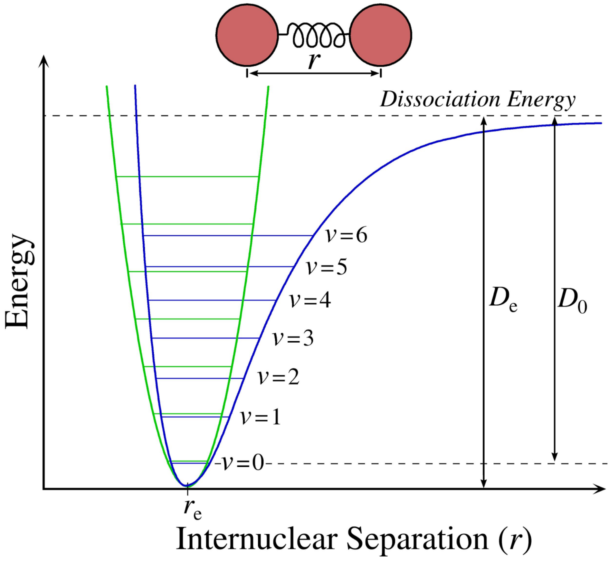
QuantumGuards 2020

Calculating thermodynamic observables of LiH using Qiskit simulations

Eric Berg, Eva Gurra, Abigail Martucci, Sam Shersher and Jerry Zhuang (Javad Shabani)

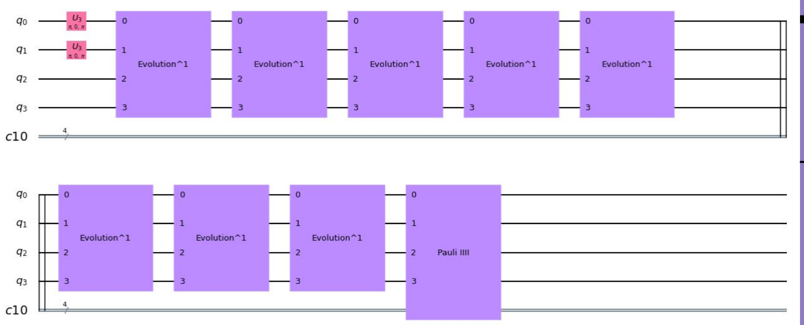
Calculation of thermodynamic observables-Entropy, Enthalpy and Gibbs Free Energy



1. Calculate the Born-Oppenheimer potential energy surface (BOPES)
   a. Can be challenging due to Schrodinger equation
2. Determine vibrational modes
3. Calculate the partition function and thermodynamic observables as a function of temperature and pressure.

*Use quantum computing to help with Schrodinger equation!

1.Qiskit Chemistry Drivers: Calculates one-body, two-body integrals



2. FermionicOperator and QMolecule convert the one body, two-body integral calculations into a qubit operator

3. Prepare initial state using HartreeFock and set up the variational form using Qiskit Chem UCCSD form. Using Qiskit VQE algorithm, we input the variatonal form, the qubit operator and the optimizer.

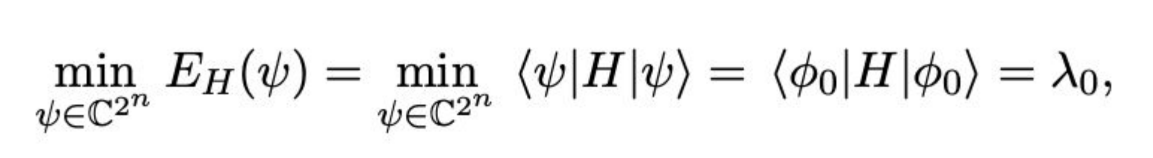
4. Plot the simulated results and fit to a Morse potential to obtain the equilibrium potential energy well depth, width and equilibrium bond length

