Simulating lattice spin models on today's quantum computers

DESY Summer Student Programme, 2022

Mateusz Salamon
The University of Manchester, United Kingdom

Supervisor
Karl Jansen, CQTA

November 20, 2022

Abstract

The project aimed at testing whether currently available quantum computers can be used to obtain meaningful physics results about lattice spin models. Therefore, the transverse-field Ising model was investigated with the VQE algorithm on both an ideal, simulated quantum computer and a real, noisy IBMQ device. Results from an ideal quantum computer showed great agreement with exact diagonalisation with relative energy error of the order of 1% and fidelities reaching above 99%. The IBMQ quantum computer achieved lower fidelities of about 90% and energy errors of the order of 10%, but the results showed correct trends and could be used to deduce the existence of a phase transition. It was concluded that current quantum computers cannot be yet used for precision computations, but can still be useful for investigating the qualitative behaviour of some lattice models.

Contents

1 Introduction

1.1 Quantum computation

Quantum computers are the future generation of computers. They utilise purely quantum phenomena of superposition and entanglement to gain an advantage over classical computers. Whether they will be able to offer a significant speed-up in practically useful computations still remains an open question, but there are many signs that quantum computers may be able to perform some tasks that would take their classical counterparts ages to complete [1]. Because of that, there has recently been a lot of research into the development of both the hardware and the software for quantum computers.

In the near future, however, quantum computers will not yet be able to perform all the computations we expect them to excel at. This is because the current pace of development will only allow us to use a restricted number of qubits in the coming years. Additionally, current quantum devices are characterised by high levels of noise, associated with different types of errors. To describe this era of quantum computers, J. Preskill coined a term of the 'NISQ' era, which stands for 'Noisy Intermediate-Scale Quantum' devices era [2]. Researchers are nevertheless trying to find ways in which these NISQ devices could still be used in the coming years in order to utilise the 'quantum advantage' as soon as possible to accelerate research.

1.2 Variational quantum algorithms

One of the most promising ways to make use of the NISQ-era quantum computers is via the use of variational quantum algorithms [3]. These are hybrid quantum-classical algorithms, meaning that part of the work is done on a classical computer and part on a quantum computer. The underlying principle here is similar to machine learning; variational quantum algorithms use a parametrised quantum circuit to perform some calculation and a classical computer is responsible for updating the parameters to find the minimum of some cost function. The most prominent of these type of algorithms are the Quantum Approximate Optimisation Algorithm (QAOA) and the Variational Quantum Eigensolver (VQE), which is the algorithm used in this project.

1.3 Aims of the project

The aim of this project was to use the hybrid VQE algorithm on real quantum computers to analyse the transverse-field Ising model and compute its ground state along with properties such as energy and magnetisation. This was firstly done using a classical simulation of an ideal quantum computer. Next, the effects of noise were investigated and then the computation was performed using real hardware from IBMQ. Finally, the Variational Quantum Deflation (VQD) algorithm was used to find the first excited state of the model. All the computations for the project were done using the Qiskit Python package within the Jupyter Notebook environment.

2 Theory

2.1 Transverse-field Ising model

The transverse-field Ising model (TFIM) can be described as a quantum version of the classical Ising model, which is the simplest model of magnetism. The classical Ising model describes a lattice of 'spins', which can only point in the positive ('up' or $|1\rangle$ state) or negative ('down' or $|0\rangle$ state) z-direction. They interact with their nearest neighbours through a coupling term and a magnetic field of strength h is applied in the z-direction.

The TFIM differs from the classical Ising model in that the 'spins' are now quantum objects, i.e. they can be in state $|0\rangle$, $|1\rangle$ or any superposition of these two. The magnetic field h is then applied in the (transverse) x-direction instead. Hence, the transverse-field Ising model can be described by the Hamiltonian

$$H = -\sum_{\langle i,j \rangle} Z_i Z_j - h \sum_i X_i, \tag{1}$$

where Z_i and X_i are, respectively, Pauli Z and X operators acting on 'spin' on site i, and $\langle i, j \rangle$ denotes a sum over nearest neighbours. Since the Pauli operators in different directions do not commute with each other, this model cannot be described classically and will exhibit some quantum phenomena. Indeed, we can observe a quantum phase transition at zero temperature; the transition is between a ferromagnetic and a paramagnetic phase as the magnetic field is varied. For high fields the magnetic term dominates, so spins prefer to be aligned in the x-direction; the magnetisation per spin in z drops to 0 and rises to 1 in the x-direction. Similarly, for low fields the coupling term dominates; the magnetisation in z-direction is close to 1 and in the x-direction it is 0.

