

Time Evolution of the Triton Toy Model

Tim Koreman* (2418541)

Abstract

In this report we look at ground state estimation and time dynamics of the triton toy model as defined by Roggero *et al.* (2019) [1]. We use a variational quantum eigensolver (VQE) and quantum phase estimation (QPE) to estimate the ground state of the model. We use these models to study the time evolution of the system using single and multi step trotterization via the three body density and find results consistent with results found in [1]. We also present some preliminary results for the linear response of the model.

1 Introduction

Many of the often mentioned applications of quantum computation require a number of qubits currently not available. In order to utilize current day devices there are a number of algorithms that require fewer qubits. In this report we use two of these algorithms, Variational Quantum Eigensolvers (VQE) and Quantum Phase Estimation (QPE), to study the time evolution of a toy model defined by Roggero *et al.* (2019) [1]. This toy model models neutrino-nucleus scattering. We first give a short description of the model and the algorithms used and then show results from simulations and compare these with results from [1].

2 Model Background

In this report we will focus on the triton toy model as defined by Roggero *et al.* (2019) [1]. This model describes a system with 3 nucleons on a 2x2 lattice, for example triton, a nucleus with 1 proton and 2 neutrons. This model is described by the following Hamiltonian (where the static nucleon is placed at site 1):

$$\begin{aligned} H = & -t \sum_{f=1}^{N_f} \sum_{\langle i,j \rangle} c_{i,f}^\dagger c_{j,f} + 2dtA + U \sum_{i=1}^{N_f} \sum_{f < f'}^{N_f} n_{i,f} n_{i,f'} \\ & + V \sum_{f < f' < f''} \sum_{i=1}^{N_f} n_{i,f} n_{i,f'} n_{i,f''} + U \sum_{f=1}^{N_f} n_{1,f} + V \sum_{f < f'}^{N_f} n_{1,f} n_{1,f'} \end{aligned} \quad (1)$$

with c^\dagger and c creation and annihilation operators respectively, $n \equiv c^\dagger c$ the number operator and t the parameter describing the kinetic energy, U , V describe the strength of two- and three-body interaction respectively. A denotes the number of active particles and d the dimensionality of the system. The sum over i,j sums over the lattice site, the sum over f sums over the fermionic species with N_f the number of fermionic species in the system.

We now focus on a system with $N_f = 2$ and $A = 2$ which reduces the Hamiltonian to:

$$H = -t \sum_{f=1}^{N_f} \sum_{\langle i,j \rangle} c_{i,f}^\dagger c_{j,f} + 8t + U \sum_{i=1}^{N_f} \sum_{f < f'}^{N_f} n_{i,f} n_{i,f'} + U \sum_{f=1}^{N_f} n_{1,f} + V \sum_{f < f'}^{N_f} n_{1,f} n_{1,f'}$$

*e-mail: t.koreman@umail.leidenuniv.nl

Now we use 2 qubits to store the position of each non static nucleon on the lattice in the following way (see figure 1):

$$|1\rangle = |00\rangle \quad |2\rangle = |01\rangle \quad |3\rangle = |10\rangle \quad |4\rangle = |11\rangle$$

where 1 through 4 label the lattice sites where here we have a static nucleon at site 1. Since we are encoding the position of our particles this is a first-quantized representation of the system.

Removed sourced image.

Figure 1: Qubit mapping as used in this report for a single fermion. The static particle is placed at site 1. Source: [1].

The first term in the Hamiltonian describes the hopping of the particles between lattices sites. It should take a particle on a site to one of the two connected sites. For example for $|2\rangle$ there should be a term mapping it to $|1\rangle$ and a term mapping it to $|4\rangle$. This condition is satisfied for all sites by the matrix:

$$H_{\text{hop}}^A = -2t \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} = -2t(X_1 \otimes \mathbf{1}_2 + \mathbf{1}_1 \otimes X_2)$$

