

## 0.1 Quantum Computing

Analogous to the digital bits used for classical computation, a quantum computer requires quantum bits, more commonly known as qubits, as the fundamental register. However, unlike the classical bits which can acquire the value of either a 0 or a 1, a qubit state can be a linear superposition of the states  $|0\rangle$  and  $|1\rangle$ , where  $|\rangle$  represents a quantum state in the Dirac notation, i.e.,

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle, \quad (1)$$

where  $a_0$  and  $a_1$  are the complex amplitudes such that  $|a_1|^2 + |a_0|^2 = 1$ . According to the principles of quantum mechanics, a measurement of the qubit state then yields either  $|0\rangle$ , with probability  $|a_0|^2$  or  $|1\rangle$ , with probability  $|a_1|^2$ .

In its simplest form, a qubit is therefore a two-level system. A spin-1/2 particle with its two levels being the up spin and the down spin, can naturally be used as a qubit. In this notation, it is conventional to associate state  $|0\rangle$  with  $|\uparrow\rangle$ , and state  $|1\rangle$  with  $|\downarrow\rangle$ . The three components of the Pauli matrices corresponding to the spin-1/2 operator  $\mathbf{S}$  spanned by these states are:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

which are then used as the fundamental elements of quantum computing.

For  $N$  such qubits, the basis state is a tensor product of the single-qubit basis states. Since each single state constitutes two basis vectors, the  $N$ -qubit state comprises of  $L = 2^N$  basis vectors. This constitutes of the computational basis -  $|00\dots 0\rangle, |00\dots 1\rangle, \dots, |11\dots 1\rangle$ . Therefore, a general  $N$ -qubit state can be represented as

$$|\psi\rangle = a_0 |00\dots 0\rangle + a_1 |00\dots 1\rangle + \dots + a_L |11\dots 1\rangle, \quad (2)$$

which requires  $L$  complex amplitudes for its description.

The computational basis can be more conveniently notated as  $|00\dots 0\rangle = |0\rangle, |00\dots 1\rangle = |1\rangle, \dots, |11\dots 1\rangle = |L\rangle$ . In this representation, eq. (2) thus becomes

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle + \dots + a_L |L\rangle. \quad (3)$$

Therefore, the Hilbert space spanned by  $N$  qubits is an  $L$ -dimensional space.

Since Pauli matrices, together with the identity operator  $I$ , form a complete basis for a vector in  $2 \times 2$  space, any gate acting on a qubit can be expressed as a linear combination of Pauli matrices [Quantum Walks and search algorithms, Renato Portugal]. The action of a Pauli matrix,  $\sigma^\alpha$ , where  $\alpha \in \{x, y, z\}$  on the  $j^{th}$  of the  $N$  bits of the basis vector is represented as  $\sigma_j^\alpha$ . Thus, the Pauli operator  $\sigma_j^\alpha$  acting on the general state  $|\psi\rangle$ , given in eq. (2), just acts on the  $j^{th}$  bit of the basis vector, altering the coefficients  $a_0, a_1, \dots, a_L$ . The other bits of the basis vector remain unchanged as a result of being acted upon by identity matrices  $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}_{i \neq j}$ .

## 0.2 Optimization Problems and Quantum Annealing

Mathematically, an optimization problem comprises of a cost function,  $f_0(\mathbf{x})$  involving  $N$  variables  $x_1, x_2, x_3, \dots, x_N$ , such that  $\mathbf{x} = (x_1, \dots, x_N)$ . The goal is then to find the solution  $\mathbf{x}_0$  that minimises (or maximises) the cost function. In the computing language the cost function is written as a Boolean expression in the form of conjunction (a Boolean AND operation) of  $r$  clauses, where each clause is a disjunction (a Boolean OR operation) of  $k$  variables or their negations ( $k$ -SAT problems)[Convex Optimization, Stephen Boyd (book)], [Ting Hsu et. al]. A 2-SAT instance is thus formulated as a Boolean expression where each clause is a disjunction of two literals. The literals,  $L_{i,j}$ , are Boolean variables or their negations. The task is then to find a truth assignment to the Boolean variables that makes the formula  $G = (L_{1,1} \vee L_{1,2}) \wedge (L_{2,1} \vee L_{2,2}) \wedge \dots \wedge (L_{r,1} \vee L_{r,2})$  true. If  $G=1$  then the 2-SAT instance is satisfiable.

Many physically inspired approaches have also been adopted to find the solution for the optimization problems. One of the widely used techniques is to encode the problem into the couplings of Ising Hamiltonians [Nonstoquastic Hamiltonians]. The ground state of this Hamiltonian, corresponding to the state of the minimum energy, then

represents the optimal solution of the problem. At low enough temperatures, the system state should eventually relax to the ground state, yielding the required solution.

