NExt ApplicationS of Quantum Computing

D4.4: Completion of first software suite (Open source software)

Document Properties

Contract Number	951821
Contractual Deadline	30/09/2022 (M25)
Dissemination Level	Public
Nature	Report
Editors	Arseny Kovyrshin, Data Science and Modeling, Pharmaceutical Sciences, R&D, AstraZeneca, Gothenburg, Sweden Giorgio Silvi, HQS Quantum Simulations GmbH
Authors	Arseny Kovyrshin, Data Science and Modeling, Pharmaceutical Sciences, R&D, AstraZeneca, Gothenburg, Sweden Giorgio Silvi, HQS Quantum Simulations GmbH Bruno Chagas, Goar Sánchez, Venkatesh Kannan, ICHEC
Reviewers	Jan Reiner, HQS Quantum Simulations GmbH Marko Rancic, TotalEnergies
Date	26/09/2022
Keywords	Automatic Fragmentation, Variational Ansatz, VQE, Pre-Born-Oppenheimer, Time Evolution
Status	Final
Release	2.0

This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 951821

History of Changes

Release	Date	Author, Organization	Description of Changes
0.1	29/10/2021	Arseny Kovyrshin AstraZeneca AB R&D, Giorgio Silvi HQS Quantum Simulations GmbH	For review version
1.0	23/11/2021	Arseny Kovyrshin AstraZeneca AB R&D, Giorgio Silvi HQS Quantum Simulations GmbH	Final version including corrections of review findings
1.1	19/09/2022	Arseny Kovyrshin AstraZeneca AB R&D, Giorgio Silvi HQS Quantum Simulations GmbH, Bruno Chagas, ICHEC, Goar Sánchez, ICHEC	For review version
2.0	26/09/2022	Arseny Kovyrshin AstraZeneca AB R&D, Giorgio Silvi HQS Quantum Simulations GmbH, Bruno Chagas, ICHEC, Goar Sánchez, ICHEC	Final version after review

Table of Contents

1 Executive Summary

Here, we present an overview of the quantum computing computational package for quantum chemistry. It is comprised of the following two repositories:

- Automatic fragmentation tool (Chagas & Sánchez, 2022)
- Variational algorithms suite (Kovyrshin & Silvi, 2021)

The first repository contains molecule analysis and fragmentation tools based on graph theory. The second contains variational Hamiltonian ansatze adapted to chemical systems is used for state preparation. Equipped with the variational algorithms for imaginary- and real-time evolution, the package can optimize and propagate in time wave functions for chemical systems. As the Pre-Born-Oppenheimer molecular structure is implemented, one can describe nuclear quantum effects.

In this report, we first give an introduction of the features of the each repository. Aferwards we explain briefly the theory behind the methods. Later, we show use cases for the developed methods on small chemical systems such as lithium hydride and hydrogen molecules. Specifically, the real- and imaginary-time evolutions have been proven to work correctly and efficiently. The Pre-Born-Oppenheimer scheme delivers results in agreement with the reference. Furthermore, it is shown that the variational Hamiltonian ansatz may approximate wave functions more efficiently than traditional variational ansatzes.

For more information about the code and for a documentation we refer the reader to the respective repositories (Chagas & Sánchez, 2022; Kovyrshin & Silvi, 2021).

2 Introduction

The simulation of quantum systems in the last decades achieved huge successes and is discussed as one of the possible applications for quantum computing: Having at hand a decent number of long-term stable logical qubits in a universal quantum computer, one could potentially solve, in polynomial time, problems in quantum chemistry intractable for a classical computer. Quantum chemistry gives access to information about the properties of molecular systems before their synthesis. This is a great advantage as otherwise a purely combinatorial experimental approach is unavoidable for finding molecules with desired properties.

