let us assume for the sake of argument that this is surmountable.

Qubit count Honeywell recently announced a record-breaking quantum computer with a quantum volume* of 64, accompanied by an optimistic declaration of 10-fold increases in quantum volume per year going forward [124]. Assuming Honeywell's ambitious projection holds true, quantum hardware with 1 million qubits may be available within a decade. Assuming this lofty goal is achieved and quantum resources cease to be the bottleneck, at what point will classical resource requirements become limiting?

Time limits The largest *classical* barriers to implementing VQE are the memory and time required to generate circuits from the parameters at each iteration. An iterative $n^{4.4}$ time step, while appearing at first to be more favorable than the hexic scaling of a traditional projection coupled-cluster calculation, is compounded by the enormous number of VQE optimization iterations necessary even for small systems. The VQE iteration counts appearing in Figure 3-17 already exceeded 10,000 for a mere 2^3 deuterium atoms having 16 basis functions. The number of iterations will grow even larger for realistic molecules in large basis sets, totalling perhaps, say, another order of magnitude. Meanwhile projection CCSD typically finishes in 20 iterations on classical hardware. VQE-UCCSD is expected to be slower than its classical analog until the system size overcomes this factor of 1000–10,000 difference in the required iterations. From this we can estimate that the crossover point will occur between 75 and 750 basis functions. For reference, our estimate of the limit of projection coupled-cluster calculations, as performed on massively-parallel classical hardware, was $\sim$ 1000 basis functions. Exclusively considering time requirements, VQE is a worthy candidate to carry the torch beyond the limitations of what is currently feasible on classical hardware (1000 functions) to cover the range of systems (up to $\sim$ 5000 basis functions) targeted by the DOE (see, e.g., Appendix A for a list).

Classical Memory Let us now consider implications of the classical memory scaling in the parameterization step. The STO-3G data in Figure 3-14 fit to a scaling relation of $1.2 \times 10^{-5}\ n^{4.2}$, which is actually steeper than the n^4 memory requirements of traditional projection CCSD. Assuming VQE-UCCSD has access to the full 600GB of RAM available on a single node of ORNL's Summit supercomputer, this corresponds to a maximum number of 128 basis functions. This is comparable to the number of basis functions for which exact diagonalization is possible on a laptop. Furthermore, 128 is only a small fraction of the 1000 basis-function calculations currently possible on the same Summit node using massively-parallel implementations of projection coupled-cluster.

Conclusions Variational quantum eigensolvers, a family of error-resilient algorithms potentially applicable on NISQ hardware, have become a topic of increasing research activity by the quantum simulation community. Whether they will enable scientific advances that were otherwise impossible is yet to be seen. The present LDRD effort came away with a generally negative outlook about the prospects of VQE-UCCSD. Even when operating under the cavalier

*Competing quantum hardware platforms are difficult to compare, so quantum volume was proposed as a universally applicable performance metric. Quantum volume simultaneously considers the number of qubits, connectivity performance, gate set performance, whole-algorithm errors, and compilers and software stack performance [124].

assumption that we will soon have access to unlimited quantum resources, classical memory requirements of the VQE loop limit the short-term applicability to small system sizes. Systems of this size are already exactly solvable on a capable laptop. While the expense of the limiting parameterization step can be reduced by switching to alternative hybrid ansätze such as UpCCD, this comes at great penalty to the associated energy accuracy. Unless the community can mitigate these classical memory requirements, we recommend prioritizing investment in fully quantum algorithms, such as the quantum phase estimation, over hybrid approaches, such as VQE.

REFERENCES

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[2] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, and R. Wisnieff. Leveraging secondary storage to simulate deep 54-qubit sycamore circuits. *arXiv preprint arXiv:1910.09534*, 2019.

[3] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[4] P. J. Ollitrault, A. Baiardi, M. Reiher, and I. Tavernelli. Hardware efficient quantum algorithms for vibrational structure calculations. *arXiv preprint; arXiv:2003.12578v1*.

[5] F. T. Cerasoli, K. Sherbert, J. Sławińska, and M. Buongiorno Nardelli1. Quantum computation of silicon electronic band structure. *arXiv preprint, arXiv:2006.03807v1*.