For finding the optimal solution, the whole spectrum of the cost function needs to be explored. This requires that the system should be able to escape from a local minimum, if it gets trapped in one, during the course. The presence of multiple local minima thus makes the determination of the optimal solution harder [Optimization using quantum mechanics: quantum annealing through adiabatic evolution: Santoro and Tosatti].

The method of simulated annealing was therefore utilised for [Kirkpatrick S, Gelatt C D Jr and Vecchi M P 1983 Science 220 671], where adding thermal fluctuations to the cost function, keeps the system from getting trapped in the local minima. Yet if the barrier potential diverges or becomes very high, this approach can no longer be helpful. It was in this spirit that the technique of quantum annealing was first employed by B.Apolloni, C.Carvalho and D.de Falco in 1988 [Quantum stochastic optimization, Appoloni, Carvalho, de Falco], wherein quantum fluctuations were used in place of thermal fluctuations. By making use of the quantum tunnelling effect, this approach can still allow for the search of global minima as the system state can tunnel through the poor local minima.

The cost function is mapped on to the Ising model of spins, making use of the external magnetic field  $h_i^z$  and the spin couplings  $J_{ij}^z$  of the model, and thus has the following form:

$$H_P = - \sum_{i=1}^N h_i^z \sigma_i^z - \sum_{\langle i,j \rangle} J_{ij}^z \sigma_i^z \sigma_j^z, \quad (4)$$

where  $\sigma_i^z$  denotes the z component of Pauli-spin matrix acting on the  $i^{th}$  spin, and the set  $\langle i, j \rangle$  represents the set of pairwise couplings.

The recipe for the annealing algorithm consists of starting with an initial Hamiltonian  $H_I$ , whose ground state can be easily determined and realised. Most commonly used is the transverse field Hamiltonian:

$$H_I = - \sum_{i=1}^N h_i^x \sigma_i^x. \quad (5)$$

The ground state for  $H_I$  is therefore the uniform superposition state:

$$|\psi\rangle = \frac{1}{(\sqrt{2})^N} (|0\rangle + |1\rangle)_1 * (|0\rangle + |1\rangle)_2 * \dots * (|0\rangle + |1\rangle)_N. \quad (6)$$

The Hamiltonian is then slowly swept towards the problem Hamiltonian, with the means of an annealing parameter, say  $s$ , defined as  $s = t/T_A$ , where  $t$  is the instantaneous time, and  $T_A$  is the total annealing time. The Hamiltonian,  $H(t)$ , corresponding to the most straightforward annealing scheme, is then given by:

$$H(t) = (1 - s(t))H_I + s(t)H_P. \quad (7)$$

Therefore, the Hamiltonian transitions from the initial Hamiltonian,  $H(t = 0) = H_I$  to the final Hamiltonian,  $H(t = T_A) = H_P$ .

According to the **quantum adiabatic theorem**, the instantaneous state of the system stays close to the ground state of Hamiltonian  $H(t)$ , if one starts with ground state of the initial Hamiltonian and if the driving from the initial Hamiltonian to the problem Hamiltonian is slow enough, which is determined by the minimum energy gap,  $\Delta_{min}$ , between the ground state and the first excited state of the Hamiltonian  $H(t)$  during the course of annealing. [ADIABATIC QUANTUM COMPUTATION, Enej Ilievsk]. Mathematically, adiabatic theorem of evolution holds when

$$T_A \propto \Delta_{min}^{-2}, \quad (8)$$

where  $T_A$  is the total annealing time. [Perspectives of quantum annealing: Methods and implementations]. Therefore, the performance of quantum annealing depends strongly on the minimum energy gap.

Thus, the problem of finding the optimal solution reduces to the problem of solving the time dependent Schrödinger equation for the resulting  $H(t)$  (see eq. 7):

$$i \frac{\partial}{\partial t} |\psi\rangle = H(t) |\psi\rangle. \quad (9)$$

However, the process of evolution, starting from the trivial ground state of the initial Hamiltonian and going to a non-trivial ground state of the problem Hamiltonian, is accompanied by a quantum phase transition. Such a transition is characterized by a vanishing energy gap in the thermodynamic limit. For the problems which are considered to be difficult, the minimum gap is found to close exponentially as a function of the system size, i.e.,  $\Delta_{min} \propto e^{-cN}$ , for a positive constant  $c$  and  $N$  number of spins. This suggests that the annealing time required to ascertain that the evolution is adiabatic also grows exponentially,  $T_A \propto e^{2cN}$ . On the other hand, if the gap closes polynomially ( $\Delta_{min} \propto N^{-l}$ , for a positive constant  $l$ ), the computation time is also polynomial,  $T_A \propto N^{2l+1}$ , and the problem is considered easy [Perspectives, Nishimori].