The Noisy Intermediate-Scale Quantum (NISQ) devices have a limited number of qubits. Thus one needs a compact wavefunction parameterization giving a reasonable accuracy for molecular energies. Provided such a parameterization, it is of extreme interest to develop methods exploring chemical dynamics and reactivity on NISQ computers. With help of an automatic fragmentation tool, KraChem, one can split complicated chemical systems into weakly interacted fragments. Thus, only fragments with high correlation are considered in the calculation.

2.1 KraChem Repository

KraChem(Chagas & Sánchez, 2022) is a beta-version Python library designed for working with chemical molecules, fragmentation, based on chemical files input and mathematical tools from graph theory. Therefore, this package has two main functionalities, and they are:

1. Intermolecular Fragmentation: given a chemical compound, where we may have a collection of disconnected molecules we want to find
2. Intramolecular Fragmentation: given a molecule defined by its atoms and connections, we want to brake this molecule into smaller parts, so we have a collection of subsets of atoms.

There are two main packages where we develop our framework for fragmenting molecules, based on our approach of analysing chemical molecules as graphs and certain arrays. In this way, we use the following packages

1. numpy: a library for the python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays:
2. networkx : a python library for studying graphs and networks, where we encode our molecules as graphs and perform several routines and algorithms in our graphs

Moreover, we implement some routines for plotting graphs and certain analysis, as well as packages for creating subsets and manipulating paths and directories, and they are:

3. matplotlib: a plotting library for the python programming language and its numerical mathematics extension numpy, where we have some features for plotting figures
4. itertools: this module implements a number of iterator building blocks inspired by constructs from APL, Haskell, and SML. We use this package for combining parts of a chemical compound or atoms in a molecule into a collection of subsets
5. pathlib: this module offers classes representing filesystem paths with semantics appropriate for different operating systems, where we can access and manipulate files in folders

2.2 VA and BBO Repository

The Github repository provided in (Kovyrshin & Silvi, 2021) collects Python scripts and Jupyter notebooks that allow the user to test different variational algorithms. It contains our custom functions (e.g. VHA ansatz, PBO Hamiltonian) that are built upon Qiskit libraries (Qiskit, 2021):

- qiskit-terra 0.18.2
- qiskit-nature 0.2.0
- qiskit-aer 0.9.0

Additionally, integration with the QLM libraries (Atos, 2016-2022):

- myqlm 1.5.1
- qlmaas 1.5.1
- myqlm-interop 0.0.6

is also provided. This allows the user to execute the Qiskit methods also on QLM quantum processing units (QPUs), both locally and on the remote QLMaaS server. This is especially useful for larger problems where one may need more computational power. In order to be able to run the example scripts, a script to build a custom environment is also provided. It automatically installs all the necessary libraries in a Conda environment.

2.3 QLM interoperability

The code in the repository (Kovyrshin & Silvi, 2021) is mainly written using the Qiskit library. To be able to run the circuits onto QLM quantum processing units (QPUs), we integrated the myqlm-interop library which enables the conversion of Qiskit circuits to QLM circuits (as well as the opposite). Additionally, the library allows wrapping QLM QPUs onto a mock Qiskit‘s quantum instance. This allows for easy and simple integration of QPUs as backends to run Qiskit‘s circuits and methods. Each circuit and observable is wrapped into a QLM job that can be submitted to either the local or the remote QLM QPU.

Unfortunately, this implementation suffers from a large overhead due to the time associated with job submissions, result retrieval, and, possibly, queue waiting times. For this reason, to minimize the overhead, we decided to use the custom plugin framework of MyQLM. In particular, we build a custom junction that is capable of handling multiple circuit runs onto a single job submission. Junctions provide a simple interface to also embed repeated, adaptive, classical computations in the middle of the execution stack. The same custom junction can be uploaded to the QLM server (using the script provided in the repo) to be available from the QLMaaS library when the remote connection is established.

