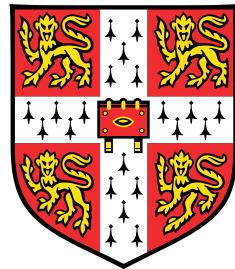


# Defining, classifying and optimising fermion–qubit mappings



**Mitchell Lee Chiew**

Department of Applied Mathematics and Theoretical Physics  
University of Cambridge

This dissertation is submitted for the degree of  
*Doctor of Philosophy*

Corpus Christi College

December 2024



## **Declaration**

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted, or, is being concurrently submitted, for any degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Mitchell Lee Chiew  
December 2024



## Acknowledgements

I would like to express my first and foremost thanks to my supervisor, Professor Sergii Strelchuk, for his consistent support and patience and over the past four years.

Thanks also go to Brent Harrison, James Whitfield, Riley Chien and other members of the Dartmouth group for having me on board this year and teaching me so much about the world of fermion–qubit mappings.

My time in the maths department was made all the richer by the CQIF renaissance: Florian, who humoured my lay interest in countercultural digital movements, Campbell for helping me stir through Chicken Tonight, Wilfred for perfectly complementing my interest in film and TV, and Josh for saving my life by introducing me to the Inkscape tool. Thanks also go to the younglings: Oliver for the late-stage (of my) PhD lunches and Orson Yi for helping carry all the snacks. A special thanks Subhayan for the encouragement and support down to the last minute, and for the unfettered access to the best office in the department.

Sincere gratitude goes to my college tutor, Ewan St. John Smith, who provided help when I needed it most, no matter the hour.

Thanks to Will for your support, food, tree-based humour and company through the last year and a half of my PhD. Elena also deserves much praise for joining us on journey through *all* of the Alien franchise.

To the Corpus crew: Satish Bharathwaj Viswanathan, we now share a bond deeper than blood now, and I hope we both make it to Rome to celebrate it. To Jenni, Astrid and Robbie; tennis??? Thanks to David, for making me get Letterboxed, and to Cat, for making sure I use it.

James and Miren, see you soon: I can't wait for more dinner and Doctor Who. And to Toby, I hope I have not let you down with my use of em- and en-dashes in this work.

Thanks to Andrea for filling Rainbow Rocket with non-maths, non-academic talk: I am very fortunate to have ended up in the same town as the OG cuber. And thanks to Ieva for the support and encouragement towards the end of my PhD.

My friends from Perth: Ryan, thank you for always being up for a chat, and Jonathan for your quick, timely advice and making sure I get out and about. I am

very grateful to Michael and Zach for providing direct lines to my true self via REMs, West Sheen and Pantheon, and I promise to get some balls back in the court now that I'm done with this.

To my family: Sally, thanks for looking after me in the UK. Tim, Vanessa, Phoebe, William and Emerson: I'm back.

## Abstract

This thesis presents a comprehensive framework for defining, classifying, and optimizing fermion–qubit mappings, a crucial step in simulating fundamental physics on quantum computers. This work surveys the catalogue of fermion–qubit mappings, exposing gaps and devising new optimisation routines to aid fermionic simulation through a consistent notation that accounts for all of the mappings in the literature.

We unify the distinct operator– and state–based notations of fermion–qubit mappings throughout the field in one definition, providing a versatile description that underpins the entire thesis. Our notation explores the intersection of two significant categories of mappings – Pauli–based mappings, which include ternary tree transformations, and the product–preserving mappings, which include classical, affine, and linear encodings of the Fock basis. By deriving the explicit formula for the Majorana operators, we show that affine encodings lie at the intersection of these two broad classifications, while also illustrating the vastness of the set of Pauli–based mappings.

We then introduce a classification system to partition the Pauli–based mappings into distinct templates, accounting for trivial labelling symmetries of Pauli operators, fermionic modes, and qubits. The classification allows us to prove that the product–preserving ternary tree transformations are equivalent to a subset of the linear encodings of the Fock basis, uniting two significant, yet disparate, areas of active research.

We then define optimisation problems for fermion–qubit mappings, resulting in new strategies to increase the efficiency of fermionic simulation. Our classification system helps establish the link between fermionic labelling and typical cost models of qubit Hamiltonians. For the Jordan–Wigner transformation, the optimal fermionic labellings for cost functions of Pauli weight correspond to solutions of well–known problems in graph theory. We use the solution to the edgesum problem to find the optimal Jordan–Wigner transformation for square fermionic lattices, revealing a latent reduction of 13.9% in the average Pauli weight of the resulting qubit Hamiltonian.

Finally, in pursuit of further efficiency, we extend our overarching definition to include ancilla–qubit mappings, introducing a method for incrementally adding ancilla qubits to further reduce the Pauli weight of qubit Hamiltonians. We demonstrate that

just two extra qubits can achieve a 37.9% reduction in average Pauli weight versus the standard ancilla-free approach.

In conclusion, this thesis addresses key mathematical and notational gaps in the field and offers a new perspective for designing cost-effective fermion–qubit mappings for fermionic simulations.

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.0.1	Acknowledgement of collaboration . . . . .	2
1.1	Quantum physics . . . . .	3
1.1.1	Fundamental particles . . . . .	5
1.1.2	Fermionic operators and modes . . . . .	6
1.1.3	Majorana operators . . . . .	9
1.1.4	Fermionic Hamiltonians . . . . .	11
1.2	Quantum computing . . . . .	13
1.2.1	Pauli operators . . . . .	14
1.2.2	Clifford operators . . . . .	16
1.2.3	The quantum resource cost of a unitary operator . . . . .	17
1.2.4	Quantum algorithms to simulate Hamiltonians . . . . .	19
1.3	Fermion–qubit mappings . . . . .	20
1.3.1	Definitions, classifications and optimisations of fermion–qubit mappings in the literature . . . . .	21
1.4	Thesis summary . . . . .	24
<b>2</b>	<b>Defining fermion–qubit mappings</b>	<b>27</b>
2.1	A mathematical definition for fermion–qubit mappings . . . . .	28
2.1.1	Applying fermion–qubit mappings to Hamiltonians . . . . .	34
2.2	Pauli–based fermion–qubit mappings . . . . .	35
2.2.1	Counting the number of Pauli–based mappings . . . . .	37
2.2.2	Ternary tree transformations . . . . .	41
2.3	Fock basis hierarchy of fermion–qubit mappings . . . . .	43
2.4	Majorana representations of affine encodings . . . . .	48
2.5	Conclusion . . . . .	58

---

<b>3 Classifying fermion–qubit mappings</b>	<b>61</b>
3.1 An equivalence relation for Pauli–based mappings . . . . .	62
3.1.1 Labelling symmetries of Pauli strings . . . . .	62
3.1.2 Three–mode product–preserving Pauli–based mappings . . . . .	66
3.2 Product–preserving ternary tree transformations and linear encodings .	68
3.2.1 Uniqueness of product–preserving ternary tree transformations .	69
3.2.2 Outline for proof of Theorem 2 . . . . .	76
3.2.3 Supporting results for Theorem 2 . . . . .	76
3.2.4 Main theorem: equivalence of ternary tree transformations to linear encodings . . . . .	89
3.2.5 The pruned Sierpinski Tree as a ternary tree transformation .	90
3.3 Conclusion . . . . .	93
3.3.1 Future work . . . . .	93
<b>4 Optimising fermion–qubit mappings</b>	<b>95</b>
4.1 Pauli–based mappings as optimisation problems with cost functions .	96
4.2 Optimising fermionic labellings . . . . .	98
4.2.1 Problem Hamiltonians and hopping terms . . . . .	99
4.2.2 Optimising fermionic labelling within templates . . . . .	102
4.2.3 Complexity–theoretical preliminaries . . . . .	107
4.2.4 Jordan–Wigner transformations of square–lattice fermionic Hamil- tonians with minimum average Pauli weight . . . . .	110
4.3 Proof of Theorem 3 . . . . .	115
4.3.1 Edgesum of the Mitchison–Durbin pattern . . . . .	128
4.4 Conclusion . . . . .	131
4.4.1 Geometric locality on qubit architectures . . . . .	132
4.4.2 Improving beyond optimality with a constant number of ancilla qubits . . . . .	134
<b>5 Further optimisation with a constant overhead of ancilla qubits</b>	<b>135</b>
5.1 The theory of ancilla–qubit mappings . . . . .	136
5.2 Incremental addition of ancilla qubits . . . . .	138
5.2.1 The first ancilla qubit . . . . .	138
5.2.2 The second ancilla qubit . . . . .	143
5.2.3 Subsequent ancilla qubits . . . . .	145
5.3 Beyond optimality with only two ancilla qubits . . . . .	146
5.3.1 Ancilla–qubit Jordan–Wigner transformations . . . . .	146

5.3.2 Beyond the ancilla-free optimality of Jordan–Wigner transformations for square-lattice fermionic systems . . . . .	150
5.4 Conclusion . . . . .	159
5.4.1 Constant ancilla mappings that are not Jordan–Wigner transformations . . . . .	160
<b>6 Conclusion</b>	<b>161</b>
<b>References</b>	<b>163</b>
<b>Appendix A Glossary</b>	<b>175</b>



# Chapter 1

## Introduction

Simulating physical systems is one of the most promising applications of quantum computers. Fermionic systems are essential components in several fields of the theoretical and experimental sciences, from quantum physics and chemistry to condensed matter and quantum field theories. Fermions pose complex, often intractable computational challenges when studied with the aid of classical computers, such as tackling the electronic structure problem, studying properties of gauge theories that govern strong interactions between quarks and gluons, and determining ground state properties of fermionic Hamiltonians.

Fermion–qubit mappings are mathematical objects that transfer all of the information and dynamics of fermionic statistics to the operators and states of quantum computers. Underpinning the core steps of any fermionic simulation algorithm, the choice of fermion–qubit mapping can affect the resource demands from gate count and circuit depth to the total number of qubits. With quantum technology in its infancy, a burgeoning supply of fermion–qubit mappings has arisen over the past twenty years to meet various demands, anticipating the limited connectivity of near–term hardware and the locality requirements for long–term simulation algorithms.

This thesis provides a comprehensive account of fermion–qubit mappings which leads to several new insights. We have dedicated considerable effort in not only discovering optimal mappings for practical applications, but also in meticulously defining and classifying fermion–qubit mappings from foundational mathematical principles. The rigorous notation and language that we establish reveals connections between disparate sections of the literature, both revealing overlaps and filling gaps left by the existing discourse in the field.

### 1.0.1 Acknowledgement of collaboration

Collaboration has played a large part in the synthesis of the ideas and results in this thesis, primarily with my supervisor Sergii Strelchuk (SS). I derived the proof for every lemma, theorem, and corollary in this thesis unless otherwise stated in the text.

**Chapter 2:** This chapter is based on material from [1] and unpublished work. The general definition for fermion–qubit mappings in Section 2.1 is my own work. The idea to distinguish Pauli–based mappings from completely general mappings in Section 2.2 arose from discussions with Brent Harrison (BH).

**Chapter 3:** Chapter 3 exclusively contains results from [1] that are my own work, and which was inspired by conversations I had with my co-authors on [2]. When we started collaborating, BH et al. had discovered the Sierpinski tree transformation while SS and I had independently discovered a very similar result while studying the ternary tree transformation. The similarity of these results encouraged us to search for the general link between linear encodings and ternary tree transformations, which resulted in [1].

**Chapter 4:** The majority of content in this chapter comes from [3]. It is my own work, with input from SS. The idea to use the Mitchison–Durbin pattern as an alternative to the existing fermionic labellings in the literature was SS’s.

**Chapter 5:** The majority of the content in this chapter comes from [3]. It is my own work, with input from SS. The idea to use a constant number of ancilla qubits was SS’s. Sections 5.1 and 5.2 are substantial revisions of the ancilla–qubit mapping definition that appears in [3], which I updated as part of unpublished work with Cameron Ibrahim, Ilya Safro, and SS.

When this thesis uses content from the papers [1–3], whether verbatim or paraphrased, it is from sections that I wrote myself. All deviation from the published works as they appear online is a result of my own work. I created all illustrations and figures in this thesis.

## 1.1 Quantum physics

Discoveries in fundamental physics at the turn of the 20th century led to the concept of wave–particle duality and the adoption of wavefunctions to describe both massless and massive fundamental particles.

The wavefunction  $\psi$  of a  $d$ –dimensional quantum system is a continuous and differentiable assignment of complex values to spacetime

$$\psi : \mathbb{R}^d \times \mathbb{R} \longrightarrow \mathbb{C}, \quad (1.1)$$

which satisfies the normalisation condition in the coordinate dimensions

$$|\psi|^2 = \int d^d \mathbf{r} \psi^*(\mathbf{r}, t) \psi(\mathbf{r}, t) = 1, \quad (1.2)$$

where  $z^*$  is the complex conjugate of  $z$  for  $z \in \mathbb{C}$ . Wavefunctions are normalised square–integrable functions belonging to the separable Hilbert space  $L^2 = \{\psi : \mathbb{R}^d \rightarrow \mathbb{C} \mid |\psi|^2 < \infty\}$ , where  $|\psi|^2 = \langle \psi, \psi \rangle$  with inner product

$$\langle \psi_1, \psi_2 \rangle = \int d^d \mathbf{r} \psi_1^*(\mathbf{r}, t) \psi_2(\mathbf{r}, t) \quad \text{for } \psi_1, \psi_2 \in L^2. \quad (1.3)$$

Dirac notation introduces a vector space consisting of *kets* or *states*  $|\psi\rangle$  for each wavefunction  $\psi \in L^2$ , with the inner product  $\langle \psi_1 | \psi_2 \rangle = \langle \psi_1, \psi_2 \rangle$ . In bra–ket notation, the wavefunction has the representation

$$|\psi\rangle_{\text{full}} := \int d^d \mathbf{r} \psi(\mathbf{r}, t) |\mathbf{r}\rangle, \quad (1.4)$$

where  $|\mathbf{r}\rangle$  is the ket representing the eigenfunction  $\phi_{\mathbf{r}}(\mathbf{r}') = \delta(\mathbf{r}' - \mathbf{r})$  of the position operator  $\hat{\mathbf{R}}$  with  $\hat{\mathbf{R}}\phi_{\mathbf{r}} = \mathbf{r}\phi_{\mathbf{r}}$ . The set  $B(L^2)$  of bounded linear operators on  $L^2$  and wavefunctions  $\psi \in L^2$  are thus interchangeable with matrix and vector counterparts.

Under the Copenhagen interpretation of quantum mechanics, an observable quantity of a system, such as energy or the momenta and positions of constituent particles, corresponds to a Hermitian operator  $\hat{O}$  acting on the discrete subspace  $\mathcal{H} := \text{span}_{\mathbb{C}}\{\phi_i\}_{i \in \mathcal{R}} \leq L^2$  of the square–integrable functions, where the countable set  $\mathcal{R} \subseteq \mathbb{N}$  indexes the eigenfunctions  $\{\phi_i\}_{i \in \mathcal{R}}$  of  $\hat{O}$ . In bra–ket notation, the operator  $\hat{O}$  acts as a Hermitian matrix on the eigenvectors  $\{|\phi_i\rangle\}_{i \in \mathcal{R}}$ ,

$$\hat{O} |\phi_i\rangle = \lambda_i |\phi_i\rangle \quad \text{for all } i \in \mathcal{R}, \quad (1.5)$$

where the eigenvalues  $\lambda_i$  of  $\hat{O}$  make up the permissible values for the observable.

Reducing the full wavefunction of a system to its support on the eigenspaces of  $\hat{O}$ , and denoting the vector representation of eigenfunctions as elements of  $\mathcal{H}$ , it is typical to define the state of a system of interest as a normalised linear combination, or *superposition*, of the eigenstates

$$|\psi\rangle := \sum_{i \in \mathcal{R}} \langle \phi_i | \psi \rangle |\phi_i\rangle = \sum_{i \in \mathcal{R}} c_i |\phi_i\rangle \in \mathcal{H}, \quad \text{where} \quad \sum_i |c_i|^2 = 1, \quad (1.6)$$

which captures all the information of the system relevant to the observable. The Born rule states that the measurement of the value  $\lambda_i$  when observing a system in state  $|\psi\rangle$  occurs with probability  $|\langle \phi_i | \psi \rangle|^2 = |c_i|^2$  and causes the wavefunction collapse

$$\text{measure } \lambda_i : |\psi\rangle \longmapsto |\phi_i\rangle. \quad (1.7)$$

Multipartite quantum systems consist of interacting and non-interacting particles. The state space of an  $n$ -particle system has the tensor product form

$$\bigotimes_{i \in [m]} \mathcal{H}_{a_i} = \mathcal{H}_{a_0} \otimes \mathcal{H}_{a_2} \otimes \cdots \otimes \mathcal{H}_{a_{m-1}} \quad (1.8)$$

where  $\mathcal{H}_{a_i}$  is the Hilbert space of wavefunctions for  $i$ th particle. A multipartite quantum system may not only be in a *product state*  $|\psi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \otimes \cdots \otimes |\psi_{m-1}\rangle$ , where  $|\psi_i\rangle \in \mathcal{H}_{a_i}$ , but also a superposition of product states

$$|\psi\rangle = \sum_{j_0, \dots, j_{n-1}=0} c_{j_0, \dots, j_{m-1}} \left( |\phi_{j_0}^{a_0}\rangle \otimes |\phi_{j_1}^{a_1}\rangle \otimes \cdots \otimes |\phi_{j_{m-1}}^{a_{m-1}}\rangle \right), \quad (1.9)$$

where  $B_i = \{|\phi_{j_i}^{a_i}\rangle\}_j$  are orthonormal bases for  $\mathcal{H}_{a_i}$ . If there is no choice of bases  $B_i$  for which  $|\psi\rangle$  is a product state, then the system is *entangled*: a physical signifier of entanglement is correlation between the measurement outcomes of different particles.

**Definition 1.1.1.** (*Operators on Hilbert spaces.*) Let  $\mathcal{H}$  be a Hilbert space. The expression  $B(\mathcal{H})$  denotes the set of bounded linear operators on  $\mathcal{H}$ , and the expression  $B_h(\mathcal{H})$  denotes the set of bounded linear *Hermitian* operators on  $\mathcal{H}$ .

### 1.1.1 Fundamental particles

Fundamental particles are indistinguishable from each other. A system consisting of two particles of the same species has state space

$$|\Psi\rangle \in \mathcal{H} \otimes \mathcal{H} =: \mathcal{H}^{\otimes 2}, \quad (1.10)$$

where  $\mathcal{H}$  is the Hilbert space of single-particle wavefunctions of that species. If  $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{H}$  and  $\hat{P}$  represents the operation that swaps the states of the two particles, then

$$\hat{P} |\psi_1\rangle \otimes |\psi_2\rangle = |\psi_2\rangle \otimes |\psi_1\rangle. \quad (1.11)$$

If the particles are to be fundamental, the indistinguishability requirement is that

$$\hat{P} |\psi_1\rangle \otimes |\psi_2\rangle = e^{i\delta} |\psi_1\rangle \otimes |\psi_2\rangle. \quad (1.12)$$

Reconciling Equations 1.11 and 1.12,

$$\hat{P}^2 |\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = e^{2i\delta} |\psi_1\rangle \otimes |\psi_2\rangle \implies e^{i\delta} = \pm 1, \quad (1.13)$$

splits the fundamental particles into two categories. The solution  $e^{i\delta} = 1$  yields *bosons*, fundamental particles for which the multipartite wavefunctions are symmetric under the particle-swapping operation. For example, if  $\mathcal{H}$  is a Hilbert space with orthonormal basis  $\{|\phi_i\rangle\}_{i \in \mathcal{R}}$ , an orthonormal basis for all symmetric two-particle wavefunctions in  $\mathcal{H}^{\otimes 2}$  is

$$\left\{ |\psi_{(i)}\rangle_b \mid i \in \mathcal{R} \right\} \cup \left\{ |\psi_{(i_0, i_1)}\rangle_b \mid i_0 \neq i_1 \in \mathcal{R} \right\}, \quad (1.14)$$

where

$$|\psi_{(i)}\rangle_b := |\phi_i\rangle \otimes |\phi_i\rangle, \quad |\psi_{(i_0, i_1)}\rangle_b := \frac{1}{\sqrt{2}} (|\phi_{i_0}\rangle \otimes |\phi_{i_1}\rangle + |\phi_{i_1}\rangle \otimes |\phi_{i_0}\rangle). \quad (1.15)$$

For example, the state vectors  $|00\rangle$  and  $|11\rangle$  of a two-boson system represent symmetric wavefunctions. Any linear combination of symmetric states can constitute the wavefunction of two bosons from the same species.

The other species of fundamental particle are the *fermions*, with  $e^{i\delta} = -1$  and wavefunctions that are antisymmetric under  $\hat{P}$ . An orthonormal basis for the antisymmetric

wavefunctions in  $\mathcal{H}^{\otimes 2}$  is

$$\left\{ |\psi_{(i_0, i_1)}\rangle_f := \frac{1}{\sqrt{2}} (|\phi_{i_0}\rangle \otimes |\phi_{i_1}\rangle - |\phi_{i_1}\rangle \otimes |\phi_{i_0}\rangle) \mid i_0 \neq i_1 \in \mathcal{R} \right\}. \quad (1.16)$$

For example, the state vector  $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$  can represent the antisymmetric wavefunction of a two-fermion system.

The Pauli Exclusion Principle is the observation that no two indistinguishable fermions may have the same state vector, which is a consequence of the antisymmetry of fermionic wavefunctions under particle exchange [4]. Slater determinants give the general form of antisymmetric  $m$ -particle wavefunctions,

$$|\psi_{(i_0, \dots, i_{m-1})}\rangle_f = \frac{1}{m!} \det \begin{pmatrix} |\phi_{i_0}\rangle & |\phi_{i_1}\rangle & \dots & |\phi_{i_{m-1}}\rangle \\ |\phi_{i_0}\rangle & |\phi_{i_1}\rangle & \dots & |\phi_{i_{m-1}}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ |\phi_{i_0}\rangle & |\phi_{i_1}\rangle & \dots & |\phi_{i_{m-1}}\rangle \end{pmatrix} \in \mathcal{H}^{\otimes m}. \quad (1.17)$$

The Pauli Exclusion Principle is apparent in the  $m$ -particle wavefunction  $|\psi_{(i_0, \dots, i_{m-1})}\rangle_f$ , which vanishes if any of the indices  $i_0, \dots, i_{m-1}$  are equal. The Hilbert space of antisymmetric  $m$ -particle wavefunctions is therefore

$$\mathcal{A}(\mathcal{H}^{\otimes m}) := \text{span}_{\mathbb{C}} \left\{ |\psi_{(i_0, \dots, i_{m-1})}\rangle \mid i_0 \neq \dots \neq i_{m-1} \in \mathcal{R} \right\} \quad (1.18)$$

$$= \text{span}_{\mathbb{C}} \left\{ |\psi_{(i_0, \dots, i_{m-1})}\rangle \mid i_0 < \dots < i_{m-1} \in \mathcal{R} \right\} \subset \mathcal{H}^{\otimes m}. \quad (1.19)$$

### 1.1.2 Fermionic operators and modes

From now on, we will only consider finite-dimensional observables of fermions, which provide an  $n$ -dimensional Hilbert space  $\mathcal{H}$  with an orthonormal basis  $\{|\phi_i\rangle\}_{i=0}^{n-1}$ . Due to the Pauli Exclusion Principle, the number  $m$  of fermions present in the system can be anywhere between 0 and  $n$ , but no two fermions may share the same eigenstate.

Considering quantum systems without fixing the number of fermions shifts the focus from the particles to their eigenstates. Fermionic eigenstates go by the name of *modes*: the Hilbert space of an  $n$ -mode fermionic system  $\mathcal{H}_f^n$  is the direct sum of Hilbert spaces of antisymmetric  $m$ -particle wavefunctions,

$$\mathcal{H}_f^n := \bigoplus_{m=0}^n \mathcal{A}(\mathcal{H}^{\otimes m}) \quad (1.20)$$

$$= \text{span}_{\mathbb{C}} \left\{ |\Omega_{\text{vac}}\rangle, |\psi_{(i_0)}\rangle_f, \dots, |\psi_{(i_0, \dots, i_{n-1})}\rangle_f \mid i_0 < \dots < i_{n-1} \in [n] \right\}. \quad (1.21)$$

where  $|\Omega_{\text{vac}}\rangle$  is the fermionic vacuum wavefunction for the case  $m = 0$ . The *Fock basis* for  $\mathcal{H}_f^n$  is the set  $\mathfrak{F}_n$  of all antisymmetric wavefunctions with fixed particle number,

$$\mathfrak{F}_n = \left\{ |\Omega_{\text{vac}}\rangle, |\psi_{(i_0)}\rangle_f, |\psi_{(i_0, i_1)}\rangle_f, \dots, |\psi_{(i_0, \dots, i_{n-1})}\rangle_f \mid i_0 < \dots < i_{n-1} \in [n] \right\}, \quad (1.22)$$

and consists of  $2^n$  distinct elements:

$$|\mathfrak{F}_n| = 1 + n + \binom{n}{2} + \binom{n}{3} + \dots + \binom{n}{n} = 2^n. \quad (1.23)$$

The Hilbert space  $\mathcal{H}_f^n$ , which is the state space of an  $n$ -mode fermionic system, is therefore  $(2^n)$ -dimensional. Fermionic kets simplify the notation: for each binary vector  $\mathbf{f} = (f_0, f_1, \dots, f_{n-1}) \in \mathbb{Z}_2^n$ , define an element of the Fock basis by

$$|\mathbf{f}\rangle := \begin{cases} |\Omega_{\text{vac}}\rangle & \mathbf{f} = (0, 0, \dots, 0, 0, 0, \dots, 0) =: \mathbf{0} \\ |\psi_{(i)}\rangle_f & \mathbf{f} = (0, 0, \dots, 0, \underset{i\text{th place}}{1}, 0, \dots, 0) =: \mathbf{1}_i, \\ |\psi_{(i_0, i_1)}\rangle_f & \mathbf{f} = \mathbf{1}_{i_0} + \mathbf{1}_{i_1} \quad \text{where } i_0 < i_1 \\ \vdots \\ |\psi_{(i_0, \dots, i_{n-1})}\rangle_f & \mathbf{f} = (1, 1, \dots, 1, 1, 1, \dots, 1) \quad \text{where } i_0 < \dots < i_{n-1}. \end{cases} \quad (1.24)$$

For example,

$$|0\rangle = |\Omega_{\text{vac}}\rangle \quad (1.25)$$

$$|1\rangle = |\psi_{(0)}\rangle_f, \quad |01\rangle = |\psi_{(1)}\rangle_f, \quad |001\rangle = |\psi_{(2)}\rangle_f, \quad \dots \quad (1.26)$$

$$|11\rangle = |\psi_{(0,1)}\rangle_f, \quad |011\rangle = |\psi_{(1,2)}\rangle_f, \quad |101\rangle = |\psi_{(0,2)}\rangle_f, \quad \dots \quad (1.27)$$

The *fermionic creation operator*  $\hat{a}_j^\dagger \in B(\mathcal{H}_f^n)$  denotes the creation of a fermion in the mode  $|\phi_j\rangle$ , and its Hermitian conjugate  $\hat{a}_j$  denotes the *annihilation operator* which removes a fermion in mode  $|\phi_i\rangle$  from the system. That is, if  $i_0 < \dots < i_{n-1}$  and

$\mathbf{f} = (f_0, \dots, f_{n-1}) = \mathbf{1}_{i_0} + \dots + \mathbf{1}_{i_{m-1}}$ , then

$$\hat{a}_j^\dagger |\psi_{(i_0, \dots, i_{m-1})}\rangle_f = \hat{a}_j^\dagger |\mathbf{f}\rangle = \begin{cases} |\psi_{(j, i_0, \dots, i_{m-1})}\rangle_f = (-1)^{\sum_{k < j} f_k} |\mathbf{f} + \mathbf{1}_j\rangle & j \notin \{i_0, \dots, i_{m-1}\} \\ 0 & \text{else,} \end{cases} \quad (1.28)$$

$$\hat{a}_j |\psi_{(i_0, \dots, i_{m-1})}\rangle_f = \hat{a}_j |\mathbf{f}\rangle = \begin{cases} 0 & j \notin \{i_0, \dots, i_{m-1}\} \\ |\psi_{(i_0, \dots, \ast, i_{m-1})}\rangle_f = (-1)^{\sum_{k < j} f_k} |\mathbf{f} + \mathbf{1}_j\rangle & \text{else,} \end{cases} \quad (1.29)$$

where the asterisk in the sequence  $(i_0, \dots, i_{m-1})$  denotes the removal of the entry  $j$ .

Completeness of the Fock basis and the antisymmetry of the wavefunctions implies the Canonical Anticommutation Relations (CARs):

$$\{\hat{a}_i, \hat{a}_j^\dagger\} = \delta_{ij} \hat{1}, \quad \{\hat{a}_i^{(\dagger)}, \hat{a}_j^{(\dagger)}\} = 0, \quad (1.30)$$

where  $\{\hat{A}, \hat{B}\} = \hat{A}\hat{B} + \hat{B}\hat{A}$  denotes the anticommutation of two operators. The formula for the Fock state  $|\mathbf{f}\rangle$  in terms of creation operators and the vacuum state is

$$(\hat{a}_{i_0}^\dagger)^{f_0} (\hat{a}_{i_1}^\dagger)^{f_1} \dots (\hat{a}_{i_{n-1}}^\dagger)^{f_{n-1}} |\mathbf{0}\rangle = |\mathbf{f}\rangle. \quad (1.31)$$

Each Fock state  $|\mathbf{f}\rangle$  is an  $(f_i)$ -eigenstate of the Hermitian *number operator*  $\hat{a}_i^\dagger \hat{a}_i \in B_h(\mathcal{H}_f^n)$ , which counts the number of fermions in mode  $|\phi_i\rangle$ .

Beginning with an ordered sequence  $(a_i^{(\dagger)})_{i=0}^{n-1} \in B(\mathcal{H}_f^n)$  of  $2n$  operators obeying the CARs in Equation 1.30, it is possible to reconstruct the entire Fock basis without reference to any particular fermionic system. Products of creation and annihilation operators generate the algebra of all bounded linear fermionic operators:

$$\langle \hat{a}_0^{(\dagger)}, \hat{a}_1^{(\dagger)}, \dots, \hat{a}_{n-1}^{(\dagger)} \rangle := \text{span}_{\mathbb{C}} \left\{ \prod_{j=0}^k \hat{a}_{i_j} \prod_{j=0}^{k'} \hat{a}_{i_j}^\dagger \mid i_j, k, k' \in [n] \right\} = B(\mathcal{H}_f^n). \quad (1.32)$$

Definition 1.1.2 contains a summary of the core information from this section.

**Definition 1.1.2.** (*Vacuum state and Fock basis of fermionic state space.*) Given an  $n$ -mode fermionic system with Hilbert space  $\mathcal{H}_f^n$  and a sequence of annihilation operators  $(\hat{a}_i)_{i=0}^{n-1}$ , choose  $|\mathbf{0}\rangle \in \mathcal{H}_f^n$  from the set of simultaneous 0-eigenstates of the number operators  $(\hat{a}_i^\dagger \hat{a}_i)_{i=0}^{n-1}$  as the *vacuum state* of the fermionic system. Any two simultaneous 0-eigenstates are proportional and differ by only a phase  $e^{i\phi}$  for some  $\phi \in [0, 2\pi]$ . The  $n$ -mode Fock basis is the orthonormal basis  $\mathfrak{F}_n = \{|\mathbf{f}\rangle \mid \mathbf{f} \in \mathbb{Z}_2^n\} \subset \mathcal{H}_f^n$ ,

where

$$|\mathbf{f}\rangle = |f_0, f_1, \dots, f_{n-2}, f_{n-1}\rangle := (\hat{a}_0^\dagger)^{f_0} (\hat{a}_1^\dagger)^{f_1} \dots (\hat{a}_{n-1}^\dagger)^{f_{n-1}} |\mathbf{0}\rangle. \quad (1.33)$$

The physical significance of the Fock basis is that each state  $|\mathbf{f}\rangle \in \mathcal{H}_f^n$  has well-defined occupancy: if the system is in state  $|\mathbf{f}\rangle$ , then the mode with label  $i$  is occupied by a fermion if and only if  $f_i = 1$ .

### 1.1.3 Majorana operators

This section introduces Majorana operators, which are integral to any mathematical description of fermion–qubit mappings. Let  $(\hat{a}_i)_{i=0}^{n-1}$  be the sequence of annihilation operators in an  $n$ –mode fermionic system with state space  $\mathcal{H}_f^n$ .

**Definition 1.1.3.** (*Majorana operators.*) The *Majorana operators* of an  $n$ –mode fermionic system make up a sequence of  $n$  pairs of fermionic operators  $((\hat{\gamma}_{2i}, \hat{\gamma}_{2i+1}))_{i=0}^{n-1}$  with definitions

$$\hat{\gamma}_{2i} = \hat{a}_i + \hat{a}_i^\dagger, \quad \hat{\gamma}_{2i+1} = i(\hat{a}_i - \hat{a}_i^\dagger). \quad (1.34)$$

Products of Majorana operators generate the entire fermionic algebra,

$$\langle \hat{\gamma}_0, \hat{\gamma}_1, \dots, \hat{\gamma}_{2n-1} \rangle := \text{span}_{\mathbb{C}} \left\{ \prod_{j=0}^k \hat{\gamma}_{i_j} \mid i_j, k \in [2n] \right\} = B(\mathcal{H}_f^n), \quad (1.35)$$

and the CARs of Equation 1.30 re-express themselves in the form

$$\{\hat{\gamma}_i, \hat{\gamma}_j\} = 2\delta_{ij}\hat{1}, \quad \hat{\gamma}_i^\dagger = \hat{\gamma}_i \quad \text{for all } i, j \in [2n]. \quad (1.36)$$

Lemma 1.1.4 rephrases the Fock states of Equation 1.33 in terms of the Majorana operators.

**Lemma 1.1.4.** (*Defining the Fock basis in terms of Majorana operators.*) Let  $\mathcal{H}_f^n$  be the Hilbert space of an  $n$ –mode fermionic system with annihilation operators  $(\hat{a}_i)_{i=0}^{2n-1}$  and vacuum state  $|\mathbf{0}\rangle$ . Then, for each  $\mathbf{f} \in \mathbb{Z}_2^n$ , the Fock basis vector  $|\mathbf{f}\rangle \in \mathfrak{F}_n$  is

$$|\mathbf{f}\rangle = (\hat{\gamma}_0)^{f_0} (\hat{\gamma}_2)^{f_1} \dots (\hat{\gamma}_{2n-2})^{f_{n-1}} |\mathbf{0}\rangle = (-i\hat{\gamma}_1)^{f_0} (-i\hat{\gamma}_3)^{f_1} \dots (-i\hat{\gamma}_{2n-1})^{f_{n-1}} |\mathbf{0}\rangle. \quad (1.37)$$

*Proof.* The inverse of Equation 1.34 is  $\hat{a}_i^{(\dagger)} = \frac{1}{2}(\hat{\gamma}_{2i} \pm i\hat{\gamma}_{2i+1})$ , and so  $\hat{a}_i^\dagger \hat{a}_i = \frac{1}{2}(\hat{1} + i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1})$ . Therefore  $|\mathbf{0}\rangle$  is a simultaneous  $(+1)$ –eigenstate of  $-i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1}$  for all  $i \in [n]$ .

Combine this with the fact that  $(\hat{\gamma}_{2i})^2 = (\hat{\gamma}_{2i+1})^2 = \hat{1}$  to obtain

$$\hat{a}_i^\dagger |\mathbf{0}\rangle = \frac{1}{2}(\hat{\gamma}_{2i} - i\hat{\gamma}_{2i+1})|\mathbf{0}\rangle = \frac{1}{2}\hat{\gamma}_{2i}(\hat{1} - i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1})|\mathbf{0}\rangle = \hat{\gamma}_{2i}|\mathbf{0}\rangle \quad (1.38)$$

$$= \frac{1}{2}(-i\hat{\gamma}_{2i+1})(-i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1} + \hat{1})|\mathbf{0}\rangle = -i\hat{\gamma}_{2i+1}|\mathbf{0}\rangle, \quad (1.39)$$

for all  $i \in [n]$ .

Let  $\mathbf{f} \in \mathbb{Z}_2^n$  have  $k$  nonzero bits  $f_{i_1}, f_{i_2}, \dots, f_{i_k} = 1$  where  $0 \leq i_1 < \dots < i_k \leq n-1$  for some  $k \in \{1, \dots, n\}$ , with all other bits zero. The Fock basis state  $|\mathbf{f}\rangle$  has the form

$$|\mathbf{f}\rangle = \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{k-1}}^\dagger \hat{a}_{i_k}^\dagger |\mathbf{0}\rangle = \left( \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{k-1}}^\dagger \right) \hat{\gamma}_{2i_k} |\mathbf{0}\rangle, \quad (1.40)$$

using Equation 1.38. Suppose that there is some  $l < k$  for which

$$|\mathbf{f}\rangle = \left( \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_l}^\dagger \right) \hat{\gamma}_{2i_{l+1}} \hat{\gamma}_{2i_{l+2}} \dots \hat{\gamma}_{2i_{k-1}} \hat{\gamma}_{2i_k} |\mathbf{0}\rangle. \quad (1.41)$$

Equation 1.40 shows that such an  $l$  exists, with value  $l = k-1$ . Since  $\hat{a}_{i_l}^\dagger$  is a linear combination of the Majorana operators  $\hat{\gamma}_{2i_l}$  and  $\hat{\gamma}_{2i_{l+1}}$ , it anticommutes with any  $\hat{\gamma}_{2j}$  where  $j \neq i_l$ , and so

$$|\mathbf{f}\rangle = (-1)^{k-(l+1)} \left( \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{l-1}}^\dagger \right) \hat{\gamma}_{2i_{l+1}} \hat{\gamma}_{2i_{l+2}} \dots \hat{\gamma}_{2i_{k-1}} \hat{\gamma}_{2i_k} \left( \hat{a}_{i_l}^\dagger \right) |\mathbf{0}\rangle \quad (1.42)$$

$$= (-1)^{k-(l+1)} \left( \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{l-1}}^\dagger \right) \hat{\gamma}_{2i_{l+1}} \hat{\gamma}_{2i_{l+2}} \dots \hat{\gamma}_{2i_{k-1}} \hat{\gamma}_{2i_k} (\hat{\gamma}_{2i_l}) |\mathbf{0}\rangle \quad (1.43)$$

$$= \left( \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{l-1}}^\dagger \right) (\hat{\gamma}_{2i_l}) \hat{\gamma}_{2i_{l+1}} \hat{\gamma}_{2i_{l+2}} \dots \hat{\gamma}_{2i_{k-1}} \hat{\gamma}_{2i_k} |\mathbf{0}\rangle, \quad (1.44)$$

again using Equation 1.38 and anticommuting  $\hat{\gamma}_{2i_l}$  back to the original position of  $\hat{a}_{i_l}^\dagger$ , which cancels out the phase of  $(-1)^{k-(l+1)}$ . Relabel  $l \leftarrow (l-1)$  and repeat steps 1.42–1.44, proceeding via induction until  $l = 0$ . At the end, we obtain the result that

$$|\mathbf{f}\rangle = \hat{\gamma}_{2i_1} \hat{\gamma}_{2i_2} \dots \hat{\gamma}_{2i_k} |\mathbf{0}\rangle, \quad (1.45)$$

which proves the first equality of Equation 1.37. If we had used Equation 1.39 rather than 1.38, we would have obtained

$$|\mathbf{f}\rangle = \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{k-1}}^\dagger \hat{a}_{i_k}^\dagger |\mathbf{0}\rangle = \left( \hat{a}_{i_1}^\dagger \hat{a}_{i_2}^\dagger \dots \hat{a}_{i_{k-1}}^\dagger \right) (-i\hat{\gamma}_{2i_{k+1}}) |\mathbf{0}\rangle, \quad (1.46)$$

instead of Equation 1.40. Following identical steps gives

$$|\mathbf{f}\rangle = (-i\hat{\gamma}_{2i_1+1})(-i\hat{\gamma}_{2i_2+1}) \dots (-i\hat{\gamma}_{2i_{k+1}+1}) |\mathbf{0}\rangle, \quad (1.47)$$

which proves the second equality of Equation 1.37.  $\square$

### 1.1.4 Fermionic Hamiltonians

The spatial and inertial coordinates of the particles in a multipartite quantum system determine its energy. The *Hamiltonian* of a physical system is a Hermitian operator that evaluates the total kinetic and potential energy of each wavefunction,

$$\hat{H}\psi(\mathbf{r}, t) = \left(-\frac{\hbar^2}{2m}\nabla^2 + \hat{V}\right)\psi(\mathbf{r}, t), \quad (1.48)$$

where  $\hat{V}(\psi(\mathbf{r}, t)) = V(\mathbf{r})\psi(\mathbf{r}, t)$  is the operator that identifies the potential energy of the wavefunction  $\psi$ . A system's Hamiltonian is the key to understanding its behaviour and physical properties: we use Hamiltonian simulation and ground-state energy as two motivating examples.

The *energy eigenstates* are the eigenfunctions of the Hamiltonian, and a broad range of quantum sciences [5–8] ultimately concern properties of the energy eigenstates or simulating time evolution itself via the time-dependent Schrödinger Equation [9],

$$i\hbar\frac{d}{dt}|\psi\rangle = \hat{H}|\psi\rangle \quad \text{for} \quad |\psi\rangle \in \mathcal{H}_{\text{system}}, \quad (1.49)$$

where  $\mathcal{H}_{\text{system}}$  is the Hilbert space of the quantum system. Equation 1.49 arises from reconciling classical physics with non-relativistic quantum phenomena.