5. Calculate thermodynamic observables in terms of these fitted parameters

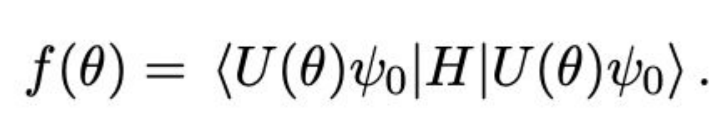
Variational Quantum Eigensolver

- Applies classical optimization to minimize the energy expectation of an initial state
- Can also be used to determine the excited states.
- Reduces the long coherence times required by the quantum phase estimation algorithm
- Here we attempted to improve on efficiency & accuracy of Li-H tutorial example using techniques from Stober et. al.

VQE- Step 1



$$\min_{\psi \in \mathbb{C}^{2^n}} E_H(\psi) = \min_{\psi \in \mathbb{C}^{2^n}} \langle \psi | H | \psi \rangle = \langle \phi_0 | H | \phi_0 \rangle = \lambda_0,$$



$$f(\theta) = \langle U(\theta)\psi_0 | H | U(\theta)\psi_0 \rangle \, .$$

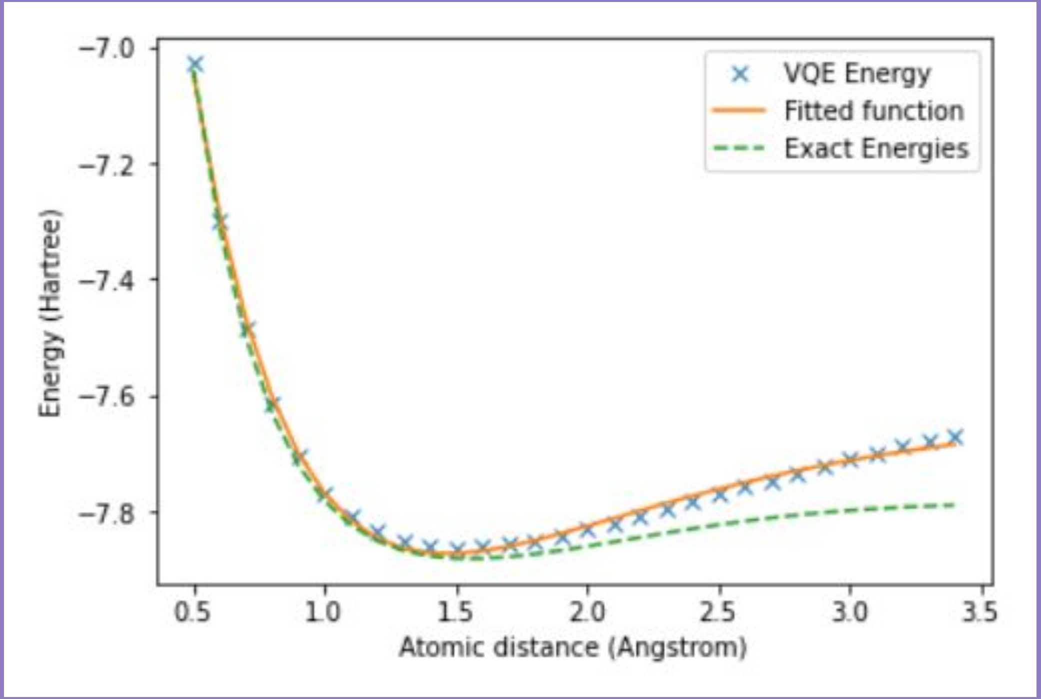
- E is minimum energy
- H is some given Hamiltonian
- ψ is a quantum state
- λ0 is the smallest eigenvalue of H
- φ0 is its corresponding eigenvector

VQE- Step 2

$$|\psi_0\rangle = U_0\,|0\rangle\,.$$

|ψ0> is a given initial state, which is then modified by the parameterized circuit U. It can simply be the state |0>, or some other quantum state (ideally, closer to the eigenstate φ0)