The custom junction is built to handle the various methods tested in the repository as input. It also modifies the function to get the energy evaluation (and gradient) inside the solver passed as input. The modification converts each circuit from Qiskit to MyQLM and takes care of the job submission and retrieval in the QLM framework.

Overall, this Qiskit-QLM integration allows us to choose which type of backend to use, and when combined with the QLMaaS server, enables this code to run for larger problems and molecules, which would not be possible using a simple laptop.

3 Theory

3.1 Variational Hamiltonian ansatz

The Variational Hamiltonian Ansatz (VHA) (Wecker et al., 2015) have been implemented and adapted specifically for the application on a quantum chemical system. Inspired by a Trotterized adiabatic evolution

$$\hat{\mathrm{T}}_{\mathrm{ad}} = \prod_{k=1}^{n} \prod_{\alpha=1}^{N} e^{-\mathbf{i}\frac{\tau}{n}\hat{\mathrm{H}}_0} e^{-\mathbf{i}\frac{\tau}{n}\frac{k}{n}\hat{\mathrm{V}}}, \tag{3.1}$$

the VHA reduces the number of parameters used by grouping similar terms in Hamiltonian together under the same variational parameter

$$\hat{\mathrm{H}} = \sum_{\alpha=1}^{N} \hat{\mathrm{H}}_{\alpha}, \tag{3.2}$$

$$\hat{\mathrm{T}}(\boldsymbol{\theta}) = \prod_{k=1}^{n} \prod_{\alpha=1}^{N} e^{-\mathbf{i}\theta_{\alpha,k}\hat{\mathrm{H}}_{\alpha}}. \tag{3.3}$$

One of the ways to split terms in the electronic Hamiltonian is as follows

$$\hat{\mathrm{H}} = \hat{\mathrm{H}}_{\mathrm{diag}} + \hat{\mathrm{H}}_{\mathrm{hop}} + \hat{\mathrm{H}}_{\mathrm{ex}}. \tag{3.4}$$

Expressing the right-hand side of Eq. (3.4) with the creation, $\hat{\mathrm{a}}_p^{\dagger}$, and the annihilation, $\hat{\mathrm{a}}_q$, operators the diagonal term reads

$$\hat{\mathrm{H}}_{\mathrm{diag}} = \sum_{p} h_{pp}\hat{\mathrm{a}}_p^{\dagger}\hat{\mathrm{a}}_p + \sum_{p,q} h_{pqpq}\hat{\mathrm{a}}_p^{\dagger}\hat{\mathrm{a}}_p\hat{\mathrm{a}}_q^{\dagger}\hat{\mathrm{a}}_q. \tag{3.5}$$

The hopping term

$$\hat{\mathrm{H}}_{\mathrm{hop}} = \sum_{pq} h_{pq}\hat{\mathrm{a}}_p^{\dagger}\hat{\mathrm{a}}_q + \sum_{p,r,q} h_{prrq}\hat{\mathrm{a}}_p^{\dagger}\hat{\mathrm{a}}_q\hat{\mathrm{a}}_r^{\dagger}\hat{\mathrm{a}}_r \tag{3.6}$$

contains normal and correlated components, while

$$\hat{\mathrm{H}}_{\mathrm{ex}} = \sum_{<p,q,r,s>} h_{pqrs}\hat{\mathrm{a}}_p^{\dagger}\hat{\mathrm{a}}_q^{\dagger}\hat{\mathrm{a}}_r\hat{\mathrm{a}}_s \tag{3.7}$$

has all other exchange terms with indexes $< p, q, r, s >$ being distinct.

3.2 Variational time evolution

Approaching chemical dynamics with quantum computing methods is challenging, as one faces the limitations of NISQ devices. The recently appeared variational time evolution algorithms (Yuan et al., 2019) have modest requirements on decoherence time and noise. That motivated us to base our chemical dynamics methods on variational evolution techniques.