[6] H.-H. Lu, N. Klco, J. M. Lukens, T. D. Morris, A. Bansal, A. Ekström, G. Hagen, T. Papenbrock, A. M. Weiner, M. J. Savage, and P. Lougovski. Simulations of subatomic many-body physics on a quantum frequency processor. *Phys. Rev. A*, 100:012320, Jul 2019. doi: 10.1103/PhysRevA.100.012320. URL `https://link.aps.org/doi/10.1103/PhysRevA.100.012320`.

[7] R. C. Clay III, M. P. Desjarlais, and L. Shulenburger. Deuterium hugoniot: Pitfalls of thermodynamic sampling beyond density functional theory. *Physical Review B*, 100(7): 075103, 2019.

[8] 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. *Nat. Commun.*, 5:4213, 2014. doi: 10.1038/ncomms5213.

[9] J. McLean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18:023023, 2016. doi: 1367-2630/18/2/023023.

[10] A. Szabo and N. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover, Mineola, first edition, 1996.

[11] Hartree-fock on a superconducting qubit quantum computer. *Science*, 369(6507): 1084–1089, 2020. doi: 10.1126/science.abb9811. URL `https://science.sciencemag.org/content/369/6507/1084`.

[12]

[13] J. M. Hutson and C. R. Le Sueur. molscat: A program for non-reactive quantum scattering calculations on atomic and molecular collisions. *Computer Physics Communications*, 241: 9 – 18, 2019. ISSN 0010-4655. doi: 10.1016/j.cpc.2019.02.014. URL http://www.sciencedirect.com/science/article/pii/S0010465519300669.

[14] Free energy calculations: Applications to chemical and biochemical phenomena. *Chemical Reviews*, 93:2395–2417, 1993. doi: 10.1021/cr00023a004. URL https://pubs.acs.org/doi/10.1021/cr00023a004#.

[15] B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus. A new basis set exchange: An open, up-to-date resource for the molecular sciences community. *J. Chem. Inf. Model.*, 59:4814–4820, 2019. doi: 10.1021/acs.jcim.9b00725.

[16] N. M. Tubman, C. D. Freeman, D. S. Levine, D. Hait, M. Head-Gordon, and K. B. Whaley. Modern approaches to exact diagonalization and selected configuration interaction with the adaptive sampling ci method. *J. Chem. Theory Comput.*, 16:2139–2159, 2020. doi: 10.1021/acs.jctc.8b00536.

[17] R. J. Bartlett. Many-body perturbation theory and coupled cluster theory for electron correlation in molecules. *Annual Review of Physical Chemistry*, 32:359, 1981. doi: 10.1146/annurev.pc.32.100181.002043.

[18] S. R. Langhoff and E. R. Davidson. *Int. J. Quantum Chem.*, 8:61, 1974.

[19] L. Meissner. *Chem. Phys. Lett.*, 146:204, 1988.

[20] S. Krebs. A review on the derivation of the spin-restricted hartree-fock (rhf) self-consistent field (scf) equations for open-shell systems. description of different methods to handle the off-diagonal lagrangian multipliers coupling closed and open shells. *Computer Physics Communications*, 116:127–277, 1999.

[21] J. Whitten. Coulombic potential energy integrals and approximations. *Journal of Chemical Physics*, 58(10):4496–4501, 1973. doi: 10.1063/1.1679012.

[22] M. Häser and R. Ahlrichs. Improvements on the direct scf method. *Journal of Computational Chemistry*, 10(1):104–111, 1988.

[23] A. Izmaylov, S. Gustavo E, and F. Michael J. Efficient evaluation of short-range hartree-fock exchange in large molecules and periodic systems. *Journal of Chemical Physics*, 125:104103, 2006.

[24] S. Maurer, D. Lambrecht, D. Flaig, and C. Ochsenfeld. Distance-dependent schwarz-based integral estimates for two-electron integrals: Reliable tightness vs. rigorous upper bounds. *Journal of Chemical Physics*, 136:144107, 2012.

[25] D. Hollman, H. Schaefer, and E. Valeev. A tight distance-dependent estimator for screening three-center coulomb integrals over gaussian basis functions. *Journal of Chemical Physics*, 142:154106, 2015.

[26] M. Beuerle, J. Kussmann, and C. Ochsenfeld. Screening methods for linear-scaling short-range hybrid calculations on cpu and gpu architectures. *Journal of Chemical Physics*, 146:144108, 2017.