Fermionic dynamics encapsulate a range of behaviours in many-body systems, including molecular electronic structure [10–13], condensed matter physics [14–19] and quantum field theory [20–23]. Making the Born–Oppenheimer approximation [24] that nuclear charges are static, the Hamiltonian of a molecule with  $m$  electrons and  $N$  nuclei is

$$\hat{H} = -\sum_{i \in [m]} \frac{\nabla_{\mathbf{r}_i}^2}{2} - \sum_{\substack{i \in [N], \\ j \in [m]}} \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}_j|} + \sum_{\substack{i, j \in [m], \\ i < j}} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (1.50)$$

where  $\mathbf{R}_i$  and  $\mathbf{r}_i$  are the coordinates of the nuclei and electrons, respectively, and  $Z_i$  are the nuclear charges [25, 26]. Choose a set of  $n \geq m$  single-electron eigenfunctions of some observables, such as the spin-orbitals of the individual atoms in the molecule. The *second-quantised electronic structure Hamiltonian* exists with respect to a basis of

products of single-particle antisymmetric wavefunctions, which reads

$$\hat{H}_{\text{fermion}} = \sum_{i,j \in [n]} (c_{ij}) \hat{a}_i^\dagger \hat{a}_j + \sum_{i,j,k,l \in [n]} (c_{kl}^{ij}) \hat{a}_i^\dagger \hat{a}_j \hat{a}_k^\dagger \hat{a}_l . \quad (1.51)$$

The coefficients  $c_{ij}$  and  $c_{kl}^{ij}$  are the density functionals

$$c_{ij} = \int d^3r \sum_{\chi=0}^1 \varphi_i^*(\mathbf{r}, \chi) \left( -\frac{\nabla_r^2}{2} - \sum_{i=0}^{M-1} \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}|} \right) \varphi_j(\mathbf{r}, \chi) , \quad (1.52)$$

$$c_{kl}^{ij} = \int d^3r d^3r' \sum_{\chi, \chi'=0}^1 \frac{\varphi_i^*(\mathbf{r}, \chi) \varphi_j^*(\mathbf{r}', \chi') \varphi_k(\mathbf{r}, \chi) \varphi_l(\mathbf{r}', \chi')}{|\mathbf{r} - \mathbf{r}'|} . \quad (1.53)$$

Evaluation of the coefficients is the concern of density functional theory [27, 28].

Second-quantised fermionic Hamiltonians throughout physics share the same general structure as  $\hat{H}_{\text{fermion}}$  in Equation 1.51. For example, the Fermi–Hubbard model in condensed matter physics considers lattices of fermionic modes with nearest-neighbour interactions between particles [29–31], using graphs to describe Hamiltonians of the form

$$\hat{H}_{\text{fermion}} = \sum_{i \in [n]} (c_i) \hat{a}_i^\dagger \hat{a}_i + \sum_{(i,j) \in G} \left( (c_{ij}) (\hat{a}_i^\dagger \hat{a}_j + \hat{a}_j^\dagger \hat{a}_i) + (\tilde{c}_{jj}^{ii}) \hat{a}_i^\dagger \hat{a}_i \hat{a}_j^\dagger \hat{a}_j \right) \quad (1.54)$$

for some  $n$ -vertex graph  $G$ .

Two computational tasks are primary motivators for the study of fermionic systems. *Hamiltonian simulation* [7, 8, 32, 33] is the process of approximating the solution to the Schrödinger equation,

$$|\psi(t)\rangle = e^{-i\hat{H}t/\hbar} |\psi(0)\rangle \quad (1.55)$$

where  $t \in \mathbb{R}$  and  $\hat{H}$  is a finite-dimensional Hamiltonian providing a basis for  $\mathcal{H}_{\text{system}}$ . The state  $|\psi(t)\rangle$  describes the system after evolution for a time  $t$  from the initial state  $|\psi(0)\rangle$ . The *ground-state energy* is the lowest eigenvalue of  $\hat{H}$ ,

$$\lambda_0 = \min_{|\psi\rangle \in \mathcal{H}} \langle \psi | \hat{H} | \psi \rangle , \quad (1.56)$$

which describes the energy of the system in equilibrium. As the gateway to exploring the rest of the spectrum of  $\hat{H}$ , finding the ground-state energy underpins many of the computational tasks of quantum science [34–36].

In fermionic systems, the dimension of the Hilbert space is exponential in the number of modes, which are single-particle basis functions in the electronic structure case, or lattice sites in the Fermi–Hubbard case. The matrix representation of the Hamiltonian thus has exponentially many entries. Existing methods to approximate the time–evolution operator  $e^{-i\hat{H}t}$  through matrix exponentiation requires access to these entries [37], and so all known classical Hamiltonian simulation algorithms have runtimes that would scale exponentially in the system size. Finding the ground–state energy of a Hamiltonian  $\hat{H}$  is similarly beyond the grasp of classical computers – and may be even more difficult than Hamiltonian simulation, because it is difficult even for quantum computers [35].

## 1.2 Quantum computing

A *qubit* is a quantum system with a two-dimensional Hilbert space  $\mathcal{H}_2 = \{a|0\rangle + b|1\rangle \mid a, b \in \mathbb{C}\}$ , which is isomorphic to  $\mathbb{C}^2$ . The orthonormal basis  $\mathfrak{C}_1 = \{|0\rangle, |1\rangle\}$  for  $\mathcal{H}_2$  consists of the  $(\pm 1)$ -eigenvectors of a Hermitian operator  $Z$ , which relates to some physical observable of the system such as the  $z$ -component of an electron’s spin or the polarisation of a photon.

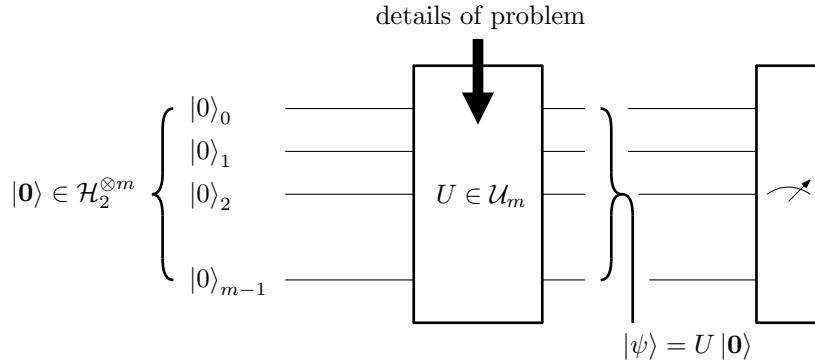
Quantum computers are an emergent technology that can process information by operating on collections of qubits. A quantum register of  $m$  qubits has the  $(2^m)$ -dimensional state space

$$\mathcal{H}_2^{\otimes m} = \text{span}_{\mathbb{C}} \left( \bigotimes_{i=0}^{m-1} \{|0\rangle_i, |1\rangle_i\} \right) = \text{span}_{\mathbb{C}} \{|\mathbf{q}\rangle \mid \mathbf{q} \in \mathbb{Z}_2^m\}, \quad (1.57)$$

where the expression  $|\psi\rangle_i$  indicates that the  $i$ th qubit is in the state  $|\psi\rangle \in \mathcal{H}_2$ , and where  $|\mathbf{q}\rangle$  denotes the product state  $|q_0\rangle_0 \otimes |q_1\rangle_1 \otimes \cdots \otimes |q_{m-1}\rangle_{m-1}$  for  $q = (q_0, q_1, \dots, q_{m-1}) \in \mathbb{Z}_2^m$ . The *computational basis* for  $\mathcal{H}_2^{\otimes m}$  is the set

$$\mathfrak{C}_m = \{|\mathbf{q}\rangle \mid \mathbf{q} \in \mathbb{Z}_2^m\}. \quad (1.58)$$

In the circuit model of quantum computing, a quantum algorithm involves the action of unitary operations, which are the linear operators that preserve wavefunctions, followed by irreversible measurements on a register of qubits. Figure 1.1 gives the general form of a *quantum circuit*, a visual diagram that represents a quantum algorithm [38]. By careful choice of the unitary operator, the measurement outcomes of the quantum register can reveal information about the problem of interest.



**Figure 1.1** The general structure of a quantum algorithm with  $m$  qubits in the circuit model consists of the initial state  $|0\rangle \in \mathcal{H}_2^{\otimes m}$  followed by a unitary operation  $U \in \mathcal{U}_m$ , producing the state  $|\psi\rangle = U|0\rangle$ . Measurements of the quantum state can reveal information in the phases and amplitudes of the terms of  $|\psi\rangle$ .

**Definition 1.2.1.** (*Notation for qubit operators.*)

- In this thesis, we distinguish fermionic and qubit operators by writing fermionic operators  $\hat{O}$  with hats and qubit operators  $O$  without hats. This stylistic choice holds throughout the entire text.
- The expression  $\mathcal{U}_n$  denotes the  $2^n \times 2^n$  unitary matrices.
- The expression  $\mathcal{H}_2$  denotes the state space of a single two-level system. The space of  $m$  qubits is  $\mathcal{H}_2^{\otimes m}$ .

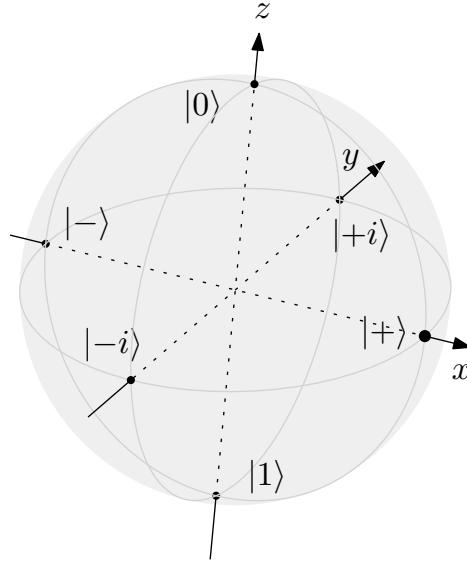
### 1.2.1 Pauli operators

In terms of the orthonormal basis  $\mathfrak{C}_1$  for the single-qubit Hilbert space  $\mathcal{H}_2$ , the Hermitian operator  $Z \in B_h(\mathcal{H}_2)$  is

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (1.59)$$

which is also a unitary matrix. The space of normalised single-qubit wavefunctions is a three-dimensional manifold in  $\mathcal{H}_2$ ,

$$\begin{aligned} & \left\{ a|0\rangle + b|1\rangle \mid a, b \in \mathbb{C} : |a|^2 + |b|^2 = 1 \right\} \\ &= \left\{ \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \mid \theta \in [0, 2\pi), \phi \in [0, \pi) \right\}, \end{aligned} \quad (1.60)$$



**Figure 1.2** The Bloch sphere representation of the single-qubit state space in  $\mathcal{H}_2$ . The eigenstates of the Pauli  $X$ ,  $Y$  and  $Z$  matrices mark unit vectors along the three cardinal directions.

which Figure 1.2 represents as the Bloch sphere. In this picture, the eigenstates  $|0\rangle$  and  $|1\rangle$  of  $Z$  are the unit vector markers along the  $z$ -axis. The qubit states

$$|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}, \quad |\pm i\rangle = \frac{|0\rangle \pm i|1\rangle}{\sqrt{2}}, \quad (1.61)$$

are, respectively, the  $(\pm 1)$ -eigenstates of the Hermitian, unitary matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad (1.62)$$

which fall at the  $\pm 1$  markers along the  $y$  and  $z$  axes on the Bloch sphere. Collectively, the set  $\{X, Y, Z\}$  contains the unique matrices in  $B(\mathcal{H}_2)$  to satisfy the anticommutation and Hermiticity relations

$$\{A, B\} = 2\delta_{A,B}\mathbb{1}, \quad A^\dagger = A, \quad \text{for all } A, B \in \{X, Y, Z\} \quad (1.63)$$

up to some global orientation of the computational basis on  $\mathcal{H}_2$ , where  $\mathbb{1}$  is the  $2 \times 2$  identity matrix.

The *single-qubit Pauli group* has the definition

$$\mathcal{P}_1 = \langle X, Y, Z \rangle = \{\pm\mathbb{1}, \pm i\mathbb{1}, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}, \quad (1.64)$$

the elements of which are *single-qubit Pauli operators*. The  $m$ -qubit *Pauli group* is the generalisation of  $\mathcal{P}_1$  to  $m$ -qubit Pauli operators:

$$\mathcal{P}_m = \langle X, Y, Z \rangle^{\otimes n} = \{ \gamma P_0 \otimes \cdots \otimes P_{m-1} \mid P_i \in \{1, X, Y, Z\}, \gamma \in \{\pm 1, \pm i\} \} . \quad (1.65)$$

An  $m$ -qubit Pauli operator  $P \in \mathcal{P}_m$  has eigenvalues  $\pm 1$ , each with degeneracy  $2^{m-1}$ . The  $(+1)$ -eigenvectors of  $P$  are the *stabiliser states* of  $P$ . The operator  $P$  commutes with exactly half of the elements of  $\mathcal{P}_m$  and anticommutes with the other half: a consequence is that the *stabiliser space* of a set of  $k$  commuting  $m$ -qubit Pauli operators  $\{P_1, P_2, \dots, P_k\}$ , which is the shared  $(+1)$ -eigenspace of the  $k$  operators, is  $(2^{m-k})$ -dimensional for  $1 \leq k \leq m$ .

Omitting the tensor product symbol, we will frequently use an abbreviation for the  $m$ -qubit Pauli operators  $P$  of the form

$$P = P_0 \otimes P_1 \otimes \cdots \otimes P_{m-1} =: (P_0)_0 (P_1)_1 \dots (P_{m-1})_{m-1} , \quad (1.66)$$

where  $(P_i)_j$  denotes the action of a single-qubit Pauli operator  $P_i \in \mathcal{H}_2$  acting on the qubit with label  $j$ .

### 1.2.2 Clifford operators

The  $m$ -qubit Clifford group  $\mathcal{C}_m$  is the set of  $m$ -qubit unitary operators that normalise the  $m$ -qubit Pauli group:

$$\mathcal{C}_m = \left\{ C \in \mathcal{U}_m \mid CPC^\dagger \in \mathcal{P}_m : P \in \mathcal{P}_m \right\} . \quad (1.67)$$

Define the Hadamard, phase and controlled-NOT gates as

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \in \mathcal{C}_1 , \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \in \mathcal{C}_2 ; \quad (1.68)$$

The  $m$ -qubit Clifford group is

$$\mathcal{C}_m = \left\langle \gamma \mathbb{1}^{\otimes m}, H_i, S_i, \text{CNOT}_{ij} \mid i \neq j \in [m], \gamma \in \mathbb{C} \right\rangle , \quad (1.69)$$

where  $H_i$  and  $S_i$  denote the action of  $H$  and  $S$  on the  $i$ th qubit, and  $\text{CNOT}_{ij}$  indicates the action of CNOT on the tensor product space of the  $i$ th and  $j$ th qubits. Quantum

circuits consisting solely of Clifford gates are classically simulable [39], and bring no advantage over classical computers. With the inclusion of the non–Clifford  $T$  gate,

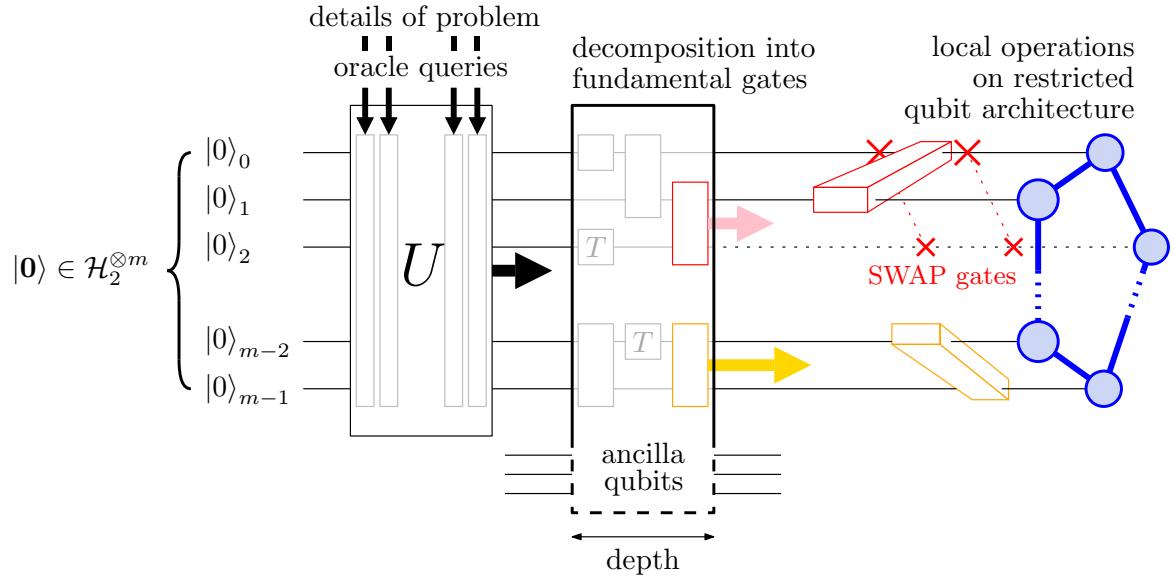
$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}, \quad (1.70)$$

the set  $\{H_i, S_i, \text{CNOT}_{ij}, T_i \mid i \neq j \in [m]\}$  forms a *universal gate set*, in that any unitary  $U \in \mathcal{U}_m$  up to a global phase has a finite-length approximation in terms of just these gates.

### 1.2.3 The quantum resource cost of a unitary operator

In quantum circuits, unitary operators transform the state of the quantum register in much the same way as a classical computer employs logic gates on arrays of bits. There are several definitions for the quantum resource cost of an  $m$ -qubit unitary  $U \in \mathcal{U}_m$ , including:

- The *query complexity* of the unitary  $U$ , which counts the number of distinct times the circuit must use external information about some function  $f : \mathbb{Z}^m \rightarrow \mathbb{Z}$  in its execution. For example, one execution of the Grover iterate in amplitude amplification requires two queries to an oracle [40, 41].
- The *number of fundamental gates* in the decomposition of the unitary  $U$  into a universal gate set: for example, how many individual gates from the set  $\{H_i, S_i, \text{CNOT}_{ij}, T_i \mid i \neq j \in [m]\}$  make up the operator  $U$  [42].
- The *number of qubits* upon which  $U$  acts in its decomposition into fundamental gates. While  $U \in \mathcal{U}_m$  can, in principle, act on only  $m$  qubits, gadgets requiring ancilla qubits may be useful in its decomposition into fundamental gates. For example, realising oracle queries as fundamental gates could require the phase kick–back technique [43].
- The *number of non-Clifford gates* in the circuit  $U$ . Non-Clifford gates, such as the  $T$  gate or Toffoli gate, boost the classically simulable Clifford operations to universality [44]. Fault–tolerant implementation of these gates is crucial to perform meaningful quantum computation, and their complexity presents a very real obstacle in quantum hardware design.



**Figure 1.3** From theoretical to laboratory-specific, there are several lenses through which to view the resource costs of a unitary operator  $U$  acting on  $m$  qubits. In the most abstract sense, the query complexity is the number of calls to oracles involved in  $U$ . On a more explicit level, the gate cost,  $T$ -count and depth of  $U$  measure complexities of its decomposition into a finite sequence of fundamental gates. Multi-qubit gates may cost additional SWAP gates to account for the limited connectivities of real-world qubit systems.

- The *depth* of the quantum circuit, which is the number of distinct layers of fundamental gates in the decomposition of  $U$  (see Figure 1.3), and is analogous to the number of distinct timesteps involved in the execution of the unitary.

One way to formalise the notion of circuit complexity is by assigning a *cost function* to a unitary operator. Suppose that the function  $C : \mathcal{U}_m \rightarrow \mathbb{R}$  denotes the cost of a quantum circuit  $U$  acting on  $m$  qubits: implementation of the unitary  $U$  is *efficient* with respect to the cost function  $C$  if  $C(U) = \text{poly}(m)$ .

In comparison to query complexity, cost functions involving the decomposition of circuits into fundamental gates are one step closer to the reality of quantum computing hardware. In the current laboratory setting, qubits have limited interconnectivity [45–49], restricting the available set of two-qubit gates to nearest-neighbours on a specific architecture. For implementation on these devices, unitary operators require decompositions with cascades of SWAP gates, as Figure 1.3 illustrates.

In this thesis we do not assume any particular qubit architecture and presume an all-to-all connectivity between qubits. However, our results yield a reduction in the complexity of unitary operators for fermionic simulation algorithms, which will be useful in any practical setting. The modifications to fermion-qubit mappings that

we suggest explicitly reduce the qubit count, depth, and total gate cost of unitary operators versus existing results in the literature. The advantages of our results thus have a clear potential to carry into the regime of restricted qubit connectivities, but we leave this for future work.

A common cost function for Pauli operators is the *Pauli weight*. Chapters 4 and 5 propose optimisations for fermion–qubit mappings that minimise the Pauli weights of the terms in qubit Hamiltonians.

**Definition 1.2.2.** (*The weight of a Pauli operator.*) Let  $P \in \mathcal{P}_n$  be an  $n$ –qubit Pauli operator. The *Pauli weight* of  $P$  is the number of qubits upon which  $P$  acts nontrivially. The function  $W : \mathcal{P}_n \rightarrow [n]$  that maps each Pauli operator  $P$  to its Pauli weight  $W(P)$  is an example of a cost function on the subset  $\mathcal{P}_n$  of the unitary matrices  $\mathcal{U}_n$ .

### 1.2.4 Quantum algorithms to simulate Hamiltonians

One of the major applications of quantum computers is studying the Hamiltonians of multipartite quantum systems [8, 50, 51]. A quantum computer can simulate the time–evolution of a quantum system under a Hamiltonian  $\hat{H} \in B_h(\mathcal{H})$  via the Schrödinger Equation 1.55 through preparing a state representing  $|\psi(0)\rangle$  and then implementing a unitary  $U \in \mathcal{U}_m$  representing  $e^{-i\hat{H}t}$ . There also exist quantum algorithms such as phase estimation [26, 52] and variational quantum eigensolvers [25, 53, 54] for estimating the ground–state energy of Hamiltonians.

It is not always straightforward to write Hamiltonians appear in nature in terms of the fundamental gate sets of quantum computers. Simulation of a qubit Hamiltonian  $H_{\text{qubit}} \in B(\mathcal{H}_2^{\otimes m})$  is the task of implementing a unitary  $U \in \mathcal{U}_m$  such that  $\|U - e^{-i\hat{H}t}\| < \epsilon$  to an arbitrarily small error  $\epsilon > 0$  with regards to some matrix norm  $\|\cdot\|$  [8]. As Section 1.1.4 mentioned, a roadblock for solving the classical problem is the exponential dimension of the Hamiltonian matrix in terms of the size of the system: using only  $m$  qubits, a quantum computer can handle  $(2^m)$ –dimensional matrices with ease. However, in the most general cases, the time evolution operator  $e^{-i\hat{H}t}$  can be any unitary operation, in which case it is unlikely that there is a unitary  $U \in \mathcal{U}_m$  that approximates  $e^{-i\hat{H}t}$  using only a polynomial number of query or gate resources [55].

Thankfully, close–range interactions dominate physical phenomena. This locality manifests in the fermionic problems of Section 1.1.4: the coefficients  $c_{ij}$  and  $c_{kl}^{ij}$  in the electronic structure Hamiltonians of Equation 1.51 are close to zero for unlikely energy transitions, and the Fermi–Hubbard model enforces locality by only including interaction terms between nearest–neighbour modes. Borrowing these real–world

notions, an *m-qubit Hamiltonian* is a  $2^m \times 2^m$  Hermitian matrix  $H_{\text{qubit}} \in B_h(\mathcal{H}_2^{\otimes m})$ , and is a *k-local Hamiltonian* [34] if there exists a decomposition into terms  $\Lambda \subseteq B(\mathcal{H}_2^{\otimes m})$ :

$$H_{\text{qubit}} = \sum_{h \in \Lambda} h, \quad (1.71)$$

where each individual term  $h \in \Lambda$  is an *m-qubit Hamiltonian* acting nontrivially on at most  $k = \mathcal{O}(1)$  qubits.

With the condition that  $|\Lambda| \in \text{poly}(m)$ , there exist numerous simulation algorithms [8, 56, 57] efficiently approximating the time evolution operator  $e^{-iH_{\text{qubit}}t}$  using a polynomial number of queries and gates in  $m$ . This places the quantum Hamiltonian simulation problem of approximating  $e^{-iH_{\text{qubit}}t}$  squarely within the complexity class BQP, which consists of tasks requiring only a polynomial number of quantum resources to solve.

In comparison, the *k-local Hamiltonian problem* is the task of estimating the lowest eigenvalue of the Hamiltonian  $H_{\text{qubit}}$ , again with the condition that  $|\Lambda| \in \text{poly}(m)$ . There are quantum algorithms solving the *k-local Hamiltonian problem* which, for several classes of  $H_{\text{qubit}}$ , use exponentially less resources than all known classical algorithms [58]. Ultimately, however, the *k-local Hamiltonian problem* remains QMA-complete in the general case [35], meaning that it is possible to verify solutions efficiently but that there is likely no efficient algorithm to solve every instance of the problem outright.

While complexity theory deals in resource costs in orders of magnitude, the exact cost breakdown into qubits, depth and number of fundamental gates will likely dictate the efficiency of near- and mid-term algorithms. In practice, access to fault-tolerant qubits with all-to-all connectivity and noiseless gates has yet to come to fruition, and polynomial-time quantum algorithms are extraordinarily difficult to implement on even modest-sized Hamiltonians. For practical purposes, there is great interest in any technique that reduces the qubit, gate and depth costs of Hamiltonian simulation tasks by constant factors, such as the novel methods we present in this thesis.

### 1.3 Fermion–qubit mappings

Fermionic Hamiltonians model nature’s fundamental forces, but simulating their dynamics and spectra lie outside the known remit of classical computing. Meanwhile, only the receding limitations of real-world quantum hardware stand in the way of algorithms for studying many physically-motivated Hamiltonians efficiently. Local

fermionic problems are an irresistible target for quantum computers, but there is a subtle distinction: fermions are not qubits, and qubits are not fermions.

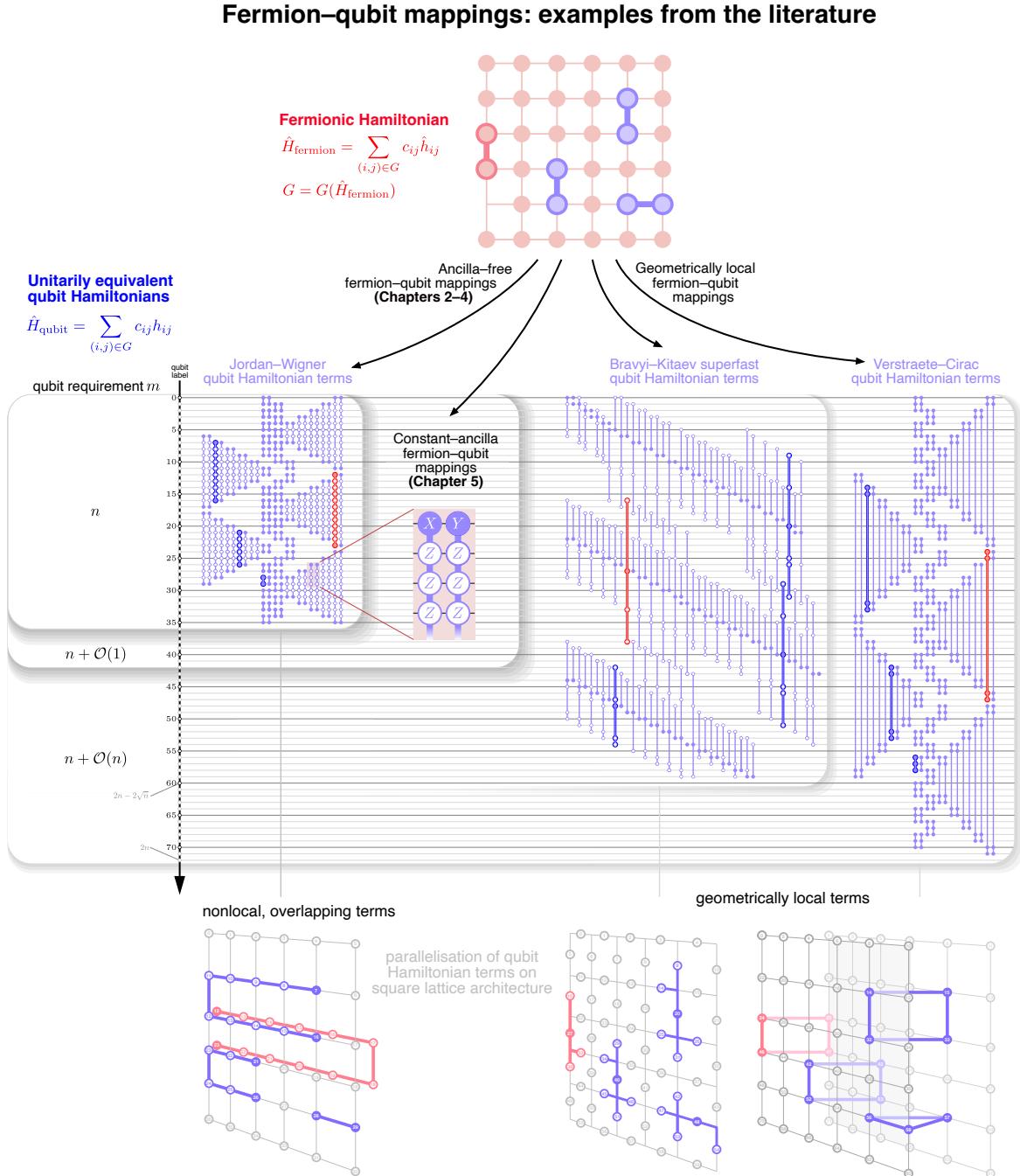
A fermionic system is not the same as a qubit system. While we may want to compute the properties of a specific fermionic Hamiltonian, with only a quantum computer at hand, it is a necessary task to construct a qubit Hamiltonian that maintains the same eigenvalues and eigenvector structure.

The term “*fermion–qubit mapping*” is a catch–all phrase to describe the solution to this task: the construction of a qubit Hamiltonian to mimic a fermionic Hamiltonian. Applying quantum simulation to the qubit Hamiltonian, as Section 1.2.4 discusses, can extract the eigenvalues or time evolution of the fermionic Hamiltonian as a result. In this thesis, details of the simulation task and intended algorithm are not of concern: our work concerns the definition of fermion–qubit mappings, how to classify them, and their optimisation. A future quantum computing scientist can pick and choose a specific fermion–qubit mapping depending on what cost model their algorithm uses.

Figure 1.4 displays the qubit Hamiltonians that arise from applying three different mappings from the literature to a  $6 \times 6$  system of interacting fermionic modes. The leftmost, the Jordan–Wigner transformation, uses only 36 qubits, but produces Hamiltonian terms that have are long, winding, and overlap with each other. The centre mapping, the superfast Bravyi–Kitaev transformation [59], uses 60 qubits: one for each edge of the fermionic interaction graph. The resulting qubit Hamiltonian terms have a constant Pauli weight and significantly less overlap; if the qubits arrange into a  $6 \times 6$  lattice, the operators preserve geometric locality. The rightmost mapping, the Verstraete–Cirac transformation [60], uses 72 qubits: two for each vertex of the fermionic interaction graph. It produces the most local qubit Hamiltonian terms of all three mappings.

### 1.3.1 Definitions, classifications and optimisations of fermion–qubit mappings in the literature

Fermion–qubit mappings have seen a flurry of activity over the past twenty years, but the methods and paradigms that have emerged share little notational consistency. From the key conceptual standpoint, simulating a fermionic Hamiltonian on a quantum computer requires finding a qubit Hamiltonian with the same eigenvalues and equivalent eigenvectors. The requirement for fermionic simulation is thus a unitarily equivalent Hamiltonian.



**Figure 1.4** Applications of different fermion–qubit mappings to a 36–mode square–lattice fermionic Hamiltonian. We explore two new directions for finding lower–cost qubit Hamiltonians: via new ancilla–free mappings (Chapters 2–4), and via mappings with a constant number of ancilla qubits (Chapter 5).

The century-old work of Jordan and Wigner [61] inspired the idea of the first fermion–qubit mapping, which Lieb et al. used to solve the  $XY$ –model Hamiltonian classically in 1961 [62]. In 2001, Ortiz employed the transformation as a fermion–qubit mapping in the first explicit proposal of quantum simulation for fermionic Hamiltonians [63]. The map that has emerged, the “Jordan–Wigner transformation” is intuitive and by far the best-known of all fermion–qubit mappings. Performing well in simulating nearest-neighbour Hamiltonians acting on 1D chains of fermionic modes, the Jordan–Wigner transformation experiences linear  $\mathcal{O}(n)$  overheads for more complicated  $n$ –mode systems.

The Bravyi–Kitaev transform [64] marked a potential improvement upon the Jordan–Wigner transformation, producing representations of the  $n$ –mode annihilation operators as sums of Pauli strings with a maximum weight of only ( $\lceil \log_2 n \rceil + 1$ ). While Bravyi and Kitaev established an elegant mathematical definition for fermion–qubit mappings as unitary embeddings of fermionic algebras, their focus on a classification of mappings that linearly encode the Fock vectors into classical bit-strings set the tone of the field for many years. In linear encodings, each Fock state  $|\mathbf{f}\rangle \in \mathcal{H}_f^n$  corresponds to a computational basis state  $|\mathbf{q}\rangle \in \mathfrak{C}_n$  through an invertible binary matrix multiplication  $\mathbf{q} = G\mathbf{f}$  for  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ . Mappings with this property, which we call “linear encodings of the Fock basis”, correspond to classical data structures [65] and invertible binary matrices [66–70]. Fermionic operations manifest through update, parity check, and flip rules on superpositions of the classical  $n$ –bit vectors representing the fermionic system. Optimisations over the space of invertible binary matrices [67, 68] has led to the discovery of linear encodings that produce Hamiltonians for molecular compounds with operators weights lower still than the Bravyi–Kitaev transformation. For example, the pruned Sierpinski tree transform [2] is another linear encoding that implements the update rules of  $n$ –mode fermionic interactions in only  $\sim \lceil \log_3(2n + 1) \rceil$  fundamental quantum gates.

Ternary tree transformations [71, 72] marked a significant paradigm shift away from the framework of linear encodings. The defining feature of ternary–tree transformations is the branching Pauli operator structure of the Majorana representations, enabling a mapping with the provably minimal average Pauli weight of  $\sim \lceil \log_3(2n + 1) \rceil$  [72]. Ternary tree transformations seemingly expanded fermion–qubit mappings into a new classification, with new optimisation routines: the Bonsai algorithm [73] reduces SWAP gate requirements of fermionic simulation on real-world quantum hardware, treespilation [74] performs a computational search over a well-behaved subset of ternary

tree transformations to minimise the CNOT cost in molecular simulation, and other methods have optimised for entanglement costs [75].

In what has evolved into a separate area of research, geometrically local fermion–qubit mappings have arisen to meet the growing demand for efficiently simulable qubit Hamiltonians of fermionic lattices [76–80]. These systems are difficult to simulate: whether by Jordan–Wigner, Bravyi–Kitaev or ternary tree transformations, local fermionic interactions map to costly non-local qubit operations. The workaround to this challenge has been a host of locality-preserving fermion–qubit mappings [59, 60, 70, 77–79, 81–83]. A common theme among these proposals is the use of  $\mathcal{O}(n)$  ancilla qubits, which are necessary for geometrically local qubit operators respecting the same commutation relations as the terms of the fermionic Hamiltonian. While geometrically local mappings provide definite advantages over the ancilla-free alternatives in the fault-tolerant regime, in near-term devices the dust has not yet settled. On the one hand, there is evidence that some geometrically local mappings outperform the Jordan–Wigner transformation for small Hubbard models [84]; on the other hand, other geometrically local mappings yield greater numerical resource overheads than their non-local counterparts [85].

While the theory has expanded to describe so many alternatives and modifications to the original Jordan–Wigner transformation, the application of fermion–qubit mappings in the quantum sciences has not caught up with this shift. Chemists and physicists have tended to remain steadfast with techniques from the Jordan–Wigner transformation, e.g. [23, 86, 87]. In addition to providing the theoretical results and advancements to the literature, another aim of this thesis is to depict the essence of fermion–qubit mappings in all their variations in a format that is accessible to a broad audience.

## 1.4 Thesis summary

Mapping fermionic systems to qubits is a crucial step in the simulation of fundamental physics on quantum computers, but, as Section 1.3 describes, the field is currently expanding on several distinct fronts without a notational consistency. This thesis details a new, complete framework for understanding the mathematical structure of fermion–qubit mappings, using a rigorous definition to classify the different approaches and detailing new methods for optimising mappings to enhance the efficiency and performance of quantum simulations. Our perspective leads intuitively to optimisation strategies that require at most a constant number of extra qubits, contrasting many recent approaches in the field.

Chapter 2 unifies the notation of fermion–qubit mappings. The past 20 years of research has produced many ancilla–free and locality–preserving mappings at the expense of a segregated classification. Descriptions of fermion–qubit mappings in the literature typically describe either the encoded operators or the encoded Fock basis. In providing a unified notation and definition that encapsulates both approaches, we centralise the key concepts of fermion–qubit mappings in Definition 2.1.7, which underpins the rest of this thesis. We summarise classical, affine, and linear encodings of the Fock basis, as well as ternary tree transformations and mappings that fall into the cracks between these definitions. Our main result, Theorem 1, derives the explicit formula for the Majorana representations of an affine encoding in a demonstration of our unified notation.

Chapter 3 uses the notation of Chapter 2 to classify Pauli–based mappings through an equivalence relation. The major technical work of this chapter, Theorem 2, uses the equivalence relation to prove that product–preserving ternary tree transformations are equivalent to linear encodings of the Fock basis, which provides new context to recent optimisations that treat these classes of mappings as distinct. We also illustrate our classification system through demonstrating the fourteen distinct classes, or ‘templates’, of product–preserving, Pauli–based mappings on three fermionic modes.

Chapter 4 presents novel optimisations for quantum simulation by exploiting fermionic labelling, a symmetry of the classification in Chapter 3. Uncovering optimal Jordan–Wigner type mappings for the 2D fermionic lattice, we use a fermionic labelling that minimises the average Pauli weight of nonlocal qubit Hamiltonian terms by 13.9% compared to previous methods.

Chapter 5 augments the idea of Chapter 4 to define new optimisation problems for fermionic simulation, using a constant overhead of ancilla qubits to produce results that outperform the optimality of the previous chapter. We first generalise Definition 2.1.7 to incorporate ancilla–qubit mappings, extending our notation to encompass all mappings in the literature. After introducing a new method for incrementally adding ancilla qubits produce lower–cost qubit Hamiltonians, we show that just two extra qubits can drastically improve the Jordan–Wigner transformation to achieve a 37.9% reduction in average Pauli weight compared to previous methods.

Appendix A includes a glossary of many terms and notations that we use in this thesis.



# Chapter 2

## Defining fermion–qubit mappings

In comparison to the many distinct descriptions of fermion–qubit mappings that appeared in the literature review of Chapter 1, the first part of this chapter proposes a general definition for fermion–qubit mappings as unitary maps between fermionic and qubit Hilbert spaces. This definition captures the dual properties of Majorana representations and Fock states on the qubit register. Persisting throughout Chapters 3 and 4, our definition culminates in Chapter 5 with an extension to include ancilla–qubit mappings, hence capturing the entire literature.

The chapter proceeds by recontextualising distinct classes of fermion–qubit mappings from the literature. Section 2.2 describes Pauli–based mappings, such as ternary tree transformations, which represent Majorana operators with Pauli matrices. There is significant algebraic structure to sets of  $2n$  anticommuting Pauli operators on  $n$  qubits, which we use to evaluate the number of distinct Pauli–based mappings. While this number grows far beyond exponentially in  $n$ , this exploration lays the foundations for our equivalence relation in Chapter 3.

While the description of Pauli–based mappings revolves entirely around the encoded Majorana operators, Section 2.3 addresses the state–based definitions of mappings from the literature. A hierarchy emerges when imposing the progressively stricter requirements that the Fock basis of a mapping contain only states that are product, computational basis, affinely or linearly encoded states. The most well–known mappings – the Jordan–Wigner, Bravyi–Kitaev and parity basis transformations – are both linear encodings and ternary tree transformations, seeding the idea to explore the relationship between these two nominally separate constructions in Chapter 3.

Section 2.4 presents Theorem 1, which proves that the Pauli–based classical encodings of the Fock basis are precisely the affine encodings, and that affine encodings only differ from linear encodings in the signs of their Majorana representations.

## 2.1 A mathematical definition for fermion–qubit mappings

This section presents the mathematical definition of a fermion–qubit mapping as a transformation between fermionic and qubit Hilbert spaces. In 2002, Bravyi and Kitaev gave a general outline for fermion–qubit mappings as unitary embeddings [64], but in the years since the tendency has been to describe a mapping in terms of the impact it has on other objects rather than as a mathematical object in its own right. For example, the qubit Hamiltonians in Figure 1.4, a list of Majorana operators, or a set of qubit–encoded Fock states are all the result of applying a mapping. In this section, we use representation theory to give a mathematical definition for fermion–qubit mappings from  $n$ –mode fermionic space to  $n$ –qubit space. Chapter 5 extends the definition naturally to include ancilla–qubit mappings, thus encapsulating all of the literature.