Therefore, the dependence of the minimum energy gap on the size of the system determines the annealing time required to make the evolution of the state of the system to be adiabatic. Thus, one method of improving the performance of quantum annealers is by controlling the closing of the gap. Altering the annealing scheme is believed to be one of the ways in which this can be achieved [Perspectives of quantum annealing: Methods and implementations, Adiabatic quantum computation: Tameem Albash, Quantum Adiabatic Evolution Algorithms with Different Paths: Farhi, Different strategies for Optimization Using the Quantum Adiabatic Algorithm: Farhi, Nonstoquastic Hamiltonians and quantum annealing of an Ising spin glass]. In this work we aim to achieve the same by including a third Hamiltonian, the Trigger Hamiltonian -  $H_T$  in the time dependent Hamiltonian  $H(t)$  (see eq. 7)[Farhi-Goldstein, Farhi (Different), Review, Non-stoquastic]. The trigger Hamiltonian should vanish at both the start and end of the annealing process, so that one can still start with the easily realizable ground state of the initial Hamiltonian, and the resulting state of the problem Hamiltonian remains unaffected. The Hamiltonian thus takes the form:

$$H(t) = (1 - s)H_I + gs(1 - s)H_T + sH_P, \quad (10)$$

where parameter  $g$  controls the strength of the added trigger.

Furthermore, for this thesis we deal with two types of trigger Hamiltonians - The ferromagnetic trigger (F) and the anti-ferromagnetic trigger (A)[Nonstoquastic...], given as

$$H_T^F = - \sum_{\langle i,j \rangle} \sigma_i^x \sigma_j^x, \quad (11)$$

and

$$H_T^A = + \sum_{\langle i,j \rangle} \sigma_i^x \sigma_j^x, \quad (12)$$

with the same pairwise coupling set  $\langle i, j \rangle$  as for the problem Hamiltonian. The trigger can also be chosen to have the form  $\pm \sum_{\langle i,j \rangle} \sigma_i^y \sigma_j^y$ . As will be seen in the following chapters, adding these triggers alters the energy spectrum considerably, which in turn affects the overlap of the final state with the ground state in different ways.

The following sections focus on two approaches that have been used in this work to solve the time-dependent Schrödinger equation. The Suzuki-Trotter product formula has been adopted to track the evolution of the state, and to compute the overlap of the final state with the already determined ground state of the problem Hamiltonian. The full diagonalization method, on the other hand, has been used to calculate the errors involved in using the Suzuki-Trotter approximation, and to determine the energy spectra and minimum gaps for  $H(t)$  for  $t \in [0, T_A]$ , where  $T_A$  is the total annealing time.

### 0.3 Exact Diagonalization

This method consists of determining the eigenvalues and eigenvectors of the Hamiltonian matrix, which is an  $L \times L$  ( $2^N \times 2^N$ ) matrix, at each time step.

For constructing the Hamiltonian matrix at time step  $t \in [0, T_A]$ , all the basis vectors in the computational basis are acted upon by the instantaneous Hamiltonian, given by eq. (10). The action of the Hamiltonian on the  $i^{th}$  basis vector corresponds to the  $i^{th}$  column of the Hamiltonian matrix, e.g. for  $|\psi\rangle = (1, 0, \dots, 0)^T$ ,  $H|\psi\rangle$  gives the first column of the Hamiltonian matrix.

The resulting matrix is then diagonalized to obtain the eigenvalues  $\Lambda$ , and unitary matrix of the eigenvectors,  $V$ . Since  $V^\dagger H V = \Lambda$ , the unitary evolution operator  $U(t) = e^{-itH} = V e^{-it\Lambda} V^\dagger$ .

This approach, however, has some serious limitations. Firstly, the memory requirement to store the Hamiltonian matrix grows exponentially with the number of qubits ( $\mathcal{O}(2^{2N})$ ). Additionally, full diagonalization takes  $\mathcal{O}(2^{3N})$  floating-point operations [Kristel-Hans paper]. Therefore, this method is rendered impractical for solving the time dependent Schrödinger equation for systems more than 20 qubits.

## 0.4 Suzuki-Trotter Product Formula

For solving the time dependent Schrödinger eq.(9), one needs to evaluate the unitary matrix exponentials of the evolution operator,  $U(t)$  given as:

$$U(t) = e^{-itH} = e^{-it(H_1+\dots+H_K)} = \lim_{m \rightarrow \infty} \left( \prod_{k=1}^K e^{-itH_k/m} \right)^m. \quad (13)$$

In this work, the Lie-Trotter-Suzuki product formula is employed to construct unitary approximations to the evolution operator. Defining  $\tau$  as  $t/m$ , to be the time step at which the evolution operator is applied, a repeated application of  $U(\tau)$  yields  $U(t)$ . For a sufficiently small time step  $\tau$ , the first order approximation for  $U(t)$  in eq. (13) is

$$\tilde{U}_1(\tau) = e^{-i\tau H_1} \dots e^{-i\tau H_K}, \quad (14)$$

which holds good for  $\tau \|H\| \ll 1$ .

