

MAPPING FERMIONS TO QUBITS

OLIVER O'BRIEN

*DAMTP, Centre for Mathematical Sciences, University of Cambridge, Cambridge CB30WA, UK
Christ's College, Cambridge, UK*

Received (received date)

Revised (revised date)

Your abstract goes here.

Keywords: The contents of the keywords

Communicated by: to be filled by the Editorial

1 Introduction

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy." - *Dr Richard Feynman* [1]

Ever since Dr Richard Feynman said these famous words at the end of his keynote speech in 1982, one of the main tasks facing quantum computers has been simulating quantum systems. Intuitively a quantum computer should be able to handle this problem better than a classic computer as they operate within the same bizarre world of entanglement and superposition. However, there are many challenges to overcome before quantum advantage can be achieved in this field. We will consider one such challenge in this essay: the optimum mapping scheme from fermions to qubits.

One of the most common classes of quantum systems we wish to simulate are those composed of fermions. In order to simulate these particles we must find a representation for them in terms a quantum computer can process, that is qubits and qubit operators. This poses a fundamental issue as fermions exhibit the non-local property of anti-commutation (**need to explain why it is non-local**), whereas qubits are independent local units. Therefore a non-trivial mapping scheme which introduces this non-local behaviour is necessary.

The first such mapping, the **Jordan-Wigner Transformation**, was invented nearly a century ago [2], and we will introduce it in Section 4. More recently, a number of new mappings have been developed including the **Bravyi-Kitaev map** and the **Derby-Klassen map**, which will be discussed in Section 5 and Section 6 respectively. Furthermore, recent research [3] has shown that the **fermionic enumeration** scheme used in a given mapping (we consider solely the Jordan-Wigner case) offers potential for increased locality with no additional

resources. This is elaborated on in Section 7.

It is important to consider how these mappings impact upon the fermionic simulation techniques used. Therefore, in Section 2 we will discuss how VQE and phase estimation can both be used to estimate the ground state energy of a system. Finally, in Section 8 we will compare the relative performance of these mappings in real-world applications (using simulated quantum computers).

2 Applications to fermionic systems

2.1 Estimating ground state energies

Computing the ground state energy of a Hamiltonian is generally the first step in computing the energetic properties of molecules and materials [4]. Chemists have developed classical computational models for estimating the ground state, however, they all rely on approximations. Quantum computing opens up the potential for an exact (*full configuration interaction*) approach, which is unfeasible on a classical computer where it scales exponentially (**verify this is true**) in the number of fermionic modes.

2.1.1 Variational Quantum Eigensolver

A VQE provides an upper bound on the ground state energy of a Hamiltonian by utilising the variational principle:

$$\frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0 \quad \forall |\psi\rangle \in \mathcal{H} \quad (1)$$

The algorithm consists of two stages. First a variational ansatz is initialised based on a set of parameters θ . Then, the Hamiltonian is measured a number of times, and the parameters of the ansatz are varied until a minimum is found. This is an example of a variational quantum algorithm, which performs a classical optimisation over a quantum oracle (**is this a correct description**). By exporting lots of the work to a classical computer, VQEs are one of the quantum algorithms that are achievable in the NISQ (Noisy Intermediate-scale Quantum) era.

The preparation of ansatz can be done without any regard for fermions or what underlying wavefunction it represents. Generally a given circuit structure is picked with some of the gates depending upon the parameters (i.e. an $R_z(\theta)$ gate). However, there are some schemes for preparing ansatz that correlate to known wavefunctions, so the wavefunction corresponding to the minimum energy estimate can be found (**check this is true, find example and see if mapping is important** - Unitary Coupled Cluser - gaussian).

It is in the measurement of the Hamiltonian that the mapping becomes important. The Hamiltonian in question will be a combination of fermionic raising and lowering operators, so needs to be mapped to a combination of Pauli operators that can be measured on a quantum computer. **consider measurement strategies that minimise repetitions**. After this mapping has been performed we are left with a qubit Hamiltonian that is a sum of Pauli strings (strings of Pauli operators). Therefore, we can measure the effectiveness of different mappings by the resource costs of implementing this qubit Hamiltonian. We will focus on three metrics (**this bit might be too similar to vqe page 31**):

- (i) **Number of qubits:** The less qubits required by the mapping the smaller the quantum computer required to run the simulation.
- (ii) **Average Pauli weight:** The average number of Pauli operators in each Pauli string in the Hamiltonian. The smaller the Pauli weight the less gates needed to measure the Hamiltonian, which reduces both the resource cost and the error due to gate infidelity. **Low-weight operators protect against barren plateau** doi:10.1038/s41467-021-21728-w... **Could potentially elaborate here as it is a bit more complex than just less gates per string it also increases parallelisation and reduces ansatz depth. Mention locality**
- (iii) **Number of Pauli-strings:** Each Pauli-string is measured separately, so the fewer Pauli-strings to measure, the fewer times the ansatz needs to be initialized and measured.

2.1.2 Quantum Phase estimation

QPE can be used to estimate the energy levels of a Hamiltonian to n bits. This is achieved by applying successive steps of time evolution as a controlled gate with the target being the input state and the control being successive Hadamard ancilla qubits (illustrated in Fig. 1). Then by performing an inverse Quantum Fourier Transform on these ancilla qubits we get a superposition of binary expressions for the energy levels.

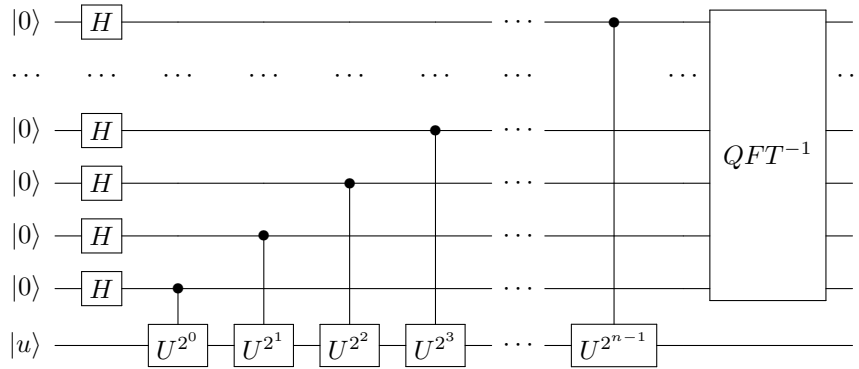


Fig. 1. Circuit diagram of QPE. In order to calculate energy levels $U = e^{-2\pi i \hat{H}}$ is used.