[27] T. Thompson and C. Ochsenfeld. Integral partition bounds for fast and effective screening of general one-, two-, and many-electron integrals. *Journal of Chemical Physics*, 150: 044101, 2019.

[28] S. Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71 (4):1085, 1999.

[29] M. Challacombe. A simplified density matrix minimization for linear scaling self-consistent field theory. *Journal of Chemical Physics*, 110:2332, 1999.

[30] M. Challacombe. A general parallel sparse-blocked matrix multiply for linear scaling scf theory. *Computer Physics Communications*, 128:93–107, 2000.

[31] E. Baerends, D. Ellis, and P. Ros. Self-consistent molecular hartree-fock-slater calculations i. the computational procedure. *Chemical Physics*, 2:41–51, 1973.

[32] H. Huang, C. D. Sherrill, and E. Chow. Techniques for high-performance construction of fock matrices. *Journal of Chemical Physics*, 152(2):024122, 2020.

[33] X. Liu, A. Patel, and E. Chow. A new scalable parallel algorithm for fock matrix construction. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, pages 902–914, 2014. doi: 10.1109/IPDPS.2014.97.

[34] E. Chow, X. Liu, M. Smelyanskiy, and J. Hammond. Parallel scalability of hartree–fock calculations. *Journal of Chemical Physics*, 142(10):104103, 2015.

[35] K. Yasuda. Two-electron integral evaluation on the graphics processor unit. *Journal of Computational Chemistry*, 29:334–342, 2008.

[36] I. Ufimtsev and T. Martinez. Quantum chemistry on graphical processing units. 1. strategies for two-electron integral evaluation. *Journal of Chemical Theory and Computation*, 4:222–231, 2008.

[37] A. Asadchev, V. Allada, J. Felder, B. Bode, M. Gordon, and T. Windus. Uncontracted rys quadrature implementation of up to g functions on graphical processing units. *Journal of Chemical Theory and Computation*, 6(3):696–704, 2010. doi: 10.1021/ct9005079.

[38] Y. Miao and K. Merz. Acceleration of electron repulsion integral evaluation on graphics processing units via use of recurrence relations. *journal of Chemical Theory and Computation*, 9(2):965–976, 2013.

[39] Y. Miao and K. Merz. Acceleration of high angular momentum electron repulsion integrals on graphics processing units. *journal of Chemical Theory and Computation*, 11(4): 1449–1462, 2015.

[40] J. Kalinowski, F. Wennmohs, and F. Neese. Arbitrary angular momentum electron repulsion integrals with graphical processing units: Application to the resolution of identity hartree–fock method. *Journal of Chemical Theory and Computation*, 13(7):3160–3170, 2017. doi: 10.1021/acs.jctc.7b00030.

[41] I. Ufimtsev and T. Martinez. Quantum chemistry on graphical processing units. 2. direct self-consistent-field implementation. *Journal of Chemical Theory and Computation*, 5: 1004—1015, 2009.

[42] G. Shi, V. Kindratenko, I. Ufimtsev, and T. Martinez. Direct self-consistent field computations on gpu clusters. In *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, pages 1–8, 2010. doi: 10.1109/IPDPS.2010.5470478.

[43] A. Asadchev and M. Gordon. New multithreaded hybrid cpu/gpu approach to hartree-fock. *Journal of Chemical Theory and Computation*, 8(11):4166–4176, 2012.

[44] J. W. Mullinax, E. Maradzike, L. N. Koulias, M. Mostafanejad, E. Epifanovsky, and E. A. Gidofalvi, Gergely DePrince III. Heterogeneous cpu + gpu algorithm for variational two-electron reduced-density matrix-driven complete active-space self-consistent field theory. *Journal of Chemical Theory and Computation*, 15(11):6164–6178, 2019.

[45] N. Luehr, U. Ivan S, and M. Todd J. Dynamic precision for electron repulsion integral evaluation on graphical processing units (gpus). *Journal of Chemical Theory and Computation*, 7:949–954, 2011. doi: 10.1021/ct100701w.

[46] E. Chow, X. Liu, S. Misra, M. Dukhan, M. Smelyanskiy, J. Hammond, Y. Du, X.-K. Liao, and P. Dubey. Scaling up hartree-fock calculations on tianhe-2. *Int. J. High Perform. Comput. Appl.*, 30(1):85—102, 2016. doi: 10.1177/1094342015592960. URL `https://doi.org/10.1177/1094342015592960`.

