1 Theory

1.1 Measurements

When we want to perform measurements on our quantum system, we need to measure the state of the qubits in a basis of choice. To properly mimic a quantum measurement, it is the *expectation value* that we should "measure". This means, that we need to make several measurements of the state, and then take the average of these measurements. A neat way to implement this will be presented in the method section.

An interesting, and important note, is that by clever usage of quantum gates - we can transform *any* Pauli measurement to a measurement in the computational basis (Z_0 measurement). This will be done by the transformations found in **Hundt˙2022** pg. 251 - 252, table 6.1 and 6.2. This is a very useful property, as making measurements in the computational basis makes our life much simpler - and then we may find expectation values of any combination of Pauli gates, by doing Z_0 measurements on the transformed ansatz. For example, if we would like to perform an X_0 measurement, we would only need to apply the Hadamard gate to the first qubit, and then perform a Z_0 measurement.

1.2 Quantum Gates

Quantum gates are the building blocks of quantum circuits. They are the quantum analogues of classical logic gates, and are used to manipulate the state of a quantum system. In this section, we will discuss some of the most important quantum gates, and how they can be used to perform quantum computations.

1.2.1 The Pauli Gates

The Pauli gates are a group of three quantum gates, based on the Pauli matrices. They are the Pauli-X, Pauli-Y, and Pauli-Z gates, and are defined as follows:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The Pauli-X gate and the Pauli-Y gate are bit-flip gates, and the Pauli-Z gate is a phase-flip gate. They are all Hermitian and unitary, and are their own inverses. The Pauli-Y gate also introduce a phase shift on the qubit. These are arguably the most important gates, as these are the ones used for encoding the Hamiltonian matrices in the VQE algorithm.

1.2.2 The Hadamard Gate

The Hadamard gate is a single-qubit gate, and is defined as follows:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The Hadamard gate is used to create superposition, and is also used to perform a change of basis.

1.2.3 Phase Gate

The phase gate is also a single-qubit gate, and is defined as follows:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

1.2.4 The CNOT Gate

The CNOT gate is a two-qubit gate, and is defined as follows:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The CNOT gate is used to create entanglement, and is also used to perform conditional operations. The CNOT gate is also known as the controlled-X gate, and the need to create entanglement arises when we are working in a system where the ground state may be an entangled state. Then we need to be able to create entanglement in our trial wavefunction, otherwise we will not be able to properly converge towards the ground state.

1.3 Computational basis

The computational basis is the basis that we can use to represent the state of a quantum system. In a quantum computer, the computational basis is the set of all possible states of a qubit. The qubits in our system can be in a superposition of states, and we represent this superposition using a lineare combination of the computational basis states. For a single qubit system, the computational basis is as follows:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}.$$

This can easily be extended to a multi-qubit system by taking the tensor product of the single-qubit basis states. E.g for a 2 qubit system,

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \quad |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix},$$
$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \quad |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

1.3.1 Bell basis

Another very useful basis is the Bell basis, which is a basis that is used to represent entangled states. The Bell basis is defined as follows:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big), \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle - |11\rangle\big),$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}\big(|01\rangle + |10\rangle\big), \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}\big(|01\rangle - |10\rangle\big).$$

1.4 Variational Quantum Eigensolver

Quantum algorithms are algorithms that are specifically designed to be run on a quantum computer, and they are constructed by manipulating a quantum system using quantum gates. The algorithm that we are going to study in this report is the VQE (Variational Quantum Eigensolver) algorithm. This is an algorithm that is used to find an estimate to the ground state energy of a given Hamiltonian, and it is based on the variational principle.