Notice though, that for low fields, the states of all spins up and all spins down (corresponding to +1 and -1 magnetisations per spin in z) are degenerate. Thus, the ground state would be an equal superposition of these two, giving zero magnetisation and leading to the model showing no signs of a phase transition. In order to avoid this, we can add a small 'bias' magnetic field h_z in the z-direction by adding a term

$$-h_z \sum_i Z_i \tag{2}$$

to the Hamiltonian H. This should guarantee (in most cases) that the spins align in the positive z-direction (for $h_z > 0$), but should not influence the high field results.

2.2 Variational Quantum Eigensolver - VQE

The VQE algorithm was first described in [4]. It is based on the variational principle from quantum mechanics, which states that for any wavefunction $|\psi(\theta)\rangle$ the expectation value of a Hamiltonian H is always greater than or equal to the ground state energy, E_0:

$$\langle \psi(\theta)|H|\psi(\theta)\rangle \geq E_0. \tag{3}$$

Here, we denote by θ a set of parameters that can be varied to change the wavefunction. Thus, by looking for a set of parameters that minimise the expectation value, we can find an estimate of the ground state $|\psi_0\rangle$ and its energy E_0.

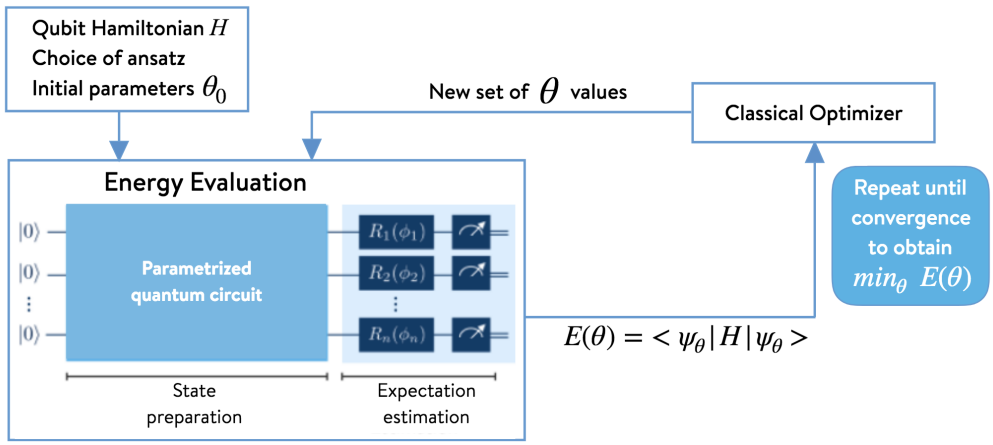

Figure 1: A schematic showing the steps of the VQE algorithm. Image taken from: *http://openqemist.1qbit.com/docs/vqe_microsoft_qsharp.html.*

A schematic of the algorithm is depicted in Fig. 1. The algorithm employs a quantum computer to evaluate the expectation value of the Hamiltonian for some trial, parametrised wavefunction, called the 'anstaz'. Next, based on the result a classical optimiser decides on a new set of parameters to be tested, which are then fed to the quantum circuit. When the global minimum is found, the energy should be close to E_0. The final wavefunction should also be similar to the true ground state wavefunction. In practice, achieving these goals requires a good, careful choice of both the ansatz and the optimiser.