An important feature of the algorithm is that the guiding state $|u\rangle$ must have overlap with the ground state. This can be seen by working through in detail (we have adapted the following treatment from [5]):

- We can express $|u\rangle$ in the basis of energy eigenstates: $|u\rangle = \sum_i c_i |E_i\rangle$. Therefore, the total state after the application of the Hadamard gates is:

$$\frac{1}{\sqrt{2^n}} \sum_i \sum_x |x\rangle c_i |E_i\rangle \quad (2)$$

- After the application of the controlled gates shown in Fig. 1, this transforms to:

$$\frac{1}{\sqrt{2^n}} \sum_i c_i (|0\rangle + e^{-2\pi i E_i 2^0} |1\rangle) (|0\rangle + e^{-2\pi i E_i 2^1} |1\rangle) \dots (|0\rangle + e^{-2\pi i E_i 2^{n-1}} |1\rangle) |E_i\rangle \quad (3)$$

as $x = x_0 2^0 + x_1 2^1 + \dots + x_{n-1} 2^{n-1}$ for $x_i \in \{0, 1\}$, this reduces to:

$$\frac{1}{\sqrt{2^n}} \sum_i \sum_x e^{-2\pi i E_i x} c_i |x\rangle |E_i\rangle \quad (4)$$

- By applying an inverse Fourier transform the phase can be extracted:

$$\frac{1}{\sqrt{2^n}} \sum_i \sum_x e^{-2\pi i E_i x} c_i |x\rangle |E_i\rangle \xrightarrow{QFT^{-1}} \sum_i c_i |\text{bin}(E_i)\rangle |E_i\rangle \quad (5)$$

- As E_i is likely not exact to n bits there will be a potential error introduced by the Quantum Fourier Transform, so to be accurate to n bits a few more ancilla qubits will be needed [6].
- By measuring the ancilla bits in the Z basis we will observe E_i to n bits with probability $|c_i|^2$. So to find the ground state energy level, the ground state eigenstate needs to be in the original expansion of the guiding state $|u\rangle$. The larger the amplitude of the ground state eigenstate (c_0) the fewer repetitions of QPE are required before E_0 is measured. Therefore, QPE works better if a guiding state close to the exact ground state is used.

The portion of this algorithm for which fermionic mapping is relevant is the construction of the controlled gates $U^{2^k} = e^{-2\pi i \hat{H} 2^k}$. Not only do the same metrics for the efficiency of the corresponding qubit Hamiltonian listed in Section 2.1 apply, now we need to consider the impact of exponentiation. As the Hamiltonian is the sum of non-commuting Pauli-strings taking the exponential is non-trivial and requires the Trotter-Suzuki approximation [8]. To first order this gives:

$$e^{\frac{-it}{\hbar} \sum_k^m \hat{H}_k} = \left(\prod_k^m e^{\frac{-it \hat{H}_k}{\hbar S}} \right)^S + O(t^2/S) \quad (6)$$

In order to achieve a desired accuracy of ϵ a sufficient number of Trotter steps $S = O(t^2/\epsilon)$ need to be used [5]. The ordering of the terms in this Trotter-Suzuki expansion greatly influences the error and therefore how many Trotter steps are required. This is important to consider as the impact the ordering has varies depending on the mapping chosen as we will discuss in Section 8.

Finally, it is useful to consider how each Trotter step is represented as a circuit. First, we consider that $e^{i(Z_1 \otimes Z_2 \otimes \dots \otimes Z_n)\theta}$ applies a phase shift of $e^{i\theta}$ if the parity of the n qubits is even and $e^{-i\theta}$ if the parity is odd. Secondly, it is possible to transform $e^{i(Z_1 \otimes Z_2 \otimes \dots \otimes Z_n)\theta}$ into the exponentiation of any Pauli-string by applying R_X or Hadamard gates to change the basis to the X or Y basis, respectively. Therefore, in general the exponential of a n -fold tensor product of Pauli matrices will require $2(n-1)$ CNOT gates centred around one phase-shift gate and enough R_X and Hadamard gates to transform into and out of the necessary basis before and after [7]. The circuit required for computing the term $e^{i(Y^X Z^Y)\theta}$ is shown in Fig. 2.

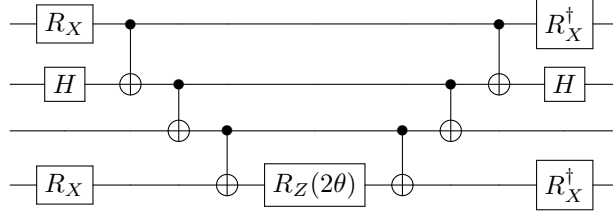


Fig. 2. The circuit required to exponentiate the Pauli-string $YXZY$ by first converting the qubits into the correct basis using R_X and H gates, then computing the parity of the four qubits before applying a single-qubit phase rotation. After this we uncompute the parity and revert to the computational (Z) basis. This is adapted from [7].

2.2 Simulating dynamics - see Nielsen Cheung

3 First and second quantization

From Quantum Field Theory (**citation needed**), we know that fermions must be antisymmetric under exchange. This property is commonly represented in two ways known as the first and second quantization. The first quantization is when the antisymmetry is retained in the wavefunction such as with:

$$|\Phi\rangle = \frac{1}{\sqrt{2}}(|\phi\rangle_1 |\psi\rangle_2 - |\psi\rangle_1 |\phi\rangle_2) \quad (7)$$

This generalises [5] to larger numbers of fermions through a Slater determinant (Eq. 8). This satisfies the antisymmetry condition as swapping any two rows of a determinant produces a sign change:

$$\phi(\mathbf{x}_0, \dots, \mathbf{x}_{N-1}) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_0(\mathbf{x}_0) & \cdots & \phi_{M-1}(\mathbf{x}_0) \\ \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_{N-1}) & \cdots & \phi_{M-1}(\mathbf{x}_{N-1}) \end{vmatrix} \quad (8)$$

Here M is the number of spin orbitals possible, and N is the number of electrons. Typically we have more spin orbitals than electrons, so the Slater determinant will only contain the N occupied spin orbitals. Acting with fermionic creation and annihilation operators on the first quantization is trivial (if long-winded) as the operators either commute with or act upon each mode.