1.4.1 The Variational Principle

The variational principle states that for any given Hamiltonian, the expectation value of the Hamiltonian in any state is always greater than or equal to the ground state energy of the Hamiltonian. This means that if we have a trial wavefunction, $|\Psi\rangle$, we can use this to estimate the ground state energy of the Hamiltonian.

$$E_0 \leq \frac{\langle\psi| H |\psi\rangle}{\langle\psi|\psi\rangle}.$$

As we can see, if we work to minimize the expectation value of the Hamiltonian, we will always find a ”better” estimate to the ground state energy - while never going below it.

1.4.2 Pauli decomposition of the Hamiltonian

In order to use the VQE algorithm (or any quantum algorithm to be run on a quantum computer), we need to express the Hamiltonian in terms of quantum gates. This is done by expressing the Hamiltonian as a linear combination of Pauli operators, what we call Pauli strings.
To find this decomposition we can use the following formula (for a n-qubit system):

$$H^{2^n \times 2^n} = \sum_{(\xi_1, \xi_2, \ldots, \xi_n) \in \{I, X, Y, Z\}^n} \frac{1}{2^n} \mathrm{Tr}\big[\big(\bigotimes_{i=1}^{n} \sigma^i_{\xi_i} \big) \cdot M \big] \cdot \bigotimes_{i=1}^{n} \sigma^i_{\xi_i} \tag{1}$$

where the tensorproduct is all possible combinations of the pauli matrices, and the first term (the trace part) yields the coefficient infront of the matrix (latter part). With this, we can easily encode our Hamiltonian using the Pauli gates introduced earlier in the section.

1.4.3 The VQE algorithm

The VQE algorithm is a hybrid quantum-classical algorithm, and is as mentioned based on the variational principle. The algorithm is based on the following steps:

1. Choose a trial wavefunction, $|\Psi(\theta, \phi)\rangle$, that depends on some parameters, θ and ϕ, which are rotation angles for the qubit(s).
2. Calculate the expectation value of the Hamiltonian, H, in the trial wavefunction, $|\Psi(\theta, \phi)\rangle$.
3. Use a classical optimization algorithm to minimize the expectation value of the Hamiltonian, and find the optimal parameters, $\theta_{\mathrm{opt}}, \phi_{\mathrm{opt}}$.
4. Use the optimal parameters, θ_{opt}, ϕ_{opt}, to calculate the ground state energy of the Hamiltonian.

The performance of the VQE algorithm strongly depends on the choice of the initial trial wavefunction, if poorly initialized, one could get "stuck" in a local minima. Another important aspect is the choice of optimization algorithm, and choice of parameters of said algorithm.

1.4.4 Wavefunction ansatz

when we want to solve the various Hamiltonians, it is important that we make a good wavefunction ansatz. In this ansatz, it is of the essence that we manage to properly capture the entanglement (or other physical properties) in the system - otherwise, our VQE algorithm will not actually converge to a good ground state estimate. In the methodsection, we will present the ansatzses we've used for the different hamiltonians.

1.5 Entanglement

A very important topic for quantum computing is the concept of entanglement. This is a property of quantum systems that do not occur in classical systems, and is widely used when we perform quantum computing. Many quantum computing algorithms would not be possible without entanglement. In our VQE algorithm, we will study Hamiltonian systems with varying interaction strengths, which will lead to entanglement in our system.
The concept of entanglement (in a quantum computer) is the fact that two (or more) qubits become correlated in such a way that the state of one qubit is directly dependent on the state of the other qubit(s). This is a property we can use, since we then need only perform a measurement on one qubit to immediately know the state of the other qubit(s)!
We will later explain how we should capture the entanglement in the systems to be studied using our wavefunction ansatz. If we'd like to make a measurement of "how" entangled a system is, one can use the Von Neumann entanglement entropy, which can be used to quantify the entanglement in a quantum system.
The von Neumann entropy is defined as:

$$S(\rho) = -\mathrm{Tr}(\rho \log(\rho)), \tag{2}$$

where ρ is the density matrix of the system. The density matrix can be found as follows:

$$\rho = |\Psi\rangle \langle\Psi|, \tag{3}$$

where $|\Psi\rangle$ is the wavefunction of the system.
If the system is in a pure state, the von Neumann entropy is zero, and if the system is composed to two subsystems, A and B, the von Neumann entropy is maximized when the two subsystems are maximally entangled. The system can then be in a pure state, but the individual subsystems may be in a mixed state. Then we use the Von Neumann entropy of one of the subsystems as a measurement of the entanglement in our composite system as such

$$S(\rho_A) = -\text{Tr}(\rho_A \log(\rho_A)), \tag{4}$$

where ρ_A is the reduced density matrix of subsystem A.