Now the total hopping term for both particles is given by:

$$\begin{aligned} H_{\text{hop}} &= H_{\text{hop}}^A \otimes \mathbf{1}_B + \mathbf{1}_A \otimes H_{\text{hop}}^B = -2t(X_1 \otimes \mathbf{1}_2 + \mathbf{1}_1 \otimes X_2) - 2t(X_3 \otimes \mathbf{1}_4 + \mathbf{1}_3 \otimes X_4) \\ &= -2t(X_1 + X_2 + X_3 + X_4) \end{aligned}$$

dropping the identity operators in the last line for notational clarity. For the other terms we look at the interaction part of the Hamiltonian. The element corresponding to both particles on site 1 is given by $3U + V$, the element for both particles on the same site other than 1 is given by U . We encode this part in the qubits by using the following sets of operators:

$$M_k = \frac{\mathbf{1} - Z_k}{2} \quad \Pi_k = \frac{\mathbf{1} + Z_k}{2}$$

We use this to write terms like:

$$(3U + V) |11\rangle \langle 11| = (3U + V) |0000\rangle \langle 0000| = (3U + V) \Pi_1 \Pi_2 \Pi_3 \Pi_4$$

and other terms follow in similar ways. Using this we can write the Hamiltonian (given by equation 1) in terms of Pauli operators (where we again drop the identity operators for notational clarity):

$$\begin{aligned} H = & 8t + \frac{3U}{4} + \frac{V}{16} - 2t \sum_k X_k + \frac{V + 4U}{16} \sum_k Z_k + \frac{V + 4U}{16} (Z_1 Z_2 + Z_3 Z_4 + Z_1 Z_3 + Z_2 Z_4) + \\ & + \frac{V}{16} (Z_1 Z_4 + Z_2 Z_3) + \frac{V}{16} \sum_{i < j < k} Z_i Z_j Z_k + \frac{V + 4U}{16} Z_1 Z_2 Z_3 Z_4 \end{aligned} \quad (2)$$

which in the limiting case $V = -4U$ reduces to:

$$H = 8t + \frac{U}{2} - 2t \sum_k X_k - \frac{U}{4} (Z_1 Z_4 + Z_2 Z_3) - \frac{U}{4} \sum_{i < j < k} Z_i Z_j Z_k \quad (3)$$

In order to quantify the time evolution of this model we define the three body contact density $C_3(t)$ as the probability of three particles to be on the same site at time t . Which in our model represents the state with both active particles on the same site as the nuclei. In our representation this corresponds to the probability our system is in the state $|0000\rangle$.

3 Technical Background

3.1 Variational Quantum Eigensolver

In order to study the time evolution of the triton model we want to find a state reasonably close to the ground state of the system. In order to do this we can use the variational quantum eigensolver (VQE) algorithm. A VQE uses a parametrised state $|\psi(\theta)\rangle$, usually called the ansatz, measure the energy of that ansatz and optimize the parameters θ to find the lowest energy for the ansatz.

To measure the energy of a state we need to find $\langle\psi(\theta)|H|\psi(\theta)\rangle$, which is the expectation value of H with respect to the state. Our Hamiltonian is given by a sum of Pauli strings so our expectation value can be found by:

$$\langle\psi(\theta)|H|\psi(\theta)\rangle = \sum_i h_i \langle\psi(\theta)|\sigma_i|\psi(\theta)\rangle$$

where h_i denotes the coefficients and σ_i the Pauli strings. Now we find the expectation values of the Pauli strings by measuring the state in the eigenbasis of the Pauli string. Since we measure our qubits in the computational basis we have to do nothing for Pauli string containing only Z gates. For X gates we have to rotate the state to the z-axis by applying a H gate. All elements of Pauli strings have the same eigenvalues (+1 and -1) so to calculate the expectation value of such a string we need to know the parity of strings (ie. on average how often do we measure the state corresponding to eigenvalue -1). To do this we encode the parity into a ancilla qubit and measure that ancilla using circuits like given in figure 2

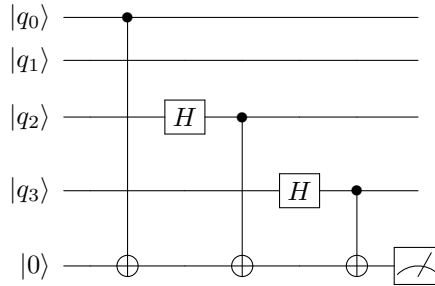


Figure 2: Circuit used to measure the parity of a term given by $Z_0 \mathbf{1}_1 X_2 X_3$. The state beginning in the state $|0\rangle$ is the ancilla qubit. Note that this string does not occur in our Hamiltonian, it is purely illustrative.