The second quantisation compresses the information of the Slater determinant by only tracking whether each orbital or fermionic mode is occupied. Therefore the above is written as:

$$\phi(\mathbf{x}_0, \dots, \mathbf{x}_{N-1}) = |f_0, \dots, f_i, \dots, f_{M-1}\rangle \quad (9)$$

where f_i (the occupation number) = 1 when ϕ_p is occupied, and 0 otherwise. This is termed the Fock basis. Acting on the Fock basis with the fermionic creation and annihilation is more complex and is given by [5]:

$$\begin{aligned} a_p |f_0, \dots, f_i, \dots, f_{M-1}\rangle &= \delta_{f_p,1} (-1)^{\sum_{i=0}^{p-1} f_i} |f_0, \dots, f_p \oplus 1, \dots, f_{M-1}\rangle \\ a_p^\dagger |f_0, \dots, f_i, \dots, f_{M-1}\rangle &= \delta_{f_p,0} (-1)^{\sum_{i=0}^{p-1} f_i} |f_0, \dots, f_p \oplus 1, \dots, f_{M-1}\rangle \end{aligned} \quad (10)$$

Therefore, second quantization can be thought of as encoding the antisymmetry of the fermions in the operators rather than the quantum states. All the mappings we will consider are examples of encoding second quantization (**should I consider first quantisation techniques**).

explain advantages of second quantisation vqe chemistry review tranter 2015

4 Jordan-Wigner Transformation

The Jordan-Wigner transformation straight forwardly stores the occupation number of the i -th orbital in the i -th qubit. Therefore, all the non-local behaviour (the parity information) will have to be encoded in the operators [3]:

$$\begin{aligned} a_i &\rightarrow \left(\bigotimes_{k=1}^{i-1} Z_k \right) \sigma_i^- \\ a_i^\dagger &\rightarrow \left(\bigotimes_{k=1}^{i-1} Z_k \right) \sigma_i^+ \end{aligned} \tag{11}$$

where

$$\begin{aligned} \sigma_i^- &= \frac{1}{2}(X_i + iY_i) = |0\rangle_i \langle 1|_i \\ \sigma_i^+ &= \frac{1}{2}(X_i - iY_i) = |1\rangle_i \langle 0|_i \end{aligned} \tag{12}$$

It is clear to see that these operators act in exactly the same way on a qubit spin basis that the fermionic creation and annihilation operators act upon the Fock basis.

5 Bravyi-Kiteav Map

The Jordan-Wigner transformation uses $O(1)$ qubits to represent each fermionic mode, however it requires $O(N)$ gates to simulate one fermionic operation. This is because it stores the occupation number locally and the parity non-locally. An alternative scheme called the parity basis stores the parity locally and the occupation number non-locally, however this still requires $O(N)$ gates to simulate one fermionic operation [7]. The Bravyi-Kiteav is a halfway house which partially stores both the occupation number and parity non-locally.

Also talk about Fenwick trees [11] The following explanation of the mapping is adapted from [9]. First, we must define a partial order \preceq over a set of binary strings. For $\alpha = \alpha_{l-1} \dots \alpha_0$ and $\beta = \beta_{l-1} \dots \beta_0$, if $\alpha_i = \beta_i$ for $i \geq i_0$ and $\beta_i = 1$ for $i < i_0$, then $\alpha \preceq \beta$. For example:

$$\left. \begin{array}{l} 000 \prec 001 \\ 010 \\ 100 \prec 101 \\ 110 \end{array} \right\} \prec 011 \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \prec 111$$

For the i -th qubit we store the parity of all modes f_k with $k \preceq i$:

$$q_i = \sum_{k \preceq i} f_k \tag{13}$$

Alternatively, the transformation matrices β_i can be described recursively as in [7]. The base case β_1 is a (1×1) matrix with a single entry of 1. Then we construct β_{2^n} by taking $I \otimes \beta_{2^n}$ and filling the bottom row of the bottom-left quadrant of this matrix with 1s:

$$\beta_{2^{n+1}} = \left(\begin{array}{c|c} \beta_{2^n} & \mathbf{0} \\ \hline \mathbf{0} & \beta_{2^n} \\ \leftarrow 1 \rightarrow & \end{array} \right) \quad (14)$$

β_i with $2^x < i < 2^{x+1}$ is just the $i \times i$ segment of $\beta_{2^{x+1}}$ that includes b_0 through b_{i-1} . The mapping in the eight qubit case with the transformation matrix β_8 is shown below (with all sums taken modulo 2) [10]:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} \quad (15)$$

Lemma 1 [9]: Define the set $L(\beta)$ such that $\alpha \in L(\beta)$ if and only if for some i_0 , $\beta_{i_0} = 1$, $\alpha_{i_0} = 0$, $\alpha_i = \beta_i$ for $i > i_0$ and $\alpha_i = 1$ for $i < i_0$, then:

$$\sum_{j \in L(i)} q_j = \sum_{j < i} f_j \quad (16)$$

The following proof is of my own invention:

$$L(\alpha) = \{\alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 0 1 1 \dots 1 : \forall i \text{ s.t. } \alpha_{i_0} = 1\} \quad (17)$$

Define $A(\beta)$ as the set of j such that $j \preceq \beta$:

$$A(\beta) = \{\beta_{i_n} \beta_{i_{n-1}} \dots \beta_{j_0} \gamma_{j_0-1} \dots \gamma_0 : \forall j_0 \text{ s.t. } \beta_i = 1 \text{ for } i < j_0 \text{ and all } \gamma\} \quad (18)$$

Let j_{max} be the maximum j such that $\beta_i = 1$ for all $i < j$.

$$A(\beta) = \{\beta_{i_n} \beta_{i_{n-1}} \dots \beta_{j_{max}} \gamma_{j_{max}-1}^1 \dots \gamma_0^1, \\ \beta_{i_n} \beta_{i_{n-1}} \dots \beta_{j_{max}-1} \gamma_{j_{max}-2}^2 \dots \gamma_0^2, \dots, \beta_{i_n} \beta_{i_{n-1}} \dots \beta_0 : \text{ for all } \gamma\} \quad (19)$$

As γ has no restrictions all the later sets are actually included within the first so:

$$A(\beta) = \{\beta_{i_n} \beta_{i_{n-1}} \dots \beta_{j_{max}} \gamma_{j_{max}-1} \dots \gamma_0 : \text{ for all } \gamma\} \quad (20)$$

For $\beta \in L(\alpha)$, we have $j_{max} = i_0$ as $\beta_i = 1$ for $i < i_0$ and $\beta_{i_0} = 0$, therefore:

$$A(L(\alpha)) = \{\{\alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 0 \gamma_{i_0-1} \dots \gamma_0\} \forall i_0 \text{ s.t. } \alpha_{i_0} = 1 \text{ and all } \gamma\} \quad (21)$$

$$A(L(\alpha)) = \{\{j : j \preceq \alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 01 \dots 1\} \forall i_0 \text{ s.t. } \alpha_{i_0} = 1\} \quad (22)$$

As $A(\beta)$ for $\beta = 11 \dots 1$ contains every integer $j \leq \beta$, Eq. 22 is an algorithm for generating the set of all the integers less than α . It achieves this by working from right to left reducing every 1 to 0 and then iterating over all the possible suffixes whilst keeping the unaltered prefix. It never attains α as the first altered bit in every element is reduced from 1 to 0, though it does attain $\alpha - 1$ in the case of the rightmost bit being switched to zero and suffixed by $11 \dots 1$. Therefore,

$$\sum_{j \in L(i)} q_j = \sum_{k \in L(i)} \sum_{j \preceq k} f_j = \sum_{S \in A(L(i))} \sum_{j \in S} f_j = \sum_{j < i} f_j \quad (23)$$

□.

