Active_Space_VQE_calculation

January 18, 2026

[11]:
```python
from qiskit_nature.second_q.drivers import PySCFDriver
from qiskit_nature.units import DistanceUnit

driver = PySCFDriver(
    atom="Li 0 0 0; H 0 0 1.5",
    basis="sto3g",
    unit=DistanceUnit.ANGSTROM,
    charge=0,
    spin=0
)

full_problem = driver.run()
print(full_problem.molecule)
print(full_problem.num_particles)
print(full_problem.num_spatial_orbitals)
```

```
Molecule:
        Multiplicity: 1
        Charge: 0
        Unit: Bohr
        Geometry:
                Li      (0.0, 0.0, 0.0)
                H       (0.0, 0.0, 2.8345891868475928)
        Masses:
                Li      7
                H       1
(2, 2)
6
```

[ ]:
```python
# from qiskit_nature.second_q.transformers import FreezeCoreTransformer

# fc_transformer = FreezeCoreTransformer(remove_orbitals=[4,5])

# fc_problem = fc_transformer.transform(full_problem)
# print(fc_problem.num_particles)
# print(fc_problem.num_spatial_orbitals)
# print(fc_problem.hamiltonian.constants)
```

```
(1, 1)
```

```
5
{'nuclear_repulsion_energy': 1.05835442184, 'FreezeCoreTransformer':
-7.84030604879426}
```

[16]:
```
from qiskit_nature.second_q.transformers import ActiveSpaceTransformer

as_transformer = ActiveSpaceTransformer(2, 2)

as_problem = as_transformer.transform(full_problem)
print(as_problem.num_particles)
print(as_problem.num_spatial_orbitals)
```

```
(1, 1)
2
```

[17]:
```
from qiskit_nature.second_q.mappers import JordanWignerMapper
mapper = JordanWignerMapper()
```

[18]:
```
from qiskit_nature.second_q.circuit.library import UCCSD, HartreeFock
ansatz = UCCSD(
    as_problem.num_spatial_orbitals,
    as_problem.num_particles,
    mapper,
    initial_state=HartreeFock(
        as_problem.num_spatial_orbitals,
        as_problem.num_particles,
        mapper,
    ),
)
```

[19]:
```
import numpy as np
from qiskit_algorithms import VQE
from qiskit_algorithms.optimizers import SLSQP
from qiskit.primitives import Estimator
vqe = VQE(Estimator(), ansatz, SLSQP())
vqe.initial_point = np.zeros(ansatz.num_parameters)
```

```
/tmp/ipykernel_1519/1451404600.py:5: DeprecationWarning: The class
``qiskit.primitives.estimator.Estimator`` is deprecated as of qiskit 1.2. It
will be removed no earlier than 3 months after the release date. All
implementations of the `BaseEstimatorV1` interface have been deprecated in favor
of their V2 counterparts. The V2 alternative for the `Estimator` class is
`StatevectorEstimator`.
  vqe = VQE(Estimator(), ansatz, SLSQP())
```

[20]:
```
from qiskit_nature.second_q.algorithms import GroundStateEigensolver
solver = GroundStateEigensolver(mapper, vqe)
```

[22]:
```
active_space_result = solver.solve(as_problem)
print(active_space_result)
print(f"Total ground state energy (Hartree) = {active_space_result.
  ↪total_energies[0]:.4f}")
```

```
=== GROUND STATE ENERGY ===

* Electronic ground state energy (Hartree): -8.921934232375
  - computed part:      -1.081628183581
  - ActiveSpaceTransformer extracted energy part: -7.840306048794
~ Nuclear repulsion energy (Hartree): 1.05835442184
> Total ground state energy (Hartree): -7.863579810535

=== MEASURED OBSERVABLES ===

  0:  # Particles: 2.000 S: 0.000 S^2: 0.000 M: 0.000

=== DIPOLE MOMENTS ===

~ Nuclear dipole moment (a.u.): [0.0  0.0  2.83458919]

  0:
  * Electronic dipole moment (a.u.): [0.0  0.0  4.726256926451]
    - computed part:      [0.0  0.0  4.730048775022]
    - ActiveSpaceTransformer extracted energy part: [0.0  0.0  -0.003791848571]
  > Dipole moment (a.u.): [0.0  0.0  -1.891667736451]  Total: 1.891667736451
                  (debye): [0.0  0.0  -4.808139339532]  Total: 4.808139339532

Total ground state energy (Hartree) = -7.8636
```

[24]:
```
from qiskit_algorithms import NumPyMinimumEigensolver
from qiskit_nature.second_q.algorithms import GroundStateEigensolver


classical_solver = GroundStateEigensolver(
    JordanWignerMapper(),
    NumPyMinimumEigensolver(),  # Exactly diagonalizes the Hamiltonian
)
```

[25]:
```
classical_result = classical_solver.solve(full_problem)
print(classical_result)
print(f"Total ground state energy (Hartree) = {classical_result.
  ↪total_energies[0]:.4f}")
```

```
=== GROUND STATE ENERGY ===

* Electronic ground state energy (Hartree): -8.940716708639
  - computed part:      -8.940716708639
```

```
~ Nuclear repulsion energy (Hartree): 1.05835442184
> Total ground state energy (Hartree): -7.882362286799

=== MEASURED OBSERVABLES ===

  0:  # Particles: 4.000 S: 0.000 S^2: 0.000 M: 0.000

=== DIPOLE MOMENTS ===

~ Nuclear dipole moment (a.u.): [0.0  0.0  2.83458919]

  0:
  * Electronic dipole moment (a.u.): [0.0  0.0  4.65747064783]
    - computed part:      [0.0  0.0  4.65747064783]
  > Dipole moment (a.u.): [0.0  0.0  -1.82288145783]  Total: 1.82288145783
                 (debye): [0.0  0.0  -4.6333020751]  Total: 4.6333020751

Total ground state energy (Hartree) = -7.8824
```