3.2.1 Real-time evolution

To evolve in time the closed quantum system represented by the trial state $\Psi(\boldsymbol{C}[t])$ with parameters $\boldsymbol{C}[t]$,

$$e^{-\mathbf{i}\delta t\hat{\mathrm{H}}}\left|\Psi(\boldsymbol{C}[t])\right\rangle \quad \Rightarrow \quad \left|\Psi(\boldsymbol{C}[t+\delta t])\right\rangle$$

one considers time-dependent Schrödinger equation

$$\frac{\partial}{\partial t}\left|\Psi(\boldsymbol{C}[t])\right\rangle = -\mathbf{i}\hat{\mathrm{H}}\left|\Psi(\boldsymbol{C}[t])\right\rangle.$$

Then applying the McLachlan variational principle (Yuan et al., 2019)

$$\delta|| \left(\partial/\partial t + \mathbf{i}\hat{\mathrm{H}}\right) |\Psi(\boldsymbol{C}[t])\rangle \, || = 0.$$

and following the steps in Ref. (Yuan et al., 2019) it is possible to derive the equation for the Variational Real-Time Evolution (VRTE)

$$\sum_j A^{\Re}_{ij} \dot{C}_j = G^{\Im}_i,$$

where the metric tensor, A_{ij}, and derivative term, G_i, have following form

$$\begin{aligned} A_{ij} &= \left(\frac{\partial \langle \Psi(\boldsymbol{C}[t])|}{\partial C_i} \frac{\partial |\Psi(\boldsymbol{C}[t])\rangle}{\partial C_j} \right), \\ G_i &= \left(\langle \Psi(\boldsymbol{C}[t])| \, \hat{\mathrm{H}} \frac{|\Psi(\boldsymbol{C}[t])\rangle}{\partial C_i} \right). \end{aligned} \tag{3.8}$$

Evaluating $A^{\Re}_{ij}$ and $G^{\Im}_i$ on quantum machine, one can update wave function parameters $\boldsymbol{C}[t]$ on the classical machine according to the following rule

$$\boldsymbol{C}[t+\delta t] = \boldsymbol{C}[t] + \delta t \left(A^{\Re}\right)^{-1} G^{\Im}.$$

3.2.2 Imaginary-time evolution

Alternatively, replacing real time with imaginary time $\tau = it$ and imposing constraints on the wavefunction norm, one can consider Wick rotated Schrödinger equation

$$\frac{\partial}{\partial \tau} |\Psi(\boldsymbol{C}[\tau])\rangle = \left[E(\tau) - \hat{\mathrm{H}} \right] |\Psi(\boldsymbol{C}[\tau])\rangle \, ,$$

whereas the McLachlan variational principle has following form

$$\delta|| \left(\partial/\partial \tau + \hat{\mathrm{H}} - E_\tau\right) |\Psi(\boldsymbol{C}[\tau])\rangle \, || = 0.$$

Similarly to the previous case the equation for the Variational Imaginary-Time Evolution (VITE) is derived

$$\sum_j A^{\Re}_{ij} \dot{C}_j = -G^{\Re}_i,$$

where $A^{\Re}_{ij}$ and $G^{\Re}_i$ are evaluated on quantum devices, while $\boldsymbol{C}[\tau]$ is update on the classical machine accordingly

$$e^{-\delta\tau\hat{\mathrm{H}}} |\Psi(\boldsymbol{C}[\tau])\rangle \quad \Rightarrow \quad \boldsymbol{C}[\tau+\delta\tau] = \boldsymbol{C}[\tau] - \delta\tau \left(A^{\Re}\right)^{-1} G^{\Re}.$$

The VITE can be viewed as an alternative to the traditional wavefunction optimization methods, providing advantages for certain cases (McArdle et al., 2019).