Lemma 2 [9]: Define the set $K(\beta)$ such that $\alpha \in K(\beta)$ if and only if for some i_0 , $\alpha_{i_0} = 0$, $\alpha_i = \beta_i$ for $i \neq i_0$ and $\beta_i = 1$ for $i < i_0$, then:

$$q_i - \sum_{j \in K(i)} q_j = \sum_{j < i} f_j \quad (24)$$

The following proof is of my own invention:

$$K(\alpha) = \{\alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 011 \dots 1 : \forall i_0 \text{ s.t. } \alpha_j = 1 \forall j < i_0\} \quad (25)$$

For $\beta \in K(\alpha)$, we have $j_{max} = i_0$ as $\beta_i = 1$ for $i < i_0$ and $\beta_{i_0} = 0$, therefore:

$$A(K(\alpha)) = \{\{\alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 0 \gamma_{i_0-1} \dots \gamma_0\} \forall i_0 \text{ s.t. } \alpha_j = 1 \forall j < i_0 \text{ and all } \gamma\} \quad (26)$$

$$A(K(\alpha)) = \{\{j : j \preceq \alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 01 \dots 1\} \forall i_0 \text{ s.t. } \alpha_j = 1 \forall j < i_0\} \quad (27)$$

Let i_{max} be the maximum i such that $\alpha_j = 1$ for all $j < i$:

$$A(K(\alpha)) = \{\{j : j \preceq \alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_{max}+1} 01 \dots 1\}, \{j : j \preceq \alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_{max}+1} 001 \dots 1\}, \dots, \{j : j \preceq \alpha_{i_n} \alpha_{i_{n-1}} \dots \alpha_{i_0+1} 01 \dots 10\}\} \quad (28)$$

$$A(K(\alpha)) = \{\{j : j \preceq \alpha^0\}, \{j : j \preceq \alpha^1\}, \dots, \{j : j \preceq \alpha^{i_{max}}\}\} \quad (29)$$

As α^i for $i \neq 0$ are all the integers one partial order below α :

$$\bigcup_{i>0} \{j : j \preceq \alpha^i\} = \{j : j < \alpha\} \quad (30)$$

As these sets are disjoint, $A(K(i))$ is equivalent to the following after applying the summation:

$$A(K(i)) = \{\{j : j \preceq i\}, \{j : j < i\}\} \quad (31)$$

So,

$$q_i - \sum_{j \in K(i)} q_j = \sum_{j \preceq i} f_j - \sum_{S \in A(K(i))} \sum_{j \in S} f_j = \sum_{j \preceq i} f_j - \left(\sum_{j \preceq i} + \sum_{j < i} \right) f_j = \sum_{j < i} f_j \quad (32)$$

□.

$K(i)$ is clearly a subset of $L(i)$ as $\alpha_{i_0} = 0$, $\alpha_i = \beta_i$ for $i > i_0$ and $\alpha_i = \beta_i = 1$ for $i < i_0$, satisfies the requirements that $\alpha_{i_0} = 0$, $\alpha_i = \beta_i$ for $i > i_0$ and $\alpha_i = 1$ for $i < i_0$ regardless of whether $\beta_{i_0} = 1$ or not.

To define the Bravyi-Kitaev operators we need to consider a more detailed treatment of how annihilating and creating fermionic modes affects the mapping. There are three different sets of qubits impacted by updating the fermionic mode f_i :

- **Parity set (P_i):** The set of qubits from which $\sum_{k < i} f_k$ can be deduced, and by Lemma 1 is equivalent to $L(i)$. This needs to be calculated so the correct phase change can be applied upon application of the fermionic operator in line with Eq. 10. (**maybe include the transformation matrix formulation**). The set $L(i)$ has at most $\log(i)$ members as there is at most one member per possible value for i_0 . Therefore, we only need to use $O(\log(N))$ Z gates to compute the parity rather than $O(N)$ gates in the Jordan-Wigner basis. Table 1 gives an example of the parity calculations in the eight qubit case:

Table 1. Example of calculating parity for eight qubit case

Qubit	i in binary	$L(i)$	Parity of qubits in $L(i)$
q_0	000		
q_1	001	000	f_0
q_2	010	001	$f_0 + f_1$
q_3	011	001, 010	$f_0 + f_1, f_2$
q_4	100	011	$f_0 + f_1 + f_2 + f_3$
q_5	101	011, 100	$f_0 + f_1 + f_2 + f_3, f_4$
q_6	110	011, 101	$f_0 + f_1 + f_2 + f_3, f_4 + f_5$
q_7	111	011, 101, 110	$f_0 + f_1 + f_2 + f_3, f_4 + f_5, f_6$

- **Update set (U_i):** The set of qubits that store the parity of f_i in their partial sum, and therefore need to be flipped when this mode is created or annihilated. This can easily be read off as the columns of β_n , or simply all j such that $i < j$. As every strict partial ordering reduces at least one binary digit from 1 to 0, the maximum number of composed strict partial orderings possible on index N is $\log_2(N)$. So, the maximum number of qubits that rely upon a given mode is $\log_2(N) + 1$ and is attained by f_0 if $N = 2^x$ for some $x \in \mathbb{N}$. Therefore, when we update an occupation number we only need to update $O(\log(N))$ qubits (using $O(\log(N))$ local X gates) rather than $O(N)$ as in the parity basis (**need to elaborate upon the parity basis earlier**).
- **Flip set (F_i):** The set of qubits from which $\sum_{k < i} f_k$ can be deduced, which is necessary as we need to know whether the qubit i has the same or flipped parity with respect to f_i . This depends on the parity of the other modes in the partial sum. For instance, if the other modes have parity 1 then $|q_i\rangle = |1\rangle \rightarrow f_i = 0$, then acting with a fermionic creation operator corresponds to applying a spin annihilation gate (σ_i^-) rather than a spin creation gate (σ_i^+). From Lemma 2, it is clear this is equivalent to $K(i)$, however it is also possible to find a formulation in terms of transformation matrices. As $\sum_j \beta_{ij}^{-1} q_j = f_i$ and β^{-1} is lower diagonal with 1 along the diagonal, we have $\sum_{j < i} \beta_{ij}^{-1} q_j = f_i - q_i = f_i - \sum_{j < i} f_j = \sum_{j < i} f_j$. Therefore, the rows of β^{-1} excluding the diagonal gives the Flip set. To compute the flip parity we need to apply $O(\log(N))$ Z gates as $K(i)$ has at most $\log(i)$ ($< \log(N)$) elements. As $K(i)$ is a subset of $L(i)$, the Flip set is a subset of the Parity set.