After determining the expectation value at certain values of the parameters we want to optimize the parameters to find a minimum in the parameter space. For this report the Nelder-Mead algorithm was implemented to perform this optimization. A detailed description of this algorithm goes beyond the scope of this report, see for example Lagarias *et al.* (1998) [2] for a further description. Important to note is that this algorithm has 4 parameters, which we call here ρ , χ , γ and σ , which control various steps of the algorithm.

3.2 Quantum Phase Estimation

A different method to perform ground state preparation is by use of Quantum Phase Estimation (QPE). The QPE algorithm uses a series of a controlled version of the time evolution unitary e^{-iHt} . This unitary is applied to some initial state and controlled by a collection of L ancilla qubits. This collection of ancillas is prepared in the $|+\rangle$ state. The repeated application of the unitary to our initial state projects it to a eigenstate of the unitary, and therefore of the Hamiltonian. In order to determine to which eigenstate the algorithm was projected we apply inverse QFT to our ancilla register and measure them. To use

QPE to prepare the groundstate we use the algorithm and only keep the states where the ancilla register measures to the desired eigenstate.

3.3 Time Evolution

To study the time evolution from 0 to a time t of our system given by Hamiltonian H we need the unitary operator given by $U = e^{-iHt}$. In order to construct this unitary we use the first order Trotter approximation to write:

$$e^{-it \sum_j h_j \sigma_j} = \Pi_j e^{-it \sigma_j} + \hat{O}$$

where the \hat{O} operator denotes a error arising from the non commutativity of Pauli strings which is of order 1 in this case. In order to visualize this error we look at the difference between time evolution by e^{-iHt} and by $\Pi_j e^{-it \sigma_j}$. We do this by using numpy to calculate the exact¹ matrix exponential of both expressions, using those matrices to do time evolution and then look at their respective three body contact densities (as defined in section 2).

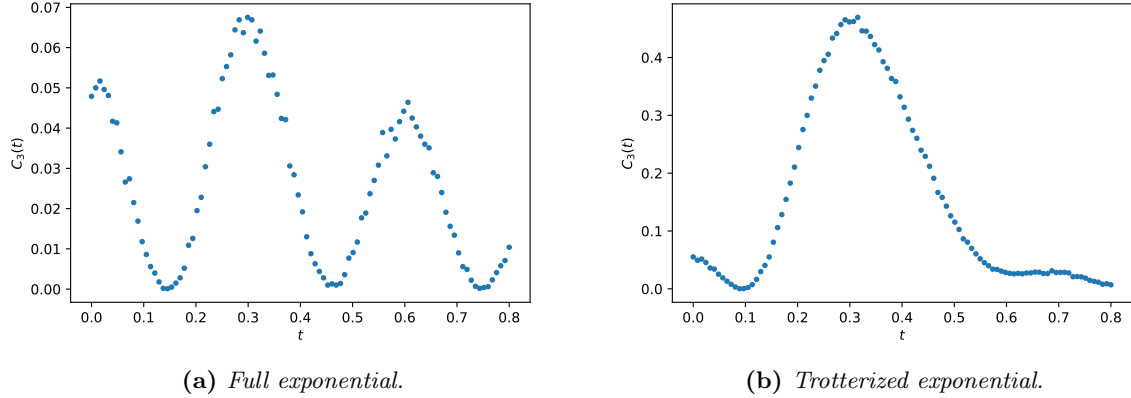


Figure 3: Three body contact density for a initial state given by the ansatz given in figure 4 with $\theta = 1.4$ and $\phi = 0.4$ under the time evolution under the Hamiltonian given by equation 3 by: a) full matrix exponential and b) trotterized exponential. Note the difference between y-axis ranges.

The trotterization however allows us to implement the time evolution using the simple gates we have access to on quantum computers. To reduce the error we can use multiple trotterization steps to arrive at the time t by using:

$$e^{A+B} = \left[e^{(A+B)/k} \right]^k = \left[e^{A/k+B/k} \right]^k = \left[e^{A/k} e^{B/k} + O_k \right]^k$$

where the error O_k now scales as $O(\|A\| \cdot \|B\|/k^2)$. So when we use multiple trotter steps the error decreases.

3.4 Sources of Error

The VQE begins with a choice of an ansatz. Since it is not given that the chosen ansatz can represent the ground state there is an error from this difference. To optimize the VQE we need to determine the exception value of the Hamiltonian. This is done by repeatedly measuring the system and determining

¹In general matrix exponentials are not exact since they are defined as a infinite (Taylor) series, but the approximation by numpy has a smaller error compared to the other elements in the experiment.

the average. In this averaging we have a statistical error. The optimization involves searching a parameter space for the lowest energy. In this search it is likely we will end up at some minimum other than the optimal minimum which also gives an error in the ground state estimation.