For an improved accuracy, a second order approximation is made to  $U(t)$  in eq. (13), using  $\tilde{U}_1(\tau)$  from eq. (14):

$$\tilde{U}_2(\tau) = \tilde{U}_1^\dagger(-\tau/2) \tilde{U}_1(\tau/2) = e^{-i\tau H_K/2} \dots e^{-i\tau H_1/2} e^{-i\tau H_1/2} \dots e^{-i\tau H_K/2}. \quad (15)$$

Since  $\tilde{U}_1(\tau)$  is unitary, as all  $H_k$  in eq. (13) are Hermitian,  $\tilde{U}_2(\tau)$  is also unitary. Thus the measure of error, calculated using the absolute difference between  $U(\tau)$  and  $\tilde{U}_2(\tau)$ , grows cubically in  $\tau$  [Kristel-Hans paper, 62 from there], i.e.

$$\|U(\tau) - \tilde{U}_2(\tau)\| \leq c\tau^3 \quad (16)$$

for a positive constant  $c$ . Since the whole annealing process requires  $m$  such time steps, the involved error becomes

$$\|U(m\tau) - \tilde{U}_2(m\tau)\| \leq m^3 c\tau^3. \quad (17)$$

Since  $m\tau = t$ , (17) is equivalent to

$$\|U(m\tau) - \tilde{U}_2(m\tau)\| \leq cm^2 t\tau^2 = c_2 t\tau^2, \quad (18)$$

where  $c_2 = cm^2$ .

The Hamiltonian  $H(t)$  is then decomposed as follows:

$$H = H_{single} + H_x + H_y + H_z, \quad (19)$$

where  $H_{single} = -(1-s) \left( \sum_{i=1}^N h_i^x \sigma_i^x \right) - s \left( \sum_{i=1}^N h_i^z \sigma_i^z \right)$ ,  $H_x = \mp(1-s) \left( \sum_{\langle i,j \rangle} J_{i,j}^x \sigma_i^x \sigma_j^x \right)$ ,  $H_y = \mp(1-s) \left( \sum_{\langle i,j \rangle} J_{i,j}^y \sigma_i^y \sigma_j^y \right)$ ,

and  $H_z = -s \left( \sum_{\langle i,j \rangle} J_{i,j}^z \sigma_i^z \sigma_j^z \right)$ . In general,

$$e^{i\mathbf{v} \cdot \boldsymbol{\sigma}} = \cos(v)I + i \frac{\sin(v)}{v} \mathbf{v} \cdot \boldsymbol{\sigma}, \quad (20)$$

where  $I$  represents the identity matrix, and  $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$ . Then,

$$e^{-itH_{single}} = \prod_{i=1}^N e^{it[(1-s) \sum_i h_i^x \sigma_i^x + s \sum_i h_i^z \sigma_i^z]} = \prod_{i=1}^N \begin{pmatrix} \cos(th_i) + i \frac{sh_i^z}{h_i} \sin(th_i) & i \frac{(1-s)h_i^x}{h_i} \sin(th_i) \\ i \frac{(1-s)h_i^x}{h_i} \sin(th_i) & \cos(th_i) - i \frac{sh_i^z}{h_i} \sin(th_i) \end{pmatrix} \quad (21)$$

where  $h_i = \sqrt{(1-s(t))^2 h_i^{x^2} + s(t)^2 h_i^{z^2}}$ .

The computational basis states are the eigenstates of the Pauli-z operator,  $\sigma_i^z$ . Thus  $e^{-itH^z}$  is a diagonal matrix in the computational basis, and its action on the input state changes the phase of each of the basis vectors. As  $H^z$  is a sum of pair interactions, it is trivial to implement this operation as a sequence of multiplications by  $4 \times 4$

diagonal matrices [Kristel-Hans paper].

The same approach can be adopted for implementing  $H_x$  operations as well, by using the rotation operators  $Y_j$  as follows. Writing  $Y = \prod_{i=1}^N Y_i$ , we obtain

$$e^{-itH^x} = \bar{Y}Y e^{-itH^x} \bar{Y}Y = \bar{Y} e^{it \sum_{\langle i,j \rangle} J_{ij}^x \sigma_i^z \sigma_j^z} Y. \quad (22)$$