Now we can define the Bravyi-Kiteav creation and annihilation operators, in a similar fashion to [7]. We will use the notation U_S to mean U applied to every qubit in set S .

Using the flip set we can alter our definition of σ_i^- and σ_i^+ from Jordan-Wigner to reflect the fact that when the flip set has parity one we act with the inverse:

$$\begin{aligned}\sigma_i^- &= \frac{1}{2}(X_i Z_{K(i)} + iY_i) \\ \sigma_i^+ &= \frac{1}{2}(X_i Z_{K(i)} - iY_i)\end{aligned}\tag{33}$$

As this will introduce an overall sign equal to the parity of $K(i)$ to the operators σ_i^- and σ_i^+ , we only need to calculate the parity of the remaining qubits in the parity set (R_i) :

$$\begin{aligned}a_i &\rightarrow X_{U(i)} \sigma_i^- Z_{R_i} = \frac{1}{2} \left(\bigotimes_{j \succ i} X_j \otimes X_i \otimes \bigotimes_{j \in L(i)} Z_j + i \bigotimes_{j \succ i} X_j \otimes Y_i \otimes \bigotimes_{j \in R_i} Z_j \right) \\ a_i^\dagger &\rightarrow X_{U(i)} \sigma_i^+ Z_{R_i} = \frac{1}{2} \left(\bigotimes_{j \succ i} X_j \otimes X_i \otimes \bigotimes_{j \in L(i)} Z_j - i \bigotimes_{j \succ i} X_j \otimes Y_i \otimes \bigotimes_{j \in R_i} Z_j \right)\end{aligned}\tag{34}$$

Therefore, every fermionic operation in the Bravyi-Kiteav mapping only requires $O(\log(N))$ gates.

6 Derby-Klassen Map

As physical systems are normally dominated by local interactions such as lattice hopping or Coulomb interactions, it is preferable to design a 'local' mapping that has low Pauli-weights for local fermionic operators. These schemes only find representations for the few local interactions that exist in the Hamiltonian, rather than between every pair of modes like the Jordan-Wigner or Bravyi-Kitaev mapping. Generally this results in using more qubits but drastically smaller Pauli weights and thus fewer gates.

One strategy used to find 'local' mappings is searching for sets of low weight Pauli-strings for representing the fermionic edge (E_{jk}) and vertex (V_j) operators. These are defined below in terms of the Majorana operators γ_j :

$$\gamma_j = a_j + a_j^\dagger, \quad \bar{\gamma}_j = \frac{a_j - a_j^\dagger}{i}\tag{35}$$

$$E_{jk} = -i\gamma_j\gamma_k, \quad V_j = -i\gamma_j\bar{\gamma}_j\tag{36}$$

The full set of these operators represent the full even fermionic algebra [12] (**proove this**):

$$\begin{aligned}a_j^\dagger a_k + a_k^\dagger a_j &= (\gamma_j - i\bar{\gamma}_j)(\gamma_k + i\bar{\gamma}_k) + (\gamma_k - i\bar{\gamma}_k)(\gamma_j + i\bar{\gamma}_j) = -\frac{i}{2}E_{jk}(V_j - V_k) \\ a_j^\dagger a_j &= (\gamma_j - i\bar{\gamma}_j)(\gamma_j + i\bar{\gamma}_j) = \frac{1}{2}(I - V_j)\end{aligned}\tag{37}$$

So if we can find Pauli-strings that satisfy the anti-commutation relations (Eq. 38) [13] and the loop condition (Eq. 39) that define these edges and vertices then we have a mapping from even fermionic operators to qubits operators.

$$\begin{aligned} \{E_{jk}, V_j\} &= 0, \{E_{ij}, E_{jk}\} = 0 \forall i \neq k \\ [V_i, V_j] &= 0, [E_{ij}, V_m] = 0, [E_{ij}, E_{mn}] = 0 \forall i \neq j \neq m \neq n \end{aligned} \quad (38)$$

$$i^n \prod_{i=1}^n E_{p_i p_{i+1}} = 1 \quad (39)$$

for a cyclic sequence of sites $p = \{p_1, p_2, \dots, p_n, p_{n+1} : p_1 = p_{n+1}\}$.

In an intuitive sense these restrictions mean:

- (i) Edge operators must anti-commute with any vertices they include
- (ii) Edge operators must anti-commute with any edges they share vertices with
- (iii) All vertices must commute with each other
- (iv) All edges that do not share a common vertex must commute
- (v) All edges must commute with vertices they do not include
- (vi) Any cycle of edges must leave the state unchanged

In a ‘local’ mapping we find a way of mapping only a selection of these edges (ones representing local interactions) to qubit operators in such a way that the full even fermionic algebra is preserved. Charles Derby and Joel Klassen [14] have developed a design strategy for creating representations that satisfy these restrictions for a given regular lattice of interactions. We will illustrate this technique by using the simplest case of the square lattice [13]:

6.0.1 Definition of mapping

For a square lattice of fermionic modes, we map every vertex to a “vertex” qubit. Consider each square of the square lattice as either a black or white face (tiled like a chess board) and associate an additional “face” qubit with every black face. Let $f(i, j)$ index the unique odd face adjacent to edge (i, j) , and assign an orientation to the edges so they circulate clockwise or counter-clockwise around white spaces alternating every row. This is illustrated below by Fig. 3:

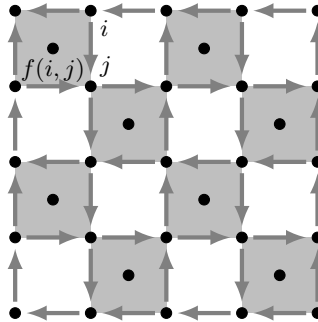


Fig. 3.

The design strategy involves mapping each edge operation to a Pauli string acting with a Y gate acting on the qubit at the head of the arrow and an X gate acting on the qubit at the tail of the arrow, and each vertex operation to a Z gate acting on the vertex. This is illustrated in Fig 4:



Fig. 4.

Straight away this can be seen to satisfy the intuitive conditions i), iii), iv) and v). Also as X and Y anti-commute, it is clear that every pair of edges that meet head-to-tail anti-commute, whereas pairs of edges that meet head-to-head or tail-to-tail commute. We can correct this behaviour by mapping every pair of these edges, those which currently commute but should anti-commute, to act upon a common ancillary qubit with X and Y gates respectively. Though this procedure is performed on a directed graph it still holds for edges in both directions as $E_{ij} = -E_{ji}$, so if E_{ij} anti-commutes with an operator so will E_{ji} . Therefore, by adding ancillary qubits we can make the mapping also satisfy restriction ii).