[47] V. Mironov, Y. Alexeev, K. Keipert, M. D'mello, A. Moskovsky, and M. S. Gordon. An efficient mpi/openmp parallelization of the hartree-fock method for the second generation of intel xeon phitm processor. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450351140. doi: 10.1145/3126908.3126956. URL `https://doi.org/10.1145/3126908.3126956`.

[48] M. Gordon, D. Fedorov, S. Pruitt, and L. Slipchenko. Fragmentation methods: A route to accurate calculations on large systems. *Chemical Reviews*, 112:632–672, 2012.

[49] A. Collins and R. Bettens. Energy-based molecular fragmentation methods. *Chemical Reviews*, 115:5607–5642, 2015.

[50] K. Raghavachari and A. Saha. Accurate composite and fragment-based quantum chemical methods for large molecules. *Chemical Reviews*, 115:5643–5677, 2015.

[51] J. M. Herbert. Fantasy versus reality in fragment-based quantum chemistry. *Journal of Chemical Physics*, 151:170901, 2019.

[52] R. J. Bartlett and M. Musiał. *Rev. Mod. Phys.*, 79:291, 2007.

[53] Perspective: Multireference coupled cluster theories of dynamical electron correlation. *J. Chem. Phys.*

[54] I. Shavitt and R. J. Bartlett. Cambridge, 2009.

[55] C. Peng, J. A. Calvin, F. Pavošević, J. Zhang, and E. F. Valeev. Massively parallel implementation of explicitly correlated coupled-cluster singles and doubles using tiled array framework. *J. Phys. Chem. A*, 120:10231–10244, 2016. doi: 10.1021/acs.jpca.6b10150.

[56] J. A. V. E. F. Peng, C.; Calvin. Coupled-cluster singles, doubles and perturbative triples with density fitting approximation for massively parallel heterogeneous platforms. *Int. J. Quantum Chem.*, 119:e25894, 2019. doi: 10.1002/qua.25894.

[57] E. Deumens, V. F. Lotrich, A. Perera, M. J. Ponton, B. A. Sanders, and R. J. Bartlett. Software design of aces iii with the super instruction architecture. *Wiley Interdiscip. Rev.:Comput. Mol. Sci.*, 1:895–901, 2011. doi: 10.1002/wcms.77.

[58] E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel. A massively parallel tensor contraction framework for coupled-cluster computations. *J. Parallel Distrib. Comput.*, 74:3176, 2014. doi: 10.1016/j.jpdc.2014.06.002.

[59] T. Shen, Z. Zhu, I. Y. Zhang, and M. Scheffler. Massive-parallel implementation of the resolution-of-identity coupled-cluster approaches in the numeric atom-centered orbital framework for molecular systems. *J. Chem. Theory Comput.*, 15:4721, 2019. doi: 10.1021/acs.jctc.8b01294.

[60] R. Kobayashi and A. P. Rendell. A direct coupled cluster algorithm for massively parallel computers. *Chem. Phys. Lett.*, 265:1–11, 1997. doi: 10.1016/S0009-2614(96)01387-5.

[61] V. M. Anisimov, G. H. Bauer, K. Chadalavada, R. M. Olson, J. W. Glenski, W. T. C. Kramer, E. Aprà, and K. Kowalski. Optimization of the coupled cluster implementation in nwchem on petascale parallel architectures. *J. Chem. Theory Comput.*, 10:4307–4316, 2014. doi: 10.1021/ct500404c.

[62] L. Gyevi-Nagy, M. Kállay, and P. R. Nagy. Integral-direct and parallel implementation of the ccsd(t) method: Algorithmic developments and large-scale applications. *Journal of Chemical Theory and Computation*, 16(1):366–384, 2020. doi: 10.1021/acs.jctc.9b00957. URL `https://doi.org/10.1021/acs.jctc.9b00957`.