2.3 Variational Quantum Deflation - VQD

The VQE algorithm can also be extended to find the low-lying excited states of a quantum Hamiltonian. This is called Variational Quantum Deflation - VQD and was first introduced in [5]. It is based on the fact that the eigenfunctions of any physical Hamiltonian are orthogonal. The algorithm is basically the same as the VQE; the only thing that changes is the Hamiltonian, which has an extra term added to penalise the ground state. If we want to find the first excited state of Hamiltonian H_0, the VQD algorithm proceeds as follows:

1. Use standard VQE to find the ground state wavefunction - $|\psi_0\rangle$
2. Compute a modified Hamiltonian:

$$H_1 = H_0 + \beta|\psi_0\rangle\langle\psi_0| \tag{4}$$

3. Use VQE again to find the ground state of the modified Hamiltonian - $|\psi_1\rangle$
4. $|\psi_1\rangle$ is the first excited state of H_0

The additional parameter β has to be chosen carefully, so that it is larger than the gap between the ground and the first excited state. If the gap is not known, β can be chosen in a self-correcting manner as described in Section 3 of [5]. The steps 1. to 4. can be repeated with $|\psi_1\rangle$ and H_1 instead of $|\psi_0\rangle$ and H_0 to find the second excited state, etc.

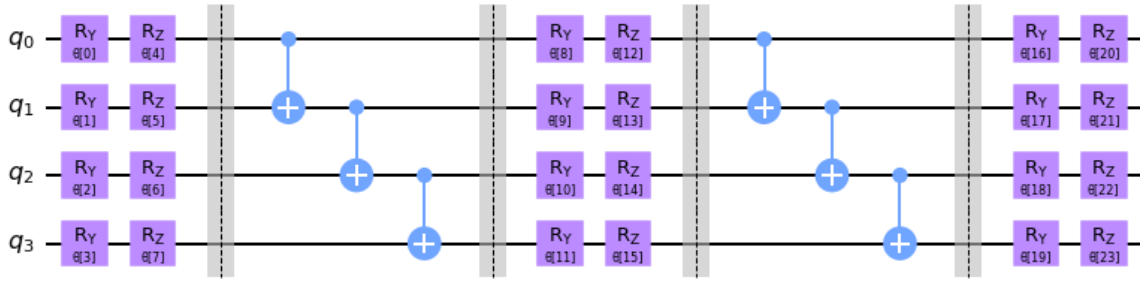

Figure 2: The 'RyRz' ansatz with 'linear' entanglement type and depth of 2 layers that was used to obtain the results shown in this report. The circuit was generated by Qiskit's `EfficientSU2` function.

The VQD algorithm has similar performance as the VQE, but requires usually about twice as many layers in the ansatz. The largest issue with VQD seems to be the fact that it cannot differentiate between degenerate or close-lying states very well, leading to low fidelities, even when the energy is as expected.

3 Methods

3.1 Ansatz

For the VQE ansatz, I have used a Qiskit's function called `EfficientSU2`, which returns a parameterised quantum circuit that can be implemented efficiently on quantum hardware. Each such circuit consists of a number of parameterised single-qubit gate layers and a number of two-qubit CNOT gates between the layers to enable entanglement and model the correlations better. The function allows the user to change many parameters; the most important of those are the gate types used in each layer, the number of layers (depth) and the type of entanglement. The gate types investigated were either only Ry gates (then we have an 'Ry' ansatz) or a combination of Ry and Rz gates (resulting in an 'RyRz' ansatz). The entanglement types could be 'linear', 'full' or 'circular'. The depth of the circuit was investigated in the range from 1 to 4. Adding more layers should in principle enable the ansatz to cover more of the Hilbert space (increase its 'expressibility'), but it also adds more parameters to the circuit, making it more difficult for the optimiser to find the global minimum.

After some investigations, I have chosen to use an 'RyRz' ansatz with 'linear' entanglement and depth of 2 layers. The circuit for this ansatz is shown in Fig. 2. This gave in total 24 parameters that needed to be found, which was still manageable for the optimiser. The 'linear' type of entanglement seemed to give the cleanest results when using noise models. This is probably due to the fact that this scheme uses the fewest two-qubit CNOT gates, which are often quite noisy. The choice of the RyRz ansatz was dictated by a bug in the Qiskit Runtime framework that enables efficient running of the VQE algorithm on IBMQ hardware. Unfortunately, the bug made it impossible to run the Ry ansatz. For consistency I had to choose RyRz ansatz also for the ideal case. Having said that, there was not much difference in terms of accuracy of the results from these two types of ansätze

in simulations with a noise model. The Ry ansatz might have performed slightly better, but the reason could well have been just the lower number of parameters.