For the time evolution by we have the error due to the trotterization. For the determination of the three body contact density we also have statistical error in the estimation of the probability for the system to be in the state $|0000\rangle$.

Since our simulations are run at a finite machine precision we also have some error due to this but this error is probably far smaller than the other errors.

3.5 Linear Response

An interesting extension to the system described above is to observe how it reacts to perturbations. We here follow the methods as described by Roggero & Carlson (2018) [3]. We study how our system reacts to perturbations given by an operator O . We are interested in the Dynamical Response Function for this operator which is defined as:

$$S_O(\omega) = \sum_{\nu} |\langle \psi_{\nu} | O | \psi_0 \rangle|^2 \delta(E_{\nu} - E_0 - \omega)$$

We estimate this function by using three unitary operators: U_G which prepares the initial state, U_O which implements time evolution under O for a short time γ en U_t which implements time evolution under the Hamiltonian. We use these unitary operators and an ancilla qubit to prepare the system in a state $|\psi_O\rangle \sim O |\psi_0\rangle$. We can then use QPE as described above to extract the probabilities $|\langle \psi_{\nu} | O | \psi_0 \rangle|^2$ and use those to estimate $S_O(\omega)$.

3.6 Implementation

The quantum circuits as described in this report were implemented and simulated using cirq 0.8.0.dev [4] where SymPy 1.5.1 [5] was used for symbolic computations. Our implementation was benchmarked against implementations found in OpenFermion 0.11.0 and OpenFermion-Cirq 0.4.0.dev [6]. Plots were made using Matplotlib 3.1.3 [7]. To perform various numerical functions NumPy 1.18.1 [8] was used.

4 Results

For all plots the averaging was performed using 10^4 measurement samples unless stated otherwise.

4.1 Ground State Preparation

We first try to find a state close to the ground state by using the variational quantum eigensolver (VQE) algorithm as described in section 3.1. We use the ansatz as postulated by Roggero *et al.* (2019) [1] given in figure 4.

Removed sourced image.

Figure 4: Ansatz as used for the optimization with parameters θ and ϕ . Source: [1].

We first start at the same values ($U = -7$, $V = 28$, $t = 1$) of the parameters in the Hamiltonian as used in [1]. We use the Nelder-Mead algorithms with parameters $\rho = 1$, $\chi = 2$, $\gamma = 1/2$ and $\sigma = 1/2$ for the optimization. Our implementation of the VQE algorithm finds a ground state energy given in table 1. In this table we compare this value with the value found by the VQE as implemented in OpenFermion and with the exact ground state energy found using numpy.

Table 1: *Ground state energy as found using our implementation of the VQE and OpenFermion compared to the exact ground state energy.*

	Implemented VQE	OpenFermion	Exact ground state
Ground state energy	-4.355	-4.415	-4.843

To see how the difference between the ground state energy found using VQE and the exact ground state energy (further called ΔE) varies for different values of U , V and t we apply the VQE at a range of values for those parameters. When we vary a parameter we keep the other two fixed at the values previously defined. We run the algorithm a number of times at each set of the parameters and pick the lowest found error to mitigate the problem of the optimization getting stuck in some local minimum.

In figure 5 we see the error for a range of U values. We see that for positive values this error increases sharply.

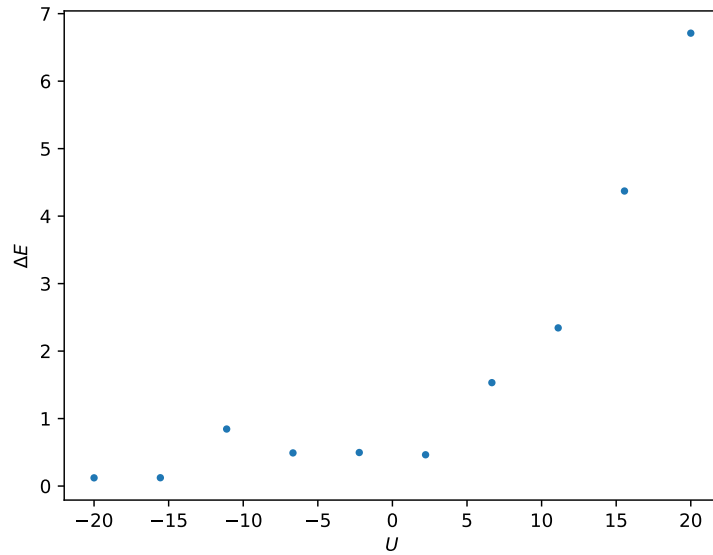


Figure 5: *Difference between the lowest energy found by the VQE and the exact ground state energy at a range of values of the parameter U with $V = 28$ and $t = 1$.*

In figure 6 we see the error for a range of V values. We see again that for positive values this error increases sharply.

