Kagome Lattice – Open Science 2022

A solution for the state preparation of the Kagome Lattice in a quantum computer

Our Solution

* Our solution is implemented as a python program where all the parameters of the VQE algorithm are fed as command line arguments.
* Ansatz Options

Name	Description	Default Value
-a, --ansatz_type	Ansatz type: • **ExcitationPreserving:** Heuristic excitation-preserving wave function. • **EfficientSU2:** A hardware efficient SU(2) 2-local circuit. • **PauliTwoDesign:** The Pauli Two-Design ansatz. • **TwoLocal:** The two-local circuit. • **RealAmplitudes:** A heuristic trial wave function.	EfficientSU2

Name	Description	Default Value
-ol, --opt_level	Circuit optimization level (1-3)	1
-ui, --uniform_interaction	Heisenberg Model uniform interaction value.	Edge weight
-up, --uniform_potential	Heisenberg Model uniform potential	0.0
-w, --weight	Lattice edge weight.	2.4
-s,--shots	Number of execution shots.	2048

Optimizer Options

Name	Description	Default Value
-o, --optimizer_type	Optimizer type: • **SPSA:** Simultaneous Perturbation Stochastic Approximation • **SLSQP:** Sequential Least Squares Programming. • **COBYLA:** Constrained Optimization By Linear Approximation. • **UMDA:** Continuous Univariate Marginal Distribution Algorithm • **GSLS:** Gaussian-smoothed Line Search. • **GradientDescent:** The gradient descent minimization routine. • **L_BFGS_B:** Limited-memory BFGS Bound optimizer. • **NELDER_MEAD:** Performs unconstrained optimization. • **POWELL:** The Powell algorithm with unconstrained optimization • **NFT:** Nakanishi-Fujii-Todo algorithm.	SPSA
-i, --max_iter	Maximum number of iterations or function evals used by the optimizer.	175

Resilience and Scalability

Name	Description	Default Value
-r, --resilience_type	Resilience type (1-3): • T-Rex: 1 • ZNE: 2 • PEC: 3	2

Scalability

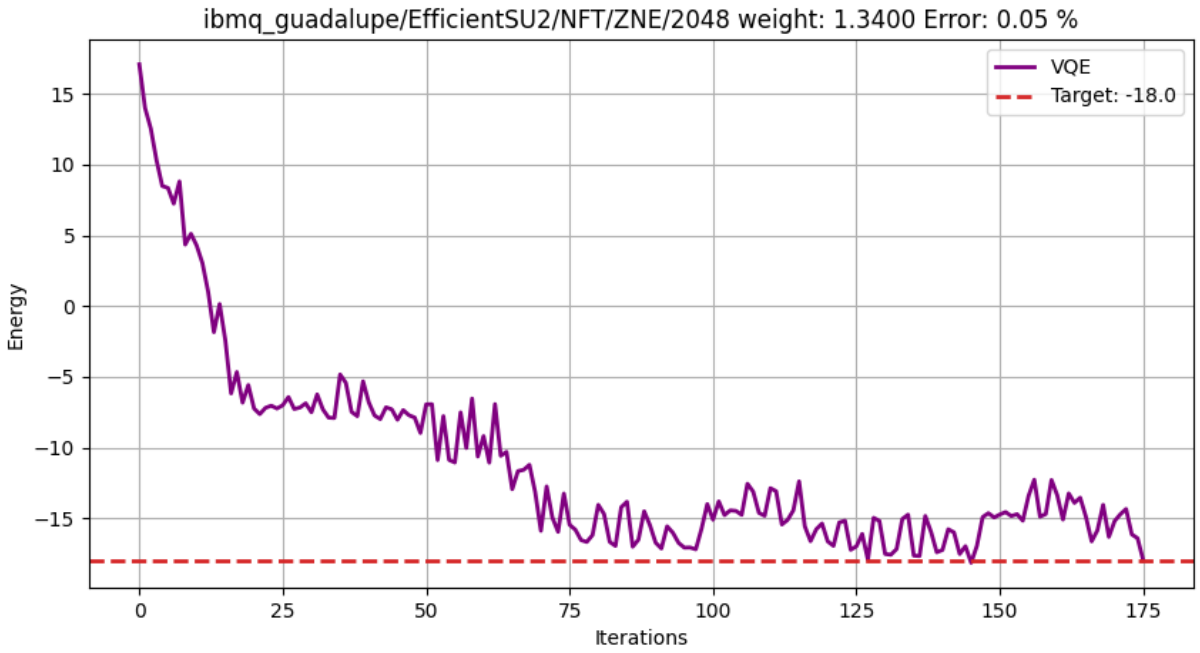
The program is designed to run in any quantum processor with any number of qubits. For example to run in Geneva (27 qubits) with an NFT optimizer and uniform potential:

$ python3 kagome_solution.py -b ibm_geneva -t ibm_geneva -q 27 -a EfficientSU2 -w 1.0 -o NFT -up -1.0

Our Results: Solution 1 - Error 0.05%

Backend: ibmq_guadalupe
Ansatz: EfficientSU2 (reps=1,entanglement='reverse_linear')
Optimizer : **NFT**(maxiter=175)
Resilience : ZNE (2)
Shots: 2048,
Edge weight: 1.34000