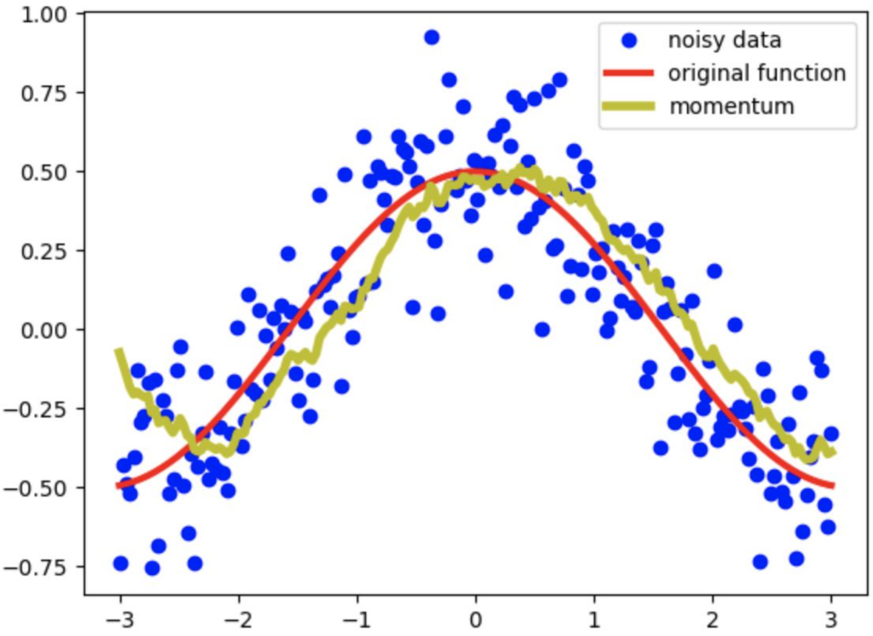
Unoptimized Energies - Variational Principle



Here we started with the initial Hartree Fock state and did not update parameters as we apply VQE.

The simulation results follow the Morse model closely but for larger interatomic distances the model and our results deviate from the the exact energies.

GRABER - Optimization



momentum — data from exponentially weighed averages.

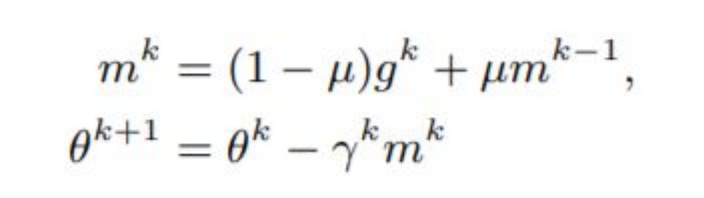
- This is a method used to classically optimize the VQE algorithm.
- It improves the precision and efficiency of the VQE algorithm, essentially smoothing the data
- Based on stochastic gradient descent method with momentum

Momentum

-Tunable parameter adjusts dependence on previous gradient

-Already implemented in built-in AQGD optimizer

-Since AQGD already used momentum, smart termination and parameter-shift rule, we modified AQGD module into GRABER



$$
\begin{aligned}
m^k &= (1-\mu)g^k + \mu m^{k-1}, \\
\theta^{k+1} &= \theta^k - \gamma^k m^k
\end{aligned}
$$