[63] B. S. Fales, E. R. Curtis, K. G. Johnson, D. Lahana, S. Seritan, Y. Wang, H. Weir, T. J. Martínez, and E. G. Hohenstein. Performance of coupled-cluster singles and doubles on modern stream processing architectures. *J. Chem. Theory Comput.* doi: 10.1021/acs.jctc.0c00336. URL `https://doi.org/10.1021/acs.jctc.0c00336`. *in press.*

[64] H. Primas. A generalized perturbation theory for quantum mechanical many-body problems. Helvetica Physica Acta*, 34(4):331–351, 1961.*

[65] H. Primas. Generalized perturbation theory in operator form. Reviews of Modern Physics*, 35(3):710–712, 1963.*

[66] D. Mukherjee, R. Moitra, and A. Mukhopadhyay. Correlation problem in open-shell atoms and molecules. Molecular Physics*, 30(6), 1975.*

[67] W. Kutzelnigg. Quantum chemistry in fock space. i. the universal wave and energy operators. Journal of Chemical Physics*, 77(6):3081–3097, 1982.*

[68] S. Pal. Use of a unitary wavefunction in the calculation of static electronic properties. Theoretica Chimica Acta*, 66(3–4):207–215, 1984.*

[69] M. Hoffmann and J. Simons. A unitary multiconfigurational coupled-cluster method: Theory and applications. The Journal of chemical physics*, 88(2):993–1002, 1988.*

[70] R. Bartlett, S. Kucharski, and J. Noga. Alternative coupled-cluster ansätze ii. the unitary coupled-cluster method. Chemical Physics Letters*, 155(1):133–140, 1989.*

[71] A. Taube and R. Bartlett. New perspectives on unitary coupled-cluster theory. International Journal of Quantum Chemistry*, 106(15):3393–3401, 2006.*

[72] G. Harsha, T. Shiozaki, and G. Scuseria. On the difference between variational and unitary coupled cluster theories. Journal of Chemical Physics*, 148:044107, 2018. doi: 10.1063/1.5011033.*

[73] S. Gulania and J. D. Whitfield. Limitations of hartree-fock with quantum resources. arXiv preprint arXiv:2007.09806*, 2020.*

[74] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. Quantum Science and Technology*, 4(1):014008, 2018.*

[75] M. Nooijen and V. Lotrich. Brueckner based generalized coupled cluster theory: Implicit inclusion of higher excitation effects. J. Chem. Phys.*, 113:4549, 2000. doi: 10.1063/1.1288912.*

[76] W. Mizukami, K. Mitarai, Y. Nakagawa, T. Yamamoto, T. Yan, and Y.-y. Ohnishi. Orbital optimized unitary coupled cluster theory for quantum computer. Phys. Rev. Research*, 2: 033421, 2020.*

[77] J. Lee, W. Huggins, M. Head-Gordon, and K. Whaley. Generalized unitary coupled cluster wave functions for quantum computation. Journal of Chemical Theory and Computation*, 15:311–324, 2019. doi: 10.1021/acs.jctc8b01004.*

[78] M. Suzuki. Decomposition formulas of exponential operators and lie exponentials with some applications to qunatum mechanics and statistical physics. J. Math. Phys.*, 26: 601–612, 1985.*

[79] R. D. Somma. A trotter-suzuki approximation for lie groups with applications to hamiltonian simulation. J. Math. Phys.*, 57:062202, 2016.*

[80] R. Babbush, J. McClean, D. Wecker, A. Aspuru-Guzik, and N. Wiebe. Chemical basis of trotter-suzuki errors in quantum chemistry simulation. Physical Review A*, 91:022311, 2015.*

[81] *A. Tranter, P. Love, F. Mintert, N. Wiebe, and P. Coveney. Ordering of trotterization: Impact on errors in quantum simulation of electronic structure.* Entropy*, 21(12):1218, 2019. doi: 10.3390/e21121218.*

[82] *A. Childs, Y. Su, and M. Tran. A theory of trotter error.* arXiv preprint arXiv:1912.08854*, 2020.*

[83] *H. Grimsley, D. Claudino, S. Economou, E. Barnes, and N. Mayhall. Is the trotterized uccsd ansatz chemically well-defined?* Journal of Chemical Theory and Computation*, 16: 1–6, 2020. doi: 10.1021/acs.jctc.9b01083.*

[84] *P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, S. Filipp, and I. Tavernelli.* Phys. Rev. A*, 98: 022322, 2018.*

[85] *P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis.* Phys. Rev. X*, 6:031007, 2016.*

