Ground State Energy of H_2 molecule using VQE

Reddi Kishore

1 Introduction

The project is about finding ground sate energy of Hydrogen molecule using Quantum computing tools and techniques. By implementing VQE on a Quantum simulator and used Qiskit runtime services to calculate Ground state of H_2 molecule.
Then simulated the molecule on a noisy simulator to replicate the Quantum hardware and applied a Probabilistic Error Cancellation to the measurement to reduce the error in the ground state energy.

1.1 Code: Calculation of Ground state energy on a Simulator

```
from qiskit import QuantumCircuit, transpile, Aer
from qiskit_nature.units import DistanceUnit
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.second_q.mappers import JordanWignerMapper
from qiskit_nature.converters.second_quantization import QubitConverter
from qiskit.algorithms import VQE
from qiskit.algorithms.optimizers import SPSA, COBYLA
from qiskit.primitives import Estimator
from qiskit_nature.second_q.circuit.library import HartreeFock, UCCSD

# Step-1: Setting up the Molecular System

driver = PySCFDriver(
    atom="H 0 0 0; H 0 0 0.735",
    basis="sto3g",
    charge=0,
    spin=0,
    unit=DistanceUnit.ANGSTROM)
problem = driver.run()

# Step-2: Hamiltonian Setup (Second Quantization)

hamiltonian= problem.hamiltonian.second_q_op()

# Step-3: Qubit Mapping

qubit_mapper= JordanWignerMapper()
qubit_hamiltonian= qubit_mapper.map(hamiltonian)

# Step-4: Initial State Preparation

initial_state= HartreeFock(
    num_spatial_orbitals= problem.num_spatial_orbitals,
    num_particles= problem.num_particles,
    qubit_mapper= qubit_mapper)

# Step-5: Ansatz Definition

ansatz = UCCSD(
    num_spatial_orbitals= problem.num_spatial_orbitals,
    num_particles= problem.num_particles,
    qubit_mapper= qubit_mapper,
    initial_state= initial_state)




# Step-6: VQE Setup

optimizer= COBYLA (maxiter=1000)
```

```
backend= Aer.get_backend("statevector_simulator")
vqe= VQE(ansatz, optimizer, quantum_instance=backend)
result=vqe.compute_minimum_eigenvalue(qubit_hamiltonian)
print(result)
```

1.2 Result

```
'optimal_point': array([ 1.89665611e-06,  4.99429076e-05, -1.11744047e-01])
'optimal_value': -1.857275027209684
```

2 Simulation of H_2 molecule on a Real Quantum Hardware

In the last section we calculated the Ground stated energy of H_2 molecule using a simulator. Now we are going to calculate by using a real Quantum noisy hardware. Lets see how the results are going to change.

2.1 Code: Without any Error Mitigation

```
from qiskit_ibm_runtime import Options, Estimator, Session
from qiskit_ibm_runtime import QiskitRuntimeService
import numpy as np

options=Options()
options.optimization_level=0
options.resilience_level=0
service= QiskitRuntimeService()
backend= service.get_backend('ibm_osaka')

with Session(service=service, backend= backend) as session:
    estimator= Estimator(options=options)
    cost_func= lambda params:estimator.run(ansatz, qubit_hamiltonian,
               parameter_values=params).result().values[0]
    result= optimizer.minimize(cost_func, x0= np.zeros(ansatz.num_parameters))
    print(result)
```

2.2 Result

```
job = service.job("cp43bnp2zy6g0084j9cg")
job.status()
```

```
<JobStatus.DONE: 'job has successfully run'>
```

```
job.result()
```

```
EstimatorResult(values=array([-1.49445814]), metadata=[{'variance': 0.19888308261667137, 'shots': 4000}
'Optimal_value': -1.49445814
```

2.3 Code: With PEC Error Mitigation

To implement Probability Error Cancellation technique, IBM providing QiskitRuntimeServices to calculate error mitigation techniques on the measurements with different Resilience levels. we need to use Resilience level=3 to implement PEC on measurements.

```
options_pec =Options()
options_pec.optimization_level=1
options_pec.resilience_level=3

service= QiskitRuntimeService()
backend= service.get_backend('ibm_osaka')
```

```
with Session(service=service, backend= backend) as session:
    estimator= Estimator(options=options_pec)
    cost_func= lambda params:estimator.run(ansatz, qubit_hamiltonian,
                parameter_values=params).result().values[0]
    result= optimizer.minimize(cost_func, x0= np.zeros(ansatz.num_parameters))
    print(result)
```

2.4 Result

```
job = service.job("cp43bx7yx18g008vkvpg")
job.status()
```

```
<JobStatus.DONE: 'job has successfully run'>
```

```
job.result()
```

```
EstimatorResult(values=array([-1.850029606645]), metadata=[{'variance': 0.1783885761667027, 'shots': 40
'Optimal_value': -1.850029606645
```

3 Summary and Results Discussion

Here we did 3 experiments, First one on a Simulator, second experiment is on real Quantum noise hardware with out any error mitigation on the measurements and tha Last experiment is on real hardware but with PEC error mitigation.
From results given below we can clearly see the difference, from these results we can conclude that error mitigation will give us better results.

Ground State Energy of H_2 molecule on a Simulator =-1.857275027209684 eV

On real Quantum hardware without any error mitigation= -1.49445814 eV

On real Quantum hardware with PEC error mitigation = -1.850029606645 eV