

Exploring molecular excited states using Qiskit software stack



Cambridge Quantum Computing

Apr 13, 2019 · 8 min read

By David Muñoz Ramo, Silas Dilkes, Seyon Sivarajah and Ross Duncan

Cambridge Quantum Computing Ltd.

CAMBRIDGE, UK — Among the many groundbreaking uses of quantum computers, calculating the properties of molecules is one of the most promising. At Cambridge Quantum Computing (CQC), we have been working hard to make this possibility a reality. In this article we will explain how we used IBM's Qiskit software stack in combination with our own proprietary $t|ket\rangle$ compiler, to compute the excited states of a simple molecule, lithium hydride (LiH).

Molecular Excited States

A fundamental discovery of quantum mechanics is that light rays — which we typically think of as behaving like waves — can also be described in terms of discrete units called photons. Each photon carries a specific quantity of energy, which is determined by the frequency (colour) of its light. Here, we are using the word “light” in a loose sense: the *visible light we see* makes up only a small part of the electromagnetic spectrum, which also includes X-rays, infrared radiation, microwaves, and radio waves.

When a beam of light is incident on a molecule, the molecule responds in a manner depending on the energy of photons in the beam. For example, microwaves may cause the molecule to rotate, while infrared light may cause the molecule to vibrate. Photons of visible (or ultraviolet) light can be absorbed by the electrons orbiting the molecule's atoms, causing them to move from a low energy orbit to a higher energy one. This change in the *electronic configuration* of the molecule leaves the molecule in an *excited state*. (See Figure 1.) Properties of electronic excited states are important in chemistry

and condensed matter physics. In particular, electronic excited states are a key component of many chemical reactions and physical processes such as fluorescence emissions. By understanding the properties of electronic excited states, we can predict reaction rates and reaction pathways, and thereby better understand the chemistry of molecules.

