DB-QITE circuit compression

Antoine COMERRE
June 19, 2025 Supervisor: Yudai SUZUKI

Abstract

The goal of this report is to implement the DB-QITE algorithm that prepares the ground state of a generic hamiltonian following the article of Marek Gluza, Zoe Holmes, Yudai Suzuki et al[1]. My contribution was to compress this algorithm to achieve the same performances with a significant lower depth.

1 Introduction

Preparing the ground state of a Hamiltonian is a foundational task in quantum computing and quantum simulation, with applications in material science [2], and chemistry[3]. This task however has an exponential complexity with respect to the number of qubits, precisely the ground state prepration is QMA-complete[4]. Nevertheless, many promising algorithm exists for ground state preparation. For this report, we are using an approach inspired by thermodynamics with imaginary time evolution (ITE) to cool the state $|\Psi_0\rangle$ with respect to the Hamiltonian $\hat{H}$:

$$|\Psi(\tau)\rangle = \frac{e^{-\tau\hat{H}}|\Psi_0\rangle}{\|e^{-\tau\hat{H}}|\Psi_0\rangle\|}. \tag{1}$$

2 Theory

The state $|\Psi(\tau)\rangle$ is guaranteed to converge to the ground state if the initial state $|\Psi_0\rangle$ has a non-zero overlap with it. The goal is now to implement the operator: $U_\tau|\Psi_0\rangle = |\Psi(\tau)\rangle$. The distinction between ITE and quantum imaginary time evolution (QITE) lies in the fact that the operator U_τ is unitary for implementation in quantum computing. U_τ is found with the resolution of double bracket flows (DBF) [1]. The unitary is recursively constructed for a trivial initial state $|0\rangle^{\otimes n}$:

$$U_{k+1} = e^{i\sqrt{s_k}\hat{H}}U_k e^{i\sqrt{s_k}}|0\rangle\langle 0|U_k^\dagger e^{-i\sqrt{s_k}\hat{H}}U_k. \tag{2}$$

with an energy reduction at each step that follows the inequality:

$$E_{k+1} \leq E_k - 2s_k V_k + \mathcal{O}(s_k^2) \tag{3}$$

with $V_k = \langle\psi|\hat{H}^2|\psi\rangle - E_k^2$ the energy fluctuation. We have a convergence for $V_k = 0$, which is true for the ground state, but also for any eigenstate. The parameter s has to be well chosen to prevent a convergence to a higher energystate. Using theorem 2 of our main article [1], this problem is prevented as the fidelity with the ground state increases exponentially when s follows:

$$s = \frac{\Delta}{12||\hat{H}||3} \tag{4}$$

with $\Delta = \lambda_1 - \lambda_0$ the energy difference between first excited energy state and ground state called the energy gap.$||\hat{H}||$ is the spectral norm, given by the highest eigenvalue of the hamiltonian. This theorem assumes that all these terms are known to compute s which is not the case for complex hamiltonians. The parameter s will be found empirically in our implementation.

3 Implementation

Hamiltonian To test the DB-QITE algorithm, I chose two differents Ising hamiltonians for an arbitrary number of qubit N. First one is defined with a central spin which is connected to all other spins, and no interaction between these other spins and an external field:

$$\hat{H}_1 = J \sum_{i=1}^{N-1} \hat{\sigma}_z^0 \hat{\sigma}_z^i + h \sum_{i=0}^{N-1} \hat{\sigma}_x^i \tag{5}$$

The second one is the classic Ising model with neighbourous interactions and an external field

$$\hat{H}_2 = J \sum_{i=1}^{N-1} \hat{\sigma}_z^i \hat{\sigma}_z^{i+1} + h \sum_{i=0}^{N-1} \hat{\sigma}_x^i \tag{6}$$

For both hamiltonians, we set $J = h = 1$. The interesting difference between these two hamiltonians is that $\hat{H}_1$ has a unique ground state for $h \neq 0$ whereas $\hat{H}_2$ is invariant under cyclic permutations of the σ^z operators, so it has N distinct ground states. We will study if we can recover one of the ground states even if it is non-unique.