3.3 Pre-Born-Oppenheimer molecular structure theory

In solving the Schrödinger equation for chemical systems, one usually considers nuclear and electronic parts separately, reasoning by the large mass difference between nuclei and electrons. Such an assumption, the so-called Born-Oppenheimer (BO) approximation, considerably simplifies the solution of the Schrödinger equation and leads to coupled electronic and nuclear equations. Generally, BO approximation fails when potential energy surfaces for electronic states are too close. Thus it is hard to study such phenomena as the excited-state dynamics, photochemistry, charge transfer, nuclear tunneling, excited vibrational-electronic states *etc.*. We establish our work on the Pre-Born-Oppenheimer (PBO) molecular structure theory (Muolo et al., 2020; Veis et al., 2016; Webb S. P., 2002)

$$\left[-\sum_i^N \frac{1}{2}\nabla^2_i - \sum_A^M \frac{1}{2M_A}\nabla^2_A - \sum_{i,A}^{M,N} \frac{Z_A}{r_{iA}} + \sum_{i<j}^N \frac{1}{r_{ij}} + \sum_{A<B}^M \frac{Z_A Z_B}{R_{AB}} \right] \Psi(\boldsymbol{r},\boldsymbol{R}) = E\Psi(\boldsymbol{r},\boldsymbol{R}),$$

as nuclear quantum effects are incorporated in the calculation without any additional corrections. Although, one has to mention that the *a posteriori* corrections from the non-adiabatic coupling theory (Ollitrault et al., 2020) are available and might deliver acceptable results.

The PBO Hamiltonian in second quantization formalism was implemented based on the electronic Hamiltonian. Specifically, the nuclear kinetic, nuclear-nuclear, as well as nuclear-electronic Coulomb interaction terms have been added, resulting in the following Hamiltonian

$$
\begin{aligned}
\hat{\mathrm{H}} = &-\sum_{ij} \overset{\text{el. kinetic}}{\left[\int \phi_i(r)\frac{1}{2}\nabla^2\phi_j(r)dr\right]} \hat{a}_i^\dagger \hat{a}_j - \sum_{MN} \overset{\text{nuc. kinetic}}{\left[\int \phi_p(M)\frac{1}{2M}\nabla^2\phi_N(R)dR\right]} \hat{a}_M^\dagger \hat{a}_N \\
&+\frac{1}{2}\sum_{ijkl} \overset{\text{el.-el. Coulomb}}{\left[\int \phi_i(r_1)\phi_k(r_2)\frac{1}{|r_1-r_2|}\phi_l(r_2)\phi_j(r_1)dr_1dr_2\right]} \hat{a}_i^\dagger \hat{a}_k^\dagger \hat{a}_l \hat{a}_j \\
&+\frac{1}{2}\sum_{MNPQ} \overset{\text{nuc.-nuc. Coulomb}}{\left[\int \psi_M(R_1)\psi_P(R_2)\frac{Z_1Z_2}{|R_1-R_2|}\psi_Q(R_2)\psi_N(R_1)dR_1dR_2\right]} \hat{a}_M^\dagger \hat{a}_P^\dagger \hat{a}_Q \hat{a}_N \\
&-\sum_{ijPQ} \overset{\text{el.-nuc. Coulomb}}{\left[\int \phi_i(r)\psi_P(R)\frac{Z}{|R-r|}\psi_Q(R)\phi_j(r_1)drdR\right]} \hat{a}_i^\dagger \hat{a}_P^\dagger \hat{a}_Q \hat{a}_j,
\end{aligned}
$$

where the regular and capital letters were used as indices for electrons and nuclei correspondingly.

4 Discussions and Conclusions

To benchmark the VRTE, VITE, and PBO Hamiltonian implementations, the H_2 molecule at equilibrium distance, 0.735 Å, was chosen as a test system. The TwoLocal hardware efficient ansatz (Kandala et al., 2017) was chosen for all calculations using an ideal state-vector simulation with the Jordan-Wigner transformation for fermionic Hamiltonians. The full entangling pattern for Cz gates was interleaved by Ry and Rz gates, repeated 3 times. The setups for calculations similar to those mentioned in Subsection 4.1 and 4.2 are available in the neoh2vqe.py, h2evo.py, neoh2evo.py python scripts.