Let  $n \in \mathbb{N}$  and suppose that  $\mathcal{H}$  is a Hilbert space of sufficiently high dimension. A *representation of the canonical anticommutation relations over  $\mathbb{R}^n$  in  $\mathcal{H}$* , or *CAR representation*, is a linear map

$$\phi : \mathbb{R}^n \longrightarrow B(\mathcal{H}), \quad \phi : \mathbf{x} \longmapsto \phi(\mathbf{x}) \quad (2.1)$$

such that, for all  $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ ,

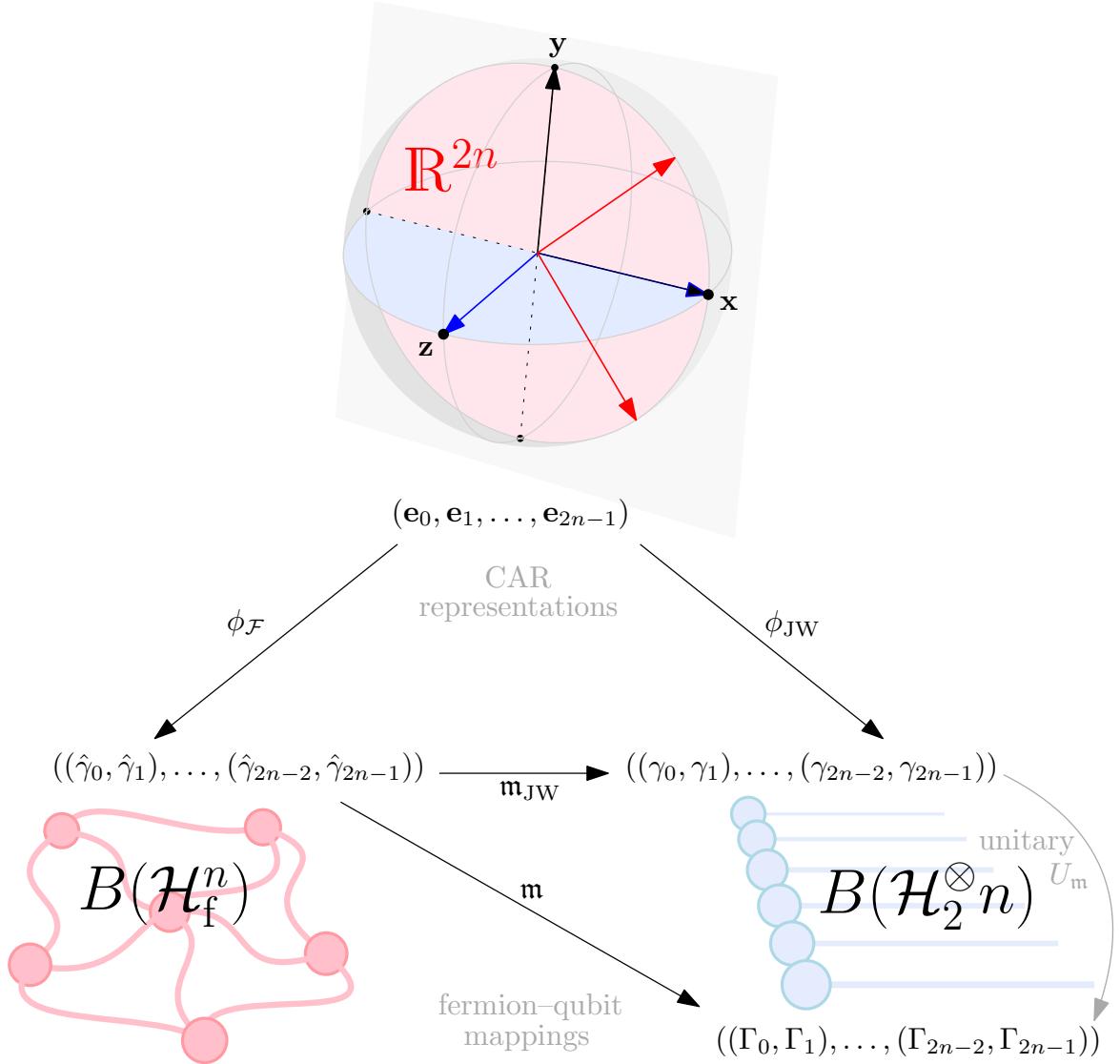
$$\phi(\mathbf{x})^\dagger = \phi(\mathbf{x}), \quad \phi(\mathbf{x})\phi(\mathbf{x}') + \phi(\mathbf{x}')\phi(\mathbf{x}) = 2(\mathbf{x}_1 \cdot \mathbf{x}_2)\hat{1}. \quad (2.2)$$

**Definition 2.1.1.** (*Irreducible, Majorana and Jordan–Wigner CAR representations.*) A CAR representation  $\phi$  over  $\mathbb{R}^n$  in a Hilbert space  $\mathcal{H}$  is *irreducible* if there is no nontrivial proper subspace  $0 < \mathcal{H}' < \mathcal{H}$  with the property that  $\phi(\mathbf{x})(\mathcal{H}') \leq \mathcal{H}'$  for all  $\mathbf{x} \in \mathbb{R}^n$ .

Let  $(\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_{2n-1})$  be an ordered basis for  $\mathbb{R}^{2n}$ . Below are two examples of irreducible CAR representations over  $\mathbb{R}^{2n}$ .

- a) The  $n$ –mode Majorana CAR representation  $\phi_{\mathcal{F}}$  is the CAR representation over  $\mathbb{R}^{2n}$  in  $\mathcal{H}_{\text{f}}^n$  that maps each basis vector to the corresponding Majorana operator:

$$\phi_{\mathcal{F}} : \mathbb{R}^{2n} \longrightarrow \mathcal{H}_{\text{f}}^n, \quad \phi_{\mathcal{F}} : \mathbf{e}_i \longmapsto \hat{\gamma}_i \quad \text{for all } i \in [2n]. \quad (2.3)$$



**Figure 2.1** CAR representations  $\phi$  are linear maps that identify Euclidean space with sets of anticommuting, Hermitian, unitary operators, which generate algebras in Hilbert spaces. Fermion–qubit mappings  $\mathfrak{m}$  are the unitary equivalences between fermionic and qubit CAR representations over even–dimensional Euclidean spaces. The Majorana representations of an  $n$ –mode fermion–qubit mapping  $\mathfrak{m}$  are the operators  $\Gamma_i = \mathfrak{m}^\dagger \hat{\gamma}_i \mathfrak{m}^\dagger \in B(\mathcal{H}_2^{\otimes n})$ , and the Fock basis of  $\mathfrak{m}$  consists of the qubit states  $|\mathbf{f}_m\rangle = \mathfrak{m} |\mathbf{f}\rangle \in \mathcal{H}_2^{\otimes n}$  for binary strings  $\mathbf{f} \in \mathbb{Z}_2^n$ . The unitary operator  $U_m \in \mathcal{U}_n$  that performs  $U_m |\mathbf{f}\rangle = |\mathbf{f}_m\rangle$  also satisfies  $U_m \gamma_i U_m^\dagger = \Gamma_i$ , where the Pauli operator  $\gamma_i$  is the Jordan–Wigner representation of  $\hat{\gamma}_i$ .

- b) The  $n$ -qubit Jordan–Wigner CAR representation  $\phi_{\text{JW}}$  is the CAR representation over  $\mathbb{R}^{2n}$  in  $B(\mathcal{H}_2^{\otimes n})$  of the form

$$\phi_{\text{JW}} : \mathbb{R}^{2n} \longrightarrow B(\mathcal{H}_2^{\otimes n}), \quad \phi_{\text{JW}} : \mathbf{e}_i \longmapsto \gamma_i \quad \text{for all } i \in [2n], \quad (2.4)$$

where the  $n$ -qubit Jordan–Wigner operators have definition

$$\gamma_{2i} := \left( \prod_{k=0}^{i-1} Z_k \right) X_i, \quad \gamma_{2i+1} := \left( \prod_{k=0}^{i-1} Z_k \right) Y_i \quad \text{for } i \in [n]. \quad (2.5)$$

The Jordan–Wigner operators form a sequence  $((\gamma_{2i}, \gamma_{2i+1}))_{i=0}^{n-1}$  comprising pairs of operators in  $B_h(\mathcal{H}_2^{\otimes n})$ .

**Proposition 1.** (*Irreducible CAR representations are unitarily equivalent to  $\phi_{\text{JW}}$ .*) Any CAR representation  $\phi$  over  $\mathbb{R}^{2n}$  in a Hilbert space  $\mathcal{H}$  with  $\dim(\mathcal{H}) \geq 2^n$  is isometrically equivalent to the  $n$ -qubit Jordan–Wigner transformation  $\phi_{\text{JW}}$ , in that there exists a Hilbert space  $\mathcal{K}$  and a unitary map  $\mathfrak{m} \in \mathcal{U}(\mathcal{H}, \mathcal{H}_2^{\otimes n} \otimes \mathcal{K})$  such that

$$\mathfrak{m} \cdot \phi(\mathbf{x}) \cdot \mathfrak{m}^\dagger = \phi_{\text{JW}}(\mathbf{x}) \otimes \mathbb{1}_{\mathcal{K}} \quad \text{for all } \mathbf{x} \in \mathbb{R}^{2n}. \quad (2.6)$$

The representation  $\phi$  is irreducible if and only if  $\mathcal{K} = \mathbb{C}$ , which is equivalent to requiring that  $\dim(\mathcal{H}) = 2^n$ . If  $\phi$  is irreducible, Schur’s Lemma states that the unitary map  $\mathfrak{m}$  is unique up to an arbitrary complex phase. That is, the only other unitary maps satisfying Equation 2.6 are the elements of the family  $\{\mathfrak{m}e^{i\theta} \mid \theta \in [0, 2\pi)\}$ .

*Proof.* See Theorem 12.27 of [88].  $\square$

Proposition 1 motivates the fundamental Definition 2.1.7 for fermion–qubit mappings as the unitary equivalences between CAR representations in fermionic and qubit space.

**Corollary 2.1.2.** The  $n$ -mode Majorana CAR representation is unitarily equivalent to the  $n$ -qubit Jordan–Wigner CAR representation; that is, there is a unitary map  $\mathfrak{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  such that  $\mathfrak{m} \cdot \hat{\gamma}_i \cdot \mathfrak{m}^\dagger = \gamma_i$  for all  $i \in [2n]$ , which is unique up to a complex phase.

*Proof.* Take  $\phi$  to be  $\phi_{\mathcal{F}}$  in Proposition 1 and note that  $\mathcal{H}_2^{\otimes n} \otimes \mathbb{C} \cong \mathcal{H}_2^{\otimes n}$ .  $\square$

**Corollary 2.1.3.** There is a unique unitary map  $\mathfrak{m}_{\text{JW}} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  that satisfies

$$\mathfrak{m}_{\text{JW}} \cdot \hat{\gamma}_i \cdot (\mathfrak{m}_{\text{JW}})^\dagger = \gamma_i \quad \text{for all } i \in [2n] \quad \text{and} \quad \mathfrak{m}_{\text{JW}}|\mathbf{f}\rangle = |\mathbf{f}\rangle \quad \text{for all } \mathbf{f} \in \mathbb{Z}_2^n. \quad (2.7)$$

*Proof.* Let  $\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  be a unitary map satisfying  $\mathbf{m} \cdot \hat{\gamma}_i \cdot \mathbf{m}^\dagger = \gamma_i$  for all  $i \in [2n]$  from Corollary 2.1.2. Take the vacuum state  $|\mathbf{0}\rangle \in \mathcal{H}_f^n$  from the Fock basis  $\mathfrak{F}_n$  along with the single-qubit operator  $Z_j \in B(\mathcal{H}_2^{\otimes n})$  for any  $j \in [n]$  and note that

$$Z_j \cdot (\mathbf{m}|\mathbf{0}\rangle) = (Z_j \cdot \mathbf{m})|\mathbf{0}\rangle = (\mathbf{m} \cdot (-i\hat{\gamma}_{2j}\hat{\gamma}_{2j+1}))|\mathbf{0}\rangle = \mathbf{m}|\mathbf{0}\rangle. \quad (2.8)$$

Therefore  $\mathbf{m}|\mathbf{0}\rangle = e^{i\phi}|\mathbf{0}\rangle$  for some  $\phi \in [0, 2\pi)$ , where  $|\mathbf{0}\rangle|0\rangle^{\otimes n} \in \mathcal{H}_2^{\otimes n}$  is the vector  $|0\rangle^{\otimes n}$  of the computational basis  $\mathfrak{C}_n$ . Let  $\mathbf{m}_{\text{JW}} = \mathbf{m}e^{-i\phi}$  so that  $\mathbf{m}_{\text{JW}}|\mathbf{0}\rangle = |\mathbf{0}\rangle$ . It follows from Equation 1.37 that

$$\mathbf{m}_{\text{JW}}|\mathbf{f}\rangle = \mathbf{m}_{\text{JW}}\left(\hat{\gamma}_0^{f_0} \dots \hat{\gamma}_{2n-2}^{f_{n-1}}\right)|\mathbf{0}\rangle = (X_0)^{f_0} \dots (X_{n-1})^{f_{n-1}}|\mathbf{0}\rangle = |\mathbf{f}\rangle. \quad (2.9)$$

for all  $\mathbf{f} \in \mathbb{Z}_2^n$ .  $\square$

**Definition 2.1.4.** (*The Jordan–Wigner transformation.*) The  $n$ -mode Jordan–Wigner transformation is the unique unitary map  $\mathbf{m}_{\text{JW}} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  to satisfy Equation 2.7.

**Corollary 2.1.5.** (*Alternatives to the Jordan–Wigner operators.*) For any sequence  $((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  comprising pairs of operators in  $B(\mathcal{H}_2^{\otimes n})$  that satisfy the CARs

$$\Gamma_i^\dagger = \Gamma_i, \quad \{\Gamma_i, \Gamma_j\} = 2\delta_{ij}\mathbb{1}^{\otimes n} \quad \text{for all } i, j \in [2n], \quad (2.10)$$

there exists a unitary matrix  $U_{\mathbf{m}} \in \mathcal{U}_n$  such that  $U_{\mathbf{m}}\gamma_i U_{\mathbf{m}}^\dagger = \Gamma_i$  for all  $i \in [2n]$ , which is unique up to a complex phase.

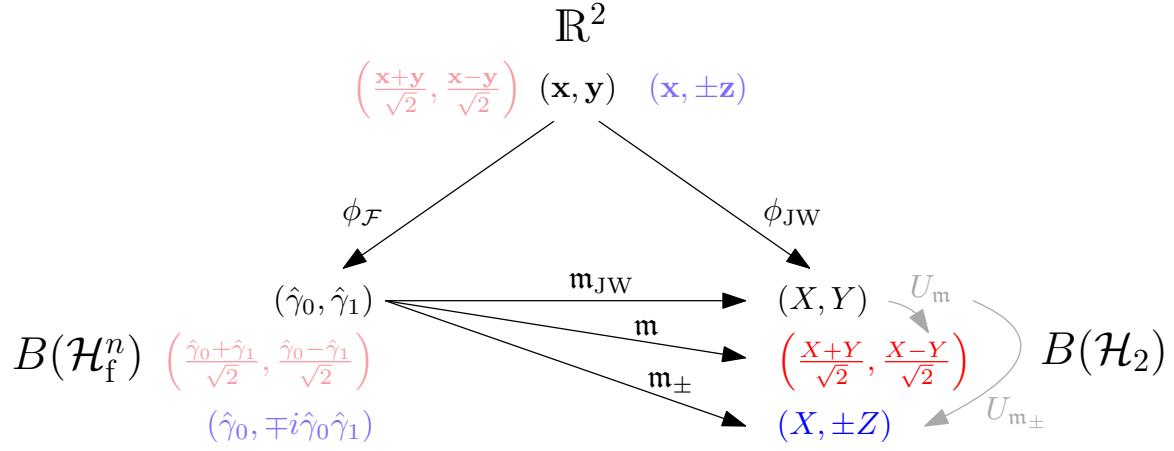
*Proof.* This follows from taking  $\phi = \phi_{\mathbf{m}}$  in Proposition 1, where the CAR representation  $\phi_{\mathbf{m}}$  satisfies  $\phi_{\mathbf{m}} : \mathbf{e}_i \mapsto \Gamma_i$  for all  $i \in [2n]$ .  $\square$

**Corollary 2.1.6.** (*Soundness of fermion–qubit mapping definition.*)

The following objects are equivalent:

- a) A unitary map  $\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  between fermionic and qubit Hilbert spaces, identifying an orthonormal basis  $\{|\mathbf{f}_{\mathbf{m}}\rangle := \mathbf{m}|\mathbf{f}\rangle \mid \mathbf{f} \in \mathbb{Z}_2^n\}$  for  $\mathcal{H}_2^{\otimes n}$ .
- b) A sequence  $((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  comprising pairs of operators in  $B(\mathcal{H}_2^{\otimes n})$  satisfying the CARs of Equation 2.10, along with any complex phase.

*Proof.* Starting with a), note that the operators  $\Gamma_i := \mathbf{m} \cdot \hat{\gamma}_i \cdot \mathbf{m}^\dagger \in B(\mathcal{H}_2^{\otimes n})$  for all  $i \in [2n]$  satisfy Equation 2.10 due to the unitarity of  $\mathbf{m}$ . From Proposition 1, the map  $\mathbf{m}$  is the unique one to implement  $\mathbf{m} : \gamma_i \mapsto \mathbf{m} \cdot \gamma_i \cdot \mathbf{m}^\dagger = \Gamma_i$  up to a complex phase.



**Figure 2.2** Example CAR representations  $\phi_F$  and  $\phi_{JW}$  over  $\mathbb{R}^2$ , which identifies the single-mode Jordan–Wigner transformation  $m_{JW}$ . The Majorana representations of the mappings  $m$  and  $m_{\pm}$  relate to the Jordan–Wigner transformation via the single-qubit unitary matrices  $U_m$  and  $U_{m\pm}$ , respectively.

Starting with b), note that the unitary map  $m := U_m \cdot m_{JW} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  satisfies  $m \cdot \hat{\gamma}_i \cdot m^\dagger = \Gamma_i$  for all  $i \in [2n]$ , where  $U_m \in \mathcal{U}_n$  is a unitary matrix, unique up to a complex phase, satisfying  $U_m \gamma_i U_m^\dagger = \Gamma_i$  for all  $i \in [2n]$  from Corollary 2.1.5.  $\square$

**Definition 2.1.7.** (*Fermion–qubit mappings.*)

An  $n$ -mode fermion–qubit mapping is a unitary map  $m \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$  between fermionic and qubit Hilbert spaces. The *Fock basis* of  $m$  is the orthonormal basis for  $\mathcal{H}_2^{\otimes n}$

$$\{|\mathbf{f}_m\rangle := m|\mathbf{f}\rangle \mid \mathbf{f} \in \mathbb{Z}_2^n\}. \quad (2.11)$$

The unitary matrix  $U_m \in \mathcal{U}_n$  that performs the map  $|\mathbf{f}\rangle \mapsto |\mathbf{f}_m\rangle$  specifies the operator pair sequence  $((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  via the relation

$$m \cdot \hat{\gamma}_i \cdot m^\dagger = \Gamma_i = U_m \gamma_i U_m^\dagger \quad \text{for all } \mathbf{f} \in \mathbb{Z}_2^n \quad \text{and all } i \in [2n]. \quad (2.12)$$

The *Majorana representations*  $\Gamma_i \in B_h(\mathcal{H}_2^{\otimes n})$  satisfy the CARs

$$\Gamma_i^\dagger = \Gamma_i, \quad \{\Gamma_i, \Gamma_j\} = 2\delta_{ij}\mathbb{1}^{\otimes n} \quad \text{for all } i, j \in [2n], \quad (2.13)$$

and are thus unitary and Hermitian.

As the Majorana representation sequence  $((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  specifies  $m$  up to a complex phase, we write  $m \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ .

Figure 2.1 illustrates the roles of  $\mathbf{m}_{\text{JW}}$ ,  $U_{\mathbf{m}}$  and  $\mathbf{m}$  from Definition 2.1.7, while Figure 2.2 illustrates Example 2.1.8.

**Example 2.1.8.** (*Some single-mode fermion–qubit mappings.*) Take the orthonormal basis  $(\mathbf{x}, \mathbf{y})$  for  $\mathbb{R}^2$ . The single-mode Majorana CAR representation is

$$\phi_{\mathcal{F}} : \mathbb{R}^2 \longrightarrow B(\mathcal{H}_f^1), \quad \phi_{\mathcal{F}} : \begin{cases} \mathbf{x} \longmapsto \hat{\gamma}_0 \\ \mathbf{y} \longmapsto \hat{\gamma}_1. \end{cases} \quad (2.14)$$

The single-mode Jordan–Wigner CAR representation is

$$\phi_{\text{JW}} : \mathbb{R}^2 \longrightarrow B(\mathcal{H}_2), \quad \phi_{\text{JW}} : \begin{cases} \mathbf{x} \longmapsto X =: \gamma_0 \\ \mathbf{y} \longmapsto Y =: \gamma_1. \end{cases} \quad (2.15)$$

The single-mode Jordan–Wigner transformation is the map  $\mathbf{m}_{\text{JW}} \in \mathcal{U}(\mathcal{H}_f, \mathcal{H}_2)$  with

$$\begin{pmatrix} \mathbf{m}_{\text{JW}} \cdot \hat{\gamma}_0 \cdot (\mathbf{m}_{\text{JW}})^\dagger = X \\ \mathbf{m}_{\text{JW}} \cdot \hat{\gamma}_1 \cdot (\mathbf{m}_{\text{JW}})^\dagger = Y \end{pmatrix}, \quad \mathbf{m}_{\text{JW}} : \begin{cases} |0\rangle \longmapsto |0\rangle \\ |1\rangle \longmapsto |1\rangle \end{cases}. \quad (2.16)$$

Let  $\phi \in [0, 2\pi)$ . One single-mode fermion–qubit mapping is  $\mathbf{m} \in \mathcal{U}(\mathcal{H}_f, \mathcal{H}_2)$ , with

$$\begin{pmatrix} \mathbf{m} \cdot \hat{\gamma}_0 \cdot \mathbf{m}^\dagger = \frac{X+Y}{\sqrt{2}} =: \Gamma_0 \\ \mathbf{m} \cdot \hat{\gamma}_1 \cdot \mathbf{m}^\dagger = \frac{X-Y}{\sqrt{2}} =: \Gamma_1 \end{pmatrix}, \quad \mathbf{m} : \begin{cases} |0\rangle \longmapsto e^{i(\phi+\pi/4)} |1\rangle =: |0_{\mathbf{m}}\rangle \\ |1\rangle \longmapsto e^{i\phi} |0\rangle =: |1_{\mathbf{m}}\rangle. \end{cases} \quad (2.17)$$

Furthermore, there is a unique unitary matrix  $U_{\mathbf{m}} \in \mathcal{U}_1$ :

$$U_{\mathbf{m}} = e^{i\phi} \begin{pmatrix} 0 & 1 \\ e^{i\pi/4} & 0 \end{pmatrix} \iff \begin{pmatrix} U_{\mathbf{m}} X U_{\mathbf{m}}^\dagger = \frac{X+Y}{\sqrt{2}} \\ U_{\mathbf{m}} Y U_{\mathbf{m}}^\dagger = \frac{X-Y}{\sqrt{2}} \end{pmatrix}, \quad U_{\mathbf{m}} : \begin{cases} |0\rangle \longmapsto e^{i(\phi+\pi/4)} |1\rangle \\ |1\rangle \longmapsto e^{i\phi} |0\rangle. \end{cases} \quad (2.18)$$

Two more single-mode fermion–qubit mappings are  $\mathbf{m}_\pm \in \mathcal{U}(\mathcal{H}_f, \mathcal{H}_2)$  with

$$\begin{pmatrix} \mathbf{m}_\pm \cdot \hat{\gamma}_0 \cdot \mathbf{m}_\pm^\dagger = X =: \Gamma_0 \\ \mathbf{m}_\pm \cdot \hat{\gamma}_1 \cdot \mathbf{m}_\pm^\dagger = \pm Z =: \Gamma_1 \end{pmatrix}, \quad \mathbf{m}_\pm : \begin{cases} |0\rangle \longmapsto e^{i\phi} |\mp i\rangle =: |0_{\mathbf{m}_\pm}\rangle \\ |1\rangle \longmapsto e^{i\phi} |\pm i\rangle =: |1_{\mathbf{m}_\pm}\rangle. \end{cases} \quad (2.19)$$

There is a unique unitary matrix  $U_{\mathbf{m}_\pm} \in \mathcal{U}_1$ :

$$U_{\mathbf{m}_\pm} = \frac{e^{i\phi}}{\sqrt{2}} \begin{pmatrix} 1 & \mp i \\ \mp i & 1 \end{pmatrix} \iff \begin{pmatrix} U_{\mathbf{m}_\pm} X U_{\mathbf{m}_\pm}^\dagger = X \\ U_{\mathbf{m}_\pm} Y U_{\mathbf{m}_\pm}^\dagger = \pm Z \end{pmatrix}, \quad U_{\mathbf{m}_\pm} : \begin{cases} |0\rangle \mapsto e^{i\phi} |\mp i\rangle \\ |1\rangle \mapsto e^{i\phi} |\pm i\rangle. \end{cases} \quad (2.20)$$

### 2.1.1 Applying fermion–qubit mappings to Hamiltonians

The purpose of a fermion–qubit mapping is to translate a fermionic Hamiltonian into a qubit Hamiltonian.

**Definition 2.1.9.** (*Applying fermion–qubit mappings to fermionic Hamiltonians.*)

Let  $\hat{H}_{\text{fermion}} \in B_{\text{h}}(\mathcal{H}_{\text{f}}^n)$  be an  $n$ –mode fermionic Hamiltonian and let  $\mathbf{m} \in \mathcal{U}(\mathcal{H}_{\text{f}}^n, \mathcal{H}_2^{\otimes n})$  be an  $n$ –mode fermion–qubit mapping. The *application of  $\mathbf{m}$  to  $\hat{H}_{\text{fermion}}$*  is the  $n$ –qubit Hamiltonian

$$H_{\text{qubit}} = \mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger; \quad (2.21)$$

we say that  $\mathbf{m}$  *produces*  $H_{\text{qubit}}$  *from*  $\hat{H}_{\text{fermion}}$ .

Different fermion–qubit mappings produce different qubit Hamiltonians from the same fermionic system. Figure 1.4 displays three different qubit Hamiltonians equivalent to fermionic hopping terms on a  $6 \times 6$  square lattice via a Jordan–Wigner transformation, the Bravyi–Kitaev superfast transformation [64], and the Verstraete–Cirac transformation [60]. While each qubit Hamiltonian is unitarily equivalent, there is no reason to expect the resource cost of simulation algorithms for distinct qubit Hamiltonians to be the same. As Section 1.2.3 outlines, typical cost functions for unitary operators involve gate count, qubit count and depth, which depend on the explicit forms of unitary operators rather than spectral properties.

From a quantum engineering perspective, resource scarcities limit the range of feasible qubit Hamiltonians. Consideration of the simulation algorithms in Section 1.2.4 suggests that the number and complexity of the operator terms in a qubit Hamiltonian are sources of computational inefficiency, and thus serve as practical measures for the quality of a qubit Hamiltonian.

Choosing an appropriate fermion–qubit mapping for a problem  $n$ –mode Hamiltonian  $\hat{H}_{\text{fermion}}$  involves the following fundamental steps:

- A selection  $\mathfrak{M} \subseteq \mathcal{U}(\mathcal{H}_{\text{f}}^n, \mathcal{H}_2^{\otimes n})$  of fermion–qubit mappings over which optimisation is feasible. For example, linear encodings [67, 69, 89], ternary tree transformations [73, 74], or Jordan–Wigner transformations with different fermionic labels [3].
- A cost function  $C : \mathcal{U}_n \rightarrow \mathbb{R}$  for unitary operators relating to the qubit Hamiltonian  $H_{\text{qubit}} = \mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger$  for each mapping  $\mathbf{m} \in \mathfrak{M}$ . For example, the time evolution operator  $e^{-iH_{\text{qubit}}t}$  or individual terms of a decomposition of  $H_{\text{qubit}}$  into a linear combination of unitary operators. While  $H_{\text{qubit}}$  itself is not a unitary

operator, the expression  $C(H_{\text{qubit}})$  is short-hand denoting the cost of simulating  $H_{\text{qubit}}$  via the method of interest.

- An optimisation routine to determine or approximate the optimal fermion–qubit mapping

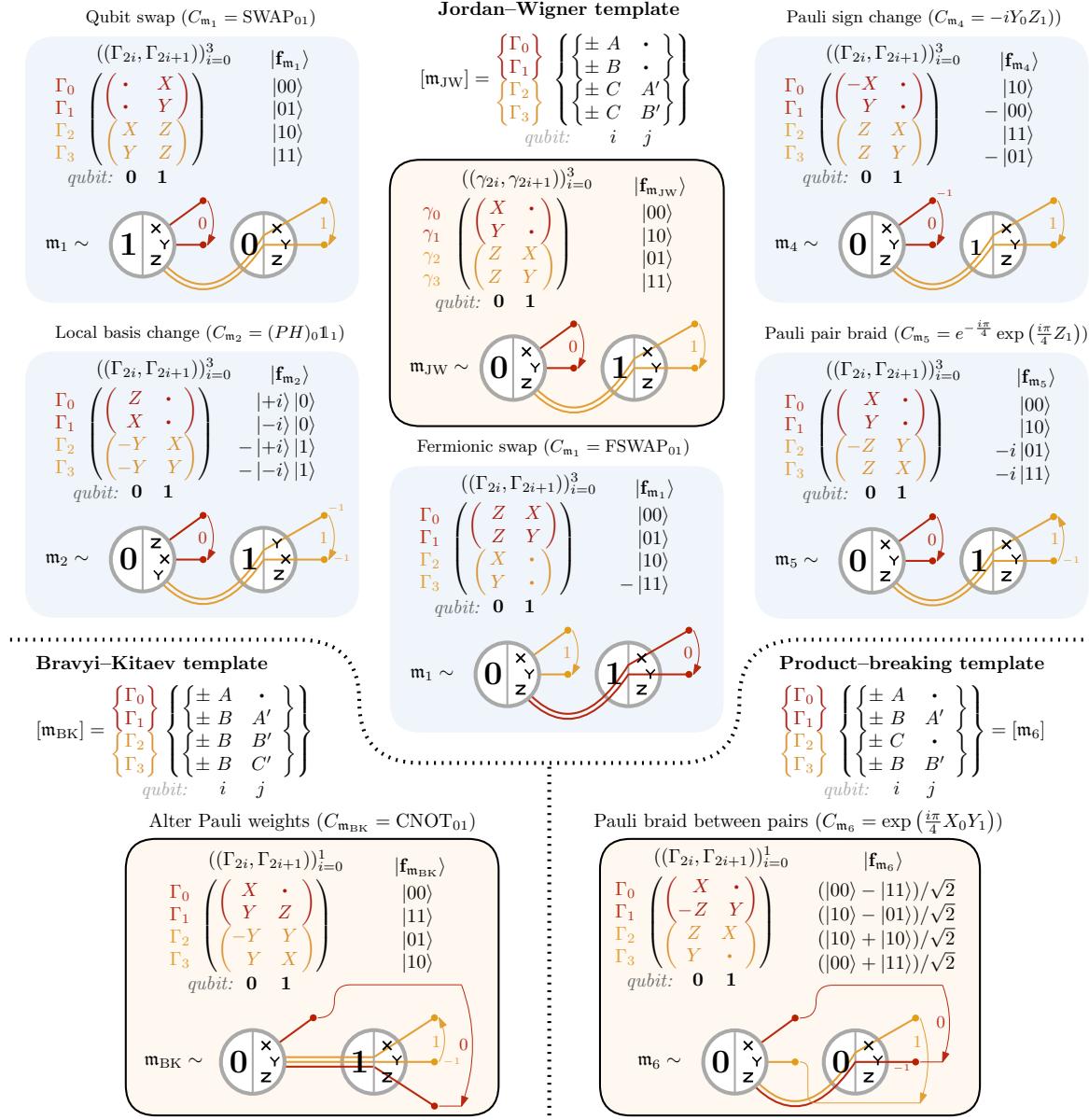
$$\mathbf{m}^* = \arg \min_{\mathbf{m} \in \mathfrak{M}} (C(\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger)). \quad (2.22)$$

## 2.2 Pauli–based fermion–qubit mappings

This section focusses the completely general Definition 2.1.7 of fermion–qubit mappings on the subset with Majorana representations that are Pauli operators. Most of the works in the literature describe only these Pauli–based mappings; a rare exception is [69]. Section 2.2.1 extends existing theory on the behaviour of anticommuting sets of Pauli operators [90] to evaluate the number of distinct  $n$ -mode Pauli–based mappings, illustrating just how many there are. Later portions of this chapter and Chapter 3 serve to classify the Pauli–based mappings, building a clear taxonomy for search routines to find efficient mappings. Section 2.2.2 formalises ternary tree transformations in terms of Definition 2.1.7.

**Definition 2.2.1.** (*Pauli–based fermion–qubit mapping.*) An  $n$ -mode fermion–qubit mapping  $\mathbf{m} : \mathcal{H}_f^n \rightarrow \mathcal{H}_2^{\otimes n}$  is *Pauli–based* if the Majorana representations  $\Gamma_i := \mathbf{m} \cdot \hat{\gamma}_i \cdot \mathbf{m}^\dagger \in \mathcal{P}_n$  are Pauli operators. The Fock basis of a Pauli–based mapping consists only of stabiliser states, and the unitary map  $|\mathbf{f}\rangle \mapsto |\mathbf{f}_m\rangle$  is the action of a Clifford matrix  $C_m \in \mathcal{C}_n$ , which also satisfies  $\mathbf{m} = C_m \cdot \mathbf{m}_{\text{JW}}$ .

**Example 2.2.2.** Figure 2.3 depicts the two–mode Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}}$  through two–qubit Pauli Majorana representations  $\gamma_i$  and the Fock basis vectors  $|\mathbf{f}\rangle$ . In accompaniment are five similar mappings  $\mathbf{m}_1$ – $\mathbf{m}_5$  in, along with the corresponding Clifford matrices  $C_{\mathbf{m}_1}$ – $C_{\mathbf{m}_5}$ . Also making an appearance is the two–mode Bravyi–Kitaev transformation  $\mathbf{m}_{\text{BK}}$  as well as a two–mode product–breaking mapping  $\mathbf{m}_6$ , so–called because its Fock states are entangled. Note that we write the order of the Fock basis states from top-to-bottom as  $|00_m\rangle, |10_m\rangle, |01_m\rangle, |11_m\rangle$  in Figure 2.3, which preserves the left-to-right ordering of qubit labels that appears throughout all other notation. This is a slight change from the conventional order of  $|00_m\rangle, |01_m\rangle, |10_m\rangle, |11_m\rangle$ .



**Figure 2.3** The operator– and state–based descriptions as well as the diagrams and templates (see Chapter 3) of the two–mode Pauli–based fermion–qubit mappings. On display is the Jordan–Wigner transformation  $\mathfrak{m}_{\text{JW}}$ , five equivalent mappings  $\mathfrak{m}_1$ – $\mathfrak{m}_5$ , the Bravyi–Kitaev transformation  $\mathfrak{m}_{\text{BK}}$  and a product–breaking transformation  $\mathfrak{m}_6$ . The similar diagram structure of  $\mathfrak{m}_{\text{JW}}$  and  $\mathfrak{m}_1$ – $\mathfrak{m}_5$  reveal their equivalence to each other and distinction from the Jordan–Wigner, Bravyi–Kitaev and product–breaking templates. For each of the mappings  $\mathfrak{m}$ , the Clifford operator  $C_{\mathfrak{m}}$  is the unique matrix to satisfy  $C_{\mathfrak{m}} |\mathbf{f}\rangle = |\mathbf{f}_{\mathfrak{m}}\rangle$ ; as a consequence, each also satisfies  $C_{\mathfrak{m}} \gamma_i C_{\mathfrak{m}}^\dagger = \Gamma_i$ .

### 2.2.1 Counting the number of Pauli-based mappings

This section illustrates the vastness of the set of Pauli-based mappings. We derive the total number of mappings with the assistance of results from [90], not accounting for physical equivalences upon which we expound in Section 3.1. Later in this chapter, we introduce the existing classifications of Pauli-based mappings that make the total set more tractable.

We first count the number of distinct sets of *unsigned* Pauli operators and then extend the count to signed, ordered lists of pairs of operators, giving the number of Pauli-based mappings. This gives an initial picture of how vast the set is, before we demonstrate how mappings in the existing body of literature relate to

**Definition 2.2.3.** (*Anticommuting and unsigned Pauli operators.*)

- a) The *unsigned Pauli operators* are the elements of the set  $\tilde{\mathcal{P}}_n := \{\mathbb{1}, X, Y, Z\}^{\otimes n} \subset \mathcal{P}_n$ . By definition, unsigned Pauli operators are Hermitian. Note that each Pauli operator  $P \in \mathcal{P}_n$  satisfies exactly one of the conditions  $\pm P \in \tilde{\mathcal{P}}_n$ ,  $\pm iP \in \tilde{\mathcal{P}}_n$ . Similarly, either  $P = \pm Q$  or  $P = \pm iQ$  for some  $Q \in \tilde{\mathcal{P}}_n$  and we say that  $P$  is *proportional to*  $Q$ , with mathematical expression  $P \propto Q$ .
- b) An *anticommuting set of  $k$  Pauli operators* is a subset  $\mathcal{G} = \{\Gamma_i\}_{i=0}^{k-1}$  of  $\mathcal{P}_n$  where  $\{\Gamma_i, \Gamma_j\} = 0$  for all  $i \neq j \in [k]$ .
- c) A *maximally anticommuting set of Pauli operators*  $\mathcal{G}$  is an anticommuting set of Pauli operators that is not a proper subset of any other anticommuting set of Pauli operators.

Let  $\prod \mathcal{G}$  denote the product  $\prod_{\Gamma \in \mathcal{G}} \Gamma$  for any subset  $\mathcal{G} \subset \mathcal{P}_n$  of the  $n$ -qubit Pauli operators.

**Lemma 2.2.4.** (*Results from [90].*)

Let  $\mathcal{G}$  be an anticommuting set of  $n$ -qubit Pauli operators.

- a) The number of elements in  $\mathcal{G}$  is  $|\mathcal{G}| \leq 2n + 1$ .
- b) If  $|\mathcal{G}| = 2n$ , then for any  $\mathcal{J} \subseteq \mathcal{G}$ , the Pauli operator  $\prod \mathcal{G} \not\propto \mathbb{1}^{\otimes n}$ ;
- c) If  $|\mathcal{G}| = 2n$ , every element of the Pauli group is proportional to a product of elements in  $\mathcal{G}$ .

*Proof.* For a), see Lemma 4.5 of [90]; for b), see Theorem 4.1 (a) and (d) of [90]; for c), see Theorem 4.1 (e) of [90].  $\square$

**Lemma 2.2.5.** (*Further properties of sets of  $2n$  anticommuting Pauli operators.*) Let  $\mathcal{G}$  be an anticommuting set of  $n$ -qubit Pauli operators with  $|\mathcal{G}| = 2n$ . Then,

- a) the Pauli operator  $\prod \mathcal{G}$  anticommutes with every element in  $\mathcal{G}$ , and hence  $\prod \mathcal{G}$  is not equal or proportional to any element of  $\mathcal{G}$ ;
- b) there are four maximally anticommuting sets of Pauli operators  $\mathcal{G}_{\max}$  such that  $\mathcal{G} \subset \mathcal{G}_{\max}$ , and they have the form

$$\mathcal{G}_{\max} = \mathcal{G} \cup \left\{ \pm \prod \mathcal{G} \right\} \quad \text{or} \quad \mathcal{G}_{\max} = \mathcal{G} \cup \left\{ \pm i \prod \mathcal{G} \right\}. \quad (2.23)$$

*Proof.* For part a), let  $\mathcal{G} = \{\Gamma_i\}_{i=0}^{2n-1}$  be the anticommuting set and choose any  $i \in [2n]$ . Note that

$$\left\{ \Gamma_i, \prod \mathcal{G} \right\} = (\Gamma_i(\Gamma_0\Gamma_1 \dots \Gamma_{2n-1}) + (\Gamma_0\Gamma_1 \dots \Gamma_{2n-1})\Gamma_i) \quad (2.24)$$

$$= \left( (-1)^{2n-1}(\Gamma_0\Gamma_1 \dots \Gamma_{2n-1})\Gamma_i + (\Gamma_0\Gamma_1 \dots \Gamma_{2n-1})\Gamma_i \right) \quad (2.25)$$

$$= 0, \quad (2.26)$$

since  $\Gamma_i$  anticommutes with each of the elements of  $\mathcal{G}$  except for itself. Since the operator  $\prod \mathcal{G}$  anticommutes with each element of  $\mathcal{G}$ , it cannot be equal or proportional to any of the elements of  $\mathcal{G}$ .

For part b), let  $Q \in \mathcal{P}_n$  be such that  $\mathcal{G} \cup \{Q\}$  is an anticommuting set of Pauli operators, where  $Q$  is not proportional to  $\prod \mathcal{G}$ . Due to Lemma 2.2.4c), there must exist a decomposition of  $Q$  into a product of the elements in a proper subset  $\mathcal{J} \subset \mathcal{G}$ ,

$$Q = \prod \mathcal{J}. \quad (2.27)$$

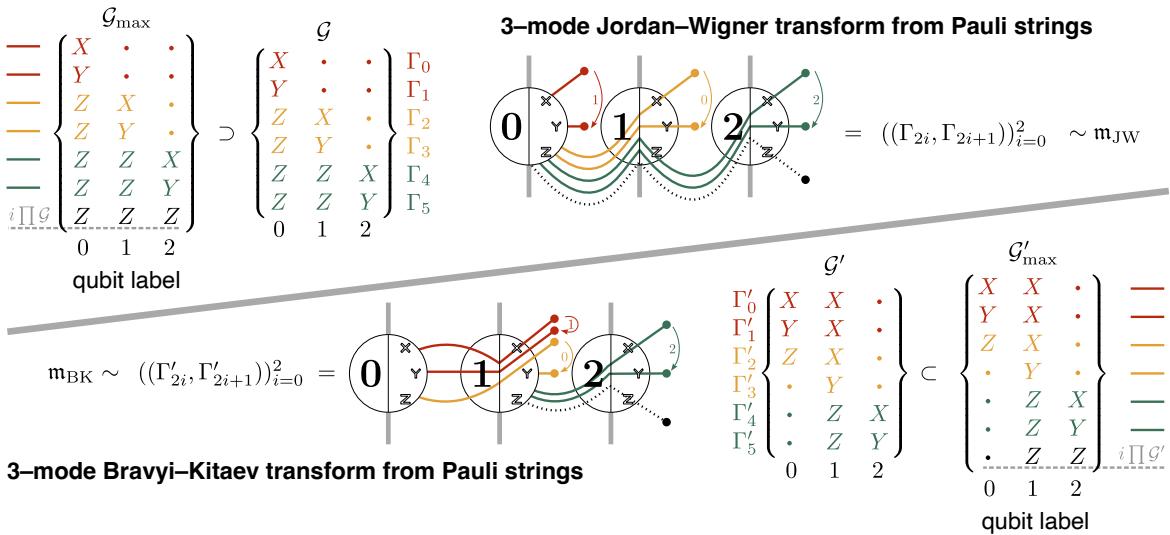
The decomposition in Equation 2.27 would have to be unique, because the equality of any two distinct products  $\prod \mathcal{J}$  and  $\prod \mathcal{J}'$  for  $\mathcal{J}, \mathcal{J}' \subset \mathcal{G}$  would imply that  $Q^2 = \prod (\mathcal{J} \Delta \mathcal{J}') = \mathbb{1}^{\otimes n}$ , where  $\Delta$  denotes the symmetric difference<sup>1</sup>, which violates Lemma 2.2.4b). An operator of the form  $Q = \prod \mathcal{J}$  satisfies

$$\Gamma_i Q = Q \Gamma_i \quad \begin{cases} \text{for all } \Gamma_i \notin \mathcal{J} \text{ if } |\mathcal{J}| \text{ is even,} \\ \text{for all } \Gamma_i \in \mathcal{J} \text{ if } |\mathcal{J}| \text{ is odd.} \end{cases} \quad (2.28)$$

Regardless of the parity of  $|\mathcal{J}|$ , Equation 2.28 contradicts the assumption that  $\mathcal{G} \cup \{Q\}$  is an anticommuting set, because it implies that there is always an operator  $\Gamma_i \in \mathcal{G}$

---

<sup>1</sup> $A \Delta B = A \cup B \setminus (A \cap B)$



**Figure 2.4** Sets of  $2n$  anticommuting, Hermitian Pauli operators define the Majorana representations of Pauli-based fermion–qubit mappings. This figure also includes the diagrammatic notation for Pauli-based mappings from Chapter 3.

that commutes with  $Q$ . The sets  $\mathcal{G} \cup \{\pm \prod \mathcal{G}\}$  and  $\mathcal{G} \cup \{\pm i \prod \mathcal{G}\}$  are anticommuting by part a); Lemma 2.2.4a) guarantees  $2n+1$  to be the maximum size of an anticommuting set.  $\square$

Remark 2.2.6 introduces a classification system for the sets of  $2n$  anticommuting Pauli operators.