DBQITE The DB-QITE algorithm is implemented following **Eq.** 2. Although this algorithm is recursive, instead of computing directly U_k, we can compute first U_1 then U_2 ... until U_k and all those steps have a polynomial complexity. The computation of the matrix U_k has then a polynomial complexity instead of a complexity of 3^k if we start by U_k.
The unitary matrix is then converted to a circuit using the qiskit built-in function UnitaryGate(). The main issue with implementation was caused by float imprecisions that caused U_k to be non unitary for large k. This issue was solved by computing the closest unitary matrix $\tilde{U_k}$ of U_k with respect to the Frobenius norm:

$$\min_{\tilde{U} \in \mathcal{U}(n)} \|U_k - \tilde{U_k}\|_F$$

We prove in **Section** 7 that $\tilde{U_k} = WX^\dagger$ where $U = W\Sigma X^\dagger$ is the singular value decomposition of U. This correction is applied at every 5 steps of k.

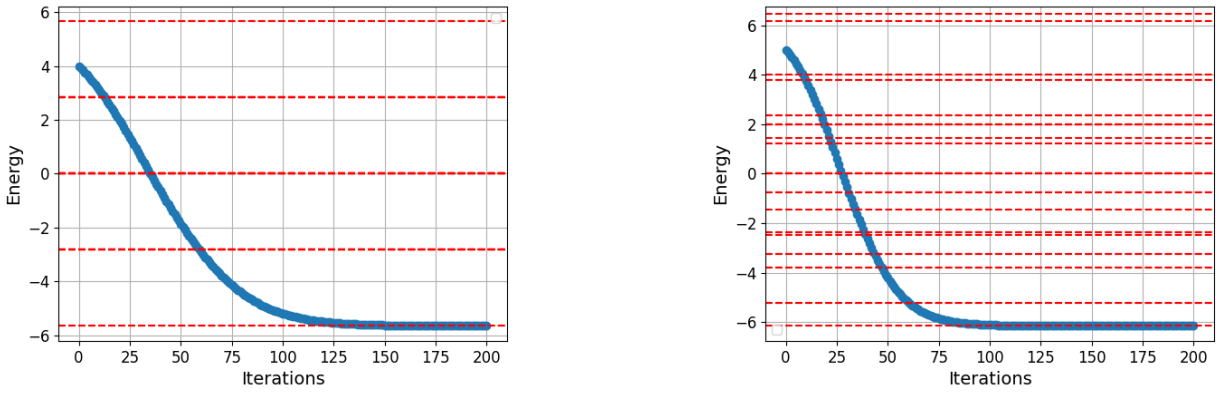
Reduction Once the circuit is transpiled, the goal is to compress it. For this purpose, I followed the article of Niall F. Robertson, Albert Akhriev, Jiri Vala and Sergiy Zhuk in [5] which was suggested to me by Yudai. This article was implemented on the qiskit library [6]. The procedure is the following: the final state of the circuit I want to compress is evaluated for the input $|0\rangle^{\otimes n}$. The output is then converted in a matrix product state (MPS):

$$|\psi\rangle = \sum_{\{j_1,...,j_N\}} c_{j_1,...,j_N} |j_1, ..., j_N\rangle = \sum_{\{j_1,...,j_N\}} A_1^{[s_1]} \cdot ... \cdot A_N^{[s_N]} |j_1, ..., j_N\rangle$$

The advantage we get from this representation is that the matrices'size of A_{j_i} is bounded by a parameter χ which is small if the wavefunction has low entanglement. This way, the state we want to construct with a compressed circuit is easy to store in memory.
Next we are using an the hardware efficient SU(2) Ansatz circuit defined in the qiskit library [7] and shown in **Fig.** 2b. The depth of the Ansatz circuit can be tuned as an extra parameter to achieve a better fidelity, as if the number of parameters is too low, we arrive at a point with zero-gradient which is far from our MPS goal state. We are computing the gradient descent with the loss function defined as the fidelity between our previous MPS and the state we generate with our ansatz. The optimizer used is the Limited-memory BFGS, and the optimizer is stopped once a fidelity of $1 - 10^{-4}$ is achieved.