4.1 Pre-Born-Oppenheimer molecular structure

The PBO Hamiltonian implementation has been tested in the Variational Quantum Eigensolver (VQE) calculation. For the reference we performed the Nuclear Electronic Orbitals Complete Active Space Correlation Interaction (NEOCASCI) (Webb S. P., 2002) calculation in 6-31G for electrons and DZSNB (Webb S. P., 2002) for nuclei basis sets. To limit the number of utilized qubits, the active space was composed of 2 electrons on 2 orbitals and 2 orbitals for 2 hydrogen nuclei (8 spin orbitals in total). The corresponding integrals for nuclei and electrons were utilized in the VQE calculation. In table 1 one can find the results of VQE energy minimizations employing solely PBO Hamiltonian as well as those with constraints on the number of particles, $\hat{\mathrm{N}}$, and the spin projection, $\hat{\mathrm{S}}_z$.

***Table 1**: The energies of H_2 molecule employing solely PBO Hamiltonian as well as with constraints on number of particles, $\hat{N}$, and spin projection, $\hat{S}_z$.*

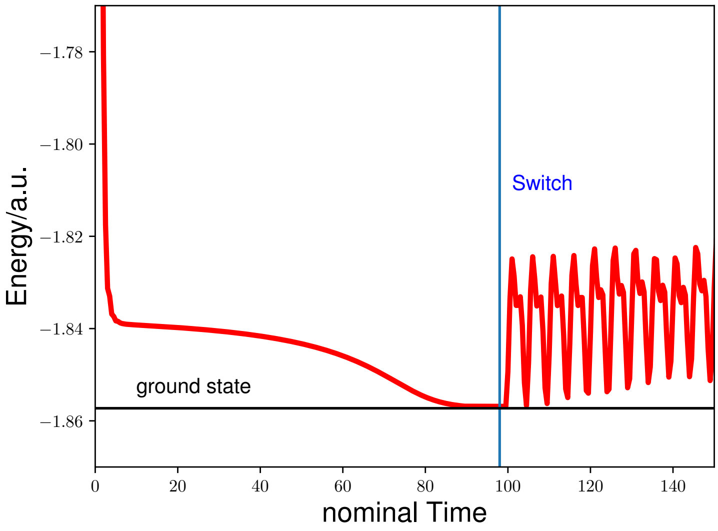
method	constraints	Energy/a.u.
Reference	-	-1.0537650432
TwoLocal	-	-1.0537606126
TwoLocal	$\hat{\mathrm{N}}$	-1.0537647634
TwoLocal	$\hat{\mathrm{N}}$, $\hat{\mathrm{S}}_z$	-1.0537650258

4.2 Variational time evolution

We tested the variational evolution methods on the electronic Hamiltonian with integrals taken from a Hartree-Fock calculation with STO-3G basis set. Initially, the VITE simulation started with the time step equal to 0.1 (the imaginary and real time are given in a.u. here). After 80 iterations, the total energy decreased close to the ground state energy, see Figure 1, and at iteration 100, the simulation was switched to VRTE with the time step equal to 0.05. One can immediately see at that point the start of the energy oscillation. It is worth noting that until the end of VITE the norm of the update vector was constantly decreasing to zero. Contrary, for VRTE the norm becomes reasonably large and keeps its value between 2.5 and 3.3 during the simulation.

The oscillation has the same repetitive pattern slightly changing at the end due to numerical noise. One can consider that as excited states exploration in the vicinity of the ground state. Currently, we are working on performing the same experiment but on the PBO Hamiltonian and introducing orbital relaxation upon changing atomic orbitals centers.

Figure 1: The variational imaginary-time and real-time evolution on H_2 at equilibrium distance.