**Remark 2.2.6.** (*Partitioning the anticommuting sets of  $2n$  Pauli operators via maximally anticommuting sets.*) Any anticommuting set of Pauli operators is the result of removing a single element from a maximally anticommuting set of  $(2n + 1)$  Pauli operators, as Figures 2.4 and 3.1 illustrate. That is, the maximally anticommuting sets of Pauli operators  $\{\mathcal{G}_{\max} : |\mathcal{G}_{\max}| = 2n + 1\}$  induce a partition on the anticommuting sets of  $2n$  Pauli operators  $\{\mathcal{G} \text{ with } |\mathcal{G}| = 2n\}$ . Each partition has the form  $\{\mathcal{G} \text{ with } \mathcal{G} \subset \mathcal{G}_{\max}\}$  and consists of  $(2n + 1)$  different sets anticommuting sets  $\mathcal{G} : |\mathcal{G}| = 2n$ .

Returning to the matter of Pauli-based fermion–qubit mappings, the  $2n$  Majorana representations of an  $n$ -mode Pauli-based mapping are anticommuting *Hermitian* Pauli operators. Hermitian Pauli operators have the form  $\pm P$  for unsigned Pauli operators  $P \in \tilde{\mathcal{P}}_n$ .

Remark 2.2.6 leads to a natural classification of Pauli-based mappings. Consider a Pauli-based fermion–qubit mapping  $\mathfrak{m}$  with Majorana representations  $\{\Gamma_i\}_{i=0}^{2n-1} \subset \mathcal{P}_n$ , which are Hermitian. Define the operators  $\tilde{\Gamma}_i = \pm \Gamma_i$  and  $\tilde{\Gamma}_{2n} = \pm (i)^n \prod_{i=0}^{2n-1} \Gamma_i$  with

signs such that  $\tilde{\Gamma}_i \in \tilde{\mathcal{P}}_n$  for all  $i \in \{0, 1, \dots, 2n\}$ . Then, the set

$$\tilde{\mathcal{G}}_{\max} := \{\tilde{\Gamma}_0, \tilde{\Gamma}_1, \dots, \tilde{\Gamma}_{2n-1}, \tilde{\Gamma}_{2n}\} \quad (2.29)$$

is the unique set of anticommuting, unsigned Pauli operators that are proportional to the Majorana representations  $\{\Gamma_i\}_{i=0}^{2n-1}$  of the mapping  $\mathfrak{m}$ . Figure 2.4 demonstrates this concept for the three-mode Jordan–Wigner and Bravyi–Kitaev transformations.

With one more result from [90], the results of this section provide a way to count the number of distinct Pauli–based mappings.

**Lemma 2.2.7.** The number of distinct maximal anticommuting subsets of unsigned Pauli operators  $\tilde{\mathcal{G}}_{\max} \subset \tilde{\mathcal{P}}_n$  of size  $k$  is

$$\left| \left\{ \tilde{\mathcal{G}}_{\max} : |\tilde{\mathcal{G}}_{\max}| = k \right\} \right| = \begin{cases} \frac{1}{k!} \left( \frac{4^n}{2^0} - 1 \right) \left( \frac{4^n}{2^1} \right) \left( \frac{4^n}{2^2} - 1 \right) \cdots \left( \frac{4^n}{2^{k-2}} \right), & k \text{ odd}, \\ 0 & k \text{ even}. \end{cases} \quad (2.30)$$

*Proof.* See Corollary 5.6 of [90].  $\square$

**Corollary 2.2.8.** The number of distinct anticommuting sets of unsigned Pauli operators  $\tilde{\mathcal{G}} \subset \tilde{\mathcal{P}}_n$  of size  $2n$  is

$$\left| \left\{ \tilde{\mathcal{G}} : |\tilde{\mathcal{G}}| = 2n \right\} \right| = (2n+1) \cdot \left| \left\{ \tilde{\mathcal{G}}_{\max} : |\tilde{\mathcal{G}}_{\max}| = 2n+1 \right\} \right| \quad (2.31)$$

$$= \frac{1}{(2n)!} \left( \frac{4^n}{2^0} - 1 \right) \left( \frac{4^n}{2^1} \right) \left( \frac{4^n}{2^2} - 1 \right) \cdots \left( \frac{4^n}{2^{2n-1}} \right). \quad (2.32)$$

*Proof.* This is a natural consequence of Remark 2.2.6 and Lemma 2.2.7.  $\square$

**Corollary 2.2.9.** The number of distinct ordered sets of anticommuting Hermitian Pauli operators  $(\Gamma_i)_{i=0}^{2n-1} \subset \mathcal{P}_n$  is

$$\left| \left\{ (\Gamma_i)_{i=0}^{2n-1} : \Gamma_i \text{ Hermitian} \right\} \right| = \underbrace{(2^{2n})}_{\pm \Gamma_i} \underbrace{(2n)!}_{\text{ordering}} \left| \left\{ \tilde{\mathcal{G}} : |\tilde{\mathcal{G}}_{\max}| = 2n \right\} \right| \quad (2.33)$$

$$= (2^{2n}) \left( \frac{4^n}{2^0} - 1 \right) \left( \frac{4^n}{2^1} \right) \left( \frac{4^n}{2^2} - 1 \right) \cdots \left( \frac{4^n}{2^{2n-1}} \right). \quad (2.34)$$

The quantity in Corollary 2.2.9 counts the number of distinct Pauli–based mappings up to a complex phase, since any such mapping is of the form  $\mathfrak{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ . For the first few numbers of qubits, the number of distinct Pauli–based mappings

trends as

$n$	1	2	3	4
$\left  \{(\Gamma_i)_{i=0}^{2n-1} : \Gamma_i \text{ Hermitian}\} \right $	12	480	129,024	336,907,468,800

(2.35)

which grows far too rapidly for any search routine to ever hope to investigate for useful mappings without reducing the search space. Chapter 3 introduces an equivalence relation on Pauli–based mappings that identifies labelling symmetries, which significantly reduce the effective search space.

### 2.2.2 Ternary tree transformations

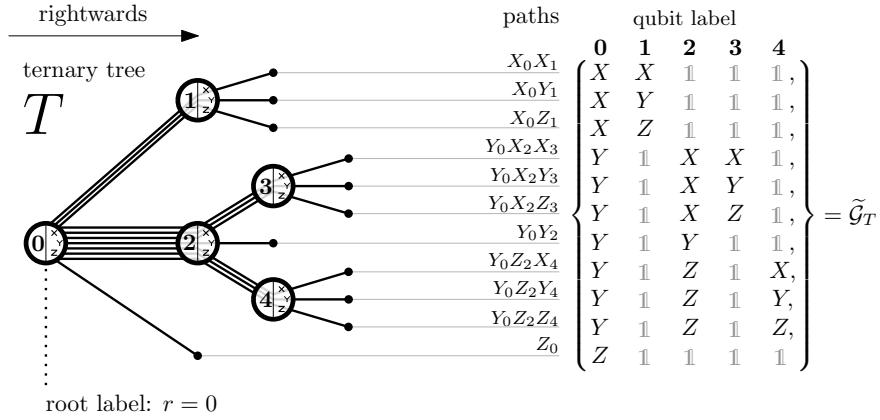
This section presents a refined and generalised definition of a special class of Pauli–based mapping, the ternary tree transformations [71–74]. Deriving from ternary tree graphs representing anticommuting Pauli operators, the original diagrammatic notation [71, 72] efficiently depicts Pauli operators with a branching tree–like structure, and describes the mapping with Majorana representations of the minimal average Pauli weight. Subsequent work [73, 74] expanded the definition to incorporate any input ternary tree graph and introduced an algorithm for a pairing scheme that ensured any ternary tree transformation would have a vacuum state of  $|0\rangle^{\otimes n}$ .

The Bonsai algorithm [73] exploits the structure of ternary tree transformations to reduce SWAP gate requirements of fermionic simulation on real–world quantum hardware. Treespilation [74] performs a computational search over a well–behaved subset of ternary tree transformations to minimise the CNOT cost for preparing qubit ansätze for molecular simulation, while another optimisation routine minimises the entanglement requirements of similar calculations [75].

**Definition 2.2.10.** (*Ternary trees*). An  $n$ –vertex ternary tree  $T$  is an  $n$ –vertex tree graph, where each vertex has a unique label in  $[n]$  and at most three children. Define the *root* as the vertex with no parent. We adopt the convention of orienting tree graphs such that the root is the leftmost vertex.

**Definition 2.2.11.** (*Ternary–tree–based set of unsigned, anticommuting Pauli operators* [71, 72].) Given an  $n$ –vertex ternary tree  $T$ , suppose the root of  $T$  has label  $r$ . We define the  *$T$ –based set of unsigned, anticommuting Pauli operators*  $\tilde{\mathcal{G}}_T$  via the following procedure.

1. Draw additional edges so that each labelled vertex has three children. The  $2n+1$  unlabelled vertices at the end of the additional edges form the *leaves* of the tree.



**Figure 2.5** A 5–vertex ternary tree  $T$  and the anticommuting set  $\tilde{\mathcal{G}}_T$  of unsigned Pauli operators.

2. For each labelled vertex of  $T$ , label its rightward edges  $X$ ,  $Y$  and  $Z$  from top-to-bottom. Associate each root-to-leaf path with the Pauli string that it spells out, in the same manner as in fermion–qubit mapping diagrams of Definition 3.1.1.

Let  $\tilde{\mathcal{G}}_T \subset \tilde{\mathcal{P}}_n$  be the set of unsigned Pauli operators that arise from each of the  $2n+1$  root-to-leaf paths. The elements of  $\tilde{\mathcal{G}}_T$  are anticommuting, because any two root-to-leaf paths diverge on a single vertex; the set  $\tilde{\mathcal{G}}_T$  is maximally anticommuting, because the maximum number of mutually anticommuting elements of  $\mathcal{P}_n$  is  $2n+1$  from Lemma 2.2.4.

**Example 2.2.12.** Figure 2.5 demonstrates a 5–vertex ternary tree  $T$  and the  $T$ –based set of unsigned, anticommuting of unsigned Pauli operators  $\tilde{\mathcal{G}}_T$ .

**Definition 2.2.13.** ( *$T$ -based mappings and ternary tree transformations.*) Let  $T$  be an  $n$ –vertex ternary tree and let  $\tilde{\mathcal{G}}_T$  be the  $T$ –based set of unsigned, anticommuting Pauli operators. Any Pauli–based mapping  $\mathfrak{m}$  of the form

$$\mathfrak{m} \sim ((\Gamma_0, \Gamma_1), (\Gamma_2, \Gamma_3), \dots, (\Gamma_{2n-2}, \Gamma_{2n-1})), \quad (2.36)$$

is a  *$T$ -based mapping*, where the Majorana representations  $\{\Gamma_i\}_{i=0}^{2n-1}$  consist of  $2n$  distinct, enumerated elements in  $\pm \tilde{\mathcal{G}}_T$ , i.e.  $\pm \Gamma_i \in \tilde{\mathcal{G}}_T$  for all  $i = 0, 1, \dots, 2n - 1$ . Collectively, for all ternary trees  $T$ , we call the set of  $T$ –based mappings the *ternary tree transformations*.

**Example 2.2.14.** (*Original proposition [71, 72].*) For any  $n \in \mathbb{N}$ , the complete  $n$ –vertex ternary tree  $T$  yields Pauli operators  $\tilde{\mathcal{G}}_T$  of equal weight  $\lceil \log_3(2n + 1) \rceil$  (see

Figure 3.9 for a modified example). The minimum value for the average Pauli weight of the Majorana representations of an  $n$ –mode Pauli–based mapping  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  is demonstrably  $\lceil \log_3(2n + 1) \rceil$  [72], so the complete ternary tee  $T$  yields Pauli–based mappings with the minimum average Pauli weight.

For an arbitrary choice of both the signs  $\pm \Gamma_i \in \tilde{\mathcal{G}}_T$  and the ordering and pairing scheme of the operators in  $((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ , there is no guarantee that the Fock basis of a  $T$ –based mapping consists of product states, let alone computational basis states. We describe existing and new methods to avoid this risk in Chapter 3. The need to avoid entanglement leads this chapter to the concept of product preservation and the Fock basis hierarchy of mappings.

## 2.3 Fock basis hierarchy of fermion–qubit mappings

This section begins by defining classical, affine and linear encodings of the Fock basis, a prominent hierarchy of fermion–qubit mappings storing fermionic occupation numbers in the bit-strings of the phaseless computational basis. Section 2.4 proves in Theorem 1 that affine encodings are the intersection of classical encodings and Pauli–based mappings, and Corollary 2.4.9 proves that affine encodings differ from linear encodings only in the signs of the Majorana representations.

Linear and affine encodings have appeared previously in the literature [67, 69, 70, 89], and this section expands the topic by introducing the explicit formula for the Majorana representatives of any affine encoding. This section lays the essential groundwork for unveiling the state–based definition of ternary tree transformations in Chapter 3.

**Definition 2.3.1.** (*Notation involving the computational basis.*) For a quantum state  $|\psi\rangle \in \mathcal{H}_2^{\otimes n}$ , we say that  $|\psi\rangle$  is *in the computational basis* if and only if  $|\psi\rangle \in \mathfrak{C}_n$ . For any  $\theta \in [0, 2\pi)$ , define

$$e^{i\theta}\mathfrak{C}_n := \{e^{i\theta}|0\rangle^{\otimes n}, e^{i\theta}|0\rangle^{\otimes(n-1)}|1\rangle, \dots, e^{i\theta}|1\rangle^{\otimes n}\}. \quad (2.37)$$

We use  $-\mathfrak{C}_n$  as shorthand to denote  $(-1)\mathfrak{C}_n$ ; similarly  $\pm\mathfrak{C}_n$  denotes  $\mathfrak{C}_n \cup (-\mathfrak{C}_n)$  and  $\pm i\mathfrak{C}_n$  denotes  $i\mathfrak{C}_n \cup (-i)\mathfrak{C}_n$ .

**Definition 2.3.2.** (*Fock basis hierarchy: product–preserving mappings; classical, affine and linear encodings of the Fock basis.*)

Let  $\mathbf{m}$  be an  $n$ –mode fermion–qubit mapping with Fock basis  $\{|\mathbf{f}_m\rangle \mid \mathbf{f} \in \mathbb{Z}_2^n\}$ .

- a) We say that  $\mathbf{m}$  is *product–preserving* if  $|\mathbf{f}_m\rangle$  is a product state for all  $\mathbf{f} \in \mathbb{Z}_2^n$  [73].

- b) We say that  $\mathbf{m}$  is a *classical encoding of the Fock basis*, or that  $\mathbf{m}$  *classically encodes the Fock basis*, if  $|\mathbf{f}_\mathbf{m}\rangle \in \mathfrak{C}_n$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ .
- c) We say that  $\mathbf{m}$  is an *affine encoding of the Fock basis*, or that  $\mathbf{m}$  *affinely encodes the Fock basis*, if  $|\mathbf{f}_\mathbf{m}\rangle = |G(\mathbf{f} \oplus \mathbf{b})\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ , where  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$  is some invertible binary matrix and  $\mathbf{b} \in \mathbb{Z}_2^n$  [69, 89].
- d) We say that  $\mathbf{m}$  is a *linear encoding of the Fock basis*, or that  $\mathbf{m}$  *linearly encodes the Fock basis*, if  $|\mathbf{f}_\mathbf{m}\rangle = |G\mathbf{f}\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ , where  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$  is some invertible binary matrix [64, 66, 67, 69, 70].

The phrase ‘classical encoding of the Fock basis’ is a new term we coin here to single out features of affine and linear encodings as opposed to the rest of the classical encodings. For example, the three-mode fermion–qubit mapping  $\mathbf{m}$  with  $U_\mathbf{m}$  equal to the Toffoli gate is a classical encoding, but it is not an affine encoding. Example 2.3.7 depicts an affine encoding that is not a linear encoding.

**Corollary 2.3.3.** (*Product-preserving Pauli-based fermion–qubit mappings.*) A Pauli-based fermion–qubit mapping  $\mathbf{m}$  is product preserving if and only if the vacuum state  $|\mathbf{0}_\mathbf{m}\rangle$  is a product state.

*Proof.* If  $|\mathbf{0}_\mathbf{m}\rangle$  is a product state, then for all  $\mathbf{f} \in \mathbb{Z}_2^n$ , the Fock basis vector  $|\mathbf{f}_\mathbf{m}\rangle = (\Gamma_0)^{f_0} \dots (\Gamma_{2n-2})^{f_{n-1}} |\mathbf{0}_\mathbf{m}\rangle$  must also be a product state as each of the Pauli operators  $\Gamma_i \in \mathcal{P}_n$  acts locally on individual qubits.  $\square$

**Example 2.3.4.** (*Fock basis of the Jordan–Wigner transformation.*) The  $n$ -mode Jordan–Wigner transformation is  $\mathbf{m}_{\mathrm{JW}} \sim ((\gamma_{2i}, \gamma_{2i+1}))_{i=0}^{n-1}$ , where the Pauli operators  $\gamma_i \in \mathcal{P}_n$  have the formula in Definition 2.1.4.

$$-i\gamma_{2i}\gamma_{2i+1} = -i \left( \left( \bigotimes_{k=0}^{i-1} Z_k \right) X_i \right) \left( \left( \bigotimes_{k=0}^{i-1} Z_k \right) Y_i \right) = Z_i. \quad (2.38)$$

The simultaneous 0-eigenstates of the vacuum stabilisers  $\{Z_i\}_{i=0}^{n-1}$  of  $\mathbf{m}_{\mathrm{JW}}$  are the states  $\{e^{i\phi} |0\rangle^{\otimes n} \mid \phi \in [0, 2\pi]\}$ . By convention, we choose  $|0\rangle^{\otimes n}$  to be  $|\mathbf{0}_{\mathbf{m}_{\mathrm{JW}}}\rangle$  without loss of generality. The Fock states of the Jordan–Wigner transformation are identical to the corresponding computational basis states:

$$|\mathbf{f}_{\mathbf{m}_{\mathrm{JW}}}\rangle = (\gamma_0)^{f_0}(\gamma_2)^{f_1} \dots (\gamma_{2n-2})^{f_{n-1}} |0\rangle^{\otimes n} \quad (2.39)$$

$$= (X_0)^{f_0}(Z_0 X_1)^{f_1} \dots (Z_0 Z_1 \dots Z_{n-2} X_{n-1})^{f_{n-1}} |0\rangle^{\otimes n} \quad (2.40)$$

$$= |\mathbf{f}\rangle, \quad (2.41)$$

where  $|\mathbf{f}\rangle = |f_0\rangle|f_1\rangle\dots|f_{n-1}\rangle$  is the computational basis state corresponding to the bit-string  $\mathbf{f}$ .

**Remark 2.3.5.** (*Fermionic mode labelling and the signs of the Fock basis.*) The only ordering of the product of  $(\gamma_{2i})^{f_i}$  operators in Equation 2.39 that yields  $|\mathbf{f}_{\mathbf{m}_{\text{JW}}}\rangle \in \mathfrak{C}_n$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$  is the existing ordering  $(\gamma_0)^{f_0} \dots (\gamma_{2n-2})^{f_{n-1}}$ . Consider a slightly altered Jordan–Wigner transformation  $\mathbf{m}'$ , with fermionic modes in a different order. In this case, there would be bit-strings  $\mathbf{f} \in \mathbb{Z}_2^n$  and indices  $i \in [n]$  for which the operator  $(\bigotimes_{k=0}^{i-1} Z_k)$  in the formula for  $|\mathbf{f}_{\mathbf{m}'}\rangle$  would act on an odd number of qubits in the state  $|1\rangle$ ; for these bit-strings, this would mean that  $|\mathbf{f}_{\mathbf{m}'}\rangle = -|\mathbf{f}\rangle \in -\mathfrak{C}_n$  and hence that  $\mathbf{m}'$  is not a classical encoding of the Fock basis.

It is a persistent result in the literature [91, 92] that unitary transformations of the form  $|\mathbf{f}\rangle \mapsto |\mathbf{G}(\mathbf{f} \oplus \mathbf{b})\rangle$  are in fact Clifford transformations, and so all affine encodings are Pauli–based mappings. While Example 2.3.6 and Figure 2.7 assume this knowledge, the purpose of Section 2.4 is to provide an explicit proof of the statement, and we go further to show that the distinction between the Majorana representations of affine and linear encodings is only in the signs of the operators.

**Example 2.3.6.** (*Linear encodings and Clifford operators.*) If  $\mathbf{m}$  is a linear encoding of the Fock basis with  $|\mathbf{f}_{\mathbf{m}}\rangle = |G\mathbf{f}\rangle$  for some invertible binary matrix  $G \in \text{GL}_n(\mathbb{Z}_2)$ , then the Clifford operator  $C_{\mathbf{m}}$  has the property  $C_{\mathbf{m}}|\mathbf{f}\rangle = |G\mathbf{f}\rangle$ . Three examples are:

- a) The Clifford transformation  $C_{\mathbf{m}_{\text{JW}}} = \mathbb{1}^{\otimes n}$  implements

$$\mathbb{1}^{\otimes n} : |\mathbf{f}\rangle \longmapsto |\mathbf{f}\rangle , \quad (2.42)$$

and corresponds to the Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}} \sim ((\gamma_{2i}, \gamma_{2i+1}))_{i=0}^{n-1}$  with  $|\mathbf{f}_{\mathbf{m}}\rangle = |\mathbf{f}\rangle$ .

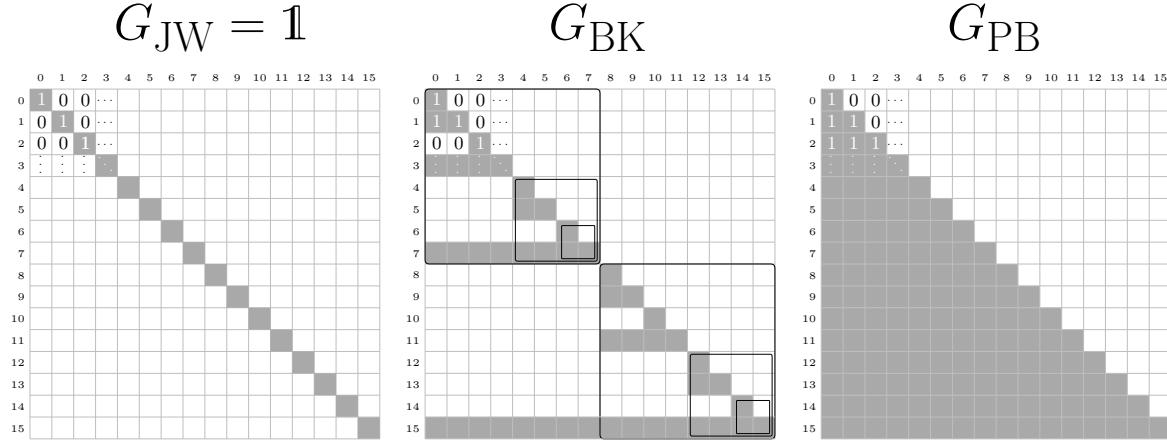
- b) If  $n$  is a power of 2, there is a unique Clifford transformation

$$C_{\mathbf{m}_{\text{BK}}} : |\mathbf{f}\rangle \longmapsto |G_{\text{BK}}\mathbf{f}\rangle , \quad (2.43)$$

where  $G_{\text{BK}}$  is a recursively-defined invertible binary matrix; see Figure 2.6 for an example when  $n = 16$ . The fermion–qubit mapping  $\mathbf{m}_{\text{BK}}$  with Fock basis vectors  $|\mathbf{f}_{\mathbf{m}_{\text{BK}}}\rangle = |G_{\text{BK}}\mathbf{f}\rangle$  has Majorana representations

$$\mathbf{m}_{\text{BK}} \sim ((C_{\mathbf{m}_{\text{BK}}}\gamma_{2i}C_{\mathbf{m}_{\text{BK}}}^\dagger, C_{\mathbf{m}_{\text{BK}}}\gamma_{2i+1}C_{\mathbf{m}_{\text{BK}}}))_{i=0}^{n-1} , \quad (2.44)$$

is exactly the Bravyi–Kitaev transformation [64] as it appears in [66].



**Figure 2.6** The invertible binary matrices  $G_{\text{JW}} = \mathbb{1}$ ,  $G_{\text{BK}}$  and  $G_{\text{PB}}$  that define the Fock bases of the Jordan–Wigner, Bravyi–Kitaev and parity basis transformations via  $|\mathbf{f}_m\rangle = |G\mathbf{f}\rangle$ , respectively, for  $n = 16$ . Shaded squares indicate entries that are equal to 1.

c) There is a Clifford transformation

$$C_{m_{\text{PB}}} : |\mathbf{f}\rangle \longmapsto |G_{\text{PB}}\mathbf{f}\rangle , \quad (2.45)$$

where  $G_{\text{PB}}$  is the matrix with lower–triangular entries equal to 1, and all other entries zero. The fermion–qubit mapping that corresponds to  $C_{m_{\text{PB}}}$  is exactly the parity basis transformation [66].

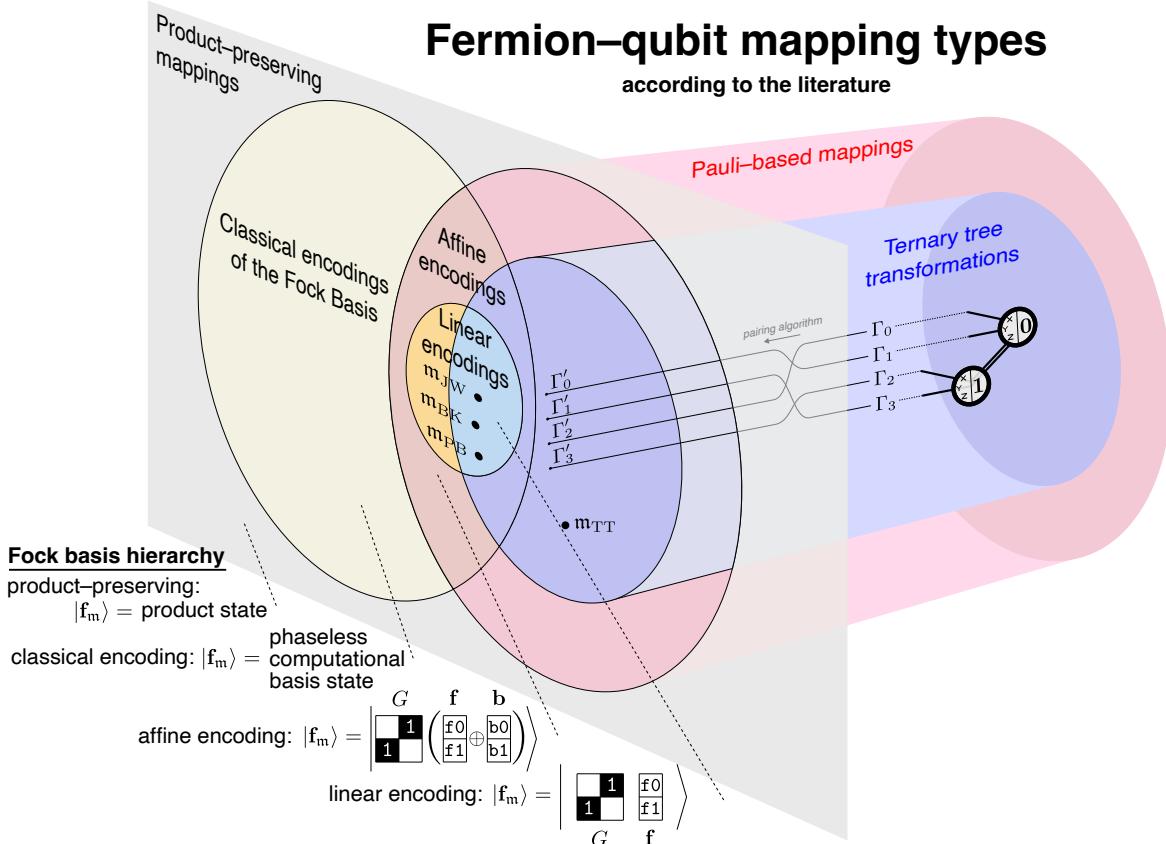
**Example 2.3.7.** (*An affine encoding of the Fock basis that is not linear.*) The two–mode fermion–qubit mapping

$$\mathbf{m} = ((\Gamma_0, \Gamma_1), (\Gamma_2, \Gamma_3)) = ((X_0, -Y_0), (-Z_0 X_1, -Z_0 Y_1)) \quad (2.46)$$

has vacuum stabilisers  $-i\Gamma_0\Gamma_1 = -Z_0$  and  $-i\Gamma_2\Gamma_3 = Z_1$ , with simultaneous 1–eigenstates  $\{e^{i\phi}|10\rangle \mid \phi \in [0, 2\pi)\}$ . By Definition 3.1.4, this mapping is equivalent to the two–mode Jordan–Wigner transformation: that is,  $\mathbf{m} \in [\mathbf{m}_{\text{JW}}]$ . Choosing  $|\mathbf{0}_m\rangle = |10\rangle$ , the Fock basis of  $\mathbf{m}$  is

$$|00_m\rangle = |10\rangle , \quad |10_m\rangle = |00\rangle , \quad |01_m\rangle = |11\rangle , \quad |11_m\rangle = |01\rangle . \quad (2.47)$$

Note that the relation  $|\mathbf{f}_m\rangle = |\mathbf{f} \oplus 10\rangle$  summarises Equation 2.47. Therefore  $\mathbf{m}$  is an affine encoding of the Fock basis, but it is not a linear encoding.



**Figure 2.7** State of fermion–qubit mapping literature, with an assortment of overlapping classes. (Figure 3.10 displays our simplified classification of fermion–qubit mappings from Chapter 3.) Classical, affine and linear encodings present as distinct classes of product–preserving mapping, while ternary tree transformations and other Pauli–based mappings may not even be product–preserving. Three well–known mappings – the Jordan–Wigner, Bravyi–Kitaev and parity basis transformations – are both linear encodings and ternary tree transformations. Theorem 1 proves that Pauli–based classical encodings of the Fock basis are affine encodings. Chapter 3 later establishes a relationship between between product–preserving ternary tree transformations and linear encodings.

## 2.4 Majorana representations of affine encodings

In Theorem 1, we prove that the intersection of classical encodings and Pauli–based mappings is equal to the set of all affine encodings. It is helpful to define the update, flip, parity and remainder sets of invertible binary matrices:

**Definition 2.4.1.** (*Generalised update, parity, flip and remainder sets from [69].*) Let  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ . For each  $i \in \{0, 1, \dots, n - 1\}$ , define the *generalised update*, *flip*, *parity* and *remainder* sets of  $i$  as follows:

1. The *update set of  $i$*  is  $U(i) = \{j \in \{0, 1, \dots, n - 1\} \mid G_{ji} = 1\}$ ; i.e. the set  $U(i)$  contains the indices of the rows of  $G$  that have non-zero elements in column  $i$ .
2. The *flip set of  $i$*  is  $F(i) = \{j \in \{0, 1, \dots, n - 1\} \mid (G^{-1})_{ik} = 1\}$ ; i.e. the set  $F(i)$  contains the indices of the columns of  $G^{-1}$  that have non-zero elements in row  $i$ .
3. The *parity set of  $i$*  is  $P(i) = \{j \in \{0, 1, \dots, n - 1\} \mid (\Pi G^{-1})_{ik} = 1\}$ , where  $\Pi$  is the lower-triangular matrix of all 1s:

$$\Pi = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 1 & 0 \end{pmatrix} \in \mathrm{M}_n(\mathbb{Z}_2); \quad (2.48)$$

i.e. the set  $P(i)$  contains the index of each column of  $G^{-1}$  for which the sum of the first  $(i-1)$  elements of that column is nonzero modulo 2. It is equivalent to define  $P(i) = F(0) \triangle F(1) \triangle \dots \triangle F(i-1)$ , where  $\triangle$  denotes the symmetric difference.

4. The *remainder set of  $i$*  is  $R(i) = F(i) \triangle P(i)$ .

**Lemma 2.4.2.** For any invertible binary matrix  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ , the update, parity, flip and remainder sets of Definition 2.4.1 satisfy

- a)  $|U(i) \cap F(i)|$  is odd for all  $i \in \{0, 1, \dots, n - 1\}$ ,
- b)  $|U(i) \cap P(i)|$  is even for all  $i \in \{0, 1, \dots, n - 1\}$ , and
- c)  $|U(i) \cap R(i)|$  is odd for all  $i \in \{0, 1, \dots, n - 1\}$ .

*Proof.* The relevant fact is that the rows of  $G^{-1}$  and the columns of  $G$  are orthonormal. Because  $G^{-1}G = \mathbb{1}$ , the matrix-vector product of the  $i$ th row of  $G^{-1}$  and the  $i$ th column of  $G$  must be  $1 \pmod{2}$ . Thus, there must be an odd number of shared elements of  $F(i)$  and  $U(i)$  for each  $i \in \{0, 1, \dots, n-1\}$ , proving a). Part b) follows from the orthonormality of the rows of  $G^{-1}$  and the columns of  $G$ : since the  $i$ th row of  $\Pi G^{-1}$  is equal to the sum of the first  $(i-1)$  rows of  $G^{-1}$ , each of which is orthogonal to the  $i$ th column of  $G$ , the  $i$ th row of  $\Pi G^{-1}$  must itself be orthogonal to the  $i$ th column of  $G$ . Therefore there must be an even number of shared elements of  $P(i)$  and  $U(i)$ . Part c) follows from parts a) and b), and the definition  $R(i) = F(i) \Delta P(i)$ .  $\square$

**Definition 2.4.3.** (*Symplectic notation for Pauli and Clifford operators [93].*)

- a) *Symplectic representation of Pauli operators.* Let  $P \in \tilde{\mathcal{P}}_n$  be an unsigned Pauli operator, and let  $P|_{\{i\}}$  denote the Pauli matrix acting on the  $i$ th qubit within the overall action of  $P$  (e.g.  $(X_0 Y_1)|_0 = X$ ). The *symplectic representation of  $P$*  is the  $2n$ -bit vector  $\mathbf{v}(P) \in \mathbb{F}_2^{2n}$  with coordinates

$$(v_i(P), v_{i+n}(P)) = \begin{cases} (0, 0) & \text{if } P|_{\{i\}} = \mathbb{1}, \\ (1, 0) & \text{if } P|_{\{i\}} = X, \\ (0, 1) & \text{if } P|_{\{i\}} = Z, \\ (1, 1) & \text{if } P|_{\{i\}} = Y \end{cases} \quad \text{for all } i \in [n]. \quad (2.49)$$

- b) *Symplectic inner product.* The *symplectic inner product* of two unsigned Pauli operators  $P, Q \in \tilde{\mathcal{P}}_n$  is the quantity

$$\mathbf{v}(P)^\top \begin{pmatrix} 0 & \mathbb{1}_n \\ \mathbb{1}_n & 0 \end{pmatrix} \mathbf{v}(Q). \quad (2.50)$$

For  $P, Q \in \tilde{\mathcal{P}}_n$ , the symplectic inner product is equal to

$$\mathbf{v}(P)^\top \begin{pmatrix} 0 & \mathbf{1}_n \\ \mathbf{1}_n & 0 \end{pmatrix} \mathbf{v}(Q) \quad (2.51)$$

$$= \left( \underbrace{\dots}_{\substack{1 \text{ iff } P|_{\{i\}}=Y \text{ or } Z}} \mid \underbrace{\dots}_{\substack{1 \text{ iff } P|_{\{i\}}=X \text{ or } Z}} \right) \begin{pmatrix} \vdots \\ \hline \vdots \end{pmatrix} \left\{ \begin{array}{l} 1 \text{ iff } Q|_{\{i\}}=X \text{ or } Y \\ 1 \text{ iff } Q|_{\{i\}}=Y \text{ or } Z \end{array} \right\} \quad (2.52)$$

$$= \left| \left\{ i \mid P|_{\{i\}} = (X \text{ or } Y) \text{ and } Q|_{\{i\}} = (Y \text{ or } Z) \right\} \right| + \left| \left\{ i \mid P|_{\{i\}} = (Y \text{ or } Z) \text{ and } Q|_{\{i\}} = (X \text{ or } Y) \right\} \right| \quad (2.53)$$

$$= \left| \left\{ i \mid P|_{\{i\}} = X \text{ and } Q|_{\{i\}} = Y \right\} \right| + \left| \left\{ i \mid P|_{\{i\}} = X \text{ and } Q|_{\{i\}} = Z \right\} \right| + \left| \left\{ i \mid P|_{\{i\}} = Y \text{ and } Q|_{\{i\}} = X \right\} \right| + \left| \left\{ i \mid P|_{\{i\}} = Y \text{ and } Q|_{\{i\}} = Z \right\} \right| \quad (2.54)$$

$$= \begin{cases} 1 & \{P, Q\} = 0 \\ 0 & [P, Q] = 0, \end{cases} \quad (2.55)$$

i.e. the symplectic inner product of  $P$  and  $Q$  is 1 if and only if  $P$  and  $Q$  anticommute.

- c) *Stabiliser tableaux of Clifford operators.* Let  $C \in \mathcal{C}_n$  be a Clifford operator. Then, there exists a vector  $\mathbf{b} \in \mathbb{Z}_2^{2n}$  and a set of unsigned Pauli operators  $\{P_i\}_{i=0}^{2n-1} \subset \tilde{\mathcal{P}}_n$  such that  $CX_iC^\dagger = (-1)^{b_i}P_i$  and  $CZ_iC^\dagger = (-1)^{b_{i+n}}P_{i+n}$ . The *stabiliser tableau* of  $C$  is the  $(2n) \times (2n+1)$  binary array

$$[C] = \left[ \begin{array}{ccc|ccc|c} \mathbf{v}(P_0) & \dots & \mathbf{v}(P_{n-1}) & \mathbf{v}(P_n) & \dots & \mathbf{v}(P_{2n-1}) & \mathbf{b} \end{array} \right], \quad (2.56)$$

where the notation in Equation 2.56 implies that the first column of  $[C]$  is  $\mathbf{v}(P_0)$ , and so on.

**Lemma 2.4.4.** Two Clifford operators with the same stabiliser tableau are identical up to a global phase.

*Proof.* This is simply a limitation of the symplectic representation for the Clifford group [91].  $\square$

Theorem 1 derives the stabiliser tableaux of affine encodings of the Fock basis. The proof integrates Lemmas 2.4.5–2.4.8.

**Theorem 1.** (*The stabiliser tableaux of affine encodings.*) For any  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$  and any  $\mathbf{b} \in \mathbb{Z}_2^n$ , the unitary transformation  $|\mathbf{f}\rangle \mapsto |G(\mathbf{f} + \mathbf{b})\rangle$  is a Clifford transformation  $C \in \mathcal{C}_n$ , where  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$  and  $\mathbf{b} \in \mathbb{Z}_2^n$ . The Clifford operator  $C$  has the following stabiliser tableau:

$$[C] = \left[ \begin{array}{c|ccc|c} & 0 & \dots & 0 & 0 \\ G & \vdots & \ddots & \vdots & \vdots \\ & 0 & \dots & 0 & 0 \\ \hline 0 & \dots & 0 & & \\ \vdots & \ddots & \vdots & (G^{-1})^\top & \mathbf{b} \\ 0 & \dots & 0 & & \end{array} \right]. \quad (2.57)$$

*Proof.* Theorem 1 follows from Lemmas 2.4.5–2.4.8. The supplementary materials of [92] have inspired this proof.