4 Results

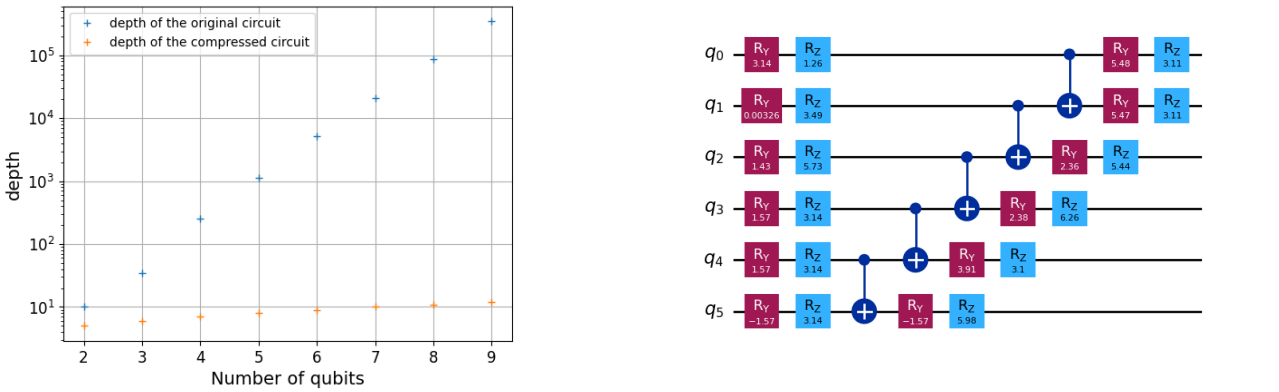
Ground-state recovery Both hamiltonians are compared with the same parameters. The number of qubits is low enough to make possible the computation of the eigenenergies of both hamiltonians. The parameter s is reduced until we observe a convergence to the ground state.



(a) (b)

Figure 1: Energies of $U_k|0\rangle^{\otimes 5}$ in function of the number of iterations k with the hamiltonians (a)$\hat{H}_1$ in **Eq.** 5 and (b)$\hat{H}_2$ in **Eq.** 6. Both hamiltonians have 5 qubits, and a parameter $\sqrt{s} = 0.1$ The red doted lines corresponds to the eigenergies of the hamiltonians.

Compression We are using the definition of the depth of qiskit, which is the number of steps needed to execute the circuit. All the parameters are tested to find how the depth is impacted. Finally the depth does not depend on k the number of iteration used to compute U_k, and it does not depend on the parameter $\sqrt{s}$. The only relevant parameter is the number of qubits in the hamiltonian where the depth has an exponential dependance with it, as shown in the **Fig.** 2a. Only one repetition of the SU(2) Ansatz circuit was needed for all qubits to achieve our threshold fidelity of 0.9999. The compress circuit has a linear dependance on the number of qubits as the depth is only increased by one when we add an extra qubit. It is a huge amelioration.



(a) Depth of the DB-QITE circuit applied on H_1 with and without compression.

(b) Compressed DB-QITE circuit using hardware efficient SU(2) for 6 qubits.

Figure 2: Performance overview of the compression algorithm applied on the H_1 hamiltonian.

The compression algorithm was also used for to compress the DB-QITE circuit for low values of k. Which implies that we are not trying to approximate the ground state but an intermediate state between initial state and ground state. The purpose of this is to construct the circuit U_{k+1} by using the ansatz circuit of U_k instead of the original U_k which can be too big. The compression algorithm didn't work as well as before but with an increase of the depth of our SU(2) harware efficient ansatz, we achieved an approximation with the same fidelity of 0.9999 as before. The **Fig.** 3 shows illustrates that more gates are needed if the state is different from ground state. As shown in **Fig.** 1, the ground state is reached for a number of iterations between 125 and 150.