[86] *H. Grimsley, S. Economou, E. Barnes, and N. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer.* Nature Communications*, 10: 3007, 2019. doi: 10.1038/s41467-019-10988-2.*

[87] *H. Tang, H. Barnes, Edwin Grimsley, N. Mayhall, and S. Economou. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansatze on a quantum processor.* arXiv prepring arXiv:1911.10205*, 2019.*

[88] *F. A. Evangelista, G. K.-L. Chan, and G. E. Scuseria. Exact parameterization of fermionic wave functions via unitary coupled cluster theory.* J. Chem. Phys.*, 151:244112, 2019. doi: 10.1063/1.5133059.*

[89] *Y. Matsuzawa and Y. Kurashige. Jastrow-type decomposition in quantum chemistry for low-depth quantum circuits.* Journal of Chemical Theory and Computation*, 2019.*

[90] *Y. Herasymenko and T. O'Brien. A diagrammatic approach to variational quantum ansatz construction.* arXiv preprint arXiv:1907.08157*, 2019.*

[91] *A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets.* Nature*, 549(7671):242, 2017.*

[92] *L. Rios and N. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations.* Journal of Global Optimization*, 56:1247–1293, 2013. doi: 10.1007/s10898-012-9951-y.*

[93] *J. McClean, S. Boixo, V. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network landscapes.* Nature Communications*, 9:4812, 2018. doi: 10.1038/s41467-018-07090-4.*

[94] *A. Cerezo, M Sone, T. Volkoff, L. Cincio, and P. Coles. Cost-function-dependent barren plateaus in shallow quantum neural networks.* arXiv preprint arXiv:2001.00550*, 2020.*

[95] *S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. Sornborger, and P. Coles. Quantum-assisted quantum compiling.* Quantum*, 3:140, 2019. ISSN 2521-327X. doi: 10.22331/q-2019-05-13-140. URL* `https://doi.org/10.22331/q-2019-05-13-140`*.*

[96] *H. Huang, K. Bharti, and P. Rebentrost. Near-term quantum algorithms for linear systems of equations.* arXiv preprint arXiv:1909.07344*, 2019.*

[97] *E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits.* Quantum*, 3:214, December 2019. ISSN 2521-327X. doi: 10.22331/q-2019-12-09-214. URL* `https://doi.org/10.22331/q-2019-12-09-214`*.*

[98] *S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz.* Algorithms*, 12:34, 2019. doi: 10.3390/a12020034.*

[99] *G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary. Quantum graph neural networks.* arXiv preprint arXiv:1909.12264*, 2019.*

[100] *Microsoft.* Microsoft Quantum Documentation*, 2020 (accessed July 4, 2020. URL* `https://docs.microsoft.com/en-us/quantum/`*.*

[101] *M. Valiev, E. Bylaska, N. Govind, K. Kowalski, T. Straatsma, H. van Dam, D. Wang, J. Nieplocha, E. Apra, T. Windus, and W. de Jong. Nwchem: a comprehensive and scalable open-source solution for large scale molecular simulations.* Comput. Phys. Commun.*, 181:1477, 2010.*

[102] *K. Sugisaki, S. Yamamoto, S. Nakazawa, K. Toyota, K. Sato, D. Shiomi, and T. Takui. Quantum chemistry on quantum computers: A polynomial-time quantum algorithm for constructing the wave functions of open-shell molecules.* J. Phys. Chem. A*, 120: 6459–6466, 2016.*

[103] *K. Sugisaki, S. Nakazawa, K. Toyota, K. Sato, D. Shiomi, and T. Takui. Quantum chemistry on quantum computers: A method for preparation of multiconfigurational wave functions on quantum computers without performing post-hartree–fock calculations.* ACS Cent. Sci.*, 5:167–175, 2019.*

[104] *R. P. Muller. Python quantum chemistry program, version 1.6.0. http://pyquante.sourceforge.net/.*

[105] *Q. Sun.* PySCF GitHub repository. *URL* `https://github.com/sunqm/pyscf`*.*

[106] *Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan. Pyscf: the python-based simulations of chemistry framework.* WIREs Comput. Mol. Sci.*, 8:1340, 2018. doi: 10.1002/wcms.1340.*

[107] P. Jordan and E. Wigner. Über das paulische äquivalenzverbot. Zeitschrift für Physik*, 47 (9-10):631–651, 1928.*