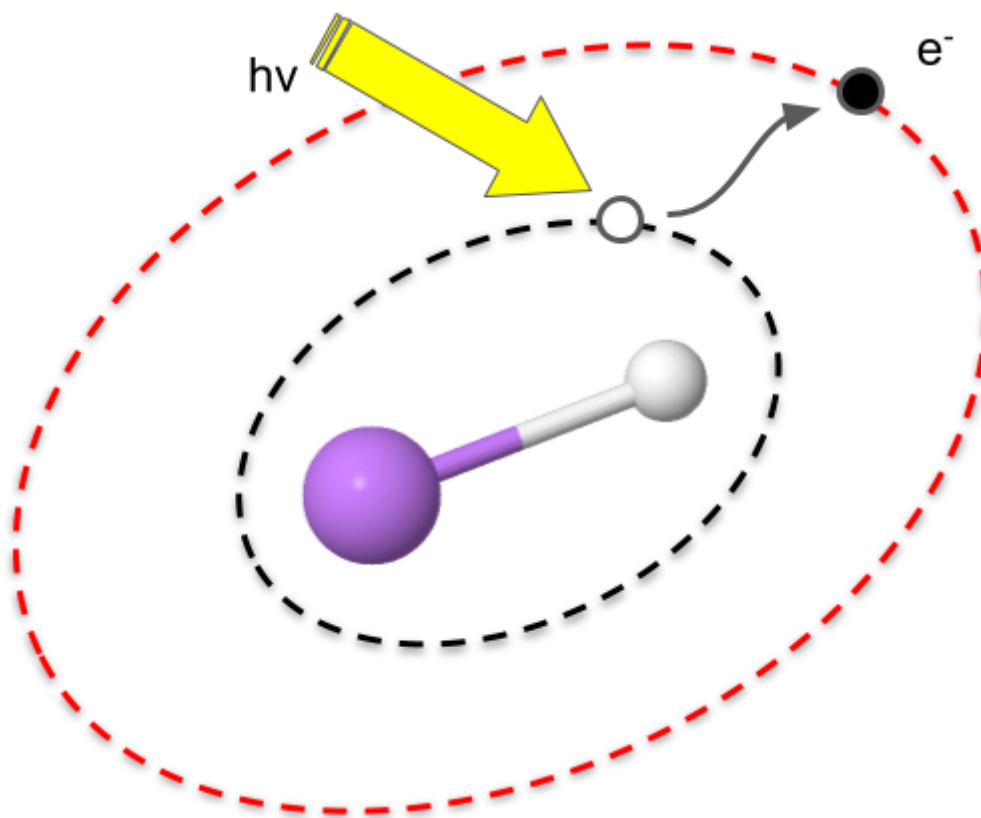


Figure 1. Upon irradiation with a quantum of light, an electron in a low-lying orbital is promoted to a higher energy orbital, changing the electronic configuration of the LiH molecule, and leaving it in an excited state.

Determining properties of excited states, such as the energy required to promote an electron from a low-energy orbital to an excited state, is difficult in general. One reason for this is that the electrons of an excited state can be highly entangled (quantum correlated). To accurately model excited states then, we require a representation of the electronic configuration that accurately captures that entanglement. In general, that representation requires a great deal of computational resources to store. In some cases, the representation could require computational resources that exceed the capabilities of the most advanced classical supercomputers. For many molecules — especially those with a large number of electrons — accurately calculating the energy of the excited states is impossible.

This is where quantum computers can help. Since they are quantum systems themselves, quantum computers are inherently capable of handling entanglement,

which means that they can provide a workable solution to the problem of simulating excited states.

Calculating the excited states of LiH with Qiskit and $t|\text{ket}\rangle$

How might this actually work? Let's look at computing the excited states of LiH. We've provided code for the numerical results below in this [Jupyter notebook tutorial](#).

For illustration purposes we'll restrict our attention to the excited states that result from promoting one electron from the ground state to a higher-energy orbital. Since this molecule has four electrons we'll consider a quantum superposition of all of the resulting excitations.

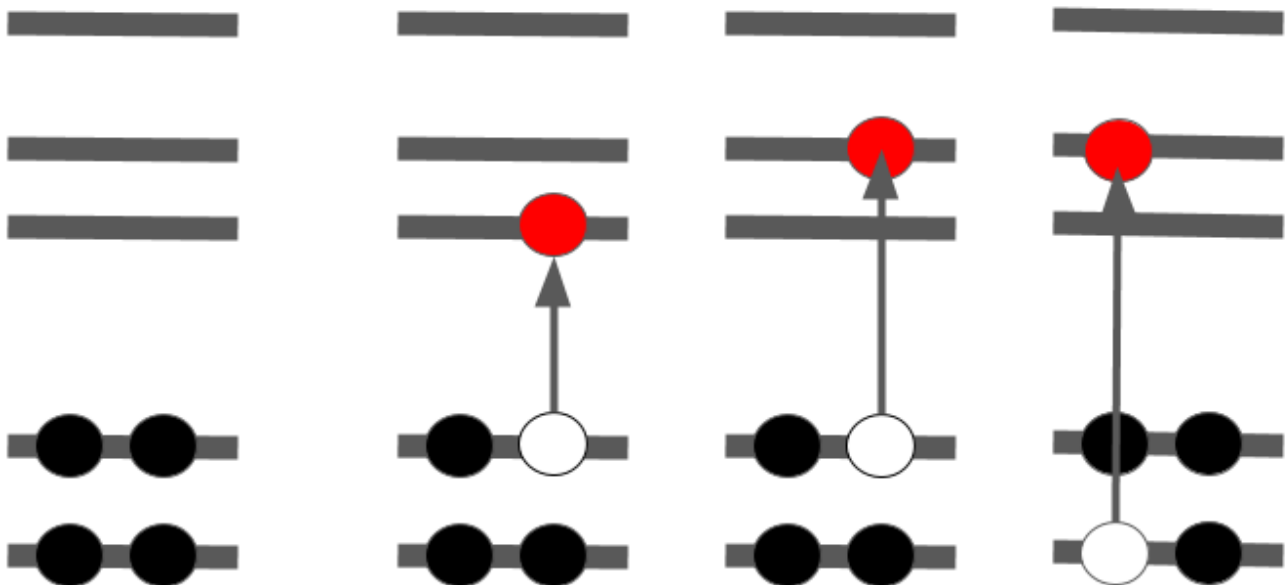


Figure 2. The ground state of LiH (left diagram) has 4 electrons in 2 orbitals. Single-electron excitations promote one electron from the lowest orbitals into higher-energy orbitals, as shown in the three other diagrams.

The energy of these excited states will generally depend on the distance between the Li and H atoms (called the *interatomic distance*). Suppose we have (artificially) fixed the interatomic distance. To compute the energy of the single-electron excitations, we use a two-step process:

- We compute the ground state of the LiH molecule at the given interatomic distance. This gives us a representation of how the electrons in the molecule distribute themselves across the atoms. To compute the ground state, we use the Variational Quantum Eigensolver (VQE) algorithm. This is a hybrid quantum-classical algorithm that uses the variational method to estimate the ground state.

- We then use the ground state, plus the quantum subspace expansion (QSE) technique, to estimate the excited state energies.

To do the first step, we use the VQE algorithm as implemented in Qiskit Aqua. The quantum part of the VQE algorithm is best understood as first preparing a parameterized *trial state* — essentially, a guess as to what the ground state is — and then measuring different terms in the Hamiltonian describing the molecule. This yields an estimate for the energy of that particular trial state. Then, the classical part of the algorithm is used to adjust the parameters of the trial state, with a goal of minimizing the energy. The algorithm then proceeds in an iterative fashion, with the classical optimizer proposing new parameter values for the trial state, and the quantum processor estimating the energy. Upon convergence, the parameter choices found by the optimizer yield an estimate of the ground state.

The trial state is prepared on the processor using a parameterized quantum circuit. Then, additional gates are added to the circuit to measure the energy of the trial state. The circuit could be run on a simulator or a real quantum processor. (In this work, we use the simulation capabilities provided by Qiskit Aer.)

One of the main challenges to running the VQE algorithm on existing processors is that common choices for the trial state (e.g., the UCCSD expansion) typically lead to circuits with a large number of gates, or high depth. Given the amount of noise in near-term processors, reducing these circuit requirements is paramount.

At CQC, we've developed the $t|ket\rangle$ compiler to reduce circuit size, depth, and other circuit resources. This makes simulations faster and reduces the impact of noise on measurement results. The $t|ket\rangle$ compiler also efficiently maps a quantum circuit to a quantum processor's connectivity constraints, a process we call routing. Because $t|ket\rangle$ handles the routing problem automatically for a user, quantum circuits for VQE can be easily constructed without prior knowledge of the hardware they will be run on.

$t|ket\rangle$ integrates with Qiskit Terra, enabling users to achieve circuit resource reduction while maintaining their existing Qiskit workflows. This is possible due to the modular and extensible nature of Terra's framework for mapping circuits to backends, as implemented in the architecture of its *transpiler*.

In Table 1, we present results showing how $t|ket\rangle$ reduces circuit resources for a randomly-chosen instance of the LiH VQE circuit. (Note that here, we are looking only at the part of the circuit that prepares the trial state.) Compared with the original

circuit (generated using the commonly-used UCCSD trial state), $t|ket\rangle$ is able to substantially simplify the circuit, reducing the total number of gates, the depth, and the overall CNOT count.

LiH VQE Circuit	Total Gates	Overall Depth	Overall CNOT Count
Input Circuit	13700	9342	8064
$t ket\rangle$ circuit optimization	7411	4416	5096
Qiskit default routing	42178	23367	17977
$t ket\rangle$ circuit optimisation + routing	19256	10022	8711

Table 1: Comparison of circuit properties for a randomly-chosen instance of a LiH VQE quantum circuits generated with Qiskit Aqua for the 'ibmq_16_melbourne' device, compiled by Qiskit and $t|ket\rangle$. CNOT count is included as they are comparatively significantly longer and more error prone than single qubit gates.

Results obtained with Qiskit Terra v0.7.1 and pytket v0.1.3

These results are dependent on the gate set used in the circuit compilation and routing processes. Here, we have used the default gate set supported by the "ibmq_16_melbourne" backend offered through the IBM Q provider. However there are other possibilities here such as the recently proposed exchange-type gates [3]. By adapting this approach even greater depth reductions may be possible, while simultaneously improving robustness against noise.

Once the VQE algorithm has converged, we now have an estimate for the ground state of the molecule. We can then move on to computing the excited state energies. To do this, we use CQC's implementation of Quantum Subspace Expansion algorithm (QSE), also referred to as linear response expansion [1, 2]. The QSE algorithm is inspired by a classical method called CI-singles, which is commonly used in conventional quantum chemistry. However the QSE algorithm is much more powerful as it is applied on an entangled ground state obtained from a VQE calculation, leaving it well suited to describe the excited states of simple molecules such as LiH.

As noted earlier, the results for the excited state energies depend on the interatomic distance between the Li and H atoms. One useful way of representing these energies is by plotting an *energy profile* of the molecule, where we sweep across different interatomic distances and plot the corresponding ground and excited state energies.

The lowest energy curve corresponds to the ground state and the higher energy curves

correspond to excited states. Figure 3 plots the energy profile for LiH, where we include estimates of the ground state and the first 3 excited states. For comparison, we have also plotted results obtained via the Equation-of Motion CCSD (EOM-CCSD) method commonly used in traditional quantum chemistry calculations:

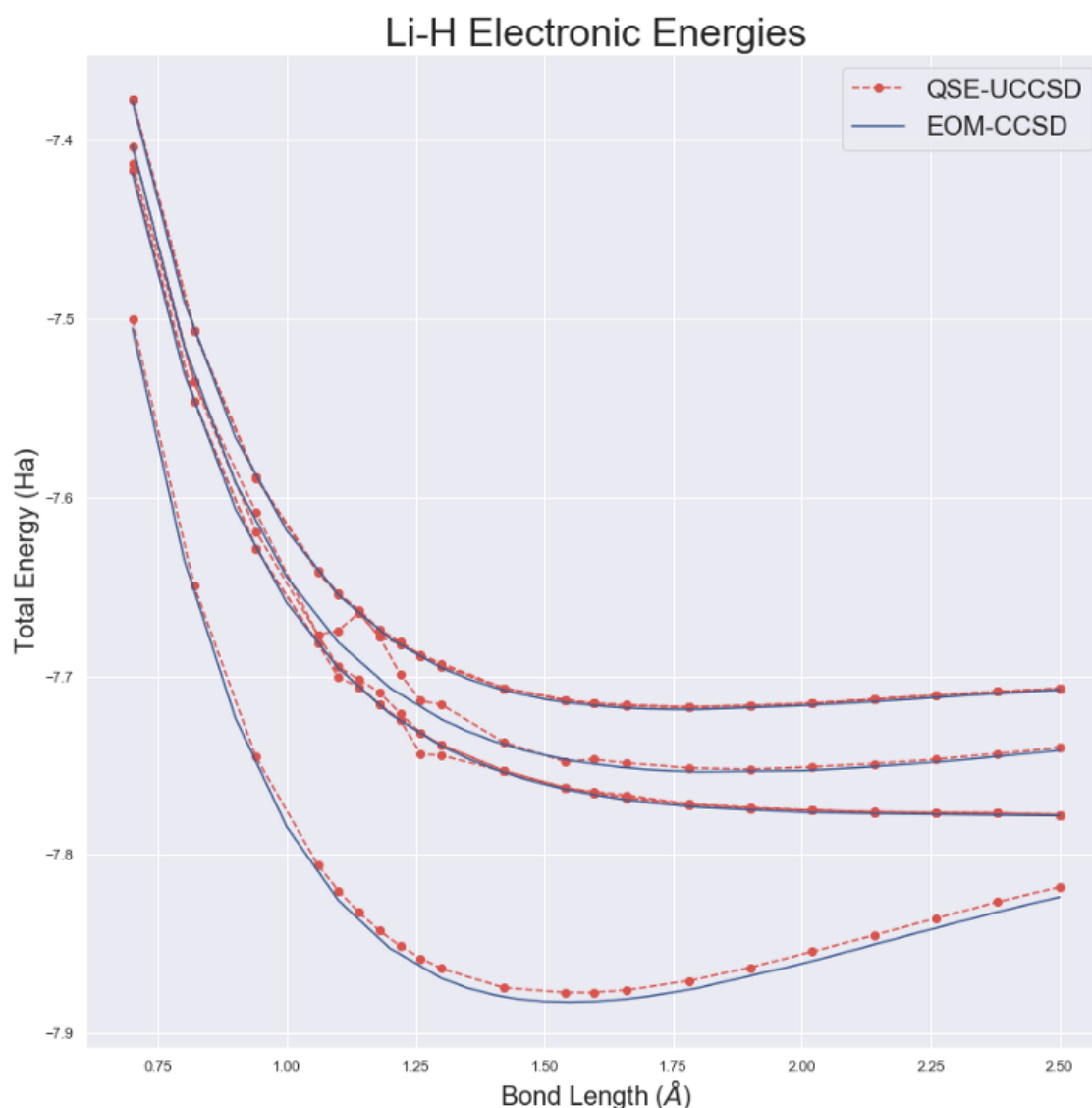


Figure 3. Graph showing the excited states of LiH at a range of bond distances calculated via two methods. The quantum subspace expansion with the unitary coupled cluster singles and doubles method (performed on a simulator) and the classically computed equation of motion coupled cluster singles and doubles method for comparison. It can be seen that the two methods agree strongly, with only some discrepancy around 1.2 Angstroms.

We find the ground state curve has a minimum at a separation of about 1.5 Å, which is in reasonable agreement with experimental data. The calculation also finds a number of excited states. Looking at the first three, we find that at the equilibrium distance,

these states are 0.11, 0.12 and 0.17 Ha higher in energy than the ground state, which is again in reasonable agreement with experimental data. Note the small kink in one of the excited states energy curve at a distance of approximately 1.2 Å. This indicates that our restriction to single electron excitations is not enough to provide an accurate description at this distance. Overall, the comparison with EOM-CCSD curves shows that this method reproduces excited state energies with good accuracy at most distances.

This example shows the potential of quantum computers to calculate excited states. While this work used a simulation of an ideal quantum computer, we could also investigate how noise affects the results using the noisy simulation capabilities of Qiskit Aer. To improve the accuracy of our idealized simulations, we could use more kinds of excitations such as double excitations where two electrons are promoted to unoccupied states. If we ran the VQE algorithm on real hardware, we could use the noise mitigation framework provided by Qiskit Ignis to improve the results.

The potential of quantum algorithms to revolutionize the study of molecular systems is well on its way to realisation, and CQC's tools in combination with IBM's Qiskit software stack offer the best way forward for quantum chemists.

[1] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter and W. A. de Jong, Phys. Rev. A 95, 042308 (2017)

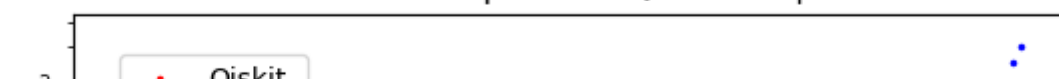
[2] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong and I. Siddiqi, Phys. Rev. X 8, 011021 (2018)

[3] M. Ganzhorn, D.J. Egger, P. Barkoutsos, P. Ollitrault, G. Salis, N. Moll, A. Fuhrer, P. Mueller, S. Woerner, I. Tavernelli and S. Filipp; Arxiv pre-print arxiv:1809.05057

Addendum:

Quantum circuits in this example were compiled using CQC's compiler `t|ket>`. Integrated into Qiskit using the `pytket` extension, `t|ket>` produces hardware-compliant quantum circuits quicker than the default Qiskit Terra compilation, and with smaller depth and size overhead.

Runtime Comparison: Qiskit vs t|ket>



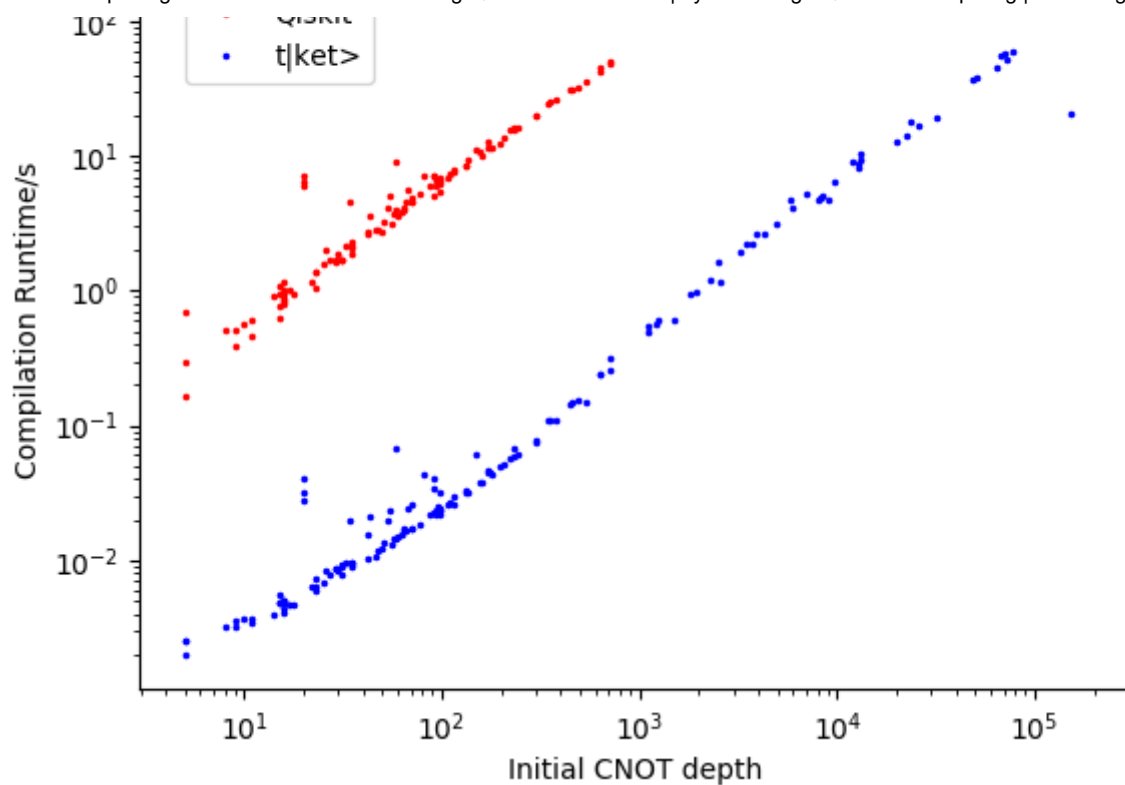


Figure 4. Runtime encompasses optimization and routing of quantum circuits for the 'ibmq_16_melbourne' device. Circuits executed on a Google Cloud Virtual Machine with 7.5GB memory and 2 vCPUs, machine type n1-standard-2.

Science Quantum Computing Quantum Chemistry

[About](#) [Help](#) [Legal](#)

Get the Medium app