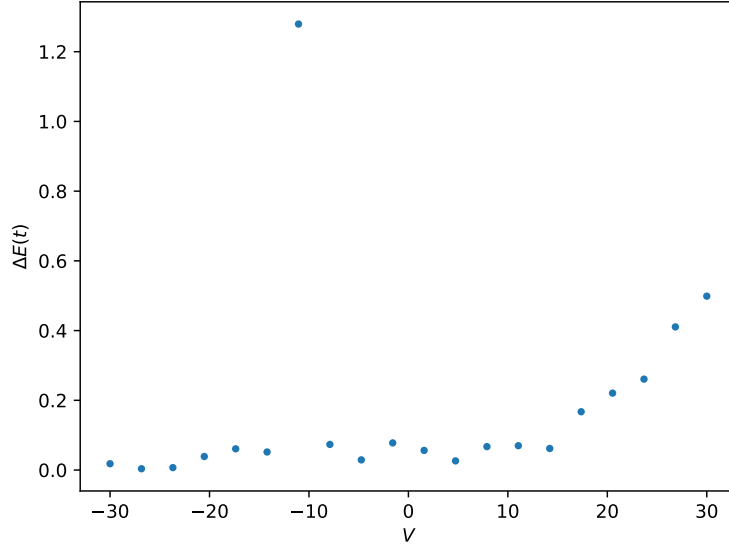


Figure 6: Difference between the lowest energy found by the VQE and the exact ground state energy at a range of values of the parameter V with $U = -7$ and $t = 1$. The outlier is probably a point where the optimizer got stuck in a local minimum.

In figure 7 we see the error for a range of t values. We see that the error peaks around $t = 0$ and decreases for increasing $|t|$.

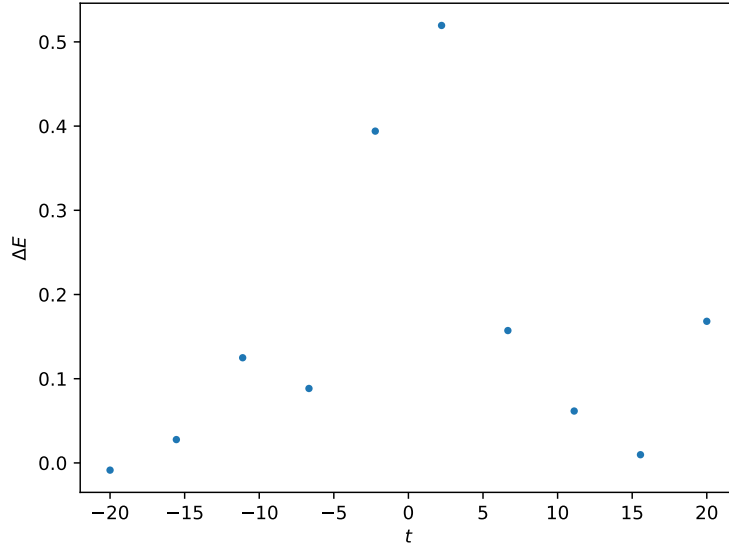


Figure 7: Difference between the lowest energy found by the VQE and the exact ground state energy at a range of values of the parameter t with $V = 28$ and $U = -7$. The error peaks around $t = 0$ and tends towards as zero as $|t|$ increases.

We can also use Quantum Phase Estimation (QPE) to prepare a groundstate. We do this by applying the algorithm as described in section 3.2. We start from the ansatz as given in figure with the parameters

as found by the VQE. In figure 8 we see a histogram for the phase as determined by measuring the ancilla qubits. We see that our QPE projects a fraction of our runs to the ground state (given by the blue line in the figure).