On a square lattice this common ancillary qubit can be the face qubit adjacent to the edges, as every pair of head-to-head or tail-to-tail edges share a common face qubit (see Fig 3). It can be seen that using an ancillary “face” qubit to introduce two separate anti-commutation relations does not break the restriction iv), as long as opposite edges of black squares act in commutable ways upon the “face” qubit (i.e. with same gate).

In order to satisfy the loop condition, applying any given loop of edges needs to correspond to an identity in the fermionic Fock space. In other words this mapping is only valid for fermionic state representations that lie in the space stabilised by cycles of edges. This is shown in detail by the following reasoning (found in [14]):

- (i) Let M_G be the group of directed edges e_{ij} and vertices v_i which satisfy Eq. 38 for a given graph G , and C_G be the directed cycles of graph G
- (ii) The group of even fermionic operators M_E is isomorphic to the quotient group M_G/C_G

Proof:

- (a) Define the following invertible transformation for all edges and vertices in G :

$$\begin{aligned}
 M_G/C_G &\longleftrightarrow M_E \\
 v_i C_G &\longleftrightarrow V_i \\
 e_{ij} C_G &\longleftrightarrow E_{ij}
 \end{aligned} \tag{40}$$

- (b) M_E contains all the possible edges not just those in G . As G is connected, all edges not in G can be constructed by composition of defined edge operators in G . Define this composition as:

$$e_{ik} = e_{ij} \circ e_{jk} \longleftrightarrow E_{ik} = iE_{ij}E_{jk} \tag{41}$$

- (c) As M_G satisfies Eq. 38 for all e_{ij} and v_i in G , this holds for M_G/C_G , and therefore for all E_{ij} and V_i in G .
- (d) All E_{ij} not in G can be decomposed into E_{ij} in G and the introduced phase difference will have no impact on Eq. 38. Therefore, all edges and vertices satisfy the anti-commutation and commutation relations
- (e) All cyclic paths in G satisfy Eq. 39 as:

$$I = f(IC_G) = f\left(i^n \prod_{i=1}^n e_{p_i, p_{i+1}} C_G\right) = i^n \prod_{i=1}^n E_{p_i, p_{i+1}} \quad (42)$$

- (f) All cyclic paths that include edges (\tilde{E}_{ij}) not in G also satisfy Eq. 39 as the definition of composition adds a factor of i for every edge that gets added so the total phase is preserved:

$$\begin{aligned} i^{n+2} \prod_{i=0}^n E_{p_i, p_{i+1}} \tilde{E}_{p_{n-1}, p_n} \tilde{E}_{p_n, p_{n+1}} &= \\ i^{n+2} \prod_{i=0}^n E_{p_i, p_{i+1}} i E_{p_{n-1}, p_n} E_{p_n, p_{n+1}} i E_{p_{n+1}, p_l} E_{p_l, p_{n+1}} &= i^{n+4} \prod_{i=1}^{n+4} E_{p_i, p_{i+1}} = I \end{aligned} \quad (43)$$

□.

- (iii) The quotient group M_G/C_G has a faithful representation in the Hilbert subspace U as:
 - (a) The group M_G has a representation σ in the Hilbert space \mathcal{H} of a system of qubits
 - (b) If a stabiliser of a non-trivial subgroup of σ contains all the elements of C_G , then can define new representation τ as:

$$\tau(mC_G) = Proj_U \circ \sigma(m) \forall m \in M_G/C_G \quad (44)$$

with U the common +1 eigenspace of $\sigma(C_G)$.

- (iv) If σ is a representation in the multi-qubit Pauli operators $\{\pm 1, \pm i\} \times_i^N \{I_i, X_i, Y_i, Z_i\}$ then any subgroup S that is abelian and does not contain $-I$ is the stabiliser of a non-trivial vector space [6].
- (v) Since C_G is abelian (**Proof**), $\sigma(C_G)$ is abelian. So if $-I \in \sigma(C_G)$, $\sigma(C_G)$ is the stabiliser of the common +1 eigenspace of $\sigma(C_G)$
- (vi) M_G/C_G (and hence M_E) has a representation in the code space of $\sigma(C_G)$

Therefore, the mapping we have found is valid as long as no cycle is mapped to $-I$. As all cycles on this square lattice can be decomposed into a combination of cycles around single black and white faces (**proof**), we only need to consider these stabiliser generators individually.

As shown in Fig 5, looping a black face produces $-I$ meaning it cannot be in a stabilizer, however this can be fixed by simply flipping the sign of one of the four edges. Then we get $X_1 Y_5 Y_2 \otimes -Y_2 X_5 X_3 \otimes X_3 Y_5 Y_4 \otimes Y_4 X_5 X_1 = -X_1^2 Y_2^2 X_3^2 Y_4^2 (Y_5 X_5)^2 = -(iZ_5)^2 = I$. It can be seen in Fig 6, that looping a white face produces a non-trivial stabilizer, which is

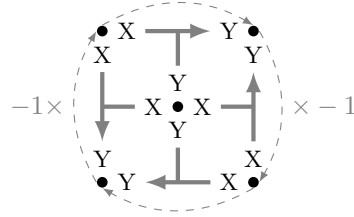


Fig. 5. Diagram of looping around black face. This gives $X^2 = Y^2 = I$ at every “vertex” qubit, but $(-X)Y(-X)Y = (XY)^2 = (iZ)^2 = -I$ at the “face” qubit. The two negative signs introduced by having to flip the orientation of the vertical edges cancel each other out.

not $-I$. Therefore, using the mapping described (with the sign flipped on one edge of every black face) will give a faithful representation of the even fermionic algebra.

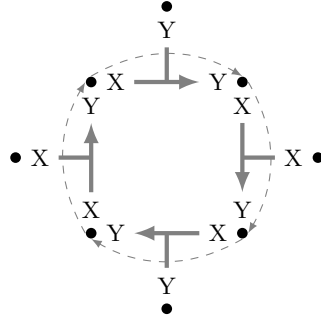


Fig. 6. Diagram of looping around white face.

Therefore, we can map the edges in the direction of their assigned orientation as:

$$E_{ij} = \begin{cases} X_i Y_j X_{f(i,j)} & (i,j) \text{ oriented downwards} \\ -X_i Y_j X_{f(i,j)} & (i,j) \text{ oriented upwards} \\ X_i Y_j Y_{f(i,j)} & (i,j) \text{ horizontal} \end{cases} \quad (45)$$

with the last term ignored if the edge lies on a boundary with no adjacent face qubit. Traversing edges in the opposite direction to their orientation just involves applying a sign change.