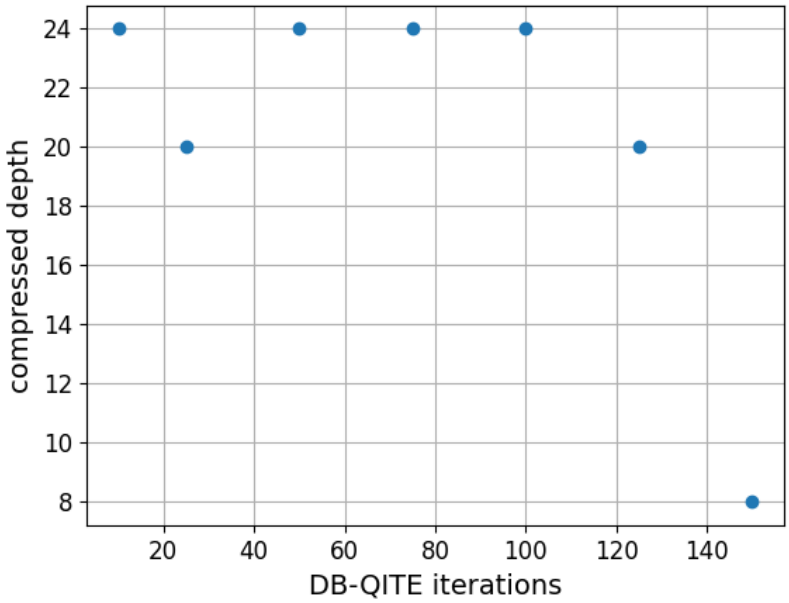


Figure 3: Ansatz depth needed to achieve a 0.9999 fidelity with $U_k|0\rangle^{\otimes 5}$ depending on the number of DB-QITE iterations k. The same parameters of **Fig.** 1 are used. A fixed number of 100 gradient descent iterations was used for each data points, and the depth was increased starting from the lowest until the fidelity threshold is reached.

5 Discussion

Ground-state recovery The DB-QITE algorithm succeeds to find a ground state for both hamiltonians, with significantly more computation time needed for the degenerate case H_2. The maximum number of qubits where the ground state was recovered with my computation ressources was $N = 9$ as the complexity grows exponentially with N. The degeneracy of the hamiltonian is not a problem for the ground state recovery and we observe that the convergence is faster for the degenerate hamiltonian H_2 than H_1 which may be explained by a higher initial fidelity with the ground state for the H_2 Hamiltonian.

Compression The compression algorithm is very promising as the number of gates needed to generate the ground state went from an exponential growth with respect to the number of qubits to a linear growth. The issue relies in the fact that we need to generate first the full circuit in order to compute its compressed version and we found a way to solve this problem by applying the compression algorithm to the DB-QITE unitary U_k for small k which needs

less computation power. A problem was identified in **Fig.** 3 about the compression of states different from ground state which needed more ressources. An explanation of this behaviour is that ground state are easier to represent with a matrix product state than other states[8].

6 Conclusion

The goal of this report was to find and construct the ground state of a generic hamiltonian. Two differents Ising based hamiltonians were used and a successful ground state preparation was achieved for both, with a limited number of qubits. The compression of the DB-QITE circuit using matrix product states worked very well achieving an exponential advantage, enabling us to think about a usage in parallel of DB-QITE to construct ground states of more complex hamiltonians. Although the DB-QITE algortihm is not be applicable to complex hamiltonians, the compression algorithm can make this algorithm implementable.