**Lemma 2.4.5.** A Clifford operator  $C \in \mathcal{C}_n$  that preserves the computational basis  $\mathfrak{C}_n$  maps single-qubit  $X$  and  $Z$  operators to Pauli strings consisting of only  $X$  or  $Z$  matrices, respectively. That is, the elements in the off-diagonal quadrants of the stabiliser tableau of  $C$  must be zero:

$$[C] = \left[ \begin{array}{c|ccc|c} & 0 & \dots & 0 & \\ \vdots & \ddots & \vdots & & \\ & 0 & \dots & 0 & \\ \hline 0 & \dots & 0 & & \\ \vdots & \ddots & \vdots & & \\ 0 & \dots & 0 & & \end{array} \right]. \quad (2.58)$$

Proof: Suppose that for some  $j \in [n]$ , the image of  $X_j$  under  $C$  is  $C(X_j) = \pm P$  where  $P \in \tilde{\mathcal{P}}_n$  is such that  $P|_{\{k\}} = Y$  or  $Z$ . For  $\mathbf{f} \in \mathbb{Z}_2^n$ ,

$$X_j |\mathbf{f}\rangle = |\mathbf{f} + \mathbf{1}_j\rangle \in \mathfrak{C}_n \implies C(X_j |\mathbf{f}\rangle) \in \mathfrak{C}_n, \quad (2.59)$$

as  $C$  preserves the computational basis. At the same time, since  $C$  preserves the computational basis, there is a permutation  $\sigma \in S_{2^n}$  of the  $n$ -bit strings such that

$$C(X_j |\mathbf{f}\rangle) = C(X_j)C |\mathbf{f}\rangle = \pm P |\sigma(\mathbf{f})\rangle, \quad (2.60)$$

and so  $\pm P |\sigma(\mathbf{f})\rangle \in \mathfrak{C}_n$ . However, this necessarily cannot hold for all bit-strings in  $\mathbb{Z}_2^n$ , as

$$P |\sigma(\mathbf{f})\rangle \in \left( \prod_{\substack{(\sigma(\mathbf{f}))_i=0 \\ P|_{\{i\}}=Y}} i \right) \left( \prod_{\substack{(\sigma(\mathbf{f}))_i=1 \\ P|_{\{i\}}=Y}} (-i) \right) \left( \prod_{\substack{(\sigma(\mathbf{f}))_i=1 \\ P|_{\{i\}}=Z}} (-1) \right) \mathfrak{C}_n . \quad (2.61)$$

Indeed, because  $C$  implements the permutation  $\sigma$  on the computational basis, there must exist a unique  $\mathbf{f}' \in \mathbb{Z}_2^n$  with  $(\sigma(\mathbf{f}'))_i = (\sigma(\mathbf{f}))_i$  for all  $i \neq k$ , and with  $(\sigma(\mathbf{f}'))_k \neq (\sigma(\mathbf{f}))_k$ . Regardless of whether  $P$  acts with a  $Y$  or  $Z$  on qubit  $k$ , the image of the state  $X_j |\mathbf{f}'\rangle$  under  $C$  is not in the computational basis:

$$C(X_j |\mathbf{f}'\rangle) = C(X_j)C |\mathbf{f}'\rangle = \pm P |\sigma(\mathbf{f}')\rangle \in -\mathfrak{C}_n , \quad (2.62)$$

because  $\sigma(\mathbf{f}')$  differs from  $\sigma(\mathbf{f})$  only on the  $k$ th bit. This contradicts the assumption that  $C$  preserves the computational basis. Therefore the images of single-qubit  $X$  operators under  $C$  must be Pauli strings consisting of  $X$  matrices or the identity, which is equivalent to stating that every entry in the bottom-left quadrant of  $[C]$  must be zero.

A similar argument follows for the top-right quadrant of  $[C]$ . Suppose that for some  $j \in [n]$ , the image of  $Z_j$  under  $C$  is  $C(Z_j) = \pm P$  where  $P \in \tilde{\mathcal{P}}_n$  is such that  $P|_{\{k\}} = X$  or  $Y$ . For  $\mathbf{f} \in \mathbb{Z}_2^n$ ,

$$Z_j |\mathbf{f}\rangle = (-1)^{f_j} |\mathbf{f}\rangle \implies C(Z_j |\mathbf{f}\rangle) = (-1)^{f_j} = (-1)^{f_j} C |\mathbf{f}\rangle = |\sigma(\mathbf{f})\rangle . \quad (2.63)$$

At the same time, because  $P$  contains at least one  $X$  or  $Y$  matrix,

$$C(Z_j |\mathbf{f}\rangle) = C(Z_j)C |\mathbf{f}\rangle = \pm P |\sigma(\mathbf{f})\rangle \neq |\sigma(\mathbf{f})\rangle , \quad (2.64)$$

which contradicts Equation 2.63. Therefore the images of single-qubit  $Z$  operators under  $C$  must be Pauli strings consisting of  $Z$  matrices or the identity, which is equivalent to stating that every entry in the top-right quadrant of  $[C]$  must be zero. ■

**Lemma 2.4.6.** The on-diagonal quadrants of the stabiliser tableau of a Clifford operator  $C \in \mathcal{C}_n$  that preserves the computational basis  $\mathfrak{C}_n$  are matrix inverse transposes of

each other. That is, for some  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ ,

$$[C] = \left[ \begin{array}{c|ccc} G & 0 & \dots & 0 \\ \hline 0 & \dots & 0 & \\ \vdots & \ddots & \vdots & \\ 0 & \dots & 0 & \end{array} \right] \quad (2.65)$$

$$\left[ \begin{array}{c|cc} & & \\ \hline & & \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \end{array} \right] \quad (2.65)$$

Proof: Note that Clifford operations preserve commutation relations: for all  $P, Q \in \mathcal{P}$ ,

$$\{P, Q\} = \{C(P)), C(Q)\}. \quad (2.66)$$

Define the Pauli operators

$$P_i := \begin{cases} X_i & 0 \leq i < n \\ Z_i & n \leq i < 2n. \end{cases} \quad (2.67)$$

Let  $G_1, G_2 \in \mathrm{GL}_n(\mathbb{Z}_2)$  be the on-diagonal entries of the stabiliser tableau of  $C$ , i.e.

$$[C] = \left[ \begin{array}{c|ccc} G_1 & 0 & \dots & 0 \\ \hline 0 & \dots & 0 & \\ \vdots & \ddots & \vdots & \\ 0 & \dots & 0 & \end{array} \right], \quad (2.68)$$

$$\left[ \begin{array}{c|cc} & & \\ \hline & & \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \end{array} \right] \quad (2.68)$$

so that  $\mathbf{v}(C(X_i))$  is equal to the  $i$ th column of  $G_1$  followed by  $n$  zeroes, and  $\mathbf{v}(C(Z_i))$  is equal to  $n$  zeroes followed by the  $i$ th column of  $G_2$ . Note that  $\{P_i, P_j\} = 0$  if and only if  $i = j \oplus n$ ; via Equation 2.66, conjugation by  $C$  preserves the anticommutation relations of the  $P_i$  operators, and so

$$\{C(P_i), C(P_j)\} = 0 \iff i = j \oplus n. \quad (2.69)$$

Note that Equation 2.69 implies that the Kronecker delta function  $\delta_{i,j \oplus n}$  is 1 if and only if  $C(P_i)$  and  $C(P_j)$  anticommute; thus, by Definition 2.4.3b), it is equal to the

symplectic inner product of  $C(P_i)$  and  $C(P_j)$ :

$$\mathbf{v}(C(P_i))^\top \begin{pmatrix} 0 & \mathbb{1}_n \\ \mathbb{1}_n & 0 \end{pmatrix} \mathbf{v}(C(P_j)) = \delta_{i,j \oplus n} \quad (2.70)$$

$$\begin{pmatrix} (G_1)^\top & 0 \\ 0 & (G_2)^\top \end{pmatrix} \begin{pmatrix} 0 & \mathbb{1}_n \\ \mathbb{1}_n & 0 \end{pmatrix} \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix} = \begin{pmatrix} 0 & \mathbb{1}_n \\ \mathbb{1}_n & 0 \end{pmatrix} \quad (2.71)$$

$$\begin{pmatrix} 0 & (G_1)^\top G_2 \\ (G_2)^\top G_1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mathbb{1}_n \\ \mathbb{1}_n & 0 \end{pmatrix} \quad (2.72)$$

$$G_2 = (G_1^{-1})^\top, \quad (2.73)$$

as required.  $\blacksquare$

**Lemma 2.4.7.** A Clifford  $C \in \mathcal{C}_n$  that preserves the computational basis also preserves the signs of single-qubit  $X$  operators. That is,

$$[C] = \left[ \begin{array}{c|cc|c} & \begin{matrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{matrix} & 0 \\ \hline G & & & \\ \hline \begin{matrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{matrix} & (G^{-1})^\top & \end{array} \right]. \quad (2.74)$$

Proof: Suppose  $C(X_i) = -P$  for some  $P \in \tilde{\mathcal{P}}_n$ . Now,

$$X_i |\mathbf{0}\rangle = |\mathbf{1}_i\rangle \in \mathfrak{C}_n \implies C(X_i |\mathbf{0}\rangle) = C(X_i)C |\mathbf{0}\rangle = -P |\sigma(\mathbf{0})\rangle \in -\mathfrak{C}_n, \quad (2.75)$$

which contradicts the assumption that  $C$  preserves the computational basis.  $\blacksquare$

**Lemma 2.4.8.** Let  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$  be an  $n \times n$  invertible binary matrix and let  $\mathbf{b} \in \mathbb{Z}_2^n$  be an  $n$ -bit binary string. The family of  $n$ -qubit unitary matrices in of the form  $\{e^{i\theta} U \mid \theta \in [0, 2\pi]\}$  where  $U : |\mathbf{f}\rangle \mapsto |G(\mathbf{f} + \mathbf{b})\rangle$  is equal to the family of Cliffords

$\{e^{i\theta}C \mid \theta \in [0, 2\pi)\}$ , which have the stabiliser tableau

$$[e^{i\theta}C] = \left[ \begin{array}{c|ccc|c} & 0 & \dots & 0 & 0 \\ G & \vdots & \ddots & \vdots & \vdots \\ & 0 & \dots & 0 & 0 \\ \hline 0 & \dots & 0 & & \\ \vdots & \ddots & \vdots & (G^{-1})^\top & \mathbf{b} \\ 0 & \dots & 0 & & \end{array} \right]. \quad (2.76)$$

Proof: Let  $C_k$  be a Clifford operator that implements  $C_k(X_i) = X_i$ ,  $C_k(Z_i) = (-1)^{\delta_{ik}} Z_i$  for all  $i \in [n]$ ; all such Clifford operators are identical up to a global phase. Note that  $C_k(Z_i |\mathbf{0}\rangle) = C_k |\mathbf{0}\rangle$ , while also

$$C_k(Z_i |\mathbf{0}\rangle) = C_k(Z_i)C_k |\mathbf{0}\rangle = (-1)^{\delta_{ik}} Z_i C_k |\mathbf{0}\rangle. \quad (2.77)$$

Therefore  $C_k |\mathbf{0}\rangle$  is a simultaneous  $(+1)$ -eigenvector of the  $n$  Pauli operators in the set  $\{(-1)^{\delta_{ik}} Z_i\}_{i=0}^{n-1}$ , and must be of the form  $C_k |\mathbf{0}\rangle = e^{i\phi} |\mathbf{1}_k\rangle$  for some  $\phi \in [0, 2\pi)$ ; fix the global phase by taking  $C_k |\mathbf{0}\rangle = |\mathbf{1}_k\rangle$ . Notice that

$$C_k |\mathbf{f}\rangle = C_k \left( \prod_{f_i=1} X_i |\mathbf{0}\rangle \right) = \prod_{f_i=1} (C_k(X_i)) C_k |\mathbf{0}\rangle = \prod_{f_i=1} X_i |\mathbf{1}_k\rangle = |\mathbf{f} + \mathbf{1}_k\rangle, \quad (2.78)$$

and so  $C_k$  is also the unitary that implements the map  $C_k : |\mathbf{f}\rangle \mapsto |\mathbf{f} + \mathbf{1}_k\rangle$ . Thus,

$$\prod_{b_k=1} C_k : |\mathbf{f}\rangle \mapsto |\mathbf{f} + \mathbf{b}\rangle. \quad (2.79)$$

The stabiliser tableau of the operator  $C_k$  reveals the tableau of the operator in Equation 2.79 via

$$[C_k] = \left[ \begin{array}{c|ccc|c} & 0 & \dots & 0 & 0 \\ \mathbb{1}_n & \vdots & \ddots & \vdots & \vdots \\ & 0 & \dots & 0 & 0 \\ \hline 0 & \dots & 0 & & \\ \vdots & \ddots & \vdots & \mathbb{1}_n & \mathbf{1}_k \\ 0 & \dots & 0 & & \end{array} \right] \Rightarrow \left[ \prod_{b_k=1} C_k \right] = \left[ \begin{array}{c|ccc|c} & 0 & \dots & 0 & 0 \\ \mathbb{1}_n & \vdots & \ddots & \vdots & \vdots \\ & 0 & \dots & 0 & 0 \\ \hline 0 & \dots & 0 & & \\ \vdots & \ddots & \vdots & \mathbb{1}_n & \mathbf{b} \\ 0 & \dots & 0 & & \end{array} \right]. \quad (2.80)$$

Consider the Clifford  $C_G$  that implements  $C_G(X_i) = X_{U(i)}$ ,  $C_G(Z_i) = Z_{F(i)}$  where  $U(i)$  and  $F(i)$  are the update and flip sets of  $G$  as in Definition 2.4.1. The operator  $C_G$  has stabiliser tableau

$$[C_G] = \left[ \begin{array}{c|ccc|c} G & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 \\ \hline 0 & \dots & 0 & & 0 \\ \vdots & \ddots & \vdots & (G^{-1})^\top & \vdots \\ 0 & \dots & 0 & & 0 \end{array} \right], \quad (2.81)$$

and so  $C_G$  preserves the computational basis, by Lemma 2.4.6. Note that  $C_G(Z_i |\mathbf{0}\rangle) = C_G |\mathbf{0}\rangle$ , while also

$$C_G(Z_i |\mathbf{0}\rangle) = C_G(Z_i)C_G |\mathbf{0}\rangle = Z_{F(i)}C |\mathbf{0}\rangle. \quad (2.82)$$

Therefore  $C_G |\mathbf{0}\rangle$  is a  $(+1)$ -eigenstate of  $Z_{F(i)}$  for all  $i \in [n]$ , which means that  $C_G |\mathbf{0}\rangle = |\mathbf{f}'\rangle$  where  $\mathbf{f}' \in \mathbb{Z}_2^n$  is such that  $\{i \mid f'_i = 1\} \cap F(i)$  is even. However, since  $G$  is invertible, the only vector  $\mathbf{f}'$  satisfying this requirement for all  $i \in [n]$  is  $\mathbf{0}$ , and hence  $C_G |\mathbf{0}\rangle = |\mathbf{0}\rangle$ . Using this, the image under  $C_G$  of an arbitrary vector  $|\mathbf{f}\rangle$  is

$$C_G |\mathbf{f}\rangle = C_G \left( \prod_{f_i=1} X_i |\mathbf{0}\rangle \right) = \prod_{f_i=1} C_G(X_{U(i)})C_G |\mathbf{0}\rangle = \prod_{f_i=1} X_{U(i)} |\mathbf{0}\rangle \quad (2.83)$$

$$= \prod_{\substack{i: f_i=1 \\ j: G_{ij}=1}} X_j |\mathbf{0}\rangle = \prod_{(G\mathbf{f})_i=1} X_i |\mathbf{0}\rangle = |G\mathbf{f}\rangle, \quad (2.84)$$

and therefore  $C_G$  implements the transformation

$$C_G : |\mathbf{f}\rangle \mapsto |G\mathbf{f}\rangle. \quad (2.85)$$

Combining Equations 2.79 and 2.85, we obtain

$$C_G \left( \prod_{b_k=1} C_k \right) : |\mathbf{f}\rangle \mapsto |G(\mathbf{f} + \mathbf{b})\rangle \implies C_G \left( \prod_{b_k=1} C_k \right) = C. \quad (2.86)$$

Comparing the stabiliser tableaux in Equations 2.80 and 2.81 reveals that

$$[C] = \left[ C_G \left( \prod_{b_k=1} C_k \right) \right] = \left[ \begin{array}{c|cc|c} G & \begin{matrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{matrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline \begin{matrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{matrix} & (G^{-1})^\top & \mathbf{b} \end{array} \right], \quad (2.87)$$

as required. ■

From Lemma 2.4.7, each Clifford that preserves the computational basis has a stabiliser tableau of the form in Equation 2.74, for some  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ . By Lemma 2.4.8, this is the tableau of the Clifford implementing the map  $|\mathbf{f}\rangle \mapsto |G(\mathbf{f} + \mathbf{b})\rangle$ , for some  $\mathbf{b} \in \mathbf{F}_2^n$ . This completes the proof. □

**Corollary 2.4.9.** Let  $G \in \mathrm{GL}_n(\mathbb{Z}_2)$  and  $\mathbf{b} \in \mathbb{Z}_2^n$ . The Majorana representations of the affine encoding of the Fock basis  $\mathfrak{m}$  with  $|\mathbf{f}_{\mathfrak{m}}\rangle = |G(\mathbf{f} + \mathbf{b})\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$  are

$$\Gamma_{2i} = (-1)^{\sum_{k=0}^{i-1} b_k} X_{U(i)} Z_{R(i)}, \quad (2.88)$$

$$\Gamma_{2i+1} = i(-1)^{\sum_{k=0}^i b_k} X_{U(i)} Z_{R(i)}, \quad (2.89)$$

where  $U(i)$ ,  $P(i)$  and  $R(i)$  are the update, parity and remainder sets of  $G$  as in Definition 2.4.1.

*Proof.* Let  $C_{\mathfrak{m}} \in \mathcal{C}_n$  be the Clifford operator implementing  $C_{\mathfrak{m}} : |\mathbf{f}\rangle \mapsto |G(\mathbf{f} + \mathbf{b})\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ . By Theorem 1, Equation 2.57 displays the stabiliser tableau of  $C_{\mathfrak{m}}$ , and therefore

$$C_{\mathfrak{m}}(Z_i) = C_{\mathfrak{m}} Z_i C_{\mathfrak{m}}^\dagger = (-1)^{b_i} Z_{F(i)} \quad (2.90)$$

$$C_{\mathfrak{m}}(X_i) = C_{\mathfrak{m}} X_i C_{\mathfrak{m}}^\dagger = X_{U(i)} \quad (2.91)$$

$$C_{\mathfrak{m}}(Y_i) = -i C_{\mathfrak{m}}(Z_i X_i) C_{\mathfrak{m}}^\dagger = (-i)(-1)^{b_i} Z_{F(i)} X_{U(i)}. \quad (2.92)$$

For Equation 2.88, observe

$$\Gamma_{2i} = C_{\mathfrak{m}} \gamma_{2i} C_{\mathfrak{m}}^\dagger = C_{\mathfrak{m}}(Z_0 Z_1 \dots Z_{i-1} X_i) \quad (2.93)$$

$$= C_{\mathfrak{m}}(Z_0) C_{\mathfrak{m}}(Z_1) \dots C_{\mathfrak{m}}(Z_{i-1}) C_{\mathfrak{m}}(X_i) \quad (2.94)$$

$$= (-1)^{\sum_{k=0}^{i-1} b_k} Z_{F(0)} Z_{F(1)} \dots Z_{F(i-1)} X_{U(i)} \quad (2.95)$$

$$= (-1)^{\sum_{k=0}^{i-1} b_k} Z_{P(i)} X_{U(i)} \quad (2.96)$$

$$= (-1)^{\sum_{k=0}^{i-1} b_k} X_{U(i)} Z_{P(i)}, \quad (2.97)$$

using Lemma 2.4.2 b) to rearrange the Paulis in the last line. Similarly, for Equation 2.89,

$$\Gamma_{2i+1} = C_{\mathfrak{m}} \gamma_{2i+1} C_{\mathfrak{m}}^\dagger = (-i) C_{\mathfrak{m}}(Z_0 Z_1 \dots Z_{i-1} Z_i X_i) \quad (2.98)$$

$$= (-i) C_{\mathfrak{m}}(Z_0) C_{\mathfrak{m}}(Z_1) \dots C_{\mathfrak{m}}(Z_i) C_{\mathfrak{m}}(X_i) \quad (2.99)$$

$$= (-i)(-1)^{\sum_{k=0}^{i-1} b_k} Z_{F(0)} Z_{F(1)} \dots Z_{F(i-1)} Z_{F(i)} X_{U(i)} \quad (2.100)$$

$$= (-i)(-1)^{\sum_{k=0}^{i-1} b_k} Z_{R(i)} X_{U(i)} \quad (2.101)$$

$$= i(-1)^{\sum_{k=0}^{i-1} b_k} X_{U(i)} Z_{R(i)}, \quad (2.102)$$

using Lemma 2.4.2 c) and  $R(i) = P(i) \triangle F(i)$ .  $\square$

## 2.5 Conclusion

The goal of this chapter was to provide a general definition for fermion–qubit mappings, and link disparate sections of the existing literature using this definition. Definition 2.1.7 achieves this goal, with only a small modification in Chapter 5 necessary to capture ancilla–qubit mappings, and Theorem 1 demonstrates the theory in action. Our definition identifies fermion–qubit mappings as unitary transformations between fermionic and qubit Hilbert spaces, and provides the link between the Majorana representations and encoded Fock states of a mapping. The Jordan–Wigner transformation acts as an anchor point between all fermion–qubit mappings.

Our definition allowed us to methodically cover the literature on Pauli–based mappings, which represent Majorana operators with Pauli matrices and includes the ternary tree transformations, and we described the sheer number of these mappings. The versatility of our definition for fermion–qubit mappings also allowed us to paint a picture of the Fock basis hierarchy of mappings with the same brush.

Theorem 1 proved that the intersection of Pauli-based mappings and classical encodings of the Fock basis – two large classes with historically different origins – is exactly the affine encodings. While this result persists in the literature, we go further to find the explicit Majorana representations of affine encodings, and also identify that the only separation between affine and linear encodings is the signs of the Majorana representations. These results will play into the proof of Theorem 2 in Chapter 3.

Figure 2.7 visualises the fermion–qubit mappings that have appeared throughout this chapter, summarising the various different definitions in the literature. Chapter 3 introduces a classification system for Pauli-based mappings that significantly simplifies this picture.



# Chapter 3

## Classifying fermion–qubit mappings

This chapter builds on the work of Chapter 2 to reveal a simple connection between two prominent classes of fermion–qubit mappings: ternary tree transformations and linear encodings of the Fock basis. Beyond the theoretical interest, this observation may be useful in unifying the emerging approaches to optimise ternary tree transformations [73–75] and linear encodings [68, 89] for practical simulation problems.

Introducing a diagrammatic language for Pauli operators, Section 3.1 establishes a practical notion of an equivalence relation for Pauli–based mappings. The symmetries of the diagrams factor out labelling choices, which represent decisions that the quantum engineer makes effectively arbitrarily at the user-end, such as the signs of individual Majorana representations. The equivalence classes of labelling symmetries sorts Pauli–based mappings into functionally distinct “templates”: we illustrate with the example of three–mode, product–preserving, Pauli–based mappings and identify fourteen templates, providing a much smaller database than the hundreds of thousands of technically distinct three–mode Pauli–based mappings.

Section 3.2 reviews ternary tree transformations to a rigorous standard using our classification system, revealing that the representation of a single Fock state completely determines the pairing scheme of Majorana representations in a ternary tree transformation. Our study builds to Theorem 2, which uses the equivalence relation in observing that product–preserving ternary tree transformations are equivalent to linear encodings of the Fock basis. One immediate observation from this result is that the complete ternary tree transformation is identical to the pruned Sierpinski tree transformation from the literature; more broadly, it confirms that any optimisation routine over linear encodings necessarily includes all product–preserving ternary tree transformations up to the equivalence relation of our classification.

### 3.1 An equivalence relation for Pauli–based mappings

In Section 3.1.1, we introduce a diagrammatic notation for sets of Pauli operators with a practical application of classifying Pauli–based mappings for optimisation routines. The symmetries of our notation factor out the Clifford operations of qubit and fermionic mode relabelling, braids within Pauli operator pairs, sign change of Pauli operators and local Pauli basis orientation, which all represent arbitrary labelling choices by the users of quantum computing hardware. To demonstrate our classification system, Section 3.1.2 displays all 14 equivalence classes of three–mode, product–preserving Pauli–based mappings.

#### 3.1.1 Labelling symmetries of Pauli strings

This section introduces diagrammatic notation for Pauli–based mappings, which extends the language of ternary tree transformations from [71, 73, 74]. Taking the symmetry group of the diagrammatic notation establishes a robust definition of equivalence between Pauli–based mappings. Our definition identifies redundancies in the sign of Pauli operators and their ordering within pairs, the labels of the single–qubit Pauli operators themselves, and the enumeration of fermionic modes and qubits. The resulting equivalence classes that strip these redundancies away become ‘templates’ for different fermion–qubit mapping types.

**Definition 3.1.1.** (*Diagrammatic notation for Pauli–based mappings.*) Let  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  be a Pauli–based mapping. The *diagram* of  $\mathbf{m}$  results from following the instructions (see Figures 2.3, 2.4, 3.1 for examples):

1. *Qubit labels:* draw a circle with label  $i \in [n]$  for each of the  $n$  qubits.
2. *Local basis labels:* inscribe the labels  $X, Y, Z$  from top to bottom in each circle.
- **Representation of Pauli operator pairs as strings:** for each Majorana representation  $\Gamma_i$ , draw the *string representing*  $\Gamma_i$  as a curve that passes from left-to-right such that it intersects and departs leftwards at the level of the label  $X, Y$  or  $Z$  from the circle with label  $j$  according to the local action of  $\Gamma_i$  on qubit  $j$ . The string representing  $\Gamma_i$  does not intersect circles with the labels of qubits that lie outside the support of  $\Gamma_i$ .
3. *Pauli ordering within pairs:* For each pair  $(\Gamma_{2i}, \Gamma_{2i+1})$ , draw a directed arrow from the end of the string representing  $\Gamma_{2i}$  to the end of the string representing  $\Gamma_{2i+1}$ ,

4. *Pauli signs*: if the Pauli operator  $\Gamma_i$  is negatively signed, i.e. if  $-\Gamma_i \in \tilde{\mathcal{P}}_n$ , indicate this by writing  $(-1)$  at the leftmost end of the string representing  $\Gamma_i$ .
5. *Fermionic labels*: mark the arrow joining the ends of the lines representing  $\Gamma_{2i}$  and  $\Gamma_{2i+1}$  with the label  $i$ .

The five numbered steps of this process correspond to labelling tasks, which are necessary to identify  $\mathbf{m}$  uniquely from its diagram.

The diagrammatic notation of Definition 3.1.1 is a generalisation of the ternary tree diagram notation from [71–74], which is limited in the sense that it can only describe mappings whose diagrams have an underlying tree-like structure. Our diagrammatic notation need not conform to this restriction, and can describe any Pauli-based mapping.

**Example 3.1.2.** (*Examples of fermion–qubit mapping diagrams.*) Each of the mappings  $\mathbf{m}_{\text{JW}}$ ,  $\mathbf{m}_{\text{BK}}$ ,  $\mathbf{m}_1$ – $\mathbf{m}_6$  in Figure 2.3 have an accompanying diagram. Note that the diagrams of the mappings  $\mathbf{m}_1$ – $\mathbf{m}_5$  all have a similar structure to the diagram for  $\mathbf{m}_{\text{JW}}$ , where the mapping  $\mathbf{m}_i$  differs from  $\mathbf{m}_{\text{JW}}$  only in the  $i$ th labelling step of Definition 3.1.1.

Example 3.1.2 points out that two distinct fermion–qubit mappings with Pauli representations of the Majorana operators may have identical diagrams if one ignores the labelling steps in Definition 3.1.1, which we will build into a notion of an equivalence relation on the set of all Pauli-based mappings.

We argue that the relative multiplication relations between Pauli representations of the Majorana operators are the only factors in Pauli-based mappings that are relevant to cost functions in quantum computation. This is because the only other factors – the labelling steps of Definition 3.1.1 – correspond to choices that a user could make essentially mentally at the start of a simulation algorithm.

Differences between mappings that are more substantial than the labelling steps can nontrivially alter the multiplication relations of the Pauli representations of the Majorana operators, which would affect the time and depth of subsequent quantum computation. For example, swapping any one Majorana representation  $\Gamma_j$  in a Pauli-based mapping with the operator  $\Gamma'_j = \prod_{i=0}^{2n-1} \Gamma_i$  yields a mapping that will likely produce qubit Hamiltonians of different costs, if the two operators  $\Gamma_j$  and  $\Gamma'_j$  have different Pauli weights.

To make this definition of equivalence explicit, we illustrate the set of mappings equivalent to the two-mode Jordan–Wigner transformation in Definition 3.1.3.

**Definition 3.1.3.** (*Template of the two-mode Jordan–Wigner transformation.*) The set of Pauli–based mappings with diagrams identical to the diagram of the two-qubit Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}} \sim ((X_0, Y_0), (Z_0 X_1, Z_0 Y_1))$  up to labelling is the *template*

$$[\mathbf{m}_{\text{JW}}] = \left\{ \begin{array}{c} \text{Pauli ordering within pairs} \\ \underbrace{\left\{ \overbrace{\{A_i, B_i\}}, \overbrace{\{C_i A'_j, C_i B'_j\}} \right\}}_{\text{Fermionic labels}} \end{array} \mid \begin{array}{c} A \neq B \neq C, A' \neq B' \in \pm\{X, Y, Z\}, \quad i \neq j \in \{0, 1\} \\ \text{Local Pauli basis labels, Pauli signs} \end{array} \right\} \quad (3.1)$$

where the annotations in Equation 3.1 indicate how the mappings in the template may differ from  $\mathbf{m}_{\text{JW}}$  by the five labelling steps in Definition 3.1.1 symbolically and with the use of unordered pairs and lists. In Figure 2.3, mappings  $\mathbf{m}_1$ – $\mathbf{m}_5$  are all members of the Jordan–Wigner template  $[\mathbf{m}_{\text{JW}}]$ .

Definition 3.1.4 formalises the notion of label–invariant templates to the set of all  $n$ –mode Pauli–based mappings, which corresponds to an equivalence relation.

**Definition 3.1.4.** (*Templates and equivalence of Pauli–based mappings.*) Let  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  be an  $n$ –mode Pauli–based mapping. The *template of  $\mathbf{m}$*  is a set of Pauli–based mappings  $[\mathbf{m}]$ , where  $\mathbf{m}' \in [\mathbf{m}]$  if and only if  $\mathbf{m}' \sim ((\Gamma'_{2i}, \Gamma'_{2i+1}))_{i=0}^{n-1}$  where there is a Clifford operator of the form  $C = \prod_k C_k \in \mathcal{C}_n$  that satisfies  $\Gamma'_i = C \Gamma_i C^\dagger$  for all  $i \in [2n]$ , and each operator  $C_k$  is one of the following Cliffords:

1. *Qubit swap:* a permutation  $\sigma \in S_n$  of the qubit labels (i.e.  $C_k : A_i \mapsto A_{\sigma(i)}$  for  $A \in \{X, Y, Z\}$  and  $i \in [n]$ ),
2. *Local basis change:* a re-orientation of the local Pauli basis of a qubit (i.e.  $C_k : (X_i, Y_i, Z_i) \mapsto (Y_i, Z_i, X_i)$ , or  $C_k : (X_i, Y_i, Z_i) \mapsto (Z_i, X_i, Y_i)$ , or  $C_k : (X_i, Y_i, Z_i) \mapsto (Y_i, X_i, -Z_i)$  and so on, for some  $i \in [n]$ ),
3. *Pauli pair braid:* a braid of a Majorana pair (i.e.  $C_k : (\Gamma_{2i}, \Gamma_{2i+1}) \mapsto (\pm \Gamma_{2i+1}, \mp \Gamma_{2i})$  for some  $i \in [n]$ ),
4. *Pauli sign change:* the change in sign of a Pauli operator (i.e.  $C_k : \Gamma_i \mapsto (-1)^{\gamma_i} \Gamma_i$  for some  $\gamma_i \in \{0, 1\}$  for each  $i \in [n]$ ),
5. *Fermionic swap:* a permutation  $\rho \in S_n$  of the fermionic mode labels (i.e.  $C_k : (\Gamma_{2i}, \Gamma_{2i+1}) \mapsto (\Gamma_{2\rho(i)}, \Gamma_{2\rho(i)+1})$  for all  $i \in [n]$ ),

We say that two Pauli-based mappings  $\mathbf{m}_1$  and  $\mathbf{m}_2$  are *equivalent* if and only if  $[\mathbf{m}_1] = [\mathbf{m}_2]$ .

With the exception of the local basis change and Pauli pair braid operations, the five types of Clifford operation in Definition 3.1.4 correspond to the five labelling steps in Definition 3.1.3. We choose to use the physically-motivated local basis change and Pauli pair braid symmetries to generate the equivalence classes of Pauli-based mappings rather than the more abstract relabelling of single-qubit Pauli matrices and reordering operators within pairs. The end result is the same because of the freedom to change the signs of Pauli operators.

For example, the sequence of the Pauli braid  $\exp\left(\frac{\pm\pi}{4}\Gamma_{2i}\Gamma_{2i+1}\right)$  implementing  $(\Gamma_{2i}, \Gamma_{2i+1}) \mapsto (-\Gamma_{2i+1}, \Gamma_{2i})$  [94], followed by the Pauli sign change  $\Gamma_{2i+1} \mapsto -\Gamma_{2i+1}$  reproduces the pair reordering symmetry  $(\Gamma_{2i}, \Gamma_{2i+1}) \mapsto (\Gamma_{2i+1}, \Gamma_{2i})$  of fermion-qubit mapping diagrams:

$$(\Gamma_{2i}, \Gamma_{2i+1}) \xrightarrow{\text{Pauli pair braid}} (-\Gamma_{2i+1}, \Gamma_{2i}) \xrightarrow{\text{Pauli sign change}} (\Gamma_{2i+1}, \Gamma_{2i}). \quad (3.2)$$

This decision is due to notational convenience in Sections 3.2. Similarly, combining the local basis change  $(X_i, Y_i, Z_i) \mapsto (Y_i, X_i, -Z_i)$  with sign changes of any Pauli operators involving  $Z_i$  will have the effect of relabelling the Pauli operators on qubit  $i$  according to  $(X_i, Y_i, Z_i) \mapsto (Y_i, X_i, Z_i)$ .

**Example 3.1.5.** (*Two-mode Pauli-based mappings.*) In addition to the Jordan–Wigner template  $[\mathbf{m}_{\text{JW}}]$  from Definition 3.1.3, there are two other templates for two-mode Pauli-based mappings. One, the Bravyi–Kitaev template, is of the form

$$[\mathbf{m}_{\text{BK}}] = \left\{ \begin{array}{l} \left\{ \{A_i, B_i A'_i\}, \{B_i B'_j, B_i C'_j\} \right\} \mid \\ A \neq B, A' \neq B' \neq C' \in \pm\{X, Y, Z\}, \quad i \neq j \in \{0, 1\} \end{array} \right\}, \quad (3.3)$$

where  $\mathbf{m}_{\text{BK}} \sim ((X_0, Y_0 Z_1), (-Y_0 Y_1, Y_0 X_1))$  is the two-qubit Bravyi–Kitaev transformation as Figure 2.3 displays. The other template, which Figure 2.3 also depicts, is the equivalence class of the mapping  $\mathbf{m}_6 \sim ((X_0, -Z_0 Y_1), (Z_0 X_1, Y_0))$ ,

$$[\mathbf{m}_6] = \left\{ \begin{array}{l} \left\{ \{A_i, B_i A'_j\}, \{C_i, B_i B'_j\} \right\} \mid \\ A \neq B \neq C, A' \neq B' \in \pm\{X, Y, Z\}, \quad i \neq j \in \{0, 1\} \end{array} \right\}, \quad (3.4)$$

which we call the ‘product-breaking template’ because the vacuum state of any mapping in  $[\mathfrak{m}_6]$  is entangled.

Corollary 3.1.6 makes the observation that affine and linear encodings of the Fock basis are equivalent, using results from Section 2.4.

**Corollary 3.1.6.** (*Affine encodings are equivalent to linear encodings.*) From Corollary 2.4.9, all affine encodings are equivalent to linear encodings, with the only difference being the signs of some of the Pauli operators.

### 3.1.2 Three-mode product-preserving Pauli-based mappings

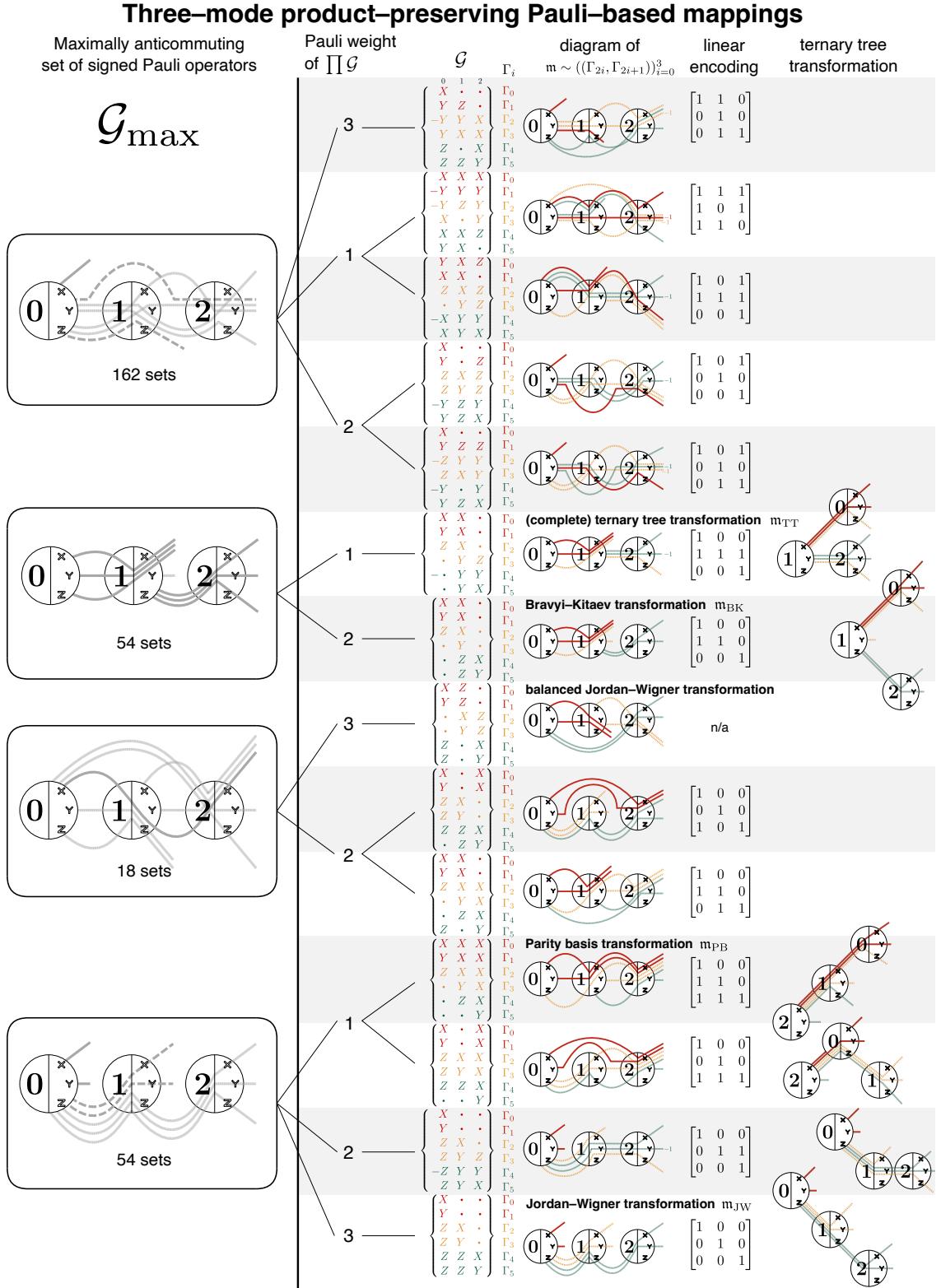
Figure 3.1 displays all 14 templates for three-mode, product-preserving, Pauli-based mappings. With only three qubits, the diagram immediately reveals that the templates of the three-mode linear encodings contain the templates of all three-mode ternary tree transformations; in Section 3.2, we prove that this is true for any number of modes. The figure also reveals the existence of a product-preserving, Pauli-based mapping that is not even a linear encoding at all.

From left-to-right, the structure of the diagram follows Remark 2.2.6, starting with the diagram of a maximally anticommuting set  $\mathcal{G}_{\max}$  of 7 Pauli operators. The set of Majorana representations  $\mathcal{G}$  for each mapping is the result of removing an element  $\Pi \mathcal{G}$  from the maximally anticommuting set  $\mathcal{G}_{\max}$ .

To check that we have not missed any mappings, Lemma 2.2.7 computes 288 distinct maximally anticommuting sets of unsigned three-qubit Pauli operators  $\tilde{\mathcal{G}}_{\max}$  with size 7. Taking all of the different Pauli, fermionic and qubit labelling combinations of the diagrams for  $\mathcal{G}_{\max}$  in Figure 3.1 into account, without considering the signs of the Pauli operators, the total number of maximally anticommuting sets is  $162 + 54 + 18 + 54 = 288$ .

Example 3.1.5 demonstrated that the only distinct classes of two-mode product-preserving Pauli-based mappings were the Jordan–Wigner and Bravyi–Kitaev templates. In Figure 3.1 the templates of the three-mode Jordan–Wigner, Bravyi–Kitaev, parity basis and complete ternary tree transformations emerge as distinct classes.

The mapping in Figure 3.1 with the label of ‘balanced Jordan–Wigner transformation’ first appeared in the appendices of [73] to demonstrate that not every mapping is a ternary tree transformation. We point out that this is the *only* three-mode Pauli-based mapping in which all Majorana representations have Pauli weight less than 3. For this reason, we conjecture that the balanced Jordan–Wigner transformation, which generalises to a  $(2n + 1)$ -mode mapping in which every Majorana representation is



**Figure 3.1** The fourteen templates of product-preserving fermion-qubit mappings. Thirteen templates contain linear encodings, a further six of which contain ternary tree transformations.

a Pauli operator of weight  $n+1$ , is an excellent candidate for low-weight fermionic simulation algorithms.

## 3.2 Product–preserving ternary tree transformations are equivalent to linear encodings

One of the advantages of linear encodings of the Fock basis is their well-behaved parameterisation in terms of invertible binary matrices, resulting in convenient databases for algorithmic searching [67, 89]. The introduction of ternary tree transformations showcase their potential by providing minimum-weight Pauli representations of the Majorana operators [71, 72]. However, the Fock states of these mappings might require entanglement to initialise on a quantum register, unlike the linear encodings of the Fock basis such as the Jordan–Wigner and Bravyi–Kitaev transformations.

A subsequent demonstration that every ternary tree  $T$  can yield a product–preserving  $T$ –based mapping in [73] removed the threat of entangled Fock basis states by defining a class of ternary–tree transformations with vacuum state  $|0\rangle^{\otimes n}$ . In Section 3.2.1, we generalise this result by algorithmically showing that it is always possible to pair the Pauli operators so that the vacuum state of a  $T$ –based mapping can be any product stabiliser state, establishing a broader class of product–preserving ternary tree transformations.