3.2 Classical optimiser

The choice of an optimiser was very difficult to make, since they perform differently under different conditions. Thus, I used one optimiser for an ideal quantum computer simulation and another one for noisy simulations or real hardware runs. Each of the optimisers had an argument that determined the maximum number of iterations allowed. For the ideal simulations, I have used either COBYLA or SLSQP optimisers (using their Qiskit's implementations, which are wrappers around scipy's functions). In general, COBYLA seemed to be able to achieve better accuracy, but only with a high number of iterations. I have used the SPSA optimiser for noisy calculations, since SLSQP failed to find a minimum in presence of noise and COBYLA required many more iterations for a comparable accuracy.

3.3 Measurement error mitigation

Since the currently available quantum computers are very noisy, their results are affected by many errors that can occur during the calculation. There exist some ways to eliminate these errors completely in the form of quantum error correction codes. However, these approaches require many more qubits than it is currently possible to use.

There is, however, another class of errors that can influence the results. These are the measurement (or readout) errors. They occur at the end of a quantum computer's calculation, during the measurement of the final state. Fortunately, these errors are easier to mitigate for and this can be implemented easily (for small systems) to increase the accuracy of the results from noisy hardware. I have implemented measurement error mitigation, using the built-in functionality of the Qiskit package. The mitigation is done by measuring the outcomes of the quantum computer (the same as the one used for actual measurements) when it was prepared in a basis state. This is repeated for every possible basis state and from these results, the calibration matrix is constructed. Finally, the inverse of the calibration matrix is calculated and applied to the vector representing the outcomes of the actual measurements. This operation gives the mitigated measurement results, with almost all of the contribution from measurement errors eliminated.

4 Results

4.1 Ground state

4.1.1 Ideal quantum computer

I used the *Aer* simulator backend from the Qiskit package to simulate the TFIM for a system of 4 spins in a chain with periodic boundary conditions. The SLSQP optimiser was chosen for the classical part of computation with maximum number of iterations set

to 1000 and tolerance of 1e-7. The applied bias magnetic field was $h_z = 0.1$, which is a bit high, but it was necessary due to a small system size. I have computed the energy as well as the magnetisations per spin in both x and z directions. Since 4 spins is a small number, the solutions to the problem can be easily obtained by a classical computer that performs diagonalisation of the Hamiltonian. This was done using the `eigh` function from the numpy package. By comparing the VQE results to the exact diagonalisation, I was able to additionally compute also the fidelities and relative energy errors.

The results are shown in Fig. 3. The solid lines correspond to the exact calculation,

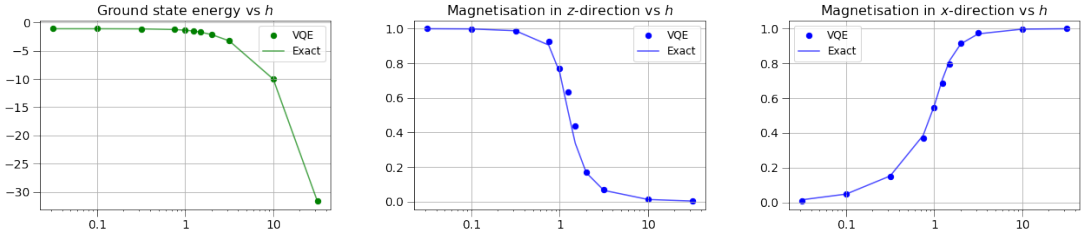

Figure 3: Results from the ideal quantum computer simulation (dots) plotted against magnetic field h and compared to the exact calculations (solid lines). Left: ground state energy; middle: magnetisation per spin in the z-direction; right: magnetisation per spin in the x-direction.

while the dots indicate the VQE results. It can be seen that the VQE results are in a very good agreement with the exact results. This is confirmed by the plots in Fig. 4, which show very low errors on energy, of the order of 1%, and high fidelities, exceeding 99% for most of the data points. These results are very promising and indicate that fault-tolerant