We also map the vertex operators to:

$$V_j = Z_j \quad (46)$$

7 Fermionic Enumeration

Mitchell Chiew and Sergii Strelchuk [3] realised that for all the mappings we have discussed so far there is the potential for reducing resource costs without changing anything other than the order we encode qubits. This is achieved by improving the locality of the mapping, so local operations have lower Pauli weights. In the following treatment (adapted from [3]) we will consider the application to the Jordan-Wigner transformation. **(waiting for new paper on applications to mappings with additional auxillary)**

In 1-D the enumeration is trivial, however, in 2D it becomes an interesting problem. For instance, consider a fermionic square lattice every mode must be assigned a qubit from a 1-D qubit array. There are many different ways of accomplishing this; most notably the S or Z pattern illustrated in Fig ???. As discussed in Section 6, we are often interested in Hamiltonians with local interactions (e.g. between adjacent modes in the lattice). Therefore, the particular enumeration you use will impact the number of qubits that lie between neighbouring modes in the lattice. This is important as in the Jordan-Wigner transformation the Pauli-weight of a fermionic operation scales with the difference in enumeration. Thus by minimising the difference in enumeration between adjacent modes we minimise the average Pauli-weight of local interactions. This reduces to the problem of simple optimal linear arrangement (or edgsum) [15] from graph theory, and is formulated as:

Simple Optimal Linear Arrangement: For a graph G with vertices V and edges E . Find an enumeration scheme $f : V \rightarrow \{1, 2, \dots, n\}$ that minimises the cost function

$$C(f) = \sum_{\alpha, \beta} |f(\alpha) - f(\beta)| \quad (47)$$

where the average Pauli weight is given by $\frac{C(f)}{|E|} + 1$.

This problem is NP-complete [15] though a number of solutions are known for a range of graphs. We will consider the case of the square lattice to illustrate the principle, but it could be equally well applied to any of the other graphs with known solutions.

The Mitchison-Durbin pattern (illustrated in Fig ??) provides the fermion enumeration scheme that minimises the average Pauli weight of hopping terms in a square lattice. The following proof of this proposition is adapted from [3] and [16]:

- **Lemma 7.1:** The minimum cost will be attained by a horizontally and vertically ordered enumeration
 - (a) Horizontal ordering means each row increases in numbering from left to right, and vertical ordering means each column increases in numbering from top to bottom
 - (b) It is clear that horizontally ordering a row will always decrease horizontal contributions to $C(f)$ (**is this clear?**)
 - (c) **Lemma 7.1.1:** Horizontally ordering an enumeration will never increase the vertical contributions to $C(f)$
 - (i) Horizontal ordering will only ever affect the vertical contributions if it acts upon adjacent rows differently, so modes that were not vertically adjacent now are. This only occurs if one row is in a different horizontal ordering to the other.
 - (ii) It suffices to consider only the generative cases of swapping two modes:
 - $f(\mu, \nu) < f(\mu, \nu + 1)$ and $f(\mu + 1, \nu) > f(\mu + 1, \nu + 1)$ or
 - $f(\mu, \nu) > f(\mu, \nu + 1)$ and $f(\mu + 1, \nu) < f(\mu + 1, \nu + 1)$
 - (iii) In both instances, before vertical ordering their vertical contributions are

- $|f(\mu, \nu) - f(\mu + 1, \nu)| + |f(\mu, \nu + 1) - f(\mu + 1, \nu + 1)|$, and after vertical ordering their vertical contributions are $|f(\mu, \nu) - f(\mu + 1, \nu + 1)| + |f(\mu, \nu + 1) - f(\mu + 1, \nu)|$
- (iv) WLOG map the four modes to $\{i_1, i_2, i_3, i_4 : i_k > i_j \text{ for } k > j\}$. Before the vertical ordering we have (I) $|i_4 - i_2| + |i_1 - i_3|$, (II) $|i_4 - i_1| + |i_2 - i_3|$ or (III) $|i_4 - i_1| + |i_3 - i_2|$, and after we have $|i_4 - i_3| + |i_1 - i_2|$, $|i_4 - i_2| + |i_3 - i_1|$ or $|i_4 - i_3| + |i_2 - i_1|$ respectively. As $|i_4 - i_2| > |i_4 - i_3|$ and $|i_1 - i_3| > |i_1 - i_2|$, vertical ordering decreases the contribution in case I. As $|i_4 - i_1| > |i_4 - i_3|$ and $|i_1 - i_2| > |i_3 - i_2|$, vertical ordering decreases the contribution in case III. Case II simplifies to $i_4 + i_3 - i_1 - i_2$ before and after so the contribution remains the same. \square .
- (d) In a symmetrical manner to Lemma 7.1.1 vertically ordering an enumeration will never increase the horizontal contributions, and it will clearly only ever decrease the vertical contributions
- (e) **Lemma 7.1.2:** Vertical ordering preserves horizontal ordering
- (i) Assume the contrary that there is a horizontally ordered enumeration h_{ij} that has been vertically ordered to enumeration v_{ij} , and now is no longer horizontally ordered
- (ii) This means that there is some $i < j$ and k such that $v_{ik} > v_{jk}$. Vertical ordering implies that $v_{jn} \leq v_{jk}$ for all $n \in [1, k]$. Therefore, as vertical ordering just reorders within columns ($v_{jn} = h_{jm}$) has k values of m for which $h_{jm} \leq v_{jk}$. Therefore, every column of h to the left of j has k values of m for which $h_{jm} < v_{jk}$. As $i < j$, column i has k values of m for which $h_{im} \leq v_{jk} < v_{ik}$, which is a contradiction as v_{ik} only has $k - 1$ values less than it in its column \square .
- (f) Given an optimal enumeration with minimum cost we can apply horizontal ordering without increasing the cost, and then apply vertical ordering without disrupting the horizontal ordering or increasing the cost. Therefore, the minimum cost will be attained by a horizontally and vertically ordered enumeration \square .
- **Lemma 7.2:** Given a horizontally and vertically ordered enumeration scheme for the lattice, the cost function for the minimum-1-sum problem is:

$$C(f) = \sum_{\alpha \in V_b} f(\alpha) + \sum_{\alpha \in V_r} f(\alpha) - \sum_{\alpha \in V_l} f(\alpha) - \sum_{\alpha \in V_t} f(\alpha) \quad (48)$$

where V_b , V_r , V_l and V_t are the vertices in the bottom row, right column, left column and top row respectively.

- (a) As horizontally and vertically ordered every absolute sign can be evaluated:

$$C(f) = \sum_{(\mu, \nu) \in V} f(\mu + 1, \nu) - f(\mu, \nu) + f(\mu, \nu + 1) - f(\mu, \nu) \quad (49)$$

Could finish this proof off but I think it might take up too much space and isn't necessary for understanding the intuition of fermionic encoding

8 Relative performance

Each of these encodings offers various advantages and disadvantages when applied to real applications such as VQEs or phase estimation discussed in Section 2. As explained in Section 2.1, the efficiency of these techniques is broadly based on three factors: number of qubits required, average Pauli-weight and the number of distinct Pauli-strings in the Hamiltonian. All of the different mappings make different trade-offs between these factors.