Despite this progress, the full relationship between ternary tree transformations and linear encodings has remained unclear. Product–preserving ternary tree transformations with vacuum states of  $|0\rangle^{\otimes n}$  are not necessarily linear encodings of the Fock basis, and can employ Fock states in both  $\pm\mathfrak{C}_n$  and  $\pm i\mathfrak{C}_n$ . A demonstration of how to apply Pauli braids to produce  $T$ –based mappings with Fock states in  $\pm\mathfrak{C}_n$  appeared in [73]. Like our study of all three–mode, product–preserving, Pauli–based mappings in Section 3.1.2, this suggests that ternary tree transformations are close in definition to linear encodings.

In this section, we combine our unified notation for fermion–qubit mappings from Chapter 2 with our notion for equivalence from Section 3.1 to show that every product–preserving ternary tree transformation is indeed equivalent to a linear encoding of the Fock basis. We detail a procedure to recover the invertible binary matrix of the linear encoding for any ternary tree input, illustrating that linear encoding templates contain ternary tree transformations within the Fock basis hierarchy of product–preserving mappings. Thus each product–preserving ternary tree transformation is equivalent to a linear encoding.

### 3.2.1 Uniqueness of product-preserving ternary tree transformations

Ternary tree transformations introduced an operator-based definition for fermion–qubit mappings [71, 72], and subsequent research devised an algorithm to pair the Pauli Majorana representations in such a way to produce mappings with vacuum states equal to  $|0\rangle^{\otimes n}$  [73]. While this method ensures a product-preserving mapping for each ternary tree, it does not explore the uniqueness of the Pauli operator pairing. In Lemma 3.2.2, we present a more general algorithm which can produce a mapping from any  $n$ -vertex ternary tree  $T$  with any  $n$ -qubit product stabiliser state as its vacuum, not just  $|0\rangle^{\otimes n}$ . Our algorithm lists all  $T$ -based mappings that have same vacuum state, and shows that they are equivalent up to the symmetries of Pauli braids and fermionic relabelling.

**Example 3.2.1.** (*Examples of product-preserving and product-breaking mappings.*)

- a) The mappings  $\mathfrak{m}_{\text{JW}}$ ,  $\mathfrak{m}_{\text{BK}}$  and  $\mathfrak{m}_1$ — $\mathfrak{m}_5$  in Figure 2.3 are product-preserving, and the mapping  $\mathfrak{m}_6$  is product-breaking.
- b) The ternary tree transformation  $\mathfrak{m}$  in Figure 3.4 (a) is a product-preserving mapping which results from applying the pairing algorithm from [73] to the ternary tree  $T$  from Example 2.2.12.

Example 3.2.1 b) made use of the Pauli operator pairing algorithm from the literature [73], which, for any ternary tree  $T$  yields a  $T$ -based mapping with vacuum state  $|0\rangle^{\otimes n}$ . While this algorithm demonstrates the existence of product-preserving ternary tree transformations, for the sake of showing equivalence between ternary tree transformations and linear encodings of the Fock basis, it is essential to find all  $T$ -based mappings with product stabiliser states as their vacuum states.

Lemma 3.2.2 generalises the pairing algorithm of [73] to produce a  $T$ -based mapping with any product vacuum state, for any ternary tree  $T$ . It also shows that all  $T$ -based mappings to satisfy this property are equivalent to the same mapping, with only the fermionic labelling, pair braiding, and Pauli sign change differentiating them. Section 3.2.2 makes use of the uniqueness in proving the equivalence of product-preserving ternary tree transformations to linear encodings of the Fock basis.

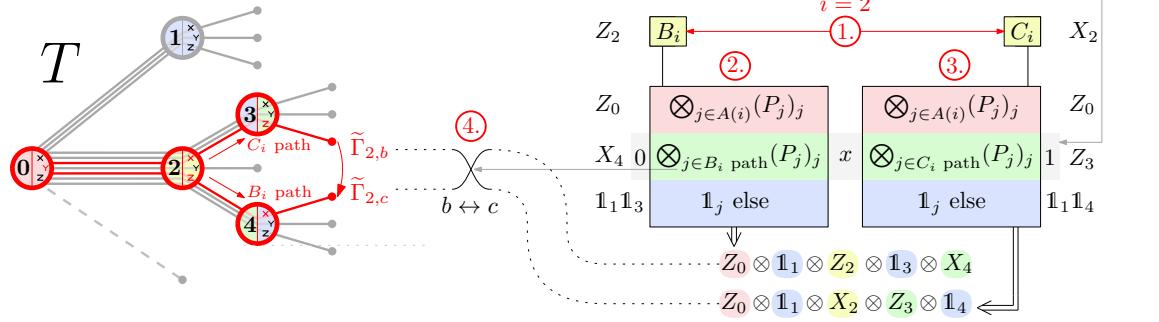
**Lemma 3.2.2.** (*Uniqueness of product-preserving ternary tree transformations.*) Let  $T$  be an  $n$ -vertex ternary tree. Then, for any product stabiliser state  $\bigotimes_{i=0}^{n-1} |a_i\rangle_i$ , where each  $|a_i\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |+i\rangle, |-i\rangle\}$  is a  $\pm 1$ -eigenstate of a Pauli matrix  $P_i \in \{Z, X, Y\}$ , the following holds:

Illustration of the pairing algorithm in **Lemma 3.2.2**

Obtain  $(\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c})$  where

- $\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c} \in \tilde{\mathcal{G}}_T$ , and

- $-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c}|0_m\rangle = |0_m\rangle$  for product state  $|0_m\rangle = \bigotimes_{j=0}^{n-1} |a_j\rangle_j$



**Figure 3.2** Demonstration of the pairing algorithm in Lemma 3.2.2 for a 5–vertex ternary tree  $T$ . The algorithm identifies the pairing of the elements of  $\tilde{\mathcal{G}}_T$  that *must* comprise the Majorana representations of any  $T$ –based mapping with the vacuum state  $|0\rangle|1\rangle|+i\rangle|1\rangle|+\rangle$ .

- There is a unique pairing  $\{(\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c})\}_{i=0}^{n-1}$  of  $2n$  elements in  $\tilde{\mathcal{G}}_T$  such that a fermion–qubit mapping  $\mathbf{m}$  with vacuum state  $|0_m\rangle = \bigotimes_{i=0}^{n-1} |a_i\rangle_i$  has the form  $\mathbf{m} \sim ((\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c}))_{i=0}^{n-1}$ . In fact, any  $T$ –based mapping  $\mathbf{m}$  with unsigned Pauli Majorana representations  $\Gamma_i \in \tilde{\mathcal{G}}_T$  and with vacuum state  $|0_m\rangle = \bigotimes_{i=0}^{n-1} |a_i\rangle_i$  must be of the form  $\mathbf{m} \sim ((\tilde{\Gamma}_{\sigma(i),b}, \tilde{\Gamma}_{\sigma(i),c}))_{i=0}^{n-1}$  where the permutation  $\sigma \in S_n$  is a fermionic labelling scheme.
- The only  $T$ –based mappings with vacuum state  $\bigotimes_{i=0}^{n-1} |a_i\rangle_i$  have the form  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  where, for some fermionic labelling scheme  $\sigma \in S_n$ , the Pauli operator pairs are

$$(\Gamma_{2i}, \Gamma_{2i+1}) = \begin{cases} (\pm \tilde{\Gamma}_{\sigma(i),b}, \pm \tilde{\Gamma}_{\sigma(i),c}), & \text{as in part a), up to a global sign, or} \\ (\pm \tilde{\Gamma}_{\sigma(i),c}, \mp \tilde{\Gamma}_{\sigma(i),b}), & \text{a braid of the } i\text{th mode from part a).} \end{cases} \quad (3.5)$$

*Proof.* For part a), the task is to find two operators  $\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c} \in \tilde{\mathcal{G}}_T$  such that

$$-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c} \left( \bigotimes_{j=0}^{n-1} |a_j\rangle_j \right) = \bigotimes_{j=0}^{n-1} |a_j\rangle_j \quad \text{for all } i \in [n]. \quad (3.6)$$

The following four-step algorithm identifies such a pair of operators.

Let  $i \in [n]$ , and recall that the operators in  $\tilde{\mathcal{G}}_T$  correspond to root-to-leaf paths in  $T$ . We will construct a pair of operators  $(\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c})$  satisfying Equation 3.6 whose corresponding root-to-leaf paths diverge on the vertex of  $T$  with label  $i$ . Note that any two paths diverging on vertex  $i$  will share vertex  $i$  and all of its ancestors in  $T$ , which we denote by  $A(i)$ , and that the paths share no other vertices.

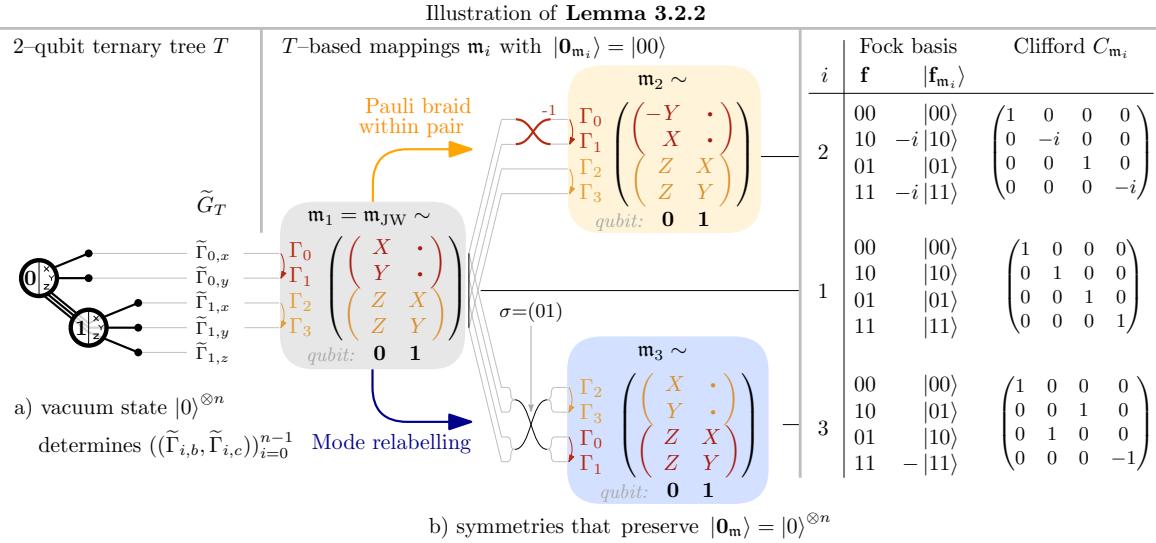
The first step of our algorithm identifies how the operators  $\tilde{\Gamma}_{i,b}$  and  $\tilde{\Gamma}_{i,c}$  must act on the  $i$ th qubit, should they exist. A necessary condition for Equation 3.6 to hold is for  $-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c}$  to locally preserve the eigenstate  $|a_i\rangle_i$ . Since the paths diverge on the  $i$ th vertex, the operators must act with different single-qubit Paulis:

- 1 Let  $B, C \in \{X, Y, Z\}$  be the single-qubit Pauli operators that satisfy  $-iBC|a_i\rangle = |a_i\rangle$ . Then, the operator  $\tilde{\Gamma}_{i,b}$  must act with  $B$  on the  $i$ th qubit, and the operator  $\tilde{\Gamma}_{i,c}$  with  $C$ .

The second and third steps of our algorithm show the existence of operators  $\tilde{\Gamma}_{i,b}$  and  $\tilde{\Gamma}_{i,c}$  that satisfy step 1 while also satisfying another necessary condition of Equation 3.6: that the product state  $\bigotimes_{j=0}^{n-1}|a_j\rangle_j$  is a  $(\pm 1)$ -eigenstate of  $-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c}$ .

- 2 Consider the unique root-to-leaf path in  $T$  that leaves the vertex with label  $i$  via its rightward edge with label  $B$ , and then passes through each subsequent vertex  $j$  via the rightward edge with label  $P_j$  until terminating at an unlabelled leaf vertex. Define the  $B_i$  path to be the segment of this root-to-leaf path to the right of vertex  $i$ , and define  $\tilde{\Gamma}_{i,b}$  to be the unsigned Pauli string that arises from this path via the construction in Definition 2.2.11.
- 3 Similarly, define  $\tilde{\Gamma}_{i,c}$  to be the Pauli string in  $\tilde{\mathcal{G}}_T$  that matches the unique root-to-leaf path in  $T$  that leaves vertex  $i$  via its rightward edge with label  $C$  and passes through each subsequent vertex  $j$  via the rightward edge with label  $P_j$ . Define the  $C_i$  path to be the segment of this root-to-leaf path to the right of vertex  $i$ .

The operator  $-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c}$  acts trivially on all qubits with labels in  $A(i)$ , as  $\tilde{\Gamma}_{i,b}$  and  $\tilde{\Gamma}_{i,c}$  act identically upon them. Steps 2 and 3 ensure that  $\bigotimes_{j=0}^{n-1}|a_j\rangle_j$  is a  $(\pm 1)$ -eigenstate



**Figure 3.3** A two-vertex ternary tree  $T$  and the Jordan–Wigner transformation  $m_1 = m_{\text{JW}}$ , which is a  $T$ -based mapping with vacuum state  $|00\rangle$ . The mappings  $m_2, m_3 \in [m_1]$  also have vacuum state  $|00\rangle$ . Lemma 3.2.2 stipulates that any  $T$ -based mapping with vacuum state  $|00\rangle$  differs from  $m_1$  only via Pauli pair braids and fermionic relabelling.

of  $-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c}$  via

$$-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c} \left( \bigotimes_{j=0}^{n-1} |a_j\rangle_j \right) = -i \left( \bigotimes_{j \in (B_i \text{ path})} (P_j)_j \right) B_i \left( \bigotimes_{j \in (C_i \text{ path})} (P_j)_j \right) C_i \left( \bigotimes_{j=0}^{n-1} |a_j\rangle_j \right) \quad (3.7)$$

$$= \left( \bigotimes_{j \in (B_i \text{ path}) \cup (C_i \text{ path})} (P_j |a_j\rangle)_j \right) (-iBC |a_i\rangle)_i \left( \bigotimes_{j \text{ else}} |a_j\rangle \right) \quad (3.8)$$

$$= (-1)^x \bigotimes_{j=0}^{n-1} |a_j\rangle_j, \quad (3.9)$$

where  $x$  is the number of Pauli  $(-1)$ -eigenstates in  $\{|a_j\rangle \mid j \in (B_i \text{ path}) \cup (C_i \text{ path})\}$ , i.e. the number of vertices in the  $B_i$  and  $C_i$  paths with labels  $j$  that satisfy  $P_j |a_j\rangle = -|a_j\rangle$ . The algorithm thus requires a corrective step:

- 4 If  $x$ , the number of Pauli  $(-1)$ -eigenstates in  $\{|a_j\rangle \mid j \in (B_i \text{ path}) \cup (C_i \text{ path})\}$ , is odd, exchange the symbols  $b$  and  $c$ .

Steps 1–4 identify a pair of operators  $\tilde{\Gamma}_{i,b}$  and  $\tilde{\Gamma}_{i,c}$  satisfying Equation 3.6, which correspond to two unique root-to-leaf paths of  $T$ . The uniqueness of these paths reflects the stronger statement that these are the only Pauli operators to anticommute on qubit  $i$  and also act with either  $P_j$  or  $\mathbb{1}_j$  on all qubits  $j \notin A(i)$ . This guarantees the

existence and uniqueness of the pair of operators  $(\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c})$  for each  $i \in [n]$ . Thus the set  $\{(\tilde{\Gamma}_{j,b}, \tilde{\Gamma}_{j,c})\}_{j=0}^{n-1}$  is the *only* set of Pauli operator pairs satisfying Equation 3.6 in which the operators of the  $i$ th pair anticommute on the  $i$ th qubit.

The assumption at the outset of this proof may seem omit the possibility of operator pair sets with a different structure, in which at least two pairs  $(\tilde{\Gamma}_1, \tilde{\Gamma}_2), (\tilde{\Gamma}_3, \tilde{\Gamma}_4)$  are such that  $\tilde{\Gamma}_1$  anticommutes with  $\tilde{\Gamma}_2$  on qubit  $i$ , and  $\tilde{\Gamma}_3$  and  $\tilde{\Gamma}_4$  also anticommute on qubit  $i$ . But because there are only three unsigned, anticommuting single-qubit Pauli matrices  $\{X, Y, Z\}$ , there would be elements from distinct pairs – say  $\tilde{\Gamma}_1$  and  $\tilde{\Gamma}_3$  – which could not anticommute on qubit  $i$  but would instead anticommute one of its children, say qubit  $i'$ . In this scenario, it is impossible for the product state  $\bigotimes_{j=0}^{n-1} |a_j\rangle_j$  to be an eigenstate of both  $-i\tilde{\Gamma}_1\tilde{\Gamma}_2$  and  $-i\tilde{\Gamma}_3\tilde{\Gamma}_4$ , because the two products act with different single-qubit Pauli operators on qubit  $i'$ .

Therefore, there is a single degree of freedom remaining in identifying a sequence of operator pairs  $\tilde{\mathcal{G}}_T$  that satisfy Equation 3.6: the order of the pairs. For any  $\sigma \in S_n$ , a  $T$ -based fermion–qubit mapping of the form  $\mathbf{m} \sim ((\tilde{\Gamma}_{\sigma(i),b}, \tilde{\Gamma}_{\sigma(i),c}))_{i=0}^{n-1}$  has vacuum state  $|\mathbf{0}_m\rangle = \bigotimes_{i=0}^{n-1} |a_i\rangle_i$ , because any permutation of the index  $i$  in  $-i\tilde{\Gamma}_{i,b}\tilde{\Gamma}_{i,c}$  upholds Equation 3.6.

Part b) relaxes the search for  $T$ -based mappings with vacuum state  $\bigotimes_{i=0}^{n-1} |a_i\rangle_i$  to require only Hermitian Pauli Majorana representations, i.e.  $\pm\Gamma_i \in \tilde{\mathcal{G}}_T$ ; the task thus amounts to investigating how to interchange and sign operators from part a) while preserving the vacuum state. Clearly the pair  $(-\tilde{\Gamma}_{\sigma(i),b}, -\tilde{\Gamma}_{\sigma(i),c})$  satisfies Equation 3.6, and due to the anticommutation  $\{\tilde{\Gamma}_{i,b}, \tilde{\Gamma}_{i,c}\} = 0$ , so do the Pauli braids of the operator pair:

$$-i(\pm\tilde{\Gamma}_{i,c})(\mp\tilde{\Gamma}_{i,b})\left(\bigotimes_{j=0}^{n-1} |a_j\rangle_j\right) = \left(\bigotimes_{j=0}^{n-1} |a_j\rangle_j\right). \quad (3.10)$$

Therefore, for any  $\sigma \in S_n$ , a mapping of the form  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  with pairs

$$(\Gamma_{2i}, \Gamma_{2i+1}) = \begin{cases} (\pm\tilde{\Gamma}_{\sigma(i),b}, \pm\tilde{\Gamma}_{\sigma(i),c}) & \text{or} \\ (\pm\tilde{\Gamma}_{\sigma(i),c}, \mp\tilde{\Gamma}_{\sigma(i),b}) \end{cases} \quad (3.11)$$

is a  $T$ -based mapping with vacuum state  $|\mathbf{0}_m\rangle = \bigotimes_{i=0}^{n-1} |a_i\rangle_i$ . These modifications of the pairs in mappings from part a) collectively describe every  $T$ -based mapping with vacuum state  $\bigotimes_{i=0}^{n-1} |a_i\rangle_i$ .  $\square$

**Example 3.2.3.** (*Demonstration of Lemma 3.2.2*). Let  $T$  be the two-vertex ternary tree in Figure 3.3. The  $T$ -based set of unsigned, anticommuting Pauli operators

is  $\tilde{\mathcal{G}}_T = \{X_0, Y_0, Z_0 X_1, Z_0 Y_1, Z_0 Z_3\}$ . Lemma 3.2.2 a) identifies the Jordan–Wigner transformation  $\mathbf{m} = ((X_0, Y_0), (Z_0 X_1, Z_0 Y_1))$  as a  $T$ –based transformation with vacuum state  $|00\rangle$  and unsigned Pauli Majorana representations; the only other  $T$ –based mappings with vacuum state  $|00\rangle$  are equivalent to  $\mathbf{m}_1$ . Two examples of equivalent  $T$ –based mappings with vacuum state  $|00\rangle$  are  $\mathbf{m}_2$  and  $\mathbf{m}_3$ , which Figure 3.3 also depicts.

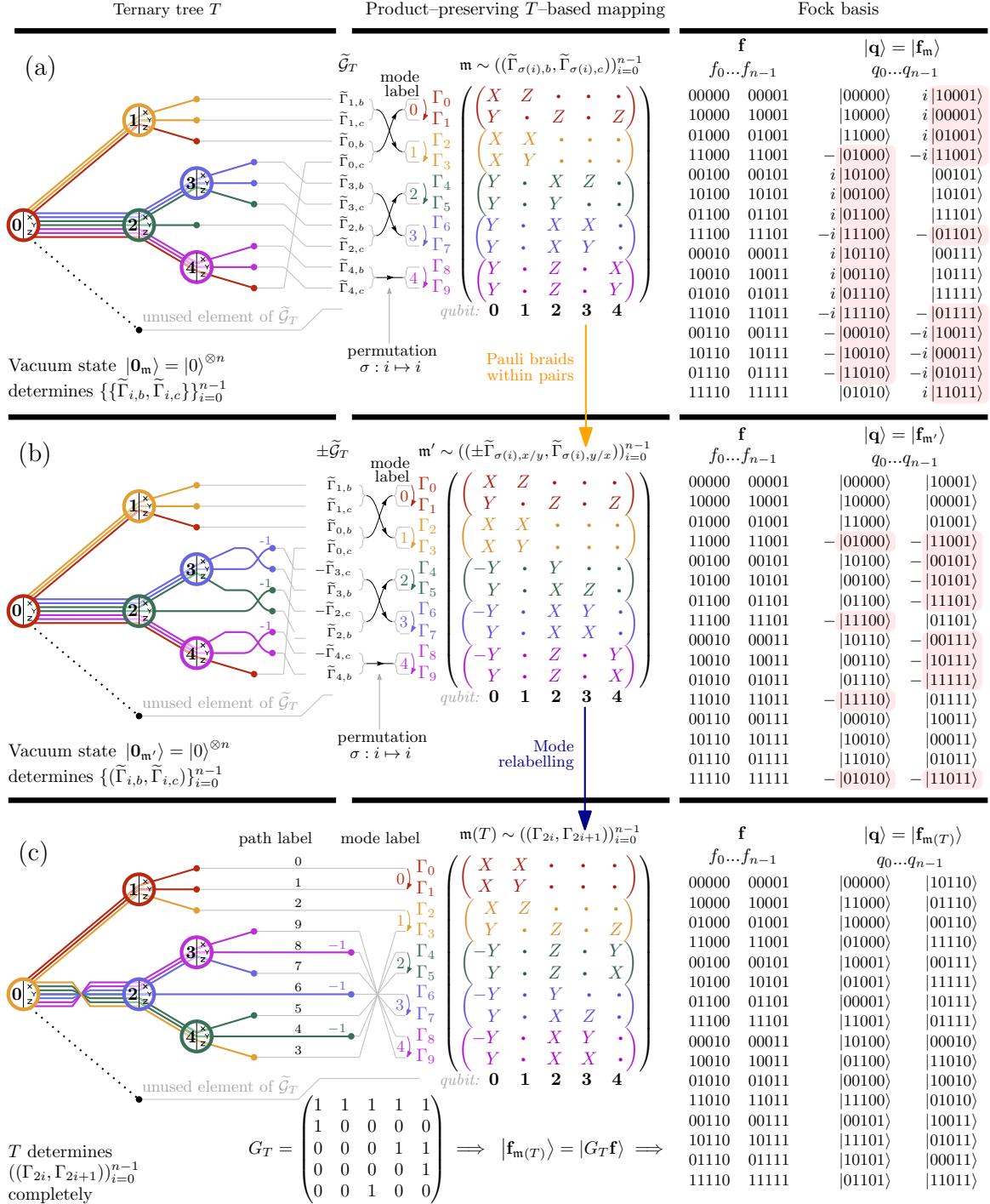
**Example 3.2.4.** (*Pauli operator pairing algorithm from [73] in the language of Lemma 3.2.2.*) Applying the algorithm for generating a product–preserving ternary–tree–based mapping from [73] to an  $n$ –vertex ternary tree  $T$  generates the unique  $T$ –based mapping of the form  $\mathbf{m} \sim ((\tilde{\Gamma}_{\sigma(i),b}, \tilde{\Gamma}_{\sigma(i),c}))_{i=0}^{n-1}$  with vacuum state  $|\mathbf{0}_m\rangle = |0\rangle^{\otimes n}$ , for some  $\sigma \in S_n$ . Figure 3.4 (a) demonstrates the mapping of this kind for the 5–vertex ternary tree from Example 2.2.12, with  $\sigma$  equal to the trivial permutation. Note that the Fock states of this mapping are computational basis states with complex coefficients, i.e.  $|\mathbf{f}_m\rangle \in \pm \mathfrak{C} \cup \pm i\mathfrak{C}$ .

The authors of [73] and its related work [74] suggest a refinement of the algorithm from 3.2.4. Example 3.2.5 details the modification, which braids Pauli operator pairs to produce mappings with Fock states in the strictly real computational basis.

**Example 3.2.5.** (*Pauli operator pairing algorithm from [73, 74].*) Let  $T$  be an  $n$ –vertex ternary tree and, for some  $\sigma \in S_n$ , consider the  $T$ –based mapping  $\mathbf{m}$  with vacuum state  $|\mathbf{0}_m\rangle = |0\rangle^{\otimes n}$  of the form  $\mathbf{m} \sim ((\tilde{\Gamma}_{\sigma(i),b}, \tilde{\Gamma}_{\sigma(i),c}))_{i=0}^{n-1}$ , which arises from the prescription in Example 3.2.4. By definition, for each  $i \in \{0, 1, \dots, n-1\}$ , the Pauli operators in the pair  $(\Gamma_{2i}, \Gamma_{2i+1}) = (\tilde{\Gamma}_{\sigma(i),b}, \tilde{\Gamma}_{\sigma(i),c})$  act with  $X$  and  $Y$  on the qubit with label  $\sigma(i)$ , and with  $Z$  operators on all qubits in the  $B_{\sigma(i)}$  and  $C_{\sigma(i)}$  paths, using notation from the proof of Lemma 3.2.2. Therefore, the pair consists of an operator that acts with a  $Y$  matrix on an even number of qubits, and an operator that acts with a  $Y$  matrix on an odd number of qubits. If  $\tilde{\Gamma}_{\sigma(i),b}$  acts with an odd number of  $Y$  operators, define a new mapping  $\mathbf{m}' \sim ((\Gamma'_{2i}, \Gamma'_{2i+1}))_{i=0}^{n-1}$  with operator pairs:

$$\begin{cases} \Gamma'_{2i} = \tilde{\Gamma}_{\sigma(i),b}, & \Gamma'_{2i+1} = \tilde{\Gamma}_{\sigma(i),c} \quad \text{if the number of } Y\text{s in } \tilde{\Gamma}_{\sigma(i),b} \text{ is even,} \\ \Gamma'_{2i} = -\tilde{\Gamma}_{\sigma(i),c}, & \Gamma'_{2i+1} = \tilde{\Gamma}_{\sigma(i),b} \quad \text{if the number of } Y\text{s in } \tilde{\Gamma}_{\sigma(i),b} \text{ is odd.} \end{cases} \quad (3.12)$$

Because  $\mathbf{m}'$  differs from  $\mathbf{m}$  only by Majorana braids, the vacuum state of  $\mathbf{m}'$  is  $|\mathbf{0}_{m'}\rangle = |\mathbf{0}_m\rangle = |0\rangle^{\otimes n}$ . Equation 3.12 braids the pair  $(\Gamma_{2i}, \Gamma_{2i+1}) \mapsto (-\Gamma_{2i+1}, \Gamma_{2i}) = (\Gamma'_{2i}, \Gamma'_{2i+1})$  from  $\mathbf{m}$  if and only if  $\Gamma_{2i}$  acts with an odd number of  $Y$  operators, which ensures that real computational bases  $\pm \mathfrak{C}_n$  contain the Fock basis  $\{|\mathbf{f}_{m'}\rangle \mid \mathbf{f} \in \mathbb{F}_2^n\}$  since  $|\mathbf{f}_m\rangle = \Gamma_0^{f_0} \Gamma_2^{f_1} \dots \Gamma_{2n-2}^{f_{n-1}} |0\rangle^{\otimes n}$ . Figure 3.4 (b) demonstrates the mapping  $\mathbf{m}'$  arising from this modification of the mapping  $\mathbf{m}$  from Example 3.2.4.



**Figure 3.4** (a) The first prescription in [73] for ternary-tree-based mappings produces a mapping  $m$  with vacuum state  $|0\rangle^{\otimes n}$  and a complex Fock basis. (b) The second prescription in [73] produces a mapping  $m'$  with vacuum state  $|0\rangle^{\otimes n}$  and a real Fock basis. (c) Our prescription produces the unique  $T$ -based mapping  $m(T)$  that linearly encodes the Fock basis.

### 3.2.2 Outline for proof of Theorem 2

Section 3.2.3 contains crucial results that build towards Theorem 2:

- **Lemma 3.2.6** shows that for any product stabiliser state  $\bigotimes_{i=0}^{n-1} |a'_i\rangle_i$ , and any  $T$ -based mapping  $\mathbf{m}$  with vacuum state  $|\mathbf{0}_\mathbf{m}\rangle = \bigotimes_{i=0}^{n-1} |a_i\rangle_i$ , there is a  $T'$ -based mapping  $\mathbf{m}'$  with vacuum state  $|\mathbf{0}_{\mathbf{m}'}\rangle$  that differs from  $\mathbf{m}$  only by locally relabelling Pauli operators, where  $T'$  only differs from  $T$  by edge relabelling.
- **Lemma 3.2.7** establishes that there is a unique Clifford operator  $C_T \in \mathcal{C}_n$  determining a  $T$ -based mapping  $\mathbf{m}(T) = ((C_T \gamma_{2i} C_T^\dagger, C_T \gamma_{2i+1} C_T^\dagger))_{i=0}^{n-1}$  with the properties that
  - 1)  $\mathbf{m}(T)$  is a  $T$ -based mapping (i.e.  $C_T \gamma_i C_T^\dagger \in \pm \tilde{\mathcal{G}}_T$  for all  $i \in [2n]$ ), and
  - 2)  $\mathbf{m}(T)$  is a classical encoding of the Fock basis (i.e.  $|\mathbf{f}_{\mathbf{m}(T)}\rangle \in \mathfrak{C}_n$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ ) with  $|\mathbf{0}_{\mathbf{m}(T)}\rangle = |0\rangle^{\otimes n}$ .

Theorem 1 thus implies that  $\mathbf{m}(T)$  is a linear encoding of the Fock basis.

- **Lemma 3.2.8** finds the invertible binary matrix  $G_T \in \mathrm{GL}_n(\mathbb{F}_2)$  such that  $|\mathbf{f}_{\mathbf{m}(T)}\rangle = |G_T \mathbf{f}\rangle$ .

Section 3.2.4 presents the main result:

- **Theorem 2:**
  - a) For each ternary tree  $T$ , there is a unique  $T$ -based mapping  $\mathbf{m}(T)$  that is also a linear encoding of the Fock basis.
  - b) Every  $n$ -mode product-preserving ternary tree transformation is equivalent to a mapping in the set  $\{\mathbf{m}(T) \mid T \text{ is an } n\text{-vertex ternary tree}\}$ .

In Section 3.2.5, we apply Theorem 2 to the complete ternary tree graph. In doing so, we recover the pruned Sierpinski tree transformation [2], demonstrating its equivalence to complete ternary tree transformation.

### 3.2.3 Supporting results for Theorem 2

Lemma 3.2.2 shows that all  $T$ -based mappings with the same product vacuum state are equivalent to the same mapping, up to fermionic labelling and the signs and braiding of the Pauli operator pairs. Lemma 3.2.6 uses this result to show that all product-preserving  $T$ -based mappings are equivalent to the same mapping, up to local Pauli relabelling.

**Lemma 3.2.6.** (*For each  $n$ -qubit product stabiliser state, there is a mapping based upon any  $n$ -vertex ternary tree with that product state as its vacuum state, up to local relabelling of edges.*) Let  $T$  be an  $n$ -vertex ternary tree, and let  $\mathbf{m}$  be a  $T$ -based mapping with vacuum state  $|\mathbf{0}_{\mathbf{m}}\rangle = \bigotimes_{i=0}^{n-1} |a_i\rangle_i$ . Then, for any product stabiliser state  $\bigotimes_{i=0}^{n-1} |a'_i\rangle_i$ , there exists an  $n$ -vertex ternary tree  $T'$  and a  $T'$ -based mapping  $\mathbf{m}'$  with vacuum state  $\bigotimes_{i=0}^{n-1} |a'_i\rangle_i$ , where the operators of  $\mathbf{m}'$  differ from those of  $\mathbf{m}$  only by local permutations of the Pauli labels  $\{X, Y, Z\}$ .

*Proof.* From Lemma 3.2.2, the mapping  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  must have Pauli Majorana representation pairs of the form in Equation 3.5 for some fermionic labelling  $\sigma \in S_n$ . The following process implements a modification of  $T$ :

Let  $i \in [n]$ . If the pair  $(\Gamma_{2i}, \Gamma_{2i+1})$  contains Pauli operators with the same sign, i.e. of the form  $(\pm \tilde{\Gamma}_{\sigma(i),b}, \pm \tilde{\Gamma}_{\sigma(i),c})$ , let  $B, C \in \{X, Y, Z\}$  be the Pauli matrices satisfying  $-iBC|a_i\rangle = |a_i\rangle$  and let  $B', C' \in \{X, Y, Z\}$  be such that  $-iB'C'|a'_i\rangle = |a'_i\rangle$ . Otherwise, if  $(\Gamma_{2i}, \Gamma_{2i+1})$  is of the form  $(\mp \tilde{\Gamma}_{\sigma(i),c}, \pm \tilde{\Gamma}_{\sigma(i),b})$ , let  $B, C \in \{X, Y, Z\}$  be the Pauli matrices satisfying  $iBC|a_i\rangle = |a_i\rangle$ , and let  $B', C'$  be such that  $iBC|a'_i\rangle = |a'_i\rangle$ .

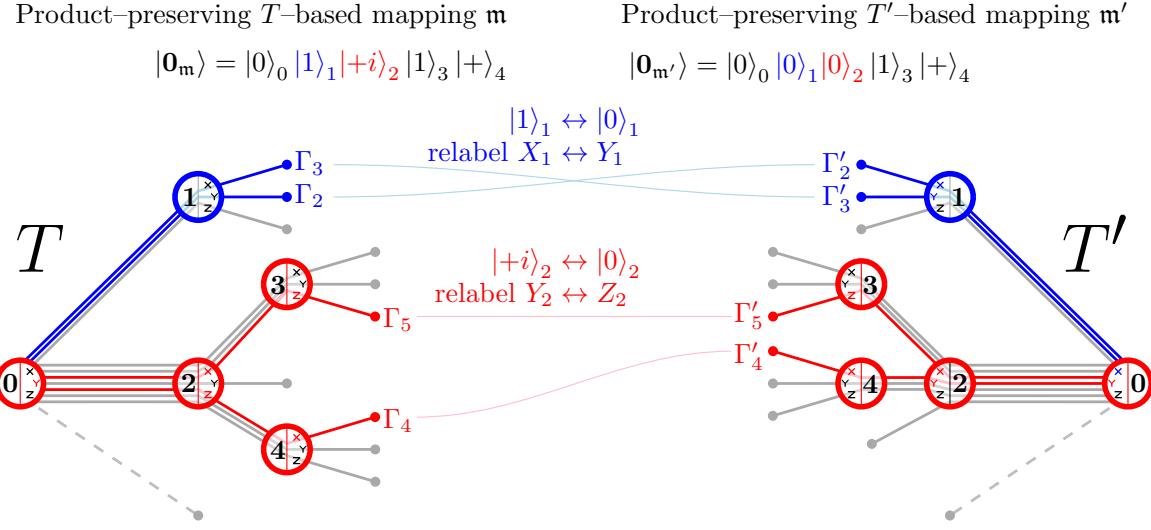
Consider the permutation  $\rho \in S_{\{X, Y, Z\}}$  of the Pauli labels with  $\rho(D) = D'$  for  $D' \in \{B, C\}$ . Define  $T'$  to be the ternary tree that arises from applying  $\rho$  to each of the labels of the immediately rightward edges of vertex  $i$  in the ternary tree  $T$ . For each  $j \in [n]$ , define  $\Gamma'_j$  to be the Pauli operators  $\tilde{\mathcal{G}}_{T'}$  that results from applying  $\rho$  to the labels of the local Pauli operators acting on the  $i$ th qubit in  $\Gamma_j$ . There is therefore a  $T'$ -based mapping  $\mathbf{m}' \sim ((\Gamma'_{2j}, \Gamma'_{2j+1}))_{j=0}^{n-1}$  with vacuum state

$$|\mathbf{0}_{\mathbf{m}'}\rangle = \bigotimes_{j \neq i} |a_j\rangle_j \otimes |a'_i\rangle_i. \quad (3.13)$$

Repeatedly defining  $\mathbf{m} \leftarrow \mathbf{m}'$ ,  $T \leftarrow T'$  and applying this modification for all  $i \in [n]$  produces a ternary tree  $T'$  and a  $T'$ -based mapping  $\mathbf{m}'$  with vacuum state  $|\mathbf{0}_{\mathbf{m}'}\rangle = \bigotimes_{i=0}^{n-1} |a'_i\rangle_i$ , as required. Figure 3.5 depicts an example of two modifications to  $T$  and  $\mathbf{m}$  that could occur during this process.  $\square$

A consequence of Lemma 3.2.6 is that, for any ternary tree  $T$ , each product-preserving  $T$ -based mapping has an equivalent mapping with any other product stabiliser vacuum state. In other words, we can choose the mapping with vacuum state  $|0\rangle^{\otimes n}$  to be the sole representative of the template of product-preserving  $T$ -based mappings; the only difference between the product-preserving  $T$ -based mappings are the labels of the Pauli matrices on each qubit. One powerful conclusion that we can draw from this is that the existing suite of mappings in the literature [71, 73, 74] is

## Illustration of Lemma 3.2.6



**Figure 3.5** Adjusting the labels of the local Pauli matrices adjusts the vacuum state of a ternary tree transformation. This figure demonstrates two such adjustments of a  $T$ -based mapping for a particular 5-vertex ternary tree: exchanging  $X$  and  $Y$  labels on the 1st qubit modifies the vacuum state locally from  $|1\rangle_1$  to  $|0\rangle_0$ , and exchanging  $Y$  with  $Z$  on the 2nd qubit modifies  $|+i\rangle_2$  to  $|0\rangle_2$ .

sufficient to describe all product-preserving ternary tree transformations. What we cannot yet conclude is whether, for an arbitrary ternary tree  $T$ , there is always a  $T$ -based mapping that is a linear encoding of the Fock basis.

Eschewing the intrinsic pairing structure of Lemma 3.2.2, Lemma 3.2.7 proves the existence and uniqueness of a Clifford operator  $C_T \in \mathcal{C}_n$  yielding a  $T$ -based mapping

$$\mathfrak{m}(T) \sim \left( (C_T \gamma_{2i} C_T^\dagger, C_T \gamma_{2i+1} C_T^\dagger) \right)_{i=0}^{n-1} \quad (3.14)$$

that linearly encodes the Fock basis: that is, the Clifford operator  $C_T$  effects the linear transformation

$$C_T : |\mathbf{f}\rangle \longmapsto |G_T \mathbf{f}\rangle \quad \text{for all } \mathbf{f} \in \mathbb{Z}_2^n \quad (3.15)$$

for some invertible binary matrix  $G_T \in \mathrm{GL}_n(\mathbb{Z}_2)$ . Lemma 3.2.8 details how to determine  $G_T$  and  $C_T$  given the ternary tree  $T$ .

**Lemma 3.2.7.** (*Unique  $T$ -based mapping  $\mathfrak{m}(T)$  to linearly encode the Fock basis.*) Let  $T$  be an  $n$ -vertex ternary tree. Then there is a unique Clifford operator  $C_T \in \mathcal{C}_n$  such that the mapping  $\mathfrak{m}(T) \sim ((C_T \gamma_{2i} C_T^\dagger, C_T \gamma_{2i+1} C_T^\dagger))_{i=0}^{n-1}$  has the following properties:

- 1)  $\mathbf{m}(T)$  is a  $T$ -based mapping, i.e.  $\pm C_T \gamma_i C_T^\dagger \in \tilde{\mathcal{G}}_T$  for all  $i \in [2n]$ , and
- 2)  $\mathbf{m}(T)$  is a classical encoding of the Fock basis, i.e.  $|\mathbf{f}_{\mathbf{m}(T)}\rangle \in \mathfrak{C}_n$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ , with vacuum state  $|\mathbf{0}_{\mathbf{m}(T)}\rangle = |0\rangle^{\otimes n}$ .

Theorem 1 proved that every classical Pauli-based mapping encoding is affine, and so these two properties of  $\mathbf{m}(T)$  guarantee that the mapping is a linear encoding of the Fock basis.

*Proof.* Inspired by [95], the following sequence of instructions details how to obtain the mapping  $\mathbf{m}(T)$  from the ternary tree  $T$ , with Figure 3.6 (a) serving as a reference. Arrange  $2n+1$  points vertically within the root vertex, labelled in ascending order from top to bottom. Draw lines representing the root-to-leaf paths of  $T$  via the following scheme, which associates an index  $i \in \{0, 1, \dots, 2n\}$  to each path:

- Starting at the point labelled  $i$  in the root, trace the line along the root-to-leaf path that ends in the  $i$ th highest unlabelled vertex on the right-hand-side of the tree.
- For each vertex, invert the subsequent top-to-bottom ordering of the lines that pass through a  $Y$  branch of that vertex for the remainder of the tree.