4.3 Variational Quantum Algorithms

One can also benchmark the capabilities of various variational algorithms on the determination of the ground state and energy for various molecules. We provide Jupyter notebooks that encompass 4 different methods to find the ground state:

- VQE (Peruzzo et al., 2014) with Unitary Coupled Clusters Singles and Doubles (UCCSD) ansatz (native to Qiskit)
- Adaptive Variational Quantum Eigensolver (AdaptVQE) (Grimsley et al., 2019) with UCCSD ansatz (native to Qiskit)
- VQE with VHA ansatz (Section 3.1) (Wecker et al., 2015)
- VITE (Section 3.2.2)(McArdle et al., 2019) with TwoLocal ansatz

List of Acronyms

Term	Definition
VRTE	Variational Real-Time Evolution
VITE	Variational Imaginary-Time Evolution
NISQ	Noisy Intermediate-Scale Quantum
VHA	Variational Hamiltonian Ansatz
BO	Born-Oppenheimer
PBO	Pre-Born-Oppenheimer
VQE	Variational Quantum Eigensolver
AdaptVQE	Adaptive Variational Quantum Eigensolver
NEOCASCI	Nuclear Electronic Orbitals Complete Active Space Correlation Interaction
UCCSD	Unitary Coupled Clusters Singles and Doubles
STO-3G	minimal Slater-Type Orbital basis with 3 Gaussian primitives
6-31G	split-valence double-zeta basis set with 6 Gaussian primitives
DZSNB	Split-valence Double-Zeta Nuclear Basis set
Li-H	lithium hydride
H_2	hydrogen molecule

Table 2: Acronyms and Abbreviations

List of Figures

List of Tables

Bibliography

Atos. (2016-2022). Myqlm library [https://myqlm.github.io/index.html].

Chagas, B., & Sánchez, G. (2022). Github repository [https://github.com/NEASQC/qfrag].

Grimsley, H. R., Economou, S. E., Barnes, E., & Mayhall, N. J. (2019). An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, *10*, 1–9.

Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J., & Gambetta, J. (2017). Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, *249*, 242.

Kovyrshin, A., & Silvi, G. (2021). Github repository [https://github.com/NEASQC/Variationals_algorithms].

McArdle, S., Jones, T., Endo, S., Li, Y., Benjamin, S., & Yuan, X. (2019). Variational ansatz-based quantum simulation of imaginary time evolution. *Npj Quantum Inf.*, *5*, 75.

Muolo, A., Baiardi, A., Feldmann, R., & Reiher, M. (2020). Nuclear-electronic all-particle density matrix renormalization group. *The Journal of Chemical Physics*, *152*(20), 204103.

Ollitrault, P. J., Mazzola, G., & Tavernelli, I. (2020). Nonadiabatic molecular quantum dynamics with quantum computers. *Phys. Rev. Lett.*, *125*, 260511.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., & O'brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, *5*, 1–7.

Qiskit. (2021). Qiskit: An open-source framework for quantum computing [http://www.qiskit.org]. https://doi.org/10.5281/zenodo.2562110

Veis, L., Visnak, J., Nishizawa, H., Nakai, H., & Pittner, J. (2016). Quantum chemistry beyond born–oppenheimer approximation on a quantum computer: A simulated phase estimation study. *Int. J. Quantum. Chem.*, *116*, 1328–1336.

Webb S. P., H.-S. S., Iordanov T. (2002). Multiconfigurational nuclear-electronic orbital approach: Incorporation of nuclear quantum effects in electronic structure calculations. *J. Chem. Phys.*, *117*, 4106–4118.

Wecker, D., Hastings, M. B., & Troyer, M. (2015). Progress towards practical quantum variational algorithms. *Phys. Rev. A*, *92*, 042303.

Yuan, X., Endo, S., Zhao, Q., Li, Y., & Benjamin, S. C. (2019). Theory of variational quantum simulation. *Quantum*, *3*, 191.