There are some techniques [17] which decrease the number of qubits required by increasing the number of distinct Pauli-strings, however we have not covered them here. All the techniques discussed in this essay leave the number of terms in the Hamiltonian unchanged.

The main trade-off at play is between the Jordan-Wigner/Bravyi-Kitaev style mappings and the 'local' (Derby-Klassen style) mappings. 'Local' mappings sacrifice an increased qubit count for drastically more localised operations. Applying a local fermionic interaction on a Jordan-Wigner or Bravyi-Kitaev mapping will take $O(N)$ and $O(\log(N))$ gates respectively, whereas on the Derby-Klassen mapping it only takes 3. This significant reduction in gate count is coupled with an increase in qubit count with Derby-Klassen requiring 1.5 qubits per fermionic mode compared to the 1 qubit per mode that Jordan-Wigner and Bravyi-Kitaev need. However, the constant gate count only holds for local interactions and performing an operation involving two arbitrary modes would scale with the diameter of the graph of local interactions. Therefore, the Derby-Klassen design scheme is only useful for Hamiltonians with local interactions, and is most effective for interactions on a regular lattice for which a mapping is known [14]. These include all uniform tilings of degree less than 4, and the cubic lattice.

There is also an important discussion to be had about the relative merits of the Jordan-Wigner and Bravyi-Kitaev transformations. Talk about how the long strings of Z increase the depth as it makes it hard to decompose in parallel

There is a typo on page 18 Optimal fermion qubit mapping [3] "Because $i < j$ and $m > n$ " not "Because $i < j$ and $m < n$ ".

9 References

pt!

1. Feynman, R.P.. *Simulating physics with computers*. International Journal of Theoretical Physics 1982;21(6-7):467–488. doi:10.1007/bf02650179.
2. Pascual Jordan and Eugene Wigner. *Über das Paulische Äquivalenzverbot*. Zeitschrift für Physik, 47(9-10):631651, September 1928.
3. Chiew, M. and Strelchuk, S.. *Optimal Fermion-Qubit Mappings*. ArXiv:2110.12792 [Quant-Ph], 25 October 2021. <http://arxiv.org/abs/2110.12792>
4. Tilly, Jules, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, et al. *The Variational Quantum Eigensolver: A Review of Methods and Best Practices*. ArXiv:2111.05176 [Quant-Ph], 9 November 2021. <http://arxiv.org/abs/2111.05176> **look for journal version**
5. McArdle, Sam, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. *Quantum Computational Chemistry*. Reviews of Modern Physics 92, no. 1 (30 March 2020): 015003. <https://doi.org/10.1103/RevModPhys.92.015003>.
6. Nielsen, M.A., Chuang, I.L.. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press; 2009. ISBN
7. Seeley, Jacob T., Martin J. Richard, and Peter J. Love. *The Bravyi-Kitaev Transformation for Quantum Computation of Electronic Structure*. The Journal of Chemical Physics 137, no. 22 (14 December 2012): 224109. <https://doi.org/10.1063/1.4768229>. 9780511976667. doi:10.1017/cbo9780511976667.
8. Suzuki, Masuo. *Generalized Trotter’s Formula and Systematic Approximants of Exponential Operators and Inner Derivations with Applications to Many-Body Problems*. Communications in Mathematical Physics 51, no. 2 (June 1976): 183–90. <https://doi.org/10.1007/BF01609348>.
9. Bravyi, Sergey, and Alexei Kitaev. *Fermionic Quantum Computation*. Annals of Physics 298, no. 1 (May 2002): 210–26. <https://doi.org/10.1006/aphy.2002.6254>.
10. Tranter, Andrew, Peter J. Love, Florian Mintert, and Peter V. Coveney. *A Comparison of the Bravyi-Kitaev and Jordan-Wigner Transformations for the Quantum Simulation of Quantum Chemistry*. Journal of Chemical Theory and Computation 14, no. 11 (13 November 2018): 5617–30. <https://doi.org/10.1021/acs.jctc.8b00450>.
11. Havlíček, Vojtěch, Matthias Troyer, and James D. Whitfield. *Operator Locality in Quantum Simulation of Fermionic Models*. Physical Review A 95, no. 3 (29 March 2017): 032332. <https://doi.org/10.1103/PhysRevA.95.032332>.
12. Setia, Kanav, Sergey Bravyi, Antonio Mezzacapo, and James D. Whitfield. *Superfast Encodings for Fermionic Quantum Simulation*. Physical Review Research 1, no. 3 (18 October 2019): 033033. <https://doi.org/10.1103/PhysRevResearch.1.033033>.
13. Derby, Charles, and Joel Klassen. *A Compact Fermion to Qubit Mapping*. Physical Review B 104, no. 3 (8 July 2021): 035118. <https://doi.org/10.1103/PhysRevB.104.035118>.
14. Derby, Charles, and Joel Klassen. *A Compact Fermion to Qubit Mapping Part 2: Alternative Lattice Geometries*. ArXiv:2101.10735 [Quant-Ph], 26 January 2021. <http://arxiv.org/abs/2101.10735>.
15. Garey, M. R., D. S. Johnson, and L. Stockmeyer. *Some Simplified NP-Complete Graph Problems*. Theoretical Computer Science 1, no. 3 (1 February 1976): 237–67. [https://doi.org/10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1).
16. Mitchison, Graeme, and Richard Durbin. *Optimal Numberings of an $N \times N$ Array*. SIAM Journal on Algebraic Discrete Methods 7, no. 4 (1 October 1986): 571–82. <https://doi.org/10.1137/0607063>.
17. Moll, Nikolaj, Andreas Fuhrer, Peter Staar, and Ivano Tavernelli. *Optimizing Qubit Resources for Quantum Chemistry Simulations in Second Quantization on a Quantum Computer*. Journal of Physics A: Mathematical and Theoretical 49, no. 29 (22 July 2016): 295301. <https://doi.org/10.1088/1751-8113/49/29/295301>.
18. P. Horodecki and R. Horodecki (2001), *Distillation and bound entanglement*, Quantum Inf. Com-

- put., Vol.1, pp. 045-075.
19. R. Calderbank and P. Shor (1996), *Good quantum error correcting codes exist*, Phys. Rev. A, 54, pp. 1098-1106.
20. M.A. Nielsen and J. Kempe (2001), *Separable states are more disordered globally than locally*, quant-ph/0105090.
21. A.W. Marshall and I. Olkin (1979), *Inequalities: theory of majorization and its applications*, Academic Press (New York).