Bootstrapping

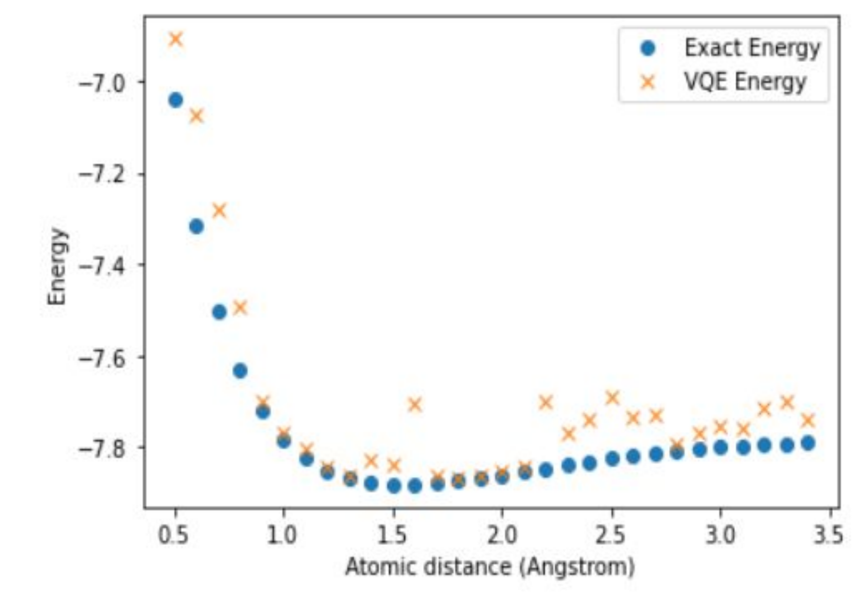
GRABER utilizes advanced starting point information (“bootstrapping”).

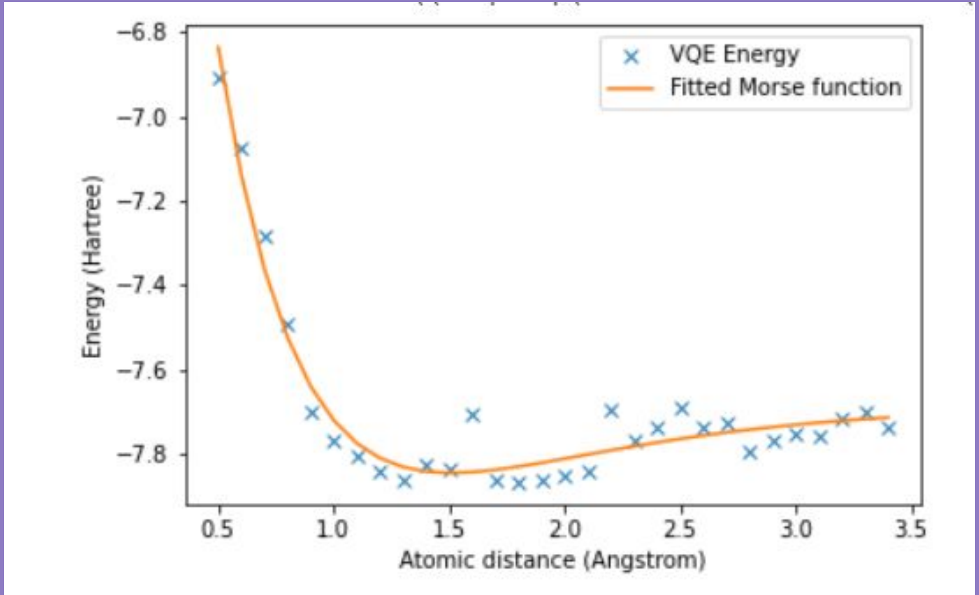
We implemented initial point updates for VQE by using the optimal_params property when running the optimizer on a specific molecule and bond length.

Epochs

-custom step length schedule with 2 parameters: offset(N) and initialEta (initial step size)

Results (Offset = 10, InitialEta = 3)





Stochastic Gradient Descent (SGD)

-Cost function (in this case hamiltonian) is sum over input dataset: ith data → Qi

-SGD computes gradient from random selection of Qi, saving time and computational resources

-We ran out of time before implementing this

Normal Gradient Descent

$$w := w - \eta \nabla Q(w) = w - \frac{\eta}{n} \sum_{i=1}^{n} \nabla Q_i(w),$$

Stochastic Gradient Descent

$$w^{new} := w^{old} - \eta \nabla Q_i(w^{new}).$$