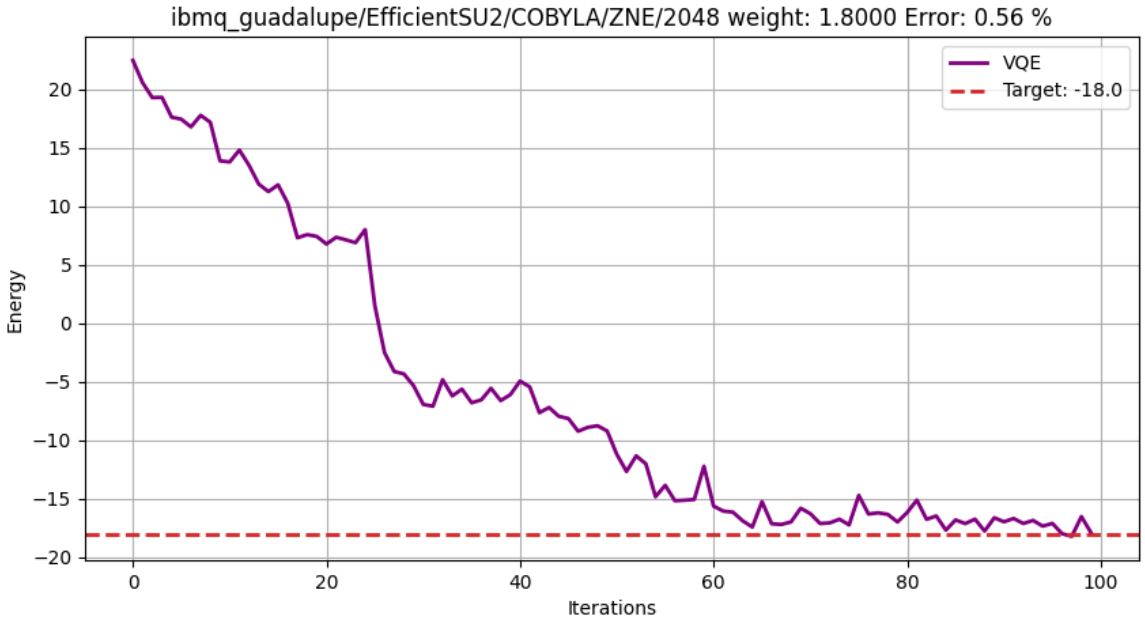
Execution time (s): 12390.63
Expected ground state energy: -18.00000000
Computed ground state energy: -17.99098307
Result eigen value: -17.99098307
Relative error: 0.05009404%



Our Results: Solution 2 - Error 0.56%

Backend: ibmq_guadalupe
Ansatz: EfficientSU2 (reps=1,entanglement='reverse_linear')
Optimizer : **COBYLA** (maxiter=100)
Resilience : ZNE (2)
Shots: 2048,
Edge weight: 1.80000

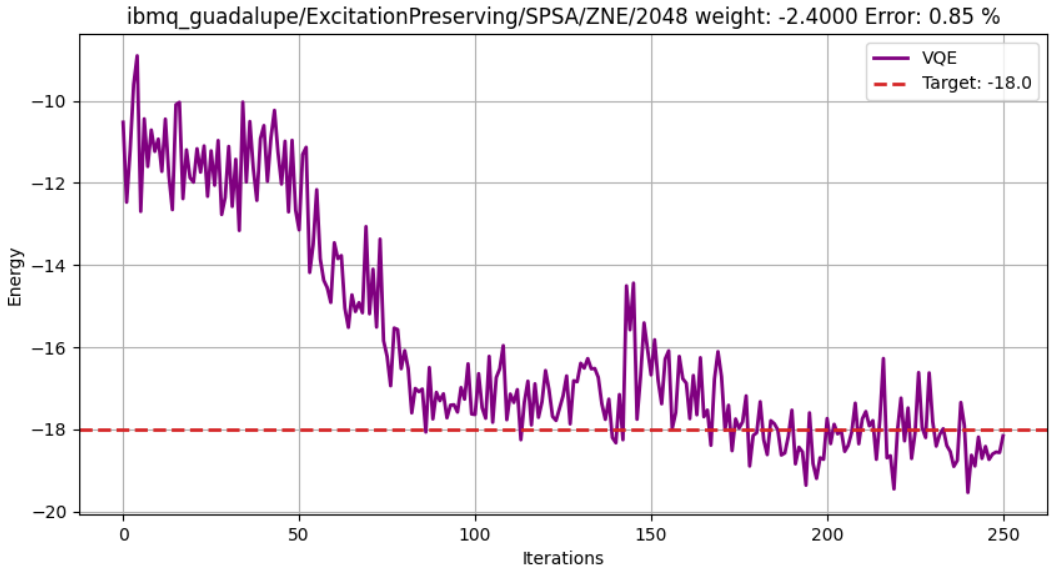
Execution time (s): 23294.63
Expected ground state energy: -18.00000000
Computed ground state energy: -17.89892578
Result eigen value: -17.89892578
Relative error: 0.5615%



Our Results: Solution 3 - Error 0.85%

Backend: ibmq_guadalupe
Ansatz: ExcitationPreserving (reps=1,entanglement='linear')
Optimizer : SPSA(maxiter=100)
Resilience : ZNE (2)
Shots: 2048
Edge weight: -2.4000

Execution time (s): 113953.17
Expected ground state energy: -18.00000000
Computed ground state energy: -18.15273438
Result eigen value: -18.15273438
Relative error: 0.848%



Other Solution - Error 7%

Backend: ibmq_guadalupe
Ansatz: EfficientSU2 (reps=1, entanglement='reverse_linear')
Optimizer : SPSA(maxiter=100)
Resilience : ZNE (2)
Shots: 2048
Edge weight: 1.9000

Execution time (s): 102658.02
Expected ground state energy: -18.00000000
Computed ground state energy: -19.38377279
Result eigen value: -19.38377279
Relative error: 7.687%