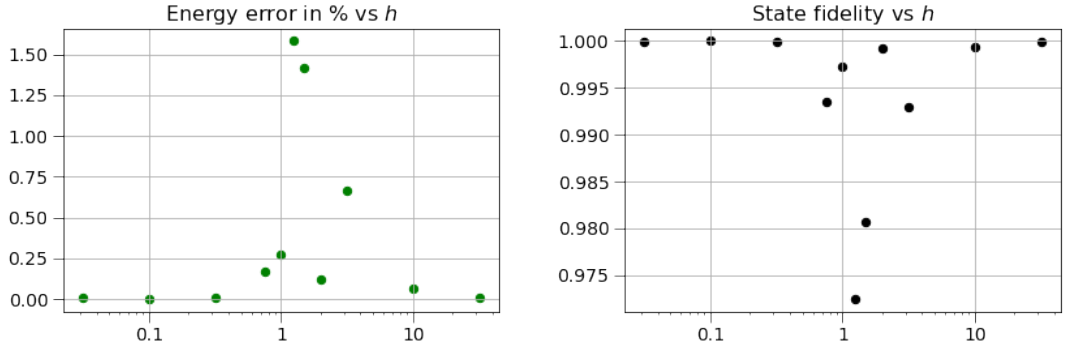

Figure 4: Plots showing the accuracy of the results from the ideal quantum computer simulation plotted against magnetic field h. Left: Relative energy errors; right: fidelities of the found ground states.

quantum computers of the future could be used to compute the properties of the ground states of some lattice Hamiltonians, which has many potential applications.

4.1.2 Real IBMQ hardware

After completing the ideal simulations, I added some custom noise models to them to see the effect that noise can have on VQE. While some low and medium levels of noise still gave acceptable results for the energy, the magnetisations were much more difficult to get right. Next, I tried simulations with the noise model of the five-qubit IBMQ *Belem* device, which is the device that was used later to run the calculations on a real quantum computer. Based in these simulations the parameters for the real hardware runs were chosen. I decided to use the SPSA optimiser with a maximum of 1000 iterations. The bias magnetic field was again set to 0.1. To run the VQE algorithm on the IBMQ quantum computers the Qikit IBMQ Runtime package was used. It allows for a more efficient and faster execution of jobs on the IBMQ quantum computers [6]. The most important advantage from a user perspective is the fact that subsequent iterations of the algorithm for the same setup are collected into a single job, which makes managing the results much easier.

In Fig. 5 the results from the real hardware runs are shown (represented by dots), again compared to the exact calculations (solid lines). The relative energy errors and

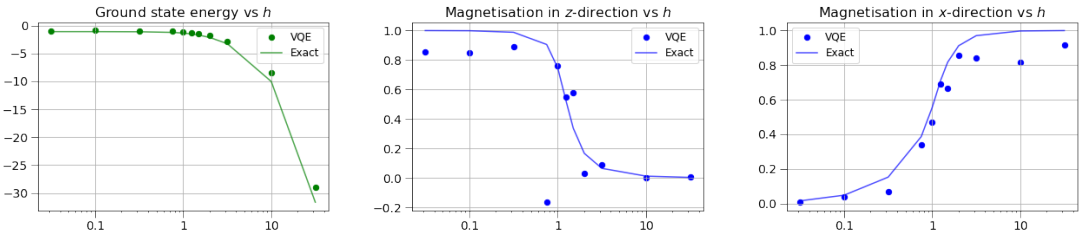

Figure 5: Results from the real IBMQ hardware run (dots) plotted against magnetic field h and compared to the exact calculations (solid lines). Left: ground state energy; middle: magnetisation per spin in the z-direction; right: magnetisation per spin in the x-direction.

fidelities are shown in Fig. 6. The results are not nearly as good as the ones from the