[108] J. T. Seeley, M. J. Richard, and P. J. Love. The bravyi-kitaev transformation for quantum computation of electronic structure. The Journal of chemical physics*, 137(22):224109, 2012.*

[109] S. B. Bravyi and A. Y. Kitaev. Fermionic quantum computation. Annals of Physics*, 298(1): 210–226, 2002.*

[110] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme. Tapering off qubits to simulate fermionic hamiltonians. arXiv preprint arXiv:1701.08213*, 2017.*

[111] A. Tranter, P. Love, F. Mintert, and P. Coveney. A comparison of the bravyi–kitaev and jordan–wigner transformations for the quantum simulation of quantum chemistry. Journal of Chemical Theory and Computation*, 14(11):5617–5630, 2018. doi: 10.1021/acs.jctc.8b00450.*

[112] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer. Improving quantum algorithms for quantum chemistry. Quantum Info. Comput.*, 15:1–21, 2015.*

[113] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Izmaylov. Qubit coupled-cluster method: a systematic approach to quantum chemistry on a quantum computer. J. Chem. Theory Comput.*, 14:6317–6326, 2018.*

[114] Comparison of unitary coupled cluster ansatz methods for the variational quantum eigensolver.

[115] M. R. Hestenes, E. Stiefel, et al. Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.*

[116] J. R. Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[117] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. The computer journal*, 7(2):149–154, 1964.*

[118] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0–fundamental algorithms for scientific computing in python. arXiv preprint arXiv:1907.10121*, 2019.*

[119] E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969.*

[120] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C. Chen, et al. Qiskit: An open-source framework for quantum computing. Accessed on: Mar*, 16, 2019.*

[121] M. J. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.*

[122] M. Powell. The newuoa software for unconstrained optimization without derivatives. Large-Scale Nonlinear Optimization*, pages 255–297, 2006.*

[123] M. J. Powell. Developments of newuoa for minimization without derivatives. IMA journal of numerical analysis*, 28(4):649–664, 2008.*

[124] S. K. Moore. Honeywell claims it has most powerful quantum computer. IEEE Spectrum*. URL* `https://spectrum.ieee.org/tech-talk/computing/hardware/honeywell-claims-it-has-most-powerful-quantum-computer`*.*

[et al.)] *H. A. et al.*

APPENDIX A. LIST OF MOLECULES

Table A-1. A list of small-to-medium-sized molecules of interest to the DOE

	Elements				Basis functions				Electrons		n_o[‡]	n_u[§]			
Molecule	H	C	N	O	STO-3G	6-31G**	cc-pVTZ	cc-pVQZ	N_{tot}[*]	N_{val}[†]		STO-3G	6-31G**	cc-pVTZ	cc-pVQZ
H_2	2				2	10	28	60	2	2	1	1	9	27	59
H_4	4				4	20	56	120	4	4	2	2	18	54	118
H_8	8				8	40	112	240	8	8	4	4	36	108	236
H_2O	2			1	7	25	58	115	10	8	4	3	21	54	111
$(H_2O)_5$	10			5	35	125	290	575	50	40	20	15	105	270	555
CH_4	4	1			9	35	86	175	10	8	4	5	31	82	171
CH_3OH	4	1		1	14	50	116	230	18	14	7	7	43	109	223
C2H5OH	2	6		1	37	115	238	445	46	32	16	21	99	222	429
CO	1			1	10	30	60	110	14	10	5	5	25	55	105
CO_2	1			2	15	45	90	165	22	16	8	7	37	82	157
CH_3NO_2	3	1	1	2	23	75	162	310	32	24	12	11	63	150	298
PETN	8	5	4	12	113	355	742	1395	162	120	60	53	295	682	1335
RDX	6	3	6	6	81	255	534	1005	114	84	42	39	213	492	963
HMX	8	4	8	8	108	340	712	1340	152	112	56	52	284	656	1284
TATB	6	6	6	6	96	300	624	1170	132	96	48	48	252	576	1122
N_2			2		10	30	60	110	14	10	5	5	25	55	105
Aspirin	8	9		4	73	235	502	955	94	68	34	39	201	468	921
Taxol	51	47	1	14	361	1185	2574	4940	452	328	164	197	1021	2410	4776
Benzene	6	6			36	120	264	510	42	30	15	21	105	249	495
Cholesterol	46	27		1	186	650	1484	2920	216	160	80	106	570	1404	2840
Caffeine	10	8	4	2	80	260	560	1070	102	74	37	43	223	523	1033