7 Annex

Orthogonal Procrustes problem We want to find the unitary matrix $\tilde{U}$ which minimizes:

$$\min_{\tilde{U}\in\mathcal{U}(n)} \|U - \tilde{U}\|_F$$

we define the Frobenius norm:

$$||U - \tilde{U}||^2 = \mathrm{Tr}((U - \tilde{U})(U^\dagger - \tilde{U}^\dagger))$$

We can then compute:

$$||U - \tilde{U}||^2 = \mathrm{Tr}(UU^\dagger) - 2\mathcal{R}[\mathrm{Tr}(U^\dagger\tilde{U})] + \mathrm{Tr}(\tilde{U}\tilde{U}^\dagger) = \mathrm{Tr}(UU^\dagger) - 2\mathcal{R}[\mathrm{Tr}(U^\dagger\tilde{U})] + n$$

where $\mathrm{Tr}(\tilde{U}\tilde{U}^\dagger) = \mathrm{Tr}(I_n) = n$. As n and $\mathrm{Tr}(UU^\dagger)$ are fixed, to minimize the norm, we have to maximize $\mathcal{R}[\mathrm{Tr}(U^\dagger\tilde{U})]$. The SVD decomposition of U is:

$$U = W\Sigma X^\dagger \qquad \Longrightarrow \qquad U^\dagger = X\Sigma W^\dagger$$

with W and X two unitary matrices. The goal is to construct $\tilde{U}$ in order to maximize:

$$\mathcal{R}[\mathrm{Tr}(X\Sigma W^\dagger\tilde{U})] = \mathcal{R}[\mathrm{Tr}(\Sigma W^\dagger\tilde{U}X)]$$

Let $\Omega = W^\dagger\tilde{U}X$. It is unitary as a product of unitary matrices We have:

$$\mathcal{R}[\mathrm{Tr}(U^\dagger\tilde{U})] = \mathcal{R}[\mathrm{Tr}(\Sigma\Omega)] = \sum_i \sigma_i\mathcal{R}[\Omega_{ii}]$$

As Ω is unitary, $|\Omega_{ii}|^2 \leq 1$ so the coefficients $\mathcal{R}[\Omega_{ii}]$ are maximum for $\Omega = I_n$. From $\Omega = W^\dagger\tilde{U}X$, we find $\tilde{U} = WX^\dagger$. □

References

[1] M. Gluza, J. Son, B. H. Tiang, Y. Suzuki, Z. Holmes, and N. H. Y. Ng, *Double-bracket quantum algorithms for quantum imaginary-time evolution*, 2024. arXiv: 2412.04554 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2412.04554.

[2] Y. Alexeev, M. Amsler, and M. A. B. et al., "Quantum-centric supercomputing for materials science: A perspective on challenges and future directions.," 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2312.09733.

[3] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, "Quantum computational chemistry," *Rev. Mod. Phys.*, vol. 92, p. 015 003, 1 Mar. 2020. DOI: 10.1103/RevModPhys.92.015003.

[4] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe, "The power of quantum systems on a line," *Communications in Mathematical Physics*, vol. 287, no. 1, pp. 41–65, Jan. 2009, ISSN: 1432-0916. DOI: 10.1007/s00220-008-0710-3.

[5] N. Robertson, A. Akhriev, J. Vala, and S. Zhuk, "Approximate quantum compiling for quantum simulation: A tensor network based approach," *ACM Transactions on Quantum Computing*, vol. 6, no. 3, pp. 1–15, May 2025, ISSN: 2643-6817. DOI: 10.1145/3731251. [Online]. Available: http://dx.doi.org/10.1145/3731251.

[6] J. R. Garrison, K. Marshall, I. Shehzad, *et al.*, *Qiskit addon: Approximate Quantum Compilation with Tensor Networks*, https://github.com/Qiskit/qiskit-addon-aqc-tensor, 2024. DOI: 10.5281/zenodo.14064353.

[7] Qiskit, *EfficientSU2*, https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.EfficientSU2.

[8] M. B. Hastings, "An area law for one-dimensional quantum systems," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2007, no. 08, P08024–P08024, Aug. 2007, ISSN: 1742-5468. DOI: 10.1088/1742-5468/2007/08/p08024. [Online]. Available: http://dx.doi.org/10.1088/1742-5468/2007/08/P08024.