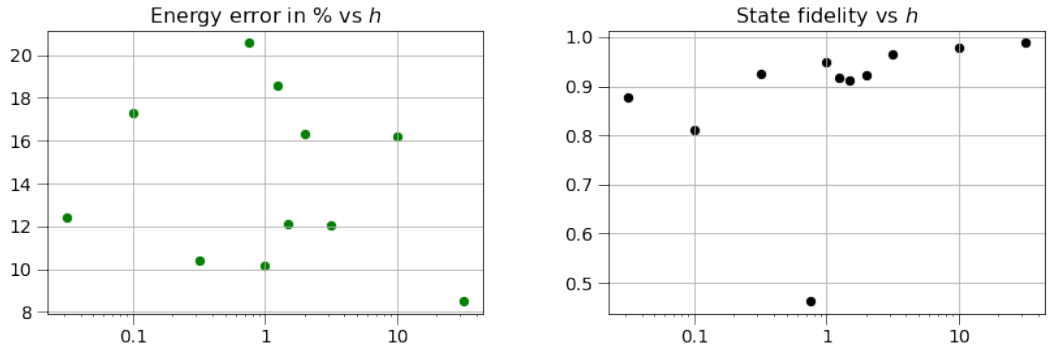

Figure 6: Plots showing the accuracy of the results from the real IBMQ hardware run plotted against magnetic field h. Left: Relative energy errors; right: fidelities of the found ground states.

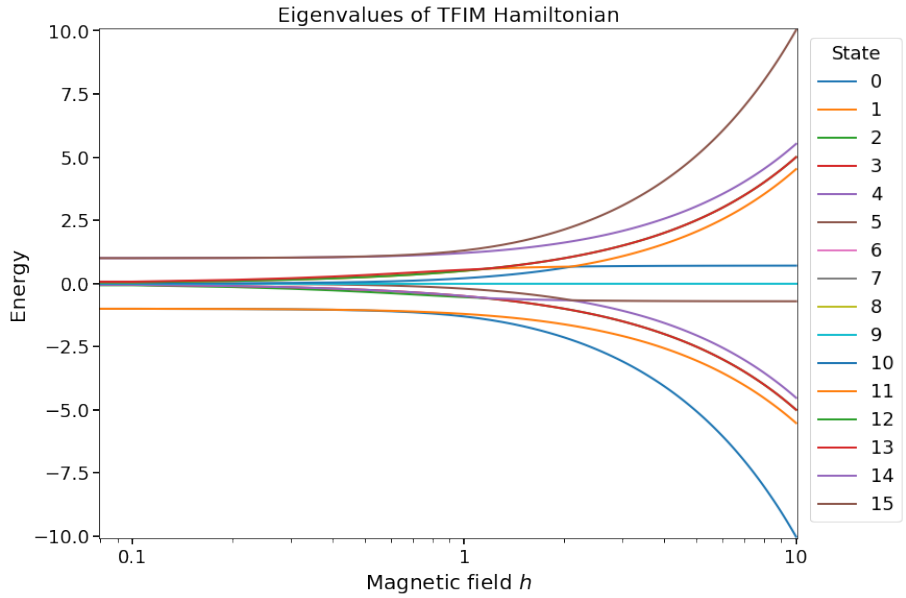

Figure 7: A plot showing the spectrum of the eigenvalues of the TFIM Hamiltonian in Eq. 1 in a chain of 4 spins for different values of the magnetic field h. Each line corresponds to a different state.

ideal simulation. The energy plot looks mostly accurate, but the relative errors are in the range of 10-20%. The fidelities are mostly above 90% though, so the results could be assumed to represent the true states quite accurately. Unfortunately, the magnetisation plots do not look as good this time. It can be seen that near the phase transition the VQE struggles, which was to be expected, but there also seems to be an issue with the fully magnetised states. The results do not reach the full +1 magnetisations for low or high fields, even though the fidelities for these data points are rather high. One possible explanation is that, by construction, the magnetisation per spin can never reach values higher than +1. Therefore, all the fluctuations and errors would result in underestimating the value at these data points, whereas for other data points the fluctuations can be both 'up' or 'down', which should balance out in the end, giving a closer estimate.

Based on the above, it can be safely concluded that current, NISQ-era quantum computers cannot be really used to do precise calculations of ground states of lattice models using the VQE algorithm as the errors are too large. However, the general trends revealed by the real hardware runs are in fact correct and agree well with the exact diagonalisation. The magnetisation plots clearly indicate the presence of a phase transition at about the value of $h = 1$. Therefore, these devices might still serve a useful purpose in investigating some lattice spin models qualitatively in regimes where classical computers struggle to obtain any results at all.