1.6 Optimization: Gradient Descent & Scipy.minimize

The optimization algorithms that we will be using in our numerical implementation of the VQE algorithm is the gradient descent algorithm, and the minimization module in the SciPy library[1]. Finding the optimal parameters, and thus the ground state energy, is a difficult task, and most gradient descent algorithms struggle with finding the global minimum for the VQE algorithm. The presence of "barren plateaus", regions in parameter space where the gradients are exponentially small, pose a big challenge for most gradient methods. However, this is not a major problem for the simple Hamiltonians we will encounter in this report - and thus we can make do with the simple gradient descent algorithm.
The gradient descent algorithm is as follows:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \nabla_\theta E(\theta_{\text{old}}), \tag{5}$$

I.e we calculate the gradient of the parameter(s) we are using to minimize the expectation value of the Hamiltonian, and then update the parameters by taking a step in the direction of the negative gradient.

1.7 Hamiltonians

As mentioned in the introduction, there are two different Hamiltonians that we will be working with in this report. The first basic Hamiltonian is a very simple Pauli string, which we will investigate for both one - and two qubits. The Hamiltonian is as follows:

$$\begin{aligned} H_0 &= \xi \mathbf{I} + \Omega \sigma_z \\ H_I &= c\mathbf{I} + \omega_z \sigma_z + \omega_x \sigma_x, \end{aligned}$$

$$H_{\text{one qubit}} = H_0 + \lambda H_I \tag{6}$$

where $\mathbf{I}$ is the identity matrix, and σ_z and σ_x are the Pauli-Z and Pauli-X matrices, respectively.
Similarly, for the two-qubit system

$$\begin{aligned} H_0 &= \epsilon_i \delta_{ijk} \\ H_I &= H_x \sigma_x \otimes \sigma_x + H_z \sigma_z \otimes \sigma_z, \end{aligned}$$

$$H_{\text{two qubit}} = H_0 + \lambda H_I \tag{7}$$

where δ_{ijk} is the Kronecker delta, ϵ_i are the corresponding diagonal elements of the non-interaction Hamiltonian.

1.8 Lipkin model

The final system, or Hamiltonian, that we are to study in this report is the Lipkin model. We will follow in particular the article in **PhysRevC.106.024319**, where the authors study the Lipkin model using the VQE algorithm. The Lipkin model is a simplified quantum system that describes interaction

[1] docs:`https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize`

between particles, in our case fermions. Due to the interaction, the system shows entanglement and phase transitions, and is thus a good model to study using the VQE algorithm.
The Hamiltonian for the Lipkin model is as follows, written in the Pauli basis[2], using $W =$, i.e only one interaction term, shown for the $J = 1, N = 2$ case:

$$H_L^{(2)} = \frac{1}{2}(Z_0 + Z_1) - \frac{V}{2}(X_0 \otimes X_1 + Y_0 \otimes Y_1), \tag{8}$$

and for the $J = 2, N = 4$ case:

$$H_L^{(4)} = -(Z_0 + Z_1) - \frac{\sqrt{6}V}{2}(X_0 + X_1 + Z_1 X_0 - X_1 Z_0), \tag{9}$$

where the subscripts indicate on to which qubit the operator acts, and V is the interaction strength.

[2] For the re-write of the Hamiltonian, see the article **PhysRevC.106.024319**