VQEDriver Documentation

v0.0.0

Python3 driver for the Variational Quantum Eigensolver (VQE) algorithm implemented in IBM's Qiskit for the calculation of the ground-state energy of a molecule. The driver reads a structured input file (commonly named `[MOLECULE].inp`) and runs a complete VQE calculation using IBM's Qiskit and PySCF libraries.

Usage of the program from terminal:

```
$ vqedriver [INPUTFILE] > vqe.out &
```

Contents

1 VQE Calculation

The input file follows a line by line structure, using keywords for each block and their options. In general, keywords are added in the format:

```
%keyword option_keyword1=option option_keyword2=option [...]
```

The input file accepts blank lines and comments beginning with the # character. The keywords can be added in any order.

1.1 Geometry

Added using the `%geometry` keyword. Currently, the geometry of the molecule has to be specified using an external XYZ file using the `xyzfile` option keyword.

Optional keywords:

- `charge` (int) – charge of the molecule.
  Default: `charge=0`
- `spin` (int) – $2S$, where S is the total spin angular momentum. Same as the number of unpaired electrons in the molecule.
  Default: `spin=0`
- `units` (str) – length units of the coordinate file.
  Available options: `angstrom`, `bohr`
  Default: `units=angstrom`

Example:

```
%geometry xyzfile=h2.xyz charge=1 spin=1 units=bohr
```

1.2 Basis set

Contains the basis set information for the molecule, added using the keyword `%basis`. Currently, only a single basis set can be added for all the atoms in the molecule using the option keyword `all=[BASIS SET]`. ECPs **are not** available.

Example:

```
%basis all=def2-TZVP
```

1.3 Self-Consistent Field (SCF)

Specifies the options for the SCF calculations, added by the `%scf` keyword. By default, the program sets a Restricted Hartree-Fock (RHF) calculation for closed-shell molecules and a Restricted-Open Hartree-Fock (ROHF) for open-shell molecule. Unrestricted Hartree-Fock (UHF) method is also

available for open- and close-shell molecules.

Optional keywords:

- `method` (str) – Hartree-Fock method for SCF calculations.

  Available options: `rhf`, `rohf`, `uhf`

  Default: `method=rhf` if `spin==0` (closed-shell), `method=rohf` if `spin!=0` (open-shell)

- `conv` (int) – convergence criteria for the SCF calculations to $10^{-\text{conv}}$ Hartree.

  Default: `conv=9`

- `maxcycles` (int) – maximum number of SCF cycles.

  Default: `maxcycles=50`

- `maxcore` (int) – maximum memory in MB that PySCF should use.

  Default: `maxcore=1000`

Example:

```
%scf method=uhf conv=6 maxcycles=100 maxcore=7500
```

1.4 VQE Settings

The VQE algorithm requires a qubit operator (mapped from the fermionic operator representing the hamiltonian in the second quantization formalism), a trial wavefunction (*ansatz*) written as a unitary variational form and a classical optimizer. VQEDriver sets up and runs the full VQE calculation on a selected digital simulator.

VQEDriver prints not only the ground-state energy, but also the eigenstate in the qubit base (along with its count histogram), and the optimal parameters. Also, it saves a PNG file of the

VQE circuit as set up, and decomposed in the base of I, $U1$, $U2$, $U3$ and $CNOT$ gates, where:

$$U1(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}$$

$$U2(\phi, \lambda) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{bmatrix}$$

$$U3(\theta, \phi, \lambda) = \begin{bmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{bmatrix}$$

Notice that $H = U2\,(0, \pi)$ and $P(\lambda) = U1(\lambda)$.

1.4.1 Qubit Operator

Contains general settings for the qubit operator setup, built mapping the fermionic operator to the qubit operator. Options are added by the `%qubitop` keyword. The qubit operator is printed as a Python3 dictionary in a text file `[INPUTFILE].qubit_operator.txt`.

Optional keywords:

- `map` (str) – mapping to convert the fermionic operator to a qubit operator.
  Available options: `jw` (Jordan-Wigner), `parity`, `bk` (Braviy-Kitaev)
  Default: `map=parity`

- `threshold` (int) – threshold for Pauli simplification, eliminate the real and imaginary parts of the weight in each Pauli Operator mapped by $10^{-\text{threshold}}$. If a weight is less than $10^{-\text{threshold}}$, the Pauli is removed from the `WeightedPauliOperator` object.
  Default: `threshold=8`

Example:

```
%qubitop map=bk threshold=9
```

1.4.2 *Ansatz*

Builds the trial wavefunction variational form to be optimized. Added using the keyword `%ansatz`. Currently, there is one option available: UCCSD (Unitary Coupled-Cluster Single and Double excitations).

Optional keywords:

- `method` (str) – variational form to be used.

  Available options: `uccsd`

  Default: `method=uccsd`

- `exctype` (str) – excitation types to include in the active space to build the variational form.

  Available options: `s` (singles), `d` (doubles), `sd` (singles and doubles)

  Default: `exctype=sd`

Example:

```
%ansatz method=uccsd exctype=sd
```

1.4.3 Optimizer

Selects the classical optimizer for the VQE algorithm using the keyword `%optimizer`. The optimizer is selected using the optional keyword `method`. All available optimizers and their particular optional keywords are detailed on table I.

Example:

```
%optimizer method=cg print=false conv=6 maxiter=1500
```

Table I: Available classical optimizers for the VQE algorithm.

Optimizer	Description	Optional Keywords
`cg`	Conjugate Gradient	`maxiter` (int) – Maximum number of iterations `print` (bool) – Whether to print convergence messages `conv` (int) – Convergence tolerance

1.5 Simulation Setup

The VQE calculation is executed on a digital simulator of the quantum circuit. The simulator is added using the keyword `%sim`.

Currently, available backends are the ideal statevector simulator and the QASM simulator (noisy quantum circuit simulator).

Optional keywords:

- `backend` (str) – simulator to be used.
  Available options: `statevector`, `qasm`
  Default: `method=qasm`

- `shots` (int) – number of repetitions of quantum circuit measurements for sampling and averaging.
  Default: `shots=8192`

- `exact` (bool) – if `true`, print the exact result calculated by classical diagonalization using the `NumPyEigensolver`. Useful for comparisons to classical techniques. Also prints the fidelity between the optimized trial state and the exact eigenstate.
  Default: `exact=true`

- `method` (str) – simulation method. For the statevector simulator, only the `statevector` CPU method is available. Several methods are available for the QASM simulator.
  Available options for QASM: `statevector`, `density_matrix`, `matrix_product_state`, `automatic`
  Default for QASM: `automatic`

Example:

```
%sim backend=qasm exact=true shots=8192 method=density_matrix
```