4.2 First excited state

In the last part of the project, I have also attempted to use the VQD algorithm in order to find the first few excited states of TFIM. This was done again on a simulation of an ideal quantum computer. According to [5], the ansatz used was changed to have 4 layers

STATE	GROUND	1ST EXCITED
Energy	-10.021	-4.926
(Exact)	*-10.025*	*-5.525*
Magnetisation in z	0.000	-0.010
(Exact)	*0.000*	*0.000*
Magnetisation in x	0.998	0.497
(Exact)	*0.997*	*0.498*
Fidelity	**0.993**	**0.001**

Table 1: An example set of results from running VQD on an ideal quantum computer simulation for magnetic field $h = 10$.

instead of just 2. For an unknown reason, the SLSQP optimiser, which worked very well for the ground state, failed to find the right minima for the modified Hamiltonian of Eq. 4. Thus, for this part, the COBYLA optimiser was used. The energy gaps between the different states of TFIM were known from the exact diagonalisation and the parameter β was chosen accordingly to be much larger than the relevant energy gap. Since the first two states of the model are very close in energies for low values of h, as can be seen in the spectrum of the TFIM Hamiltonian shown in Fig. 7, the VQD algorithm was used for $h = 10$.

Unfortunately, it was observed that it was much harder to obtain accurate estimates of the excited states using VQD compared to the VQE algorithm for the ground state. Even though the number of iterations was increased to 5000, the fidelities of the found excited states were extremely low, almost 0 in many cases. Having said that, the results for magnetisations were in good agreement with the exact results and the energies were also quite close to the exact calculations. A typical set of results from the VQD algorithm is shown in Table 1. It is suggested that this is due to the first few excited states being close in energies, even for $h = 10$, as is confirmed by the plot in Fig. 7. Although, it was easy to find the ground state, which is well separated from the other states, the next four states are quite close to each other and it may be problematic for the VQD algorithm to differentiate between them. The correct outcomes for energy and the magnetisations suggest that the state that is found may be a mixture of these four excited states. There is certainly more work to be done on the VQD algorithm and its usefulness for close-lying excited states. It seems possible, however, that it could be used to estimate the energy gap between the ground state and the first few excited states, although more fine-tuning of the parameters would be required.

5 Conclusions

In this project I applied the VQE algorithm to investigate the ground state of the transverse-field Ising model on quantum computers. Both a simulation of an ideal quantum computer as well as a real quantum computer from IBMQ were used. The results from the ideal case show that fault-tolerant quantum computers of the future could be employed to simulate some lattice spin models with good accuracy. On the other hand, currently available

NISQ-era quantum computers, like the IBMQ device used for this project, do not seem to allow for such precise computations. However, they might possibly be a great tool to investigate the qualitative behaviour of some lattice models in regimes inaccessible to classical computers. More research needs to be done in this area to confirm this supposition. Finally, an attempt was made to use the VQD algorithm to find the low-lying excited states of the model using the ideal quantum computer. This was only partially successful due to the proximity in energies of the first few excited states in the transverse-field Ising model. Future work could focus on finding ways to use the VQD more reliably in cases with (almost) degenerate excited states. An ultimate goal would be to implement VQD on real hardware and increase the number of qubits, so that the energy gap could be computed for system sizes which are difficult to solve on classical computers.

Acknowledgements

I would like to thank my supervisor, Karl Jansen, for allowing me to have this invaluable experience of an internship at DESY. I am also grateful to Cenk Tüysüz and Maria Demidik for their continuous help throughout the project. I have learnt a lot during my stay in Zeuthen thanks to your support.

References

[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* (Cambridge University Press, 2010)

[2] J. Preskill, *Quantum Computing in the NISQ era and beyond*, Quantum **2**, 79 (2018).

[3] M. Cerezo, A. Arrasmith, R. Babbush *et al.*, *Variational quantum algorithms*, Nat Rev Phys **3**, 625644 (2021)

[4] A. Peruzzo, J. McClean, P. Shadbolt *et al.*, *A variational eigenvalue solver on a photonic quantum processor*, Nat Commun **5**, 4213 (2014)

[5] O. Higgott, D. Wang, S. Brierley, *Variational Quantum Computation of Excited States*, Quantum **3**, 156 (2019)

[6] Qiskit Development Team, "Qiskit 0.38.0 Documentation" qiskit.org. *https://qiskit.org/documentation/apidoc/ibmq_runtime.html* (accessed 21/09/2022).