[*] Total number of electrons
[†] Number of valence electrons
[‡] Number of occupied orbitals
[§] Number of unoccupied orbitals

APPENDIX B. DESCRIPTION OF SOFTWARE STACK

The VQE software stack provides a test environment for benchmarking algorithmic components to determine their hardware requirements. It also enables evaluation of the algorithm's sensitivity to error sources, and of techniques for reducing that sensitivity. The stack, shown as a flow-chart in Figure 1-1, consists of components that implement each step, as follows:

1. *Hartree-Fock*: We use the open-source PySCF software suite as a driver to generate non-relativistic HF integrals. For more information on quantum chemistry drivers, see Section 3.1.

2. *Unitary Coupled Cluster*: We use IBM's Qiskit Aqua package to generate unitary coupled cluster ansätz circuits. For more information on hybrid VQE ansäze, see Section 2.4.

3. *State Preparation*: We use either IBM's Qiskit Terra package or SNL's JaqalPaq package to compile the ansatz state preparation circuit to superconducting or trapped-ion gates.

4. *Measurement*: We use either IBM's Qiskit Aer noiseless statevector emulator or our modified version of Delft's quantumsim noisy process matrix simulator to sample from the measurement distribution of the circuit. For the latter, we use process matrices supplied by SNL's ionsim package for modeling noisy trapped-ion gates.

5. *Classical Optimization*: We use either scipy's implementation of the conjugate gradient or COBYLA optimizers, Qiskit Aqua's implementation of the SPSA optimizer, or our own implementation of M. J. D. Powell's NEWUOA optimizer to update the ansatz parameters. For more information on classical optimizers, see Section 3.5.

Our resource estimations were performed with a version of this stack built primarily on the Qiskit package[et al.)]; that is, using PySCF, Qiskit Aqua, Terra, and Aer, and a variety of optimizers. This version of the stack can be found at `https://gitlab.sandia.gov/tsmetod/marmalade`. It requires Qiskit 0.19 to run, which can be installed via pip. To generate the relevant benchmarking data, go to the `qiskitVQE` subfolder and run `python3 benchmark_hN.py [idx] [stage]`. The `stage` argument specifies which stage is to be benchmarked:

0. Generating HF integrals in PySCF.

1. Generating HF integrals in PySCF and constructing a Qiskit `FermionicOperator` object from them.

2. Generating HF integrals in PySCF, constructing a Qiskit `FermionicOperator` object from them, and mapping them to a qubit operator (see Section 8).
3. Constructing the UCCSD ansatz.
4. Constructing the UCCSD ansatz and building a state preparation circuit with specified parameters.
5. All steps.

The `idx` argument specifies which test case to run, both system size and basis set. The two are combined into a single parameter for convenience when issuing batch jobs on HPC machines to benchmark many different molecules and basis sets at once. Each of basis sets described in Section 2.2 are stepped through. The base system size is H_8 for stage 0 or 1 and H_2 for stage 2–5. Each increment of 7 to `idx` increases the size of the system by another multiple of the system size; so for example `python3 benchmark_hN.py 9 0` would time how long it took to calculate the HF integrals for H_{16} in the 3-21G Pople basis. Total CPU time used at each step is output directly by the script; the Gnu `time` utility was used for benchmarking memory usage.

Qiskit's UCCSD ansatz circuit builder uses an inefficient dense data structure to store the fermionic representations of the excitation operators. As detailed in Section 3.3, we constructed a sparse representation; `sparse_fermionic_operator.py` contains both that data model and a subclass of Qiskit's `UCCSD` object, `OptimizedUCCSD`, which uses it in place of the default dense representation. This new version can be found at `https://gitlab.sandia.gov/tsmetod/marmalade`.

Finally, as we describe in Section 3.5, we ported M. J. D. Powell's NEWUOA optimizer from Fortran to Python; it can be found in `newuoa.py`, and a Qiskit-Aqua-compatible wrapper class over it, called `NEWUOA`, can be found in `q_newuoa.py`.

DISTRIBUTION

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop

Email—Internal [redacted]

Name	Org.	Sandia Email Address
Technical Library	01177	libref@sandia.gov

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.