Define the unsigned Pauli operator  $\tilde{\Gamma}_i$  to be the element of  $\tilde{\mathcal{G}}_T$  that corresponds to the  $i$ th root-to-leaf path according to this numbering system. Let  $\#y : \tilde{\mathcal{P}}_n \rightarrow [n]$  denote the number of single-qubit  $Y$  matrices appearing in an unsigned Pauli operator. Finally, define the Pauli operators

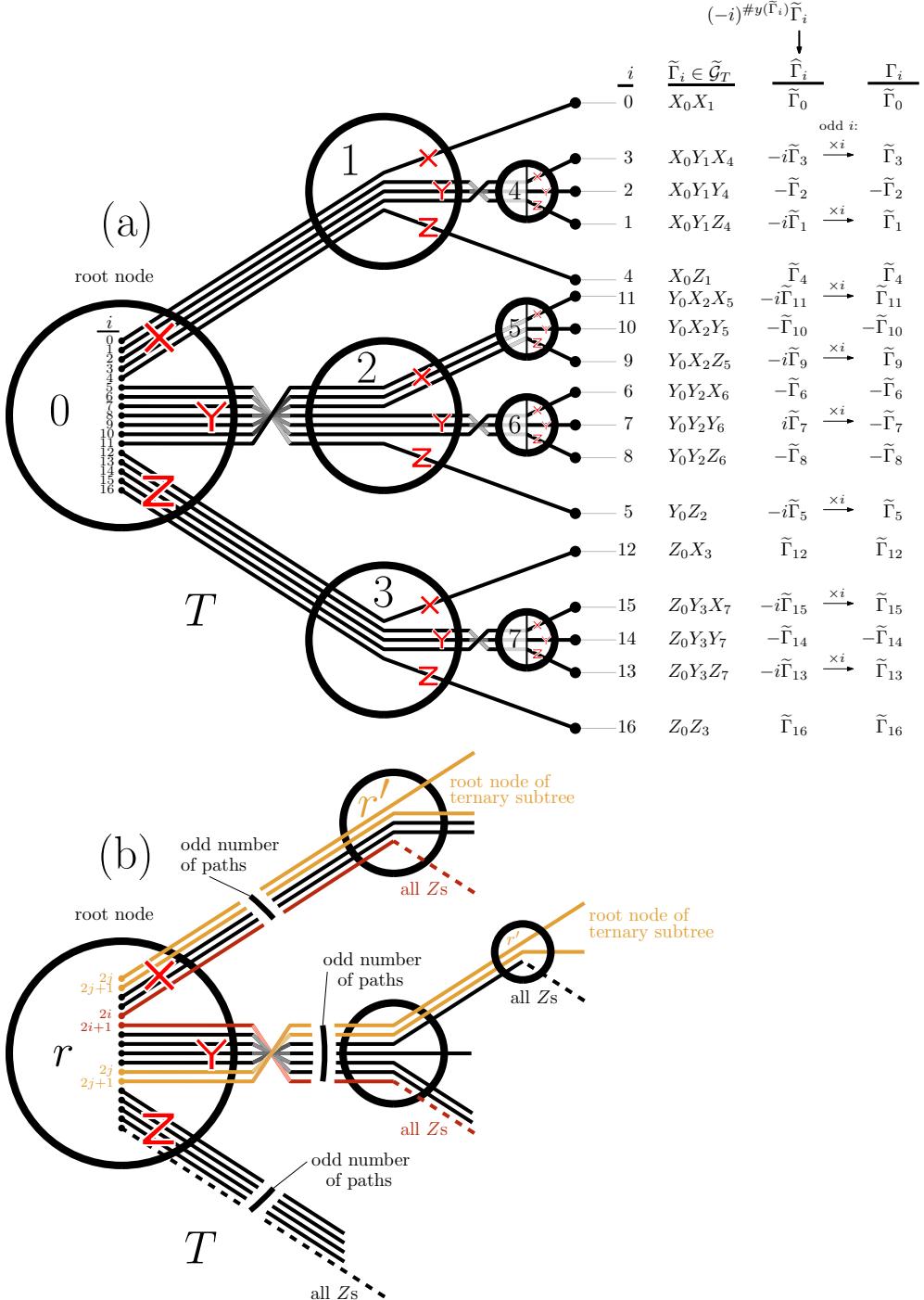
$$\hat{\Gamma}_i := (-i)^{\#y(\tilde{\Gamma}_i)} \tilde{\Gamma}_i \in \{1, X, -iY, Z\}^{\otimes n}. \quad (3.16)$$

The operator  $\hat{\Gamma}_i$  is anti-Hermitian if  $\#y(\tilde{\Gamma}_i)$  is odd, and so cannot be a Majorana representation in a  $T$ -based mapping. However the action of  $\hat{\Gamma}_i$  on  $|0\rangle^{\otimes n}$  is always in the computational basis, a property that we will exploit throughout this proof:

$$\hat{\Gamma}_i |0\rangle^{\otimes n} \in (-i)^{\#y(\tilde{\Gamma}_i)} \left( \prod_{j : \tilde{\Gamma}_i|_{\{j\}}=Y} i \right) \mathfrak{C}_n = \mathfrak{C}_n. \quad (3.17)$$

Claim: The Pauli operators  $\{\hat{\Gamma}_i\}_{i=0}^{2n}$  satisfy

$$\hat{\Gamma}_{2i} \hat{\Gamma}_{2i+1} |0\rangle^{\otimes n} = |0\rangle^{\otimes n} \text{ for all } i = 0, 1, \dots, n-1, \quad (3.18)$$



**Figure 3.6** (a) Deriving the Pauli operators  $\{\Gamma_i\}_{i=0}^{2n-1}$  from a ternary tree  $T$  such that  $\mathfrak{m}(T) \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  is a classical encoding of the Fock basis. (b) Visual guide to the proof that  $\hat{\Gamma}_{2i} \hat{\Gamma}_{2i+1} |0\rangle^{\otimes n} = |0\rangle^{\otimes n}$ .

Proof: Note that the bottom-most root-to-leaf path of a ternary tree takes only  $Z$  branches. Suppose that the paths with labels  $2i$  and  $2i+1$  diverge on the root vertex. Because there are an odd number of paths departing the root via each of the  $X$ ,  $Y$  and  $Z$  branches, the only way for the consecutive paths  $2i$  and  $2i+1$  to diverge at the root is for the  $(2i)$ th path to be the bottom-most path taking the  $X$  branch, and for the  $(2i+1)$ th path to be the topmost to take the  $Y$  branch, as Figure 3.6 (b) illustrates. These paths become the bottom-most root-to-leaf paths of their respective ternary subtrees of  $T$ , and therefore subsequently to the root, the paths take only  $Z$  branches. This means that:

$$\hat{\Gamma}_{2i} \hat{\Gamma}_{2i+1} |0\rangle^{\otimes n} = (-iXY)_r |0\rangle_r \otimes \left( \bigotimes_{s \neq r} |0\rangle_s \right) = |0\rangle^{\otimes n}. \quad (3.19)$$

Consider another pair of paths, with labels  $2j$  and  $2j+1$ , that do not diverge at the root of  $T$ . Figure 3.6 (b) highlights two possible candidates for  $j$ . Apply the argument of the previous paragraph to the ternary subtree of  $T$  whose root is the vertex  $r'$  upon which paths  $2j$  and  $2j+1$  *do* diverge. The vertical ordering of the two paths depends on the number of  $Y$  branches in their shared path up to  $r'$ ; let this number be  $y$ . Then, since the topmost of the two paths departs vertex  $r'$  via an  $X$  branch, and the bottommost via a  $Y$  branch,

$$\hat{\Gamma}_{2j} \hat{\Gamma}_{2j+1} |0\rangle^{\otimes n} = ((-1)^y)^2 (-iXY)_{r'} |0\rangle_{r'} \otimes \left( \bigotimes_{s \neq r'} |0\rangle_s \right) = |0\rangle^{\otimes n}, \quad (3.20)$$

where one factor of  $(-1)^y$  is due to anticommuting  $\hat{\Gamma}_{2j}$  and  $\hat{\Gamma}_{2j+1}$  so that the action on qubit  $r'$  is  $-iXY$ , and the other factor of  $(-1)^y$  is due to the shared  $(-iY)$  operators in  $\hat{\Gamma}_{2j}$  and  $\hat{\Gamma}_{2j+1}$ . This proves the claim.  $\blacksquare$

Claim: The Pauli operators  $\{\hat{\Gamma}_i\}_{i=0}^{2n}$  satisfy

$$\begin{aligned} \hat{\Gamma}_{i_1} \hat{\Gamma}_{i_2} \dots \hat{\Gamma}_{i_k} |0\rangle^{\otimes n} &\in \mathfrak{C}_n & \text{for all } 0 \leq i_1 < i_2 < \dots < i_k \leq 2n \\ &\text{and all } k = 1, 2, \dots, 2n. \end{aligned} \quad (3.21)$$

Proof: Observe the repeated action of  $(-iY)$  operators on a single-qubit state  $|0\rangle$ :

$$(-iY)^l |0\rangle = (-1)^{t_{l-1}} |l \bmod 2\rangle, \quad (-iY)^l |1\rangle = (-1)^{t_l} |(l+1) \bmod 2\rangle. \quad (3.22)$$

Therefore, for general  $l_1, l_2 \in \mathbb{N}$ ,

$$\begin{aligned} (-iY)^{l_1} |l_2 \bmod 2\rangle &= (-1)^{t_{(l_1+(l_2 \bmod 2)-1)}} |(l_1 + l_2) \bmod 2\rangle \\ &\in (-1)^{t_{(l_1+(l_2 \bmod 2)-1)}} \mathfrak{C}_1. \end{aligned} \quad (3.23)$$

As part of a proof by induction, assume the following is true for some  $k$ : *For any  $n$ -vertex ternary tree  $T$ , the Pauli operators  $\{\hat{\Gamma}_i\}_{i=0}^{2n}$  in Equation 3.16 satisfy the following statement: let  $k \in \{1, 2, \dots, 2n\}$  and let the integer sequence  $i_1, i_2, \dots, i_k$  be ascending, i.e.  $0 \leq i_1 < i_2 < \dots < i_k \leq 2n$ . Then, the action of the operators in order of descending labels  $\hat{\Gamma}_{i_k}, \hat{\Gamma}_{i_{k-1}}, \dots, \hat{\Gamma}_{i_2}, \hat{\Gamma}_{i_1}$  upon the state  $|0\rangle^{\otimes n}$  satisfies*

$$\hat{\Gamma}_{i_1} \hat{\Gamma}_{i_2} \dots \hat{\Gamma}_{i_k} |0\rangle^{\otimes n} \in \mathfrak{C}_n. \quad (3.24)$$

By Equation 3.17, the statement is true for  $k = 1$ . Note that, as a consequence of Equation 3.24, reversing the order of the operators introduces a sign of  $(-1)^{t_{k-1}}$ , where  $t_k$  is the  $k$ th triangular number and  $t_1 = 1$ :

$$\hat{\Gamma}_{i_k} \hat{\Gamma}_{i_{k-1}} \dots \hat{\Gamma}_{i_2} \hat{\Gamma}_{i_1} |0\rangle^{\otimes n} = (-1)^{t_{k-1}} \hat{\Gamma}_{i_1} \hat{\Gamma}_{i_2} \dots \hat{\Gamma}_{i_k} |0\rangle^{\otimes n} \in (-1)^{t_{k-1}} \mathfrak{C}_n, \quad (3.25)$$

Now, suppose the inductive statement is true for some  $k \in \{1, 2, \dots, 2n\}$  and for some sequence of integers  $i_1, i_2, \dots, i_k$  with  $0 < i_1 < i_2 < \dots < i_k \leq 2n$ . Let  $i_0$  be an integer in the range  $\{0, 1, \dots, i_1 - 1\}$ . Consider the quantity

$$\hat{\Gamma}_{i_0} \hat{\Gamma}_{i_1} \hat{\Gamma}_{i_2} \dots \hat{\Gamma}_{i_k} |0\rangle^{\otimes n}. \quad (3.26)$$

In order to determine whether the state in Formula 3.26 is in the computational basis, introduce the following notation. Let  $\mathcal{S} \subseteq \{0, 1, \dots, n - 1\}$  be a subset of qubit labels. Recall that each  $\hat{\Gamma}_i = (-i)^{\#y(\tilde{\Gamma}_i)} \tilde{\Gamma}_i$  for some  $\tilde{\Gamma}_i \in \tilde{\mathcal{G}}_T$ . Define  $\mathcal{S}_X(\hat{\Gamma})$ ,  $\mathcal{S}_Y(\hat{\Gamma})$  and  $\mathcal{S}_Z(\hat{\Gamma})$  to be the subsets of  $\mathcal{S}$  containing the labels of qubits upon which  $\tilde{\Gamma}$  acts with  $X$ ,  $Y$  and  $Z$  matrices, respectively. This allows a well-defined notion of the restriction  $\hat{\Gamma}|_{\mathcal{S}}$  of  $\hat{\Gamma}$  to the subset of qubits with labels in  $\mathcal{S}$ ; specifically,

$$\hat{\Gamma}|_{\mathcal{S}} := \left( \bigotimes_{s \in \mathcal{S}_X(\hat{\Gamma})} X_s \right) \left( \bigotimes_{s \in \mathcal{S}_Y(\hat{\Gamma})} (-iY)_s \right) \left( \bigotimes_{s \in \mathcal{S}_Z(\hat{\Gamma})} Z_s \right). \quad (3.27)$$

Using the notation of Equation 3.27, define  $|0\rangle_{\mathcal{S}} = \bigotimes_{s \in \mathcal{S}} |0\rangle_s$  and define the product states  $|\mathcal{S}\rangle$  via

$$|\mathcal{S}\rangle := \left( \widehat{\Gamma}_{i_0} \widehat{\Gamma}_{i_1} \widehat{\Gamma}_{i_2} \dots \widehat{\Gamma}_{i_k} \right) \Big|_{\mathcal{S}} |0\rangle_{\mathcal{S}} . \quad (3.28)$$

Then, for any partition  $\Pi$  of the qubit labels  $\{0, 1, \dots, n - 1\}$ , the state in Equation 3.26 is equal to

$$\widehat{\Gamma}_{i_0} \widehat{\Gamma}_{i_1} \widehat{\Gamma}_{i_2} \dots \widehat{\Gamma}_{i_k} |0\rangle^{\otimes n} = \bigotimes_{\mathcal{S} \in \Pi} |\mathcal{S}\rangle . \quad (3.29)$$

Let the root of  $T$  have label  $r$ , and let  $r'$  be the label of the rightmost vertex through which all  $k+1$  of the root-to-leaf paths pass such that their labels are in ascending order from top-to-bottom. This vertex is the last upon which all paths are ‘in order’. Either  $r = r'$ , as is the case in Figure 3.6, or all  $k+1$  paths pass identically through the vertices of a segment of the tree, as in Figure 3.7. Let  $\mathcal{R}$  be the possibly empty subset of vertex labels in this tree segment, up to and excluding  $r'$ ; let  $\mathcal{S}$  be the remaining subset of vertices including and to the right of  $r'$ , so that  $\mathcal{R} \cup \mathcal{S} = \{0, 1, \dots, n - 1\}$  partitions the vertex labels. The state of interest is thus

$$\widehat{\Gamma}_{i_0} \widehat{\Gamma}_{i_1} \dots \widehat{\Gamma}_{i_k} |0\rangle^{\otimes n} = |\mathcal{R}\rangle \otimes |\mathcal{S}\rangle . \quad (3.30)$$

Consider the product state  $|\mathcal{R}\rangle$ . Because  $X^{k+1}|0\rangle = |(k+1) \bmod 2\rangle \in \mathfrak{C}_1$  and  $Z^{k+1}|0\rangle = |0\rangle \in \mathfrak{C}_1$ , any vertex  $r_a \in \mathcal{R}$  upon which the paths take an  $X$  or  $Z$  branch corresponds to a qubit that remains in the computational basis:  $(\widehat{\Gamma}_{i_0} \widehat{\Gamma}_{i_1} \dots \widehat{\Gamma}_{i_k})|_{\{r_a\}} |0\rangle_{\{r_a\}} \in \mathfrak{C}_1$ . The  $k+1$  paths appear in their original ascending order from top-to-bottom on vertex  $r'$ , and so there must be an even number of vertices in  $\mathcal{R}$  upon which the paths take a  $Y$  branch, since each such occurrence inverts the order of the paths. For any pair of vertices  $\{r_a, r_b\} \in \mathcal{R}$  upon which all  $k+1$  paths take a  $Y$  branch,

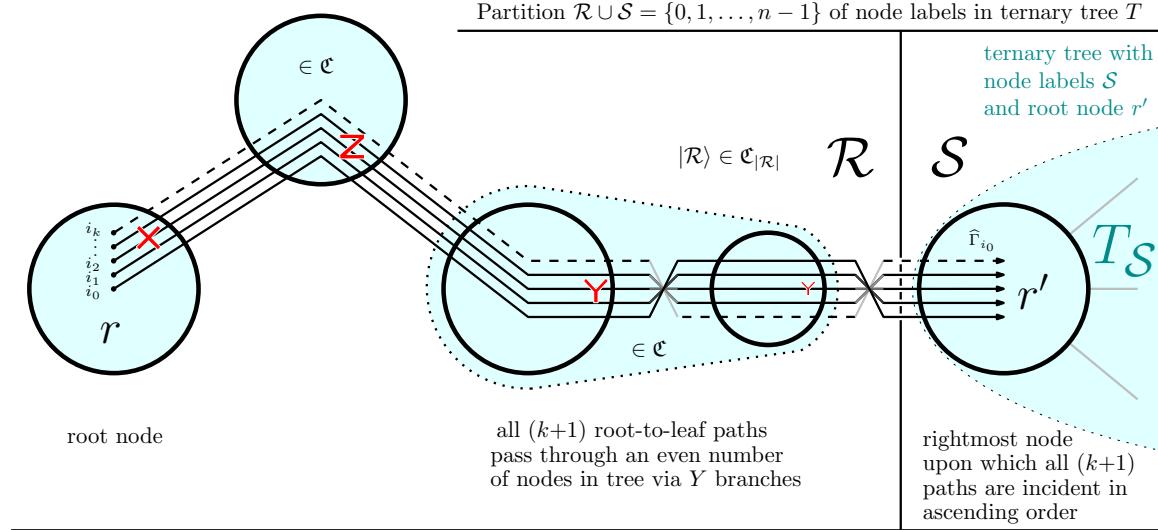
$$(-iY)_{r_a}^{k+1} \otimes (-iY)_{r_b}^{k+1} |0\rangle_{\{r_a, r_b\}} = \left( (-1)^{t_k} \right)^2 |(k+1) \bmod 2\rangle_{\{r_a, r_b\}} \quad (3.31)$$

$$= |(k+1) \bmod 2\rangle_{\{r_a, r_b\}} \quad (3.32)$$

$$\in \mathfrak{C}_2 , \quad (3.33)$$

and so  $\widehat{\Gamma}_{i_0} \widehat{\Gamma}_{i_1} \dots \widehat{\Gamma}_{i_k} |0\rangle_{\{r_a, r_b\}} \in \mathfrak{C}_2$ . Therefore  $|\mathcal{R}\rangle \in \mathfrak{C}_{|\mathcal{R}|}$ .

$k+1$  root-to-leaf paths in ternary tree  $T$

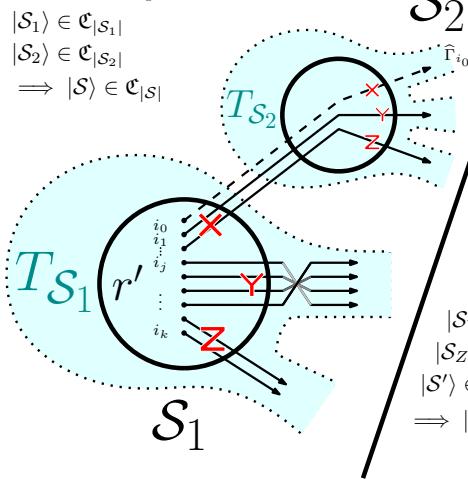


Case 1: use partition  $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$

$$|\mathcal{S}_1\rangle \in \mathfrak{C}_{|\mathcal{S}_1|}$$

$$|\mathcal{S}_2\rangle \in \mathfrak{C}_{|\mathcal{S}_2|}$$

$$\Rightarrow |\mathcal{S}\rangle \in \mathfrak{C}_{|\mathcal{S}|}$$

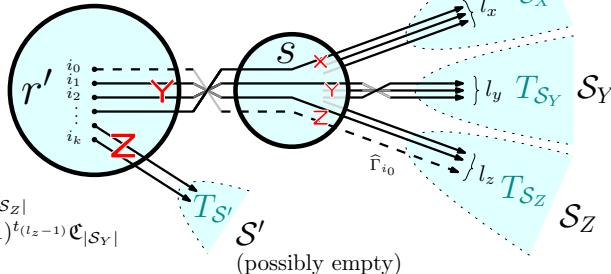


Case 2: use partition  $\mathcal{S} = \{r'\} \cup \{s\} \cup \mathcal{S}_X \cup \mathcal{S}_Y \cup \mathcal{S}_Z \cup \mathcal{S}'$

$$|\{r'\}\rangle \in (-1)^{t(l_x+l_y+l_z-1)} \mathfrak{C}_1$$

$$|\{s\}\rangle \in (-1)^{t(l_y+(l_x \bmod 2)-1)+l_z(l_x+l_y)} \mathfrak{C}_1$$

$$|\mathcal{S}_X\rangle \in (-1)^{t(l_x-1)} \mathfrak{C}_{|\mathcal{S}_X|}$$



**Figure 3.7** Part of the proof of Lemma 3.2.7. The inductive statement is that for any  $n$ -vertex ternary tree  $T$ , the state  $\hat{\Gamma}_{i_1} \hat{\Gamma}_{i_2} \dots \hat{\Gamma}_{i_k} |0\rangle^{\otimes n}$  resides in the computational basis, where the Pauli operators  $\hat{\Gamma}_i \in \{\mathbb{1}, X, -iY, Z\}^{\otimes n}$  act with labels in descending order  $0 \leq i_1 < i_2 < \dots < i_k \leq 2n$ . This statement thus applies to any sub-tree of  $T$  on the vertex subset  $\mathcal{S} \subseteq \{0, 1, \dots, n-1\}$  and any product of at most  $k$  operators, with labels in descending order, acting on the state  $|0\rangle_{\mathcal{S}}$ .

Now consider the product state  $|\mathcal{S}\rangle$ . There are two scenarios describing how the paths might continue on from vertex  $r'$ , as Figure 3.7 highlights. In both cases, we demonstrate that  $|\mathcal{S}\rangle \in \mathfrak{C}_{|\mathcal{S}|}$ .

- **Case 1:**  $(\hat{\Gamma}_{i_0}|_{\{r'\}} = X)$ : there are root-to-leaf paths extending along the  $X$  branch from vertex  $r'$ .
- **Case 2:**  $(\hat{\Gamma}_{i_0}|_{\{r'\}} = -iY)$ : there are no root-to-leaf paths extending along the  $X$  branch from vertex  $r'$ .

In **Case 1**, partition  $\mathcal{S}$  into the set  $\mathcal{S}_1$  of all vertices in the root-to-leaf paths leaving vertex  $r'$  via the  $Y$  or  $Z$  branches, and the set  $\mathcal{S}_2$  consisting of the vertices in the root-to-leaf paths leaving vertex  $r'$  via the  $X$  branch. This partitions the ternary tree with root  $r'$  into two ternary subtrees of  $T$ : the ternary tree  $T_{\mathcal{S}_1}$  with root  $r'$ , which contains vertices with labels in  $\mathcal{S}_1$ , and the ternary tree  $T_{\mathcal{S}_2}$  containing vertices with labels in  $\mathcal{S}_2$ .

We wish to investigate  $|\mathcal{S}\rangle = |\mathcal{S}_1\rangle \otimes |\mathcal{S}_2\rangle$ . The state  $|\mathcal{S}_2\rangle$  is in the computational basis by the inductive statement Equation 3.24 applied to the  $|\mathcal{S}_2|$ -vertex ternary tree  $T_{\mathcal{S}_2}$ , because  $|\mathcal{S}_2\rangle$  is the result of at most  $k$  strings with descending labels in  $\{\mathbb{1}, X, -iY, Z\}^{\otimes |\mathcal{S}_2|}$  acting on the state  $|0\rangle_{\mathcal{S}_2}$ . To show that the state  $|\mathcal{S}_1\rangle$  is also in the computational basis, suppose that the first  $j \in \{1, 2, \dots, k-1\}$  operators  $\hat{\Gamma}_{i_0}, \hat{\Gamma}_{i_1}, \hat{\Gamma}_{i_2}, \dots, \hat{\Gamma}_{i_{j-1}}$  act with  $X$  upon qubit  $r'$ , with the remaining operators acting on qubit  $r'$  with either  $(-iY)$  or  $Z$ . Then,

$$|\mathcal{S}_1\rangle = (\hat{\Gamma}_{i_0} \hat{\Gamma}_{i_1} \hat{\Gamma}_{i_2} \dots \hat{\Gamma}_{i_k}) \Big|_{\mathcal{S}_1} |0\rangle_{\mathcal{S}_1} = (X_r)^j \underbrace{(\hat{\Gamma}_{i_j} \hat{\Gamma}_{i_{j+1}} \dots \hat{\Gamma}_{i_k})}_{\in \mathfrak{C}_{|\mathcal{S}_1|} \text{ by inductive statement on } T_{\mathcal{S}_1}} \Big|_{\mathcal{S}_1} |0\rangle_{\mathcal{S}_1} \in \mathfrak{C}_{|\mathcal{S}_1|}, \quad (3.34)$$

applying the inductive statement to the  $|\mathcal{S}_1|$ -vertex ternary tree  $T_{\mathcal{S}_1}$  as there are at most  $k$  operators in descending order acting on the state  $|0\rangle_{\mathcal{S}_1}$ , and using the fact that the  $(X_r)^j$  operators fix the computational basis. Therefore  $|\mathcal{S}\rangle = |\mathcal{S}_1\rangle \otimes |\mathcal{S}_2\rangle \in \mathfrak{C}_{|\mathcal{S}|}$ .

In **Case 2**, partition  $\mathcal{S}$  into six sets: the sets  $\{r'\}$  and  $\{s\}$ , where  $s$  is the label of the vertex immediately to the right of  $r'$  along the  $Y$  branch, the possibly empty sets  $\mathcal{S}_X$ ,  $\mathcal{S}_Y$  and  $\mathcal{S}_Z$  containing the labels of the vertices in the root-to-leaf paths leaving  $s$  via its  $X$ ,  $Y$  and  $Z$  branches, and the possibly empty set  $\mathcal{S}'$  containing the labels of the vertices in the root-to-leaf paths leaving  $r'$  via its  $Z$  branch. Note that this partition induces ternary subtrees  $T_{\mathcal{S}_X}$ ,  $T_{\mathcal{S}_Y}$ ,  $T_{\mathcal{S}_Z}$  and  $T_{\mathcal{S}'}$  of the ternary tree, with vertex sets  $\mathcal{S}_X$ ,  $\mathcal{S}_Y$ ,  $\mathcal{S}_Z$  and  $\mathcal{S}'$ , respectively.

We wish to investigate

$$|\mathcal{S}\rangle = |\{r'\}\rangle \otimes |\mathcal{S}'\rangle \otimes |\{s\}\rangle \otimes |\mathcal{S}_X\rangle \otimes |\mathcal{S}_Y\rangle \otimes |\mathcal{S}_Z\rangle . \quad (3.35)$$

Applying the inductive statement to  $T_{\mathcal{S}'}$ , Equation 3.24 implies that  $|\mathcal{S}'\rangle \in \mathfrak{C}_{|\mathcal{S}'|}$ . Suppose that  $l_x$ ,  $l_y$  and  $l_z$  of the root-to-leaf paths pass through the  $X$ ,  $Y$  and  $Z$  edges of the vertex with label  $s$ , respectively. Applying Equation 3.25 in the inductive statement to ternary trees  $T_{\mathcal{S}_X}$ ,  $T_{\mathcal{S}_Y}$  and  $T_{\mathcal{S}_Z}$  then implies that  $|\mathcal{S}_X\rangle \in (-1)^{t_{(l_x-1)}}\mathfrak{C}$ ,  $|\mathcal{S}_Y\rangle \in \mathfrak{C}$  and  $|\mathcal{S}_Z\rangle \in (-1)^{t_{(l_z-1)}}\mathfrak{C}$ .

Equation 3.23 establishes that  $|\{r'\}\rangle \in (-1)^{t_{(l_x+l_y+l_z-1)}}\mathfrak{C}_1$  and that the state of the qubit with label  $s$  is

$$|\{s\}\rangle = \underbrace{\tilde{\Gamma}_{i_0} \hat{\Gamma}_{i_1} \dots \hat{\Gamma}_{i_{(l_z-1)}}}_{l_z \text{ operators}} \underbrace{\hat{\Gamma}_{i_{l_z}} \dots \hat{\Gamma}_{i_{(l_z+l_y-1)}}}_{l_y \text{ operators}} \underbrace{\hat{\Gamma}_{i_{(l_z+l_y)}} \dots \hat{\Gamma}_{i_{(l_x+l_y+l_z-1)}}}_{l_x \text{ operators}} \Big|_{\{s\}} |0\rangle_s \quad (3.36)$$

$$= Z^{l_z}(-iY)^{l_y} X^{l_x} |0\rangle_s \quad (3.37)$$

$$= Z^{l_z}(-iY)^{l_y} |l_x \bmod 2\rangle \quad (3.38)$$

$$= Z^{l_z}(-1)^{t_{(l_x+(l_x \bmod 2)-1)}} |(l_x + l_y) \bmod 2\rangle \quad (3.39)$$

$$= (-1)^{t_{(l_y+(l_x \bmod 2)-1)}+l_z(l_x+l_y)} |(l_x + l_y) \bmod 2\rangle \quad (3.40)$$

$$\in (-1)^{t_{(l_y+(l_x \bmod 2)-1)}+l_z(l_x+l_y)} \mathfrak{C}_1 . \quad (3.41)$$

Substituting the factors of  $|\mathcal{S}\rangle$  that are not necessarily in the computational basis into Equation 3.35,

$$|\mathcal{S}\rangle \in \underbrace{(-1)^{t_{(l_x+l_y+l_z-1)}}}_{\text{from } |\{r'\}\rangle} \underbrace{(-1)^{t_{(l_y+(l_x \bmod 2)-1)}}}_{\text{from } |\{s\}\rangle} + l_z(l_x + l_y) \underbrace{(-1)^{t_{(l_x-1)}}}_{\text{from } |\mathcal{S}_X\rangle} \underbrace{(-1)^{t_{(l_z-1)}}}_{\text{from } |\mathcal{S}_Z\rangle} \mathfrak{C}_{|\mathcal{S}|} \quad (3.42)$$

$$= (-1)^A \mathfrak{C}_{|\mathcal{S}|} , \quad (3.43)$$

where, using the property  $t_{l_1+l_2} = t_{l_1} + t_{l_2} + l_1 l_2$  of the triangular numbers,

$$A \equiv (t_{(l_x+l_y+l_z-1)} + t_{(l_y+(l_x \bmod 2)-1)} + l_z(l_x + l_y) + t_{(l_x-1)} + t_{(l_z-1)}) \quad \text{mod } 2 \quad (3.44)$$

$$\equiv \underbrace{(t_{(l_x+l_y+l_z)} + (l_x + l_y + l_z))}_{t_{l-1}=t_l-l} \quad (3.45)$$

$$+ \underbrace{t_{l_y+(l_x \bmod 2)} + (l_y + l_x)}_{t_{l-1}=t_l-l} + l_z(l_x + l_y) + \underbrace{t_{l_x} + t_{l_z}}_{t_{l-1}=t_l-l} + \underbrace{t_{l_z} + l_z}_{t_{l-1}=t_l-l} \quad \text{mod } 2$$

$$\equiv \underbrace{(t_{l_y} + t_{(l_x \bmod 2)} + l_y(l_x \bmod 2))}_{t_{(l_1+l_2)}=t_1+t_2+l_1l_2} + t_{(l_x+l_y+l_z)} + t_{l_x} + t_{l_z} + l_z(l_x + l_y) + l_x \quad (3.46)$$

$$\equiv (t_{(l_x+l_y+l_z)} + \underbrace{t_{l_x} + t_{l_y} + t_{l_z} + (l_x l_y + l_y l_z + l_z l_x)}_{=t_{(l_x+l_y+l_z)}} + \underbrace{t_{(l_x \bmod 2)} + l_x}_{=l_x \bmod 2}) \quad \text{mod } 2 \quad (3.47)$$

$$= 0, \quad (3.48)$$

and so  $|\mathcal{S}\rangle \in \mathfrak{C}_{|\mathcal{S}|}$ . Therefore the state of interest, which Equation 3.30 established to be equal to  $|\mathcal{R}\rangle \otimes |\mathcal{S}\rangle$ , is in the computational basis. This proves the inductive step.

Since Equation 3.18 shows that the inductive statement is true for  $k = 1$ , for any sequence of  $(k + 1)$  integers  $0 \leq i_0 \leq i_1 < \dots < i_k \leq 2n$ ,

$$\widehat{\Gamma}_{i_0} \widehat{\Gamma}_{i_1} \dots \widehat{\Gamma}_{i_k} |0\rangle^{\otimes n} \in \mathfrak{C}_n. \quad (3.49)$$

■

Define the operators  $\Gamma_{2i} = \widehat{\Gamma}_{2i}$  and  $\Gamma_{2i+1} = -i\widehat{\Gamma}_{2i+1}$  for  $i \in [n]$ . Operators of the form  $\widehat{\Gamma}_{2i}$  act on an even number of qubits with  $(-iY)$ , and operators of the form  $\widehat{\Gamma}_{2i+1}$  act on an odd number of qubits with  $(-iY)$ ; thus  $\Gamma_i$  is Hermitian for all  $i \in [2n]$ . Define the fermion-qubit mapping  $\mathfrak{m}(T) \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ . Via Equation 3.18, the vacuum state of the mapping is  $|\mathbf{0}_{\mathfrak{m}(T)}\rangle = |0\rangle^{\otimes n}$ , and the occupation number basis states are

$$|\mathbf{f}_{\mathfrak{m}(T)}\rangle = \Gamma_0 \Gamma_2 \dots \Gamma_{2n-2} |0\rangle^{\otimes n} = \widehat{\Gamma}_0 \widehat{\Gamma}_2 \dots \widehat{\Gamma}_{2n-2} |0\rangle^{\otimes n} \in \mathfrak{C}_n \quad \text{for all } \mathbf{f} \in \mathbb{Z}_2^n. \quad (3.50)$$

Therefore  $\mathfrak{m}(T)$  is not only a  $T$ -based mapping but also a classically encoding of the Fock basis. Because ternary tree transformations are Pauli-based mappings, the mapping  $\mathfrak{m}(T)$  must be an affine encoding of the Fock basis, via Theorem 1. Since the vacuum state  $|\mathbf{0}_{\mathfrak{m}(T)}\rangle = |0\rangle^{\otimes n}$ , the mapping must in fact be a linear encoding of the Fock basis, and therefore the Clifford operator  $C_T := C_{\mathfrak{m}(T)}$  with  $C_T : |\mathbf{f}\rangle \mapsto |\mathbf{f}_{\mathfrak{m}(T)}\rangle$  satisfies  $C_T \gamma_i C_T^\dagger = \Gamma_i$  for all  $i \in [2n]$ , as required. □

Since  $\mathbf{m}(T)$  is a linear encoding of the Fock basis, there is an invertible binary matrix  $G_T \in \mathrm{GL}_n(\mathbb{Z}_2)$  with  $|\mathbf{f}_{\mathbf{m}(T)}\rangle = |G_T \mathbf{f}\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ ; Lemma 3.2.8 demonstrates how to find this matrix.

**Lemma 3.2.8.** (*Determining  $G_T$  and  $C_T$  from  $\mathbf{m}(T)$ .*) Given an  $n$ –vertex ternary tree  $T$ , let  $G_T \in \mathrm{GL}_n(\mathbb{Z}_2)$  be the invertible binary matrix satisfying  $|\mathbf{f}_{\mathbf{m}(T)}\rangle = |G_T \mathbf{f}\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^n$ . Then, the elements of  $G_T$  are

$$(G_T)_{ij} = \begin{cases} 1 & \text{if } \Gamma_{2j} \text{ acts on qubit } i \text{ with } X_i \text{ or } Y_i \\ 0 & \text{if } \Gamma_{2j} \text{ acts on qubit } i \text{ with } \mathbb{1}_i \text{ or } Z_i \end{cases} \quad (3.51)$$

The Clifford operator  $C_T \in \mathcal{C}_n$  has the following stabiliser tableau<sup>1</sup>, which identifies  $C_T$  up to a global phase:

$$[C_G] = \left[ \begin{array}{c|cc|c} G_T & \begin{matrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{matrix} \\ \hline \begin{matrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{matrix} & & \begin{matrix} 0 \\ (G_T^{-1})^\top \\ \vdots \\ 0 \end{matrix} \end{array} \right]. \quad (3.52)$$

Taking the phaseless representative of this tableau, which implements  $|\mathbf{f}\rangle \mapsto |G_T \mathbf{f}\rangle$ , identifies  $C_T$  exactly.

*Proof.* As a linear transformation on  $\mathbb{Z}_2^n$ , the  $j$ th column of  $G_T$  is the image of  $\mathbf{1}_j$  under  $G_T$ . Note that

$$|G_T \mathbf{1}_j\rangle = |(\mathbf{1}_j)_{\mathbf{m}(T)}\rangle = \Gamma_{2j} |0\rangle^{\otimes n} = \left( \bigotimes_{\widehat{\Gamma}_{2j}|_{\{i\}} = X_i \text{ or } Y_i} |1\rangle_i \right) \otimes \left( \bigotimes_{\widehat{\Gamma}_{2j}|_{\{i\}} = \mathbb{1}_i \text{ or } Z_i} |0\rangle_i \right), \quad (3.53)$$

which corroborates Equation 3.51. The result of Equation 3.52 appears in the literature [91].  $\square$

---

<sup>1</sup>see Definition 2.4.3.

### 3.2.4 Main theorem: equivalence of ternary tree transformations to linear encodings

Theorem 2 summarises all the results of Section 3.2 into one statement.

**Theorem 2.** (*Every product-preserving ternary tree transformation is equivalent to a linear encoding of the Fock basis.*)

- a) *Existence and uniqueness of  $\mathbf{m}(T)$ :* For every  $n$ -vertex ternary tree  $T$ , there is a unique ancilla-free fermion-qubit mapping that both is a  $T$ -based mapping and linearly encodes the Fock basis.
- b) *Completeness of  $\mathbf{m}(T)$ :* Every  $n$ -mode product-preserving ternary tree transformation is equivalent to a mapping in the set  $\{\mathbf{m}(T) \mid T \text{ is an } n\text{-vertex ternary tree}\}$ .

*Proof.* Let  $T$  be an  $n$ -vertex ternary tree.