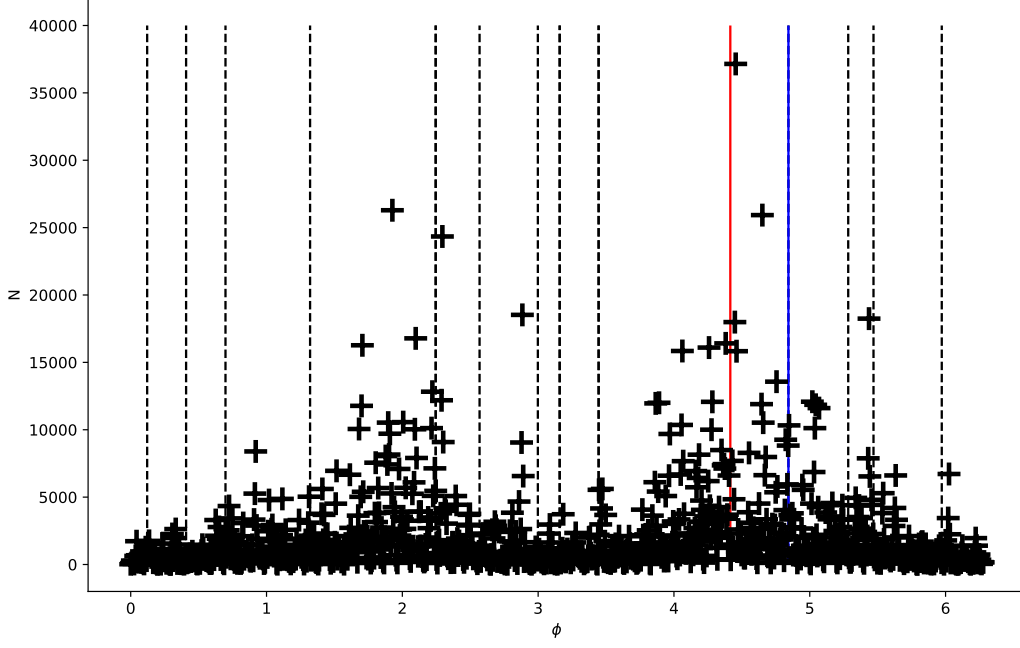


Figure 8: Histogram of the ancilla measurements for the QPE algorithm for our Hamiltonian for $L = 10$ ancillas. The dashed black lines denote eigenvalues of the Hamiltonian, the red blue line denotes the lowest eigenvalue and the red line denotes the energy as found using the VQE algorithm.

4.2 Time Evolution

In order to evaluate the time evolution of the model we look at the three body contact density as defined in section 2. We begin by looking at the time evolution by a single trotter step with the initial state as found by the VQE, this is given in figure 9. We see that our results are consistent with the result found by Roggero *et al.* (2019) [1].

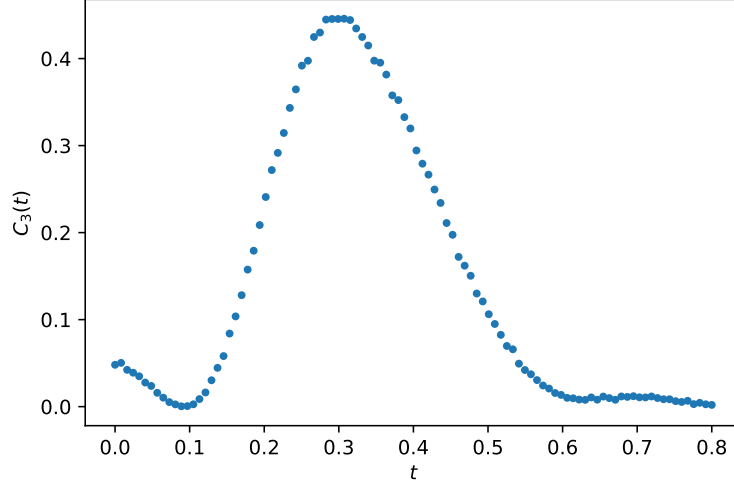


Figure 9: *Three body contact density as a function of time. Time evolution performed using a single trotter step per time point.*

When increasing the number of trotter steps we use we see that the plot behaviour changes drastically (figure 10). We see that for increasing number of trotter steps the behaviour approaches that of the exact exponential as given in figure 3a. Which is expected since the error due to trotterization should decrease with additional trotter steps.

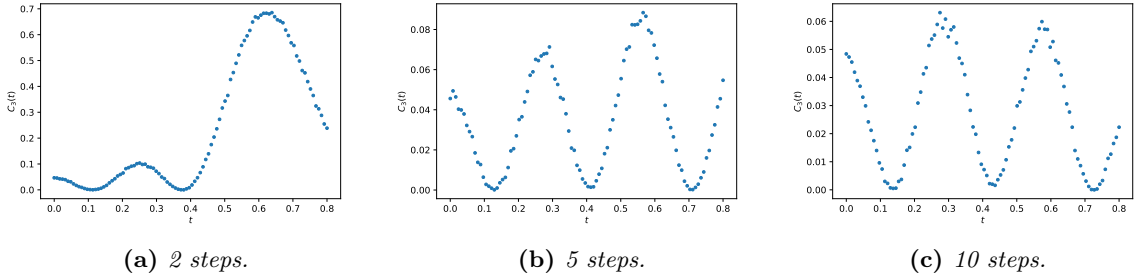


Figure 10: *Three body contact density as a function of time. Time evolution performed using a) 2, b) 5, c) 10, trotter steps per time point.*

We can also look at time evolution with a state we prepared using QPE, first using a single trotter step. This is given in figure 11. We see that this result differs from the result using the VQE, which gives an indication that it prepared a different initial state.

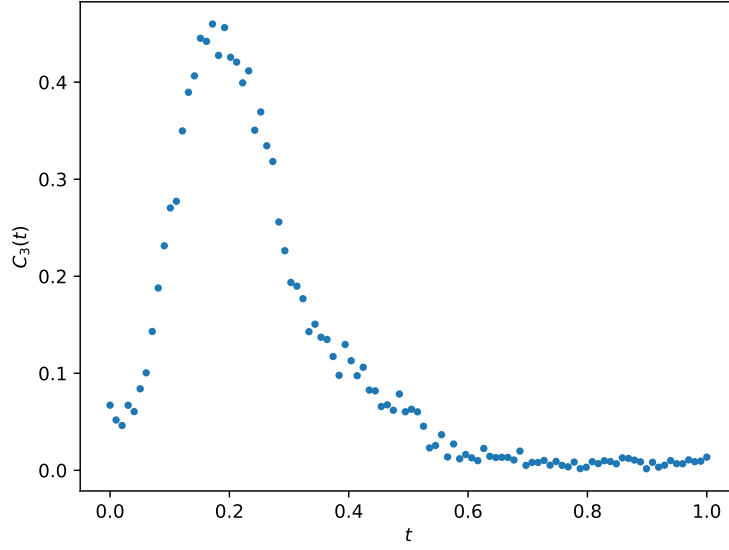


Figure 11: *Three body contact density as a function of time for a state initialised using QPE. Time evolution performed using a single trotter step per time point.*

For 10 trotter steps this result is given in figure 12.

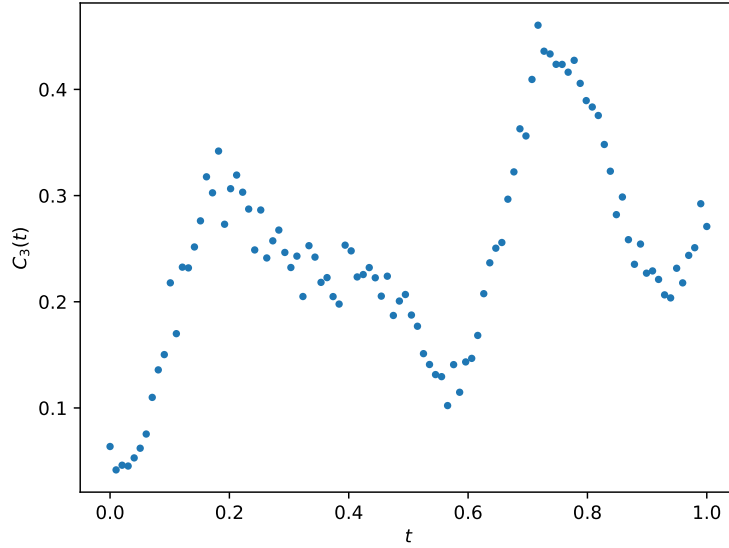


Figure 12: *Three body contact density as a function of time for a state initialised using QPE. Time evolution performed using 10 trotter steps per time point.*

4.3 Linear Response

For the linear response we only have some preliminary results.

We look at perturbations given by $O = Z_3$ and apply the algorithm as described in section 3.5 and by Roggero & Carlson (2018) [3]. We have $\gamma = 0.1$ and use the same parameters in the Hamiltonian

as described above. We get an estimation of $S_O(\omega)$ given in figure 13. We see that the rough structure seems to be similar to that in [3] however there are mainly disagreements. Due to time constraint we weren't able to improve results.

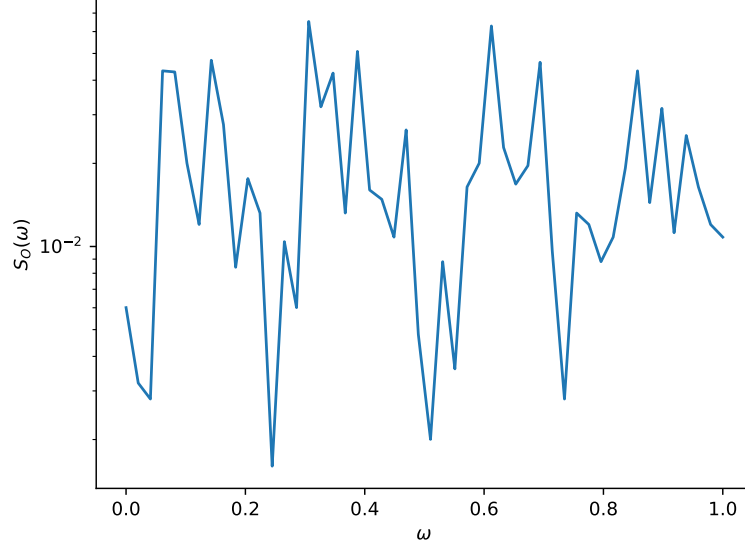


Figure 13: Estimation of the response function $S_O(\omega)$. The rough structure seems to be consistent with expectations.

5 Conclusion

We implemented the triton toy model as defined by Roggero *et al.* (2019) [1] and approximated the ground state using the variational quantum eigensolver (VQE) and quantum phase estimation (QPE) algorithm. Using the VQE we found a ground state energy approximation consistent with the results found in [1] and with the results found using the VQE as implemented in OpenFermion.

When looking at the time evolution using a single trotter step we found results consistent with [1]. When implementing time evolution with multiple trotter steps we find different results that do match with results using exact matrix exponentials using numpy.

References

- [1] Alessandro Roggero, Andy C.Y. Li, Joseph Carlson, Rajan Gupta & Gabriel N. Perdue (2019), *Quantum Computing for Neutrino-nucleus Scattering*, Preprint at <https://arxiv.org/abs/1911.06368>.
- [2] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright & Paul E. Wright (1998), *Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions*, SIAM J. Optim., 9(1), 112–147.
- [3] Alessandro Roggero & Joseph Carlson (2019), *Dynamic linear response quantum algorithm*, PHYSICAL REVIEW C 100, 034610.
- [4] The Cirq Developers, *Cirq, A python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits*, <https://github.com/quantumlib/Cirq>.
- [5] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondrej Certik, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake,

Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Stepan Roucka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman & Anthony Scopatz (2017), *SymPy: symbolic computing in Python*, PeerJ Computer Science 3:e103.

- [6] Jarrod R. McClean, Kevin J. Sung, Ian D. Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, E. Schuyler Fried, Craig Gidney, Brendan Gimby, Pranav Gokhale, Thomas Häner, Tarini Hardikar, Vojtěch Havlíček, Oscar Higgott, Cupjin Huang, Josh Izaac, Zhang Jiang, William Kirby, Xinle Liu, Sam McArdle, Matthew Neeley, Thomas O’Brien, Bryan O’Gorman, Isil Ozfidan, Maxwell D. Radin, Jhonathan Romero, Nicholas Rubin, Nicolas P. D. Sawaya, Kanav Setia, Sukin Sim, Damian S. Steiger, Mark Steudtner, Qiming Sun, Wei Sun, Daochen Wang, Fang Zhang & Ryan Babbush (2017), *OpenFermion: The Electronic Structure Package for Quantum Computers*, Preprint at <https://arxiv.org/abs/1710.07629>.
- [7] John D. Hunter (2007), *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, 9, 90-95.
- [8] Travis E. Oliphant (2006), *A guide to NumPy*, Trelgol Publishing, USA.