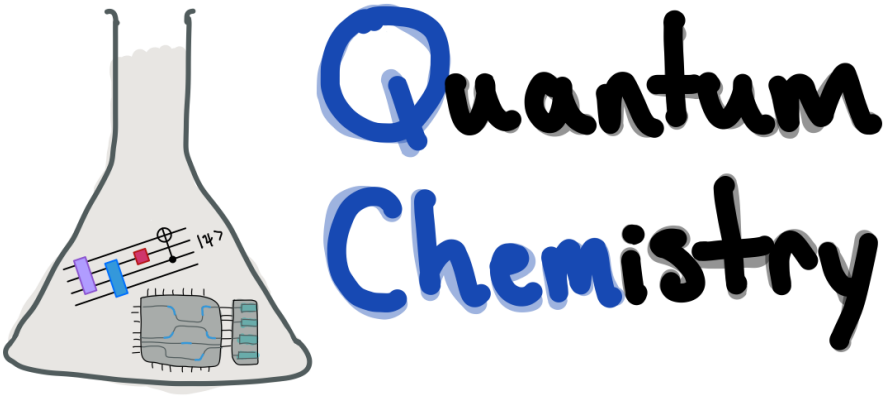

Quantum Chemistry: Mind the Gap [500 points]

Version: 1

Quantum Chemistry

Numerical techniques for determining the structure and chemical properties of molecules are a juggernaut area of research in the physical sciences. *Ab initio* methods like density functional theory have been a staple method in this field for decades, but have been limited in scalability and accuracy. As such, chemistry applications and problems are desirable candidates for demonstrating a quantum advantage.

It is therefore no surprise that quantum chemistry is one of the leading application areas of quantum computers. In the **Quantum Chemistry** category, you will be using PennyLane's core quantum chemistry functionalities to become familiarized with concepts and tools developed in this sub-field of quantum computing, like mapping molecular Hamiltonians to qubit Hamiltonians and quantum gates that preserve the electron number. Beyond these five questions in this category, there is a plethora of informative tutorials on the PennyLane website that will boost your understanding of topics that we will cover in this category. Let's get started!

Problem statement [500 points]

In this challenge, you will venture into the world of excited states. The variational quantum eigensolving (VQE) routine is a powerful numerical method for ground state calculations. In the "variational" subspace of numerical methods, ground states are usually the entity of interest. But, any variational algorithm based on optimizing the energy expectation value

$$E(\theta) = \langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle$$

can also be used to find excited states. A conceptually simple procedure for this is the following. Note that throughout this discussion, it will be assumed that the ground state energy of $\hat{H}$ is the largest negative eigenvalue of $\hat{H}$.

1. Esimate the ground state $|\lambda_0\rangle \approx |\psi(\theta^*)\rangle$ of the Hamiltonian $\hat{H}$ via traditional VQE.
2. In the computational basis $\{|\sigma\rangle\}$, construct the operator $\hat{H}_1 = \hat{H} + \beta|\lambda_0\rangle\langle\lambda_0|$. Here, β is a real number that is larger than the ground-first-excited energy gap.
3. Perform *another* VQE procedure by using $\hat{H}_1$ in place of the original Hamiltonian to obtain an estimate on the first-excited state $|\lambda_1\rangle$.

To see why this method works, let us write the $\hat{H}_1$ operator as

$$\hat{H}_1 = (E_0 + \beta)|\lambda_0\rangle\langle\lambda_0| + \sum_i E_i|\lambda_i\rangle\langle\lambda_i|.$$

Clearly, if $E_0 + \beta > E_1$, then the largest negative eigenvalue of $\hat{H}_1$ is now E_1, the excited state energy! Therefore, optimizing a variational circuit with respect to

$$E(\theta) = \langle\psi(\theta)|\hat{H}_1|\psi(\theta)\rangle$$

should give us an approximate form of the first excited state.

Your job is to perform steps 1-3 listed above for the H_2 molecule given a H–H distance (see Figure 1). The template file `mind_the_gap_template.py` contains three functions that you must define. The `ground_state_VQE` function must optimize a variational circuit and return the calculated ground state energy and the ground state itself. The `excited_state_VQE` function must optimize a different variational circuit and return the calculated excited state energy. Finally, you must manually create the $\hat{H}_1$ operator's matrix representation in the function `create_H1`.

At the end of it all, your code will be able to map out the "vertical" separation between the ground state and excited state potential energy surfaces (PES) (see Figure 1).

NOTE: You *must* do this problem without the use of the `qchem` library. Please use `qml.hf` instead.

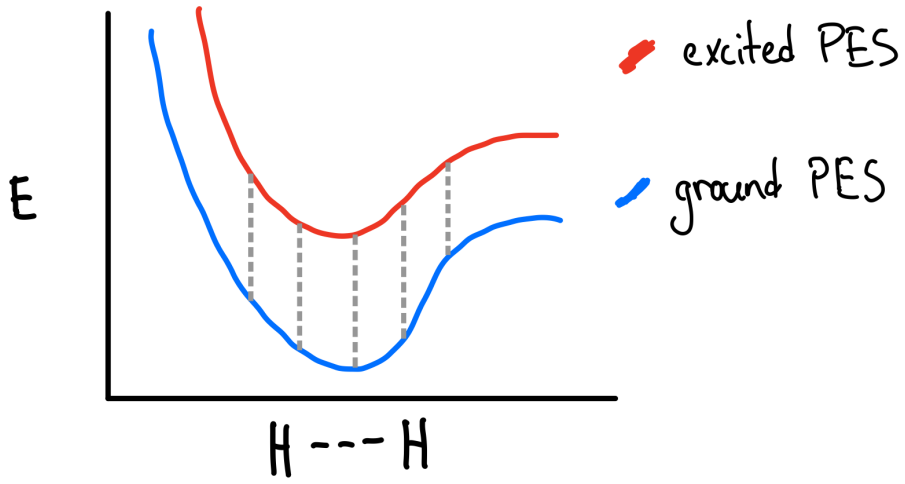

Figure 1: Excited state energy gaps for various distances. (PES = potential energy surface)

Input

- `float`: Half of the H–H distance. The individual atom coordinates are assigned as `geometry = np.array([[0.0, 0.0, - <input value>], [0.0, 0.0, <input value>]])`.

Output

- `list(float)`: A list containing the ground state and excited state energies in that order.

Acceptance Criteria

In order for your submission to be judged as "correct":

- The outputs generated by your solution when run with a given `.in` file must match those in the corresponding `.ans` file to within the `0.01` tolerance specified below. To clarify, your answer must satisfy

$$\text{tolerance} \geq \left| \frac{\text{your solution} - \text{correct answer}}{\text{correct answer}} \right| .$$

- Your solution must take no longer than the `80s` specified below to produce its outputs.

You can test your solution by passing the `#.in` input data to your program as stdin and comparing the output to the corresponding `#.ans` file:

```
python3 {name_of_file}.py < 1.in
```

WARNING: Don't modify the code outside of the `# QHACK #` markers in the template file, as this code is needed to test your solution. Do not add any print statements to your solution, as this will cause your submission to fail.

Specs
Tolerance: **0.01**
Time limit: **80 s**

Version History

Version 1: Initial document.