- a) Lemma 3.2.7 guarantees the existence of  $\mathbf{m}(T)$ . Because  $\mathbf{m}(T)$  has vacuum state  $|\mathbf{0}_{\mathbf{m}(T)}\rangle = |0^{\otimes n}\rangle$ , all other product-preserving  $T$ -based mappings with vacuum state  $|0\rangle^{\otimes n}$ , and hence all  $T$ -based mappings that are linear encodings of the Fock basis, are equivalent to  $\mathbf{m}(T)$  up to Pauli braids, sign change of pairs, and fermionic labelling, via Lemma 3.2.2. However, performing any of these operations would produce a mapping  $\mathbf{m} \mapsto \mathbf{m}'$  that is not a linear encoding:

- Pauli braid  $\Gamma_{2j} \mapsto \pm \Gamma_{2j+1}$ :

$$|(\mathbf{1}_j)_{\mathbf{m}(T)}\rangle = \Gamma_{2j} |0\rangle^{\otimes n} \mapsto |(\mathbf{1}_j)_{\mathbf{m}'}\rangle = \pm \Gamma_{2j+1} |0\rangle^{\otimes n} \in (\pm i)\mathfrak{C}_n. \quad (3.54)$$

- Sign change  $\Gamma_{2j} \mapsto -\Gamma_{2j}$ :

$$|(\mathbf{1}_j)_{\mathbf{m}(T)}\rangle = \Gamma_{2j} |0\rangle^{\otimes n} \mapsto |(\mathbf{1}_j)_{\mathbf{m}'}\rangle = -\Gamma_{2j} |0\rangle^{\otimes n} \in -\mathfrak{C}_n. \quad (3.55)$$

For any permutation  $\sigma \in S_n$  of the fermionic mode labels, there exist  $i, j \in [n]$  with  $i < j$  such that  $\sigma(i) > \sigma(j)$ . In this situation:

- Fermionic labelling:

$$|(\mathbf{1}_i + \mathbf{1}_j)_{\mathbf{m}(T)}\rangle = \Gamma_{2i} \Gamma_{2j} |0\rangle^{\otimes n} \mapsto |\mathbf{f}_{\mathbf{m}'}\rangle = \Gamma_{2\sigma(i)} \Gamma_{2\sigma(j)} |0\rangle^{\otimes n} \in -\mathfrak{C}_n. \quad (3.56)$$

Therefore  $\mathbf{m}(T)$  is the unique  $T$ -based mapping to linearly encode the Fock basis.

b) The only product-preserving  $T$ -based mappings that are not equivalent to  $\mathbf{m}(T)$  via Pauli braids, sign changes of pairs, and fermionic relabelling are those that feature the operator  $\tilde{\Gamma}_{2n} := \prod_{i=0}^{2n-1} \tilde{\Gamma}_i$  or  $-\tilde{\Gamma}_{2n}$  as one of the Majorana representations, from Lemma 2.2.5b). As a tensor product of only the single-qubit matrices  $\{\mathbb{1}, Z\}$ , this operator arises from the bottom-most root-to-leaf path in  $T$ . By Lemma 3.2.2, a  $T$ -based mapping with  $\pm\tilde{\Gamma}_{2n}$  as one of the Majorana representations will have a vacuum state that is not equal to  $|0\rangle^{\otimes n}$ . By Lemma 3.2.6, any such mapping is equivalent to a  $T'$ -based mapping with vacuum state  $|0\rangle^{\otimes n}$  for some  $n$ -vertex ternary tree  $T'$  which differs from  $T$  only by local Pauli relabelling of the edges of  $T$ . Therefore, any such mapping is equivalent to  $\mathbf{m}(T')$ ; hence the set of mappings  $\{\mathbf{m}(T) \mid T \text{ is an } n\text{-vertex ternary tree}\}$  describes all product-preserving  $n$ -qubit ternary tree transformations.  $\square$

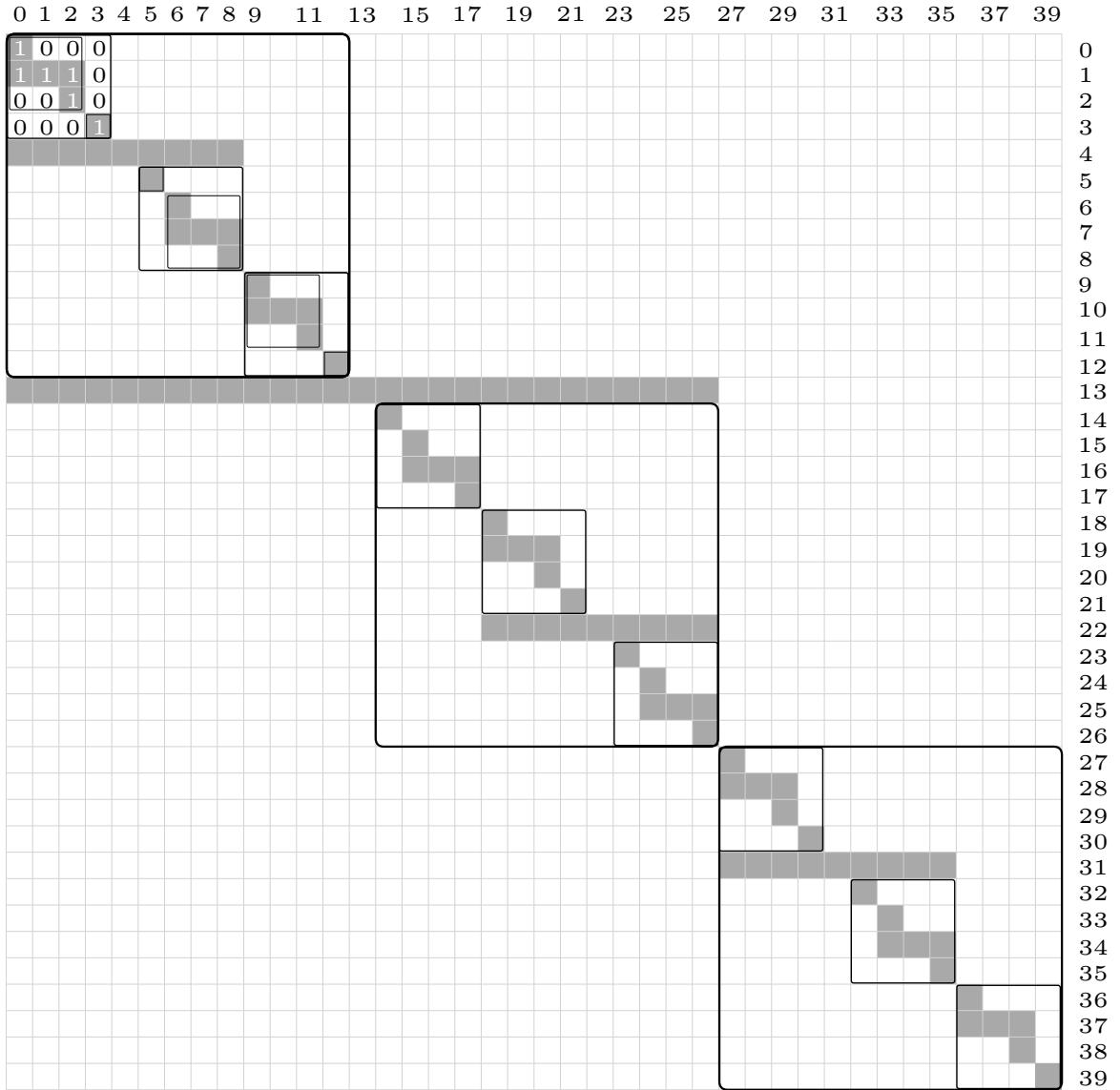
**Example 3.2.9.** (*A ternary tree transformation that linearly encodes the Fock basis.*) Figure 3.4 (c) demonstrates the mapping  $\mathbf{m}(T)$  that arises from the 5-vertex ternary tree from Example 2.2.12. Note that all Fock basis vectors are strictly in the computational basis, and that the matrix  $G_T$  is the one that arises from Lemma 3.2.8.

### 3.2.5 The pruned Sierpinski Tree as a ternary tree transformation

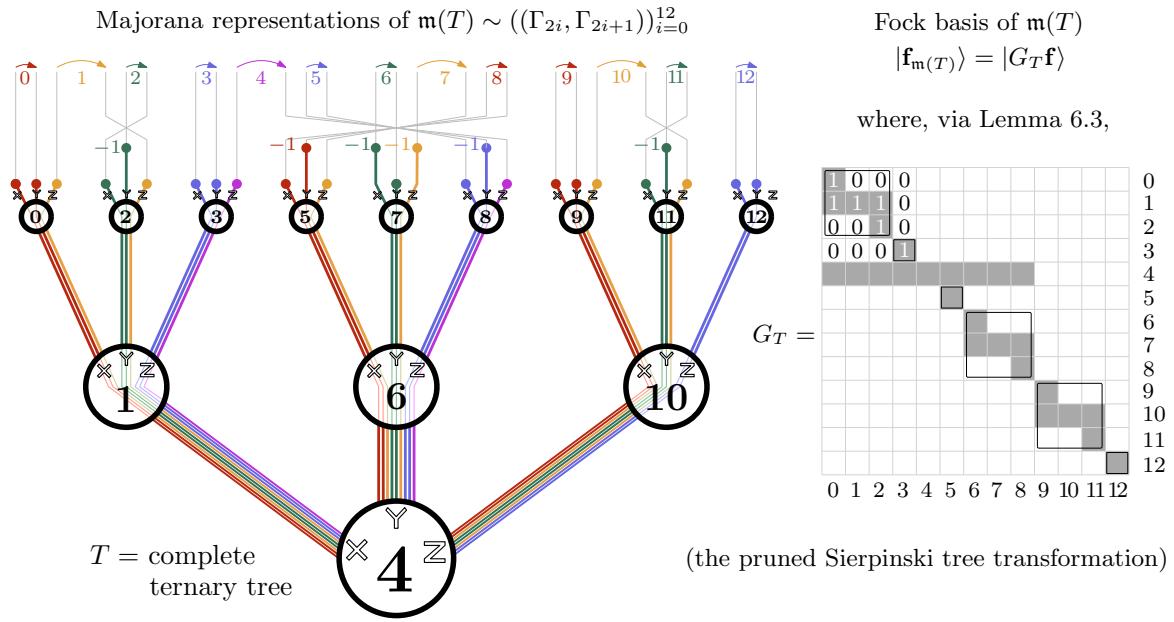
An initial value proposition of ternary tree transformations was that the complete ternary tree graph yielded Pauli operators with the minimum possible average Pauli weight of  $\sim \lceil \log_3(2n+1) \rceil$  qubits [72]. Examples 3.2.4 and 3.2.5 demonstrated a caveat: that ternary tree transformations are not necessarily linear encodings of the Fock basis. However, Theorem 2 has now allayed this concern by showing that every ternary tree transformation is *equivalent* to a linear encoding, in the spirit of Definition 3.1.4. In particular, this means that there is some invertible binary matrix that implements mappings where every Pauli operator has weight  $\sim \lceil \log_3(2n+1) \rceil$ .

In [2], the authors find a classical encoding of the Fock basis via the pruned Sierpinski tree data structure [65] which produces Pauli operators with the minimum average Pauli weight  $\sim \lceil \log_3(2n+1) \rceil$ , matching the promise of ternary tree transformations. Using Theorem 2 shows that these two classes of mappings are equivalent: for the complete ternary tree  $T$ , we find the invertible binary matrix  $G_T$  that implements the  $T$ -based linear encoding  $\mathbf{m}(T)$ , and note that it exactly matches the definition of the pruned Sierpinski tree transform. Figure 3.9 displays the equivalence for  $n = 13$ , and Figure 3.8 shows the recursive definition of the matrix  $G_T$  for increasing  $n = 1, 4, 13, 40, \dots$ .

# $G_{\mathfrak{m}_{TT}}$



**Figure 3.8** The invertible binary matrix  $G_{\mathfrak{m}_{TT}}$  for the 40–vertex complete ternary tree  $T$ , which describes the linear encoding  $\mathfrak{m}(T)$  of the Fock basis  $|\mathbf{f}_{\mathfrak{m}(T)}\rangle = |G_{\mathfrak{m}_{TT}} \mathbf{f}\rangle$  for all  $\mathbf{f} \in \mathbb{Z}_2^{40}$ . Complete ternary trees have  $1, 4, 13, 40, \dots, 3k+1$  vertices for  $k \in \mathbb{N}$ . The outlined squares are a visual guide to the recursive definition of  $G_{\mathfrak{m}_{TT}}$  as the number of vertices in the complete ternary tree  $T$  grows.



**Figure 3.9** The mapping  $\mathfrak{m}(T)$ , where  $T$  is a complete 13–vertex ternary tree, is equal to the pruned Sierpinski triangle transform on 13 qubits. The diagram for the mapping reveals its Majorana representations as the Pauli operator pairs  $((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{12}$ . The state–based definition reveals that  $\mathfrak{m}(T)$  is a linear encoding of the Fock basis of the form  $|\mathbf{f}_{\mathfrak{m}(T)}\rangle = |G_T \mathbf{f}\rangle$  for the invertible binary matrix  $G_T \in \text{GL}_{13}(\mathbb{Z}_2)$  on the right of the figure, which is the Sierpinski tree transform.

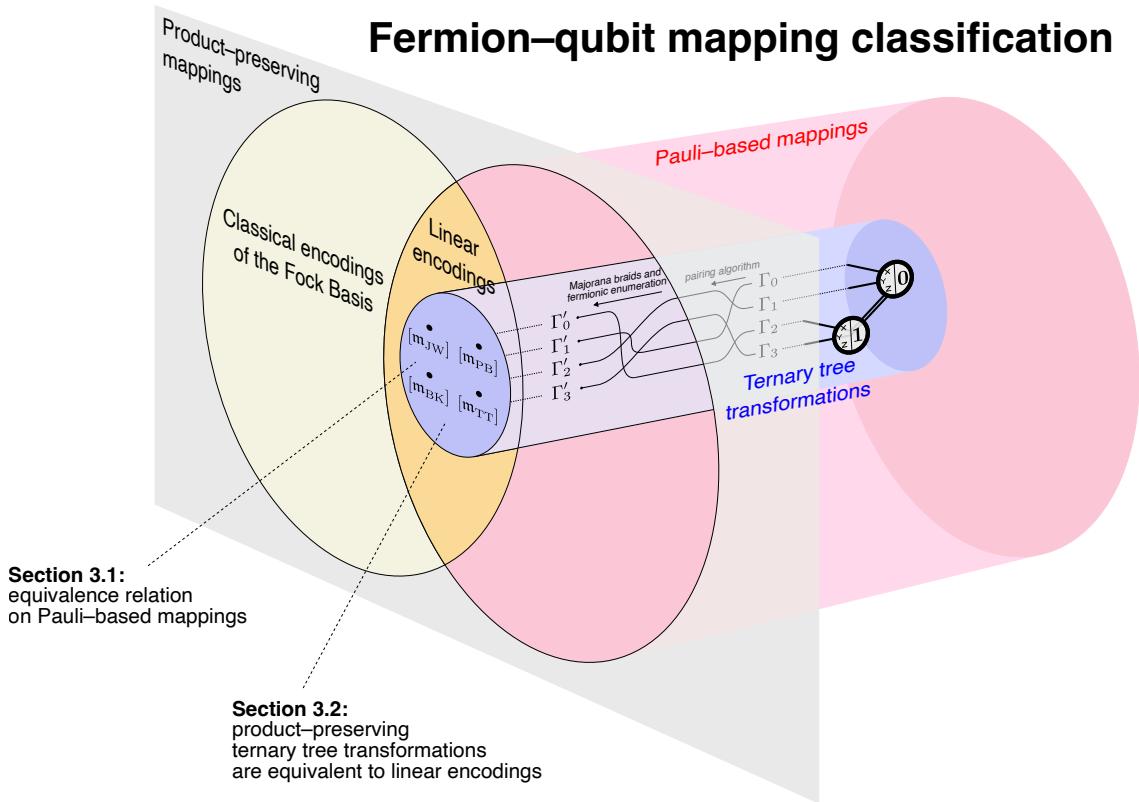
### 3.3 Conclusion

This chapter presents a comprehensive classification of Pauli–based mappings, introducing what we believe are several practical insights to the field. In Section 3.1, we established a practical equivalence relation for Pauli–based mappings, which make up virtually all of the mappings of interest in the field, addressing the redundancies in their notation and highlighting the symmetries that should not affect fermionic simulation algorithms. The notion of equivalence partitions these mappings into templates, distinguishing mappings that differ meaningfully in the Pauli structure of the Majorana representations. In contrast to the first impressions one might gather from the literature in Figure 2.7, our perspective proffers a satisfying and immediate simplification, which Figure 3.10 summarises.

The second half of this chapter combined our definitions and notation from Chapter 2 with this new concept of equivalence upon ternary tree transformations. While ternary tree transformations are already notable for offering the minimum average Pauli operator weight and optimisation algorithms for restricted qubit architectures, we showed that product–preserving ternary tree transformations also benefit from being equivalent to linear encodings of the Fock basis. Our proof not only improves the existing product–preserving Pauli operator pairing algorithm from [73], but also provides a formula to determine the unique linear encoding for each ternary tree graph.

#### 3.3.1 Future work

This chapter opens new directions for discovery and optimisation of fermion–qubit mappings. In the theoretical direction of Section 3.1, separating Pauli–based mappings into templates under our equivalence relation could lead to a deeper understanding of fermion–qubit mappings in general. We suspect that the group of symmetries of this equivalence relation is undocumented in existing literature. Figure 2.3 demonstrates that there are only two two–mode, product–preserving, Pauli–based mappings up to equivalence; Figure 3.1 shows that there are fourteen three–mode mappings of this kind. Creating a database of the templates of  $n$ –mode fermion–qubit mappings would produce a practical classification scheme, where rather than searching over all configurations of Pauli strings or linear encodings of the Fock basis, one need only search through the catalogue of templates for the best mapping on a case-by-case scenario. The existence of the balanced Jordan–Wigner transformation, with its practical advantage of offering the least Pauli weight of all three–mode mappings, also suggests that additional insights lie in exploring mappings that are *not* equivalent to linear encodings. In general, classical



**Figure 3.10** Introducing an equivalence relation for Pauli–based mappings eliminates the distinction between affine and linear encodings and reveals that all product–preserving ternary tree transformations are linear encodings. (Compare with Figure 2.7, which displays the classification before this work.)

encodings or even product–breaking mappings themselves could have unique properties with unforeseen advantages in quantum simulation.

On the practical side, our findings in Section 3.2 provide important context for computational searches for optimal fermion–qubit mappings. We have demonstrated that any search for linear encodings inherently includes ternary tree transformations as a subset, indicating that current algorithms might already be identifying these mappings without explicitly realising. Our results provide the necessary tools to identify whether any linear encoding is a ternary tree transformation. Computational searches [67, 73–75] are often tailored to real–world problems, such as studying Hamiltonians in quantum chemistry applications, and it would be interesting to see if ternary tree transformations tended to outperform other types of linear encodings in practical applications.

# Chapter 4

## Optimising fermion–qubit mappings

This chapter uses the language of our universal definition in Chapter 2 and the templates of our classification in Chapter 3 to define formal optimisation problems for fermion–qubit mappings. As opposed to many computational optimisations in the literature, our formalisation for Jordan–Wigner transformations reduces to graph problems with known solutions, allowing us to provide an exact prescription for the optimal mapping for fermions interacting in a square lattice system from within the Jordan–Wigner template.

Simulating fermionic systems on a quantum computer requires a high–performing fermion–qubit mapping. A key characteristic of an efficient mapping is its ability to translate fermionic interactions into local qubit interactions: the more effectively the mapping achieves this task, the less costly the qubit Hamiltonian. Notions of fermion–qubit mappings that are optimal with regard to many practical cost functions emerge as a result. Section 4.1 formalises optimisation problems for fermion–qubit mappings, introducing Pauli weight functions that serve as cost functions of fermionic Hamiltonians for the rest of this thesis.

In Section 4.2, we show that fermionic relabelling is the only labelling of our classification system that alters the value of Pauli weight functions. Within each template, fermionic labelling thus completely determines the optimal fermion–qubit mapping. We establish that the hopping terms of fermionic Hamiltonians correspond to non–local qubit Hamiltonians, and that optimising fermionic labelling of Jordan–Wigner transformations to minimise Pauli weight functions corresponds to well–known “ $p$ –sum” vertex labelling problems in graph theory.

Our main example of optimisation is Theorem 3, the minimisation of the average Pauli weight in Jordan–Wigner transforms of square–lattice fermionic systems. We replicate Mitchison and Durbin’s forty–year–old solution to the 1–sum, or “edgesum”

problem for square lattice graphs in Section 4.3, and repurpose the result to determine the analytically optimal mapping within the Jordan–Wigner template. The novelty of this approach is its provable optimality, and that it introduces a way to improve the choice of fermion–qubit mappings without using additional resources such as ancilla qubits. This leads to qubit Hamiltonians consisting of terms with average Pauli weights  $\sim 13.9\%$  less than the previous standard.

## 4.1 Pauli–based mappings as optimisation problems with cost functions

This section formalises fermionic Hamiltonians as sums of products of Majorana operators, and qubit Hamiltonians as sums of Pauli operators, for the sake of defining cost functions of qubit Hamiltonians. This, in turn, positions fermion–qubit mappings as search parameters for optimisation problems. In the quantum sciences, the typical fermionic interactions of interest are first– and second–order interaction terms between particles in physical modes. Since establishing that Majorana operators are the building blocks of fermion–qubit mappings in Definition 2.1.7, this thesis has continued to work in the Majorana framework rather than with creation and annihilation operators, which are more common in the computational and experimental literature of fermionic simulation, e.g. [66, 87]. Definition 4.1.1 sets the standard for fermionic and qubit Hamiltonians for both this chapter and Chapter 5.

**Definition 4.1.1.** (*Majorana product decomposition of fermionic Hamiltonians and Pauli decomposition of qubit Hamiltonians.*) Let  $\hat{H}_{\text{fermion}}$  be an  $n$ –mode fermionic Hamiltonian. The *Majorana product decomposition* of  $\hat{H}_{\text{fermion}}$  is the sum

$$\hat{H}_{\text{fermion}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}, \quad (4.1)$$

where the nonzero coefficients  $c_{\hat{h}} \in \mathbb{C}$  ensure the Hermiticity of the Hamiltonian and the set of terms  $\Lambda$  is a subset of the products of  $n$ –mode Majorana operators,

$$\Lambda \subseteq \left\{ \hat{1}, \hat{\gamma}_0, \dots, \hat{\gamma}_{2n-1}, (\hat{\gamma}_0 \hat{\gamma}_1), \dots, (\hat{\gamma}_0 \hat{\gamma}_1 \dots \hat{\gamma}_{2n-1}) \right\} \subset B(\mathcal{H}_f^n). \quad (4.2)$$

Let  $\mathbf{m}$  be an  $n$ -mode Pauli-based mapping, and let  $H_{\text{qubit}}$  be the qubit Hamiltonian that  $\mathbf{m}$  produces from  $\hat{H}_{\text{fermion}}$ . The *Pauli decomposition* of  $H_{\text{qubit}}$  is the sum

$$H_{\text{qubit}} = \mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h, \quad (4.3)$$

where, for each term  $\hat{h} \in \Lambda$  of the fermionic Hamiltonian, there is a corresponding  $n$ -qubit Pauli term  $h = \mathbf{m} \cdot \hat{h} \cdot \mathbf{m}^\dagger \in \mathcal{P}_n$  in the qubit Hamiltonian.

The weights or geometric locality of the terms in a Pauli decomposition make up a class of cost functions we call the *Pauli weight functions* of qubit Hamiltonians.

**Definition 4.1.2.** (*Pauli weight functions.*) A *Pauli weight function* is a cost function  $C : \mathcal{U}_n \rightarrow \mathbb{R}$  that extends to the linear combinations of unitary terms in a qubit Hamiltonian via

$$C(\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger) = C\left(\{W(\mathbf{m} \cdot \hat{h} \cdot \mathbf{m}^\dagger) \mid \hat{h} \in \Lambda\}\right), \quad (4.4)$$

where  $W : \mathcal{P} \rightarrow [n]$  returns the weight of a Pauli operator and  $\mathbf{m}$  is any Pauli-based mapping.

Minimising Pauli weight functions encapsulates the mindset that led to the Bravyi–Kitaev [64] and ternary tree transformations [71, 72]. In this chapter, we use the equivalence relation of Pauli-based mappings from Chapter 3 to show that optimisation problems for Jordan–Wigner transformations have a natural correspondence to well-known graph problems with many known solutions, exploring overlooked opportunities for improvements in Jordan–Wigner techniques.

**Example 4.1.3.** (*Pauli weight and locality cost functions for qubit Hamiltonians.*) This chapter discusses three cost functions for qubit Hamiltonians of Pauli-based mappings. Given a fermionic Hamiltonian  $\hat{H}_{\text{fermion}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}$  for some set of products of Majorana operators  $\Lambda \subset B(\mathcal{H}_f^n)$  and a Pauli-based mapping  $\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$ :

- a) The *average Pauli weight* of the qubit Hamiltonian  $\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger$  is the Pauli weight function

$$C_{\text{avg}}(\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger) = \frac{1}{|\Lambda|} \sum_{\hat{h} \in \Lambda} W(\mathbf{m} \cdot \hat{h} \cdot \mathbf{m}^\dagger). \quad (4.5)$$

The average Pauli weight of a qubit Hamiltonian could be a limiting resource in near-term algorithms such as the variational quantum eigensolver (VQE) [53],

where  $C_{\text{avg}}(H_{\text{qubit}})$  represents the total number of single-qubit measurements involved in each step of the VQE.

- b) The *maximum Pauli weight* of the qubit Hamiltonian  $\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger$  is the Pauli weight function

$$C_{\max}(\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger) = \max_{\hat{h} \in \Lambda} \left\{ W(\mathbf{m} \cdot \hat{h} \cdot \mathbf{m}^\dagger) \right\}. \quad (4.6)$$

The maximum Pauli weight of a qubit Hamiltonian is a complexity-theoretic parameter in the cost of long-term simulation algorithms involving phase estimation, such as Trotterisation [32]. For example, a  $k$ -local Hamiltonian has  $C_{\max}(H_{\text{qubit}}) = k$ .

- c) The *Pauli depth* of the qubit Hamiltonian  $\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger$  is the function

$$C_{\text{depth}}(\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger) = \min_k \left\{ k \in \mathbb{N} \mid \bigsqcup_{i=0}^{k-1} \Lambda_i = \Lambda, \right. \\ \left. \{\mathbf{m} \cdot \hat{h} \cdot \mathbf{m}^\dagger\}_{\hat{h} \in \Lambda_i} \text{ non-overlapping for all } i \in [k] \right\}, \quad (4.7)$$

the size of the smallest partition of qubit Hamiltonian terms  $\{\mathbf{m} \cdot \hat{h} \cdot \mathbf{m}^\dagger\}_{\hat{h} \in \Lambda}$  into non-overlapping sets. The Pauli depth of a qubit Hamiltonian might correspond to the number of distinct timesteps required to run all Hamiltonian terms in parallel [96], indicating, for example, the total clock time required to perform all measurements during an iteration of a VQE algorithm.

## 4.2 Optimising fermionic labellings

This section introduces an optimisation problem for the fermionic labellings of Pauli-based mappings, which leads to qubit Hamiltonians that minimise practical Pauli weight cost functions. Section 4.2.1 establishes a broad class of problem Hamiltonians, arising both from general mappings and then specifically the Jordan–Wigner transformation, defining optimality in terms of practical cost functions for quantum computers. Section 4.2.2 derives expressions for the cost functions of the search, which relate to the graph theory problems in from Section 4.2.3. Section 4.2.4 states Theorem 3 for minimising the average Pauli weight of a Jordan–Wigner transformation of a system of fermions interacting in a square lattice, with proof in Section 4.3.

### 4.2.1 Problem Hamiltonians and hopping terms

The most general second-quantised Hamiltonians of the form in Equation 1.51 have a Majorana product decomposition of the form:

$$\hat{H}_{\text{fermion}} = \sum_{i,j \in [n]} (c_{ij}) \hat{a}_i^\dagger \hat{a}_j + \sum_{i,j,k,l \in [n]} (c_{kl}^{ij}) \hat{a}_i^\dagger \hat{a}_j \hat{a}_k^\dagger \hat{a}_l \quad (4.8)$$

$$\begin{aligned} &= \sum_{i \in [n]} \left( (c_{ii}) + (c_{ii}^{ii}) + \sum_{j \neq i} (c_{ji}^{ij}) \right) \hat{a}_i^\dagger \hat{a}_i \\ &\quad + \sum_{i < j \in [n]} \left( (c_{ij}) \hat{a}_i^\dagger \hat{a}_j + (c_{ij})^* \hat{a}_j^\dagger \hat{a}_i \right) + \sum_{i < j} (\tilde{c}_{jj}^{ii}) \hat{a}_i^\dagger \hat{a}_i \hat{a}_j^\dagger \hat{a}_j \\ &\quad + \sum_{i \neq j \in [n]} \hat{a}_i^\dagger \hat{a}_i \left( (c_{ij}^{ii}) \hat{a}_i^\dagger \hat{a}_j + (c_{ii}^{ji})^* \hat{a}_j^\dagger \hat{a}_i \right) + \sum_{\substack{k \in [n] \\ i < j \in [n] \\ k \neq i, j}} \hat{a}_k^\dagger \hat{a}_k \left( (\tilde{c}_{kk}^{ij}) \hat{a}_i^\dagger \hat{a}_j + (\tilde{c}_{kk}^{ij})^* \hat{a}_j^\dagger \hat{a}_i \right) \\ &\quad + \sum_{i < j < k < l \in [n]} \left( \hat{a}_i^\dagger \hat{a}_j \left( (\tilde{c}_{kl}^{ij}) \hat{a}_k^\dagger \hat{a}_l + (\tilde{c}_{lk}^{ij}) \hat{a}_l^\dagger \hat{a}_k \right) + \hat{a}_j^\dagger \hat{a}_i \left( (\tilde{c}_{kl}^{ij})^* \hat{a}_l^\dagger \hat{a}_k + (\tilde{c}_{lk}^{ij})^* \hat{a}_k^\dagger \hat{a}_l \right) \right. \\ &\quad \left. + (\tilde{c}_{jk}^{il}) \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k \hat{a}_l + (c_{jk}^{il})^* \hat{a}_i \hat{a}_j \hat{a}_k^\dagger \hat{a}_l \right) \end{aligned} \quad (4.9)$$

$$\begin{aligned} &= \frac{1}{2} \sum_{i \in [n]} \left( (c_{ii}) + (c_{ii}^{ii}) + \sum_{j \neq i} (c_{ji}^{ij}) \right) (\hat{1} + i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1}) + \text{quartic terms} \quad (4.10) \\ &\quad + \frac{1}{2} \sum_{i < j \in [n]} (\text{Re}(c_{ij})(\hat{\gamma}_{2i}\hat{\gamma}_{2j} + \hat{\gamma}_{2i+1}\hat{\gamma}_{2j+1}) + i\text{Im}(c_{ij})(\hat{\gamma}_{2i}\hat{\gamma}_{2j+1} - \hat{\gamma}_{2i+1}\hat{\gamma}_{2j})) , \end{aligned}$$

omitting the terms that are quartic in the Majorana operators for brevity, and featuring the coefficients  $\tilde{c}_{kl}^{ij} := (c_{kl}^{ij} - c_{kj}^{il} - c_{il}^{kj} + c_{ij}^{kl}) = c_{rs}^{pq} \delta_{ik}^{pr} \delta_{jk}^{qs}$ .

The quartic interactions that appear in nature are two-body interactions, such as the repulsion forces of electrons [54] or adjacent fermions on the lattice in the Hubbard model [29]. Rather than involving four separate modes, physically-motivated quartic terms describe interactions between pairs of operators on only two modes, taking the form

$$\hat{a}_i^\dagger \hat{a}_i \hat{a}_j^\dagger \hat{a}_j = \frac{1}{4} (\hat{1} + i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1})(\hat{1} + i\hat{\gamma}_{2j}\hat{\gamma}_{2j+1}) . \quad (4.11)$$

Quartic terms make diagonalisation of the fermionic Hamiltonian  $\hat{H}_{\text{fermion}}$  exponentially difficult using known classical algorithms. Restricting Hamiltonians of interest to include only quartic interactions of Equation 4.11, the problem Hamiltonians in Equation 4.10

take the form

$$\hat{H}_{\text{fermion}} \quad (4.12)$$

$$= \underbrace{\frac{1}{2} \sum_{i \in [n]} (c_{ii} + c_{ii}^{ii} + \sum_{j \neq i} c_{ji}^{ij}) (\hat{1} + i\hat{\gamma}_{2i}\hat{\gamma}_{2j+1}) + \frac{1}{4} \sum_{(i,j) \in G} \tilde{c}_{jj}^{ii} (\hat{1} + i\hat{\gamma}_{2i}\hat{\gamma}_{2i+1})(\hat{1} + i\hat{\gamma}_{2j}\hat{\gamma}_{2j+1})}_{\text{on-site}}$$

$$+ \underbrace{\frac{1}{2} \sum_{(i,j) \in G} (\text{Re}(c_{ij})(\hat{\gamma}_{2i}\hat{\gamma}_{2j} + \hat{\gamma}_{2i+1}\hat{\gamma}_{2j+1}) + i \text{Im}(c_{ij})(\hat{\gamma}_{2i}\hat{\gamma}_{2j+1} - \hat{\gamma}_{2i+1}\hat{\gamma}_{2j}))}_{\text{hopping terms}}$$

$$= \hat{H}_{\text{fermion}}^{\text{on-site}} + \hat{H}_{\text{fermion}}^{\text{hopping}}, \quad (4.13)$$

where  $G = G(\hat{H}_{\text{fermion}})$  is the *interaction graph of*  $\hat{H}_{\text{fermion}}$ , which has vertex set  $[n]$  of fermionic modes and edges  $\{(i,j) \mid c_{ij} \neq 0 \text{ or } \tilde{c}_{jj}^{ii} \neq 0\}$ . Figure 4.1 depicts an example of a fermionic interaction graph. The collection of terms under the banners of ‘on–site’ and ‘hopping’ refers to the physical interpretation of the Hamiltonian terms: hopping terms represent the movement of a fermion between modes that correspond to adjacent vertices of the graph, and on–site terms reflect the occupation and repulsion energies of fermions present in the system.

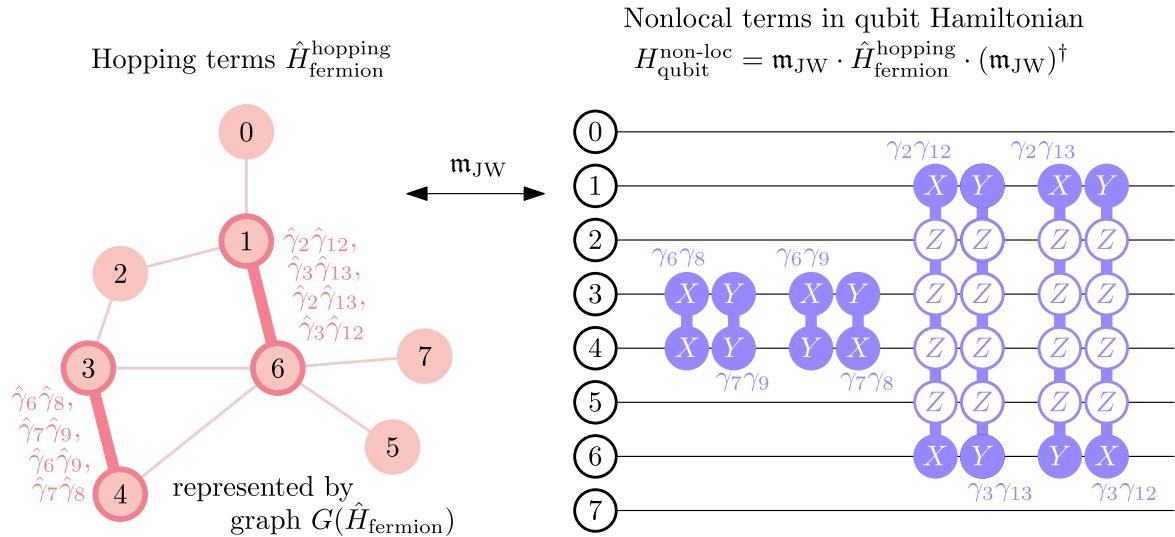
Applying a Pauli–based mapping of the form  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  to the fermionic Hamiltonian  $\hat{H}_{\text{fermion}}$  produces the qubit Hamiltonian

$$\begin{aligned} H_{\text{qubit}} &= \mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger \\ &= \frac{1}{2} \sum_{i \in [n]} (c_{ii} + c_{ii}^{ii} + \sum_{j \neq i} c_{ji}^{ij}) (\mathbb{1}^{\otimes n} + i\Gamma_{2i}\Gamma_{2i+1}) \\ &\quad + \frac{1}{4} \sum_{(i,j) \in G} \tilde{c}_{jj}^{ii} (\mathbb{1}^{\otimes n} + i\Gamma_{2i}\Gamma_{2i+1})(\mathbb{1}^{\otimes n} + i\Gamma_{2j}\Gamma_{2j+1}) \\ &\quad + \frac{1}{2} \sum_{(i,j) \in G} (\text{Re}(c_{ij})(\Gamma_{2i}\Gamma_{2j} + \Gamma_{2i+1}\Gamma_{2j+1}) + i \text{Im}(c_{ij})(\Gamma_{2i}\Gamma_{2j+1} - \Gamma_{2i+1}\Gamma_{2j})) . \end{aligned} \quad (4.14)$$

The weights of the nontrivial terms in  $H_{\text{qubit}}$  parameterise the cost of the Hamiltonian under any Pauli weight function  $C : \mathcal{U}_n \rightarrow \mathbb{R}$ . The weights of these terms are ultimately a function of the fermionic labels  $(i, j)$  of each edge in  $G$ , which dictate the cost of the Hamiltonian:

$$C(H_{\text{qubit}}) = C \left( \{W(\Gamma_{2i}\Gamma_{2j}), W(\Gamma_{2i}\Gamma_{2j+1}), W(\Gamma_{2i+1}\Gamma_{2j+1}), W(\Gamma_{2i}\Gamma_{2i+1})\}_{(i,j) \in G} \right) \quad (4.15)$$

$$\equiv C(\{(i, j)\}_{(i,j) \in G}) . \quad (4.16)$$



**Figure 4.1** Application of the Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}}$  to hopping terms in a fermionic Hamiltonian.

Using a Pauli-based mapping that differs from  $\mathbf{m}$  via fermionic relabelling corresponds to varying the vertex labelling of the graph  $G$ , and can produce qubit Hamiltonians of different costs. None of the other four labelling choices of the diagrammatic notation for Pauli-based mappings – qubit labels, local Pauli basis labels, ordering of Paulis within pairs, or the signs of the Pauli operators – alter the cost function in Equation 4.16.

In particular, the Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}} \sim ((\gamma_{2i}, \gamma_{2i+1}))_{i=0}^{n-1}$  produces the qubit Hamiltonian

$$H_{\text{qubit}} = \mathbf{m}_{\text{JW}} \cdot \hat{H}_{\text{fermion}} \cdot (\mathbf{m}_{\text{JW}})^{\dagger} \quad (4.17)$$

$$= \underbrace{\frac{1}{2} \sum_{i \in [n]} (c_{ii} + c_{ii}^{ii} + \sum_{j \neq i} c_{ji}^{ij})(\mathbb{1}^{\otimes n} - Z_i)}_{\text{local}} + \frac{1}{4} \sum_{(i,j) \in G} (\tilde{c}_{jj}^{ii})(\mathbb{1}^{\otimes n} - Z_i)(\mathbb{1}^{\otimes n} - Z_j) \quad (4.18)$$

$$+ \underbrace{\frac{1}{2} \sum_{(i,j) \in G} \left( \text{Re}(c_{ij})(X_i X_j + Y_i Y_j) + \text{Im}(c_{ij})(X_i Y_j + Y_i X_j) \right) \left( \bigotimes_{k=i+1}^{j-1} Z_k \right)}_{\text{non-local}} \quad (4.19)$$

$$= H_{\text{qubit}}^{\text{loc}} + H_{\text{qubit}}^{\text{non-loc}},$$

where  $H_{\text{qubit}}^{\text{loc}} = \mathbf{m}_{\text{JW}} \cdot \hat{H}_{\text{fermion}}^{\text{on-site}} \cdot (\mathbf{m}_{\text{JW}})^{\dagger}$  consists of entirely local terms with Pauli weights between 0 and 2 independent of fermionic labelling, and  $H_{\text{qubit}}^{\text{non-loc}} = \mathbf{m}_{\text{JW}} \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathbf{m}_{\text{JW}})^{\dagger}$  consists of non-local terms with Pauli weights that can take values between 2 and  $n$ .

The only terms that influence a Pauli weight cost function  $C$  of the Jordan–Wigner qubit Hamiltonian in Equation 4.17 originate in  $H_{\text{qubit}}^{\text{non-loc}}$ , giving the cost function a dependence on the differences between the labels of adjacent fermionic modes:

$$C(H_{\text{qubit}}^{\text{loc}}) = \mathcal{O}(1) \quad (4.20)$$

$$C(H_{\text{qubit}}^{\text{non-loc}}) = C\left(\left\{W\left(\bigotimes_{k=i+1}^{j-1} Z_k\right)\right\}_{(i,j)\in G}\right) \equiv C(|i-j|)_{(i,j)\in G} \quad (4.21)$$

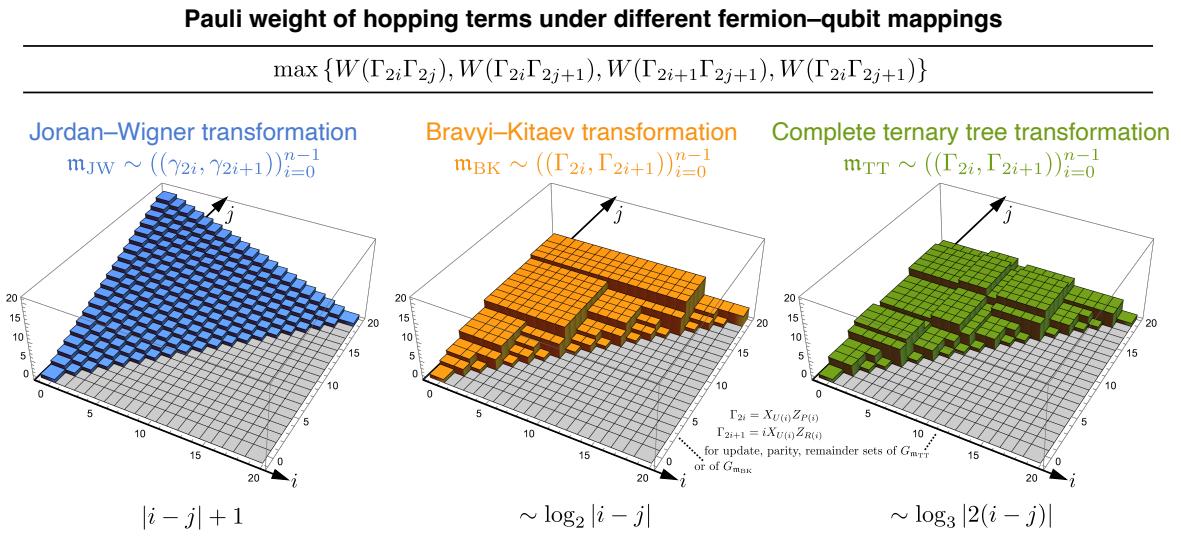
$$\implies C(H_{\text{qubit}}) = C(H_{\text{qubit}}^{\text{non-loc}}) + C(H_{\text{qubit}}^{\text{loc}}) = C(|i-j|)_{(i,j)\in G} + \mathcal{O}(1). \quad (4.22)$$

The simple dependence of Pauli weight functions of  $H_{\text{qubit}}$  on the difference in labels of interacting fermionic modes, which makes an ideal playground for classical optimisation, is unique to the Jordan–Wigner transformation. For example, the Bravyi–Kitaev and complete ternary tree transformations map products of Majorana operators  $\hat{\gamma}_{2i}\hat{\gamma}_{2j}$  into Pauli operators  $\Gamma_{2i}\Gamma_{2j}$  that act on  $\sim \lceil \log_2 n \rceil$  and  $\sim \lceil \log_3(2n+1) \rceil$  qubits, respectively [66, 72]. While the Pauli weights of these terms under the Bravyi–Kitaev and complete ternary transformations are cheaper than via the Jordan–Wigner transformation, the cost function  $W(\Gamma_{2i}\Gamma_{2j})$  itself is more complex to evaluate. Figure 4.2 summarises the simplicity of evaluating the Pauli weights of hopping terms under the Jordan–Wigner transformation. The optimisation routines in this chapter and Chapter 5 exploit this simplicity.

### 4.2.2 Optimising fermionic labelling within templates

The only labelling symmetry of Pauli–based mappings from Definition 3.1.1 that does not fix the Pauli weight cost of a fermionic Hamiltonian is fermionic labelling. Remark 4.2.1 makes the observation that the labelling scheme is thus the only necessary parameter when optimising within the template of a fermion–qubit mapping. This section then phrases optimisation problems for Jordan–Wigner transformations as graph problems.

**Remark 4.2.1.** (*Fermionic labelling is sufficient to determine the optimal mapping within a template.*) Let  $\mathbf{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$  be an  $n$ –mode Pauli–based mapping and let  $C : \mathcal{U}_n \rightarrow \mathbb{R}$  be a Pauli weight function. For any  $n$ –mode fermionic Hamiltonian  $\hat{H}_{\text{fermion}}$  of the form in Equation 4.12, all fermion–qubit mappings  $\mathbf{m}' \in [\mathbf{m}]$  in the template of  $\mathbf{m}$  yield qubit Hamiltonians with the same cost  $C(\mathbf{m}' \cdot \hat{H}_{\text{fermion}} \cdot (\mathbf{m}')^\dagger) = C(\mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger)$  unless the mapping differs from  $\mathbf{m}$  via fermionic labelling. When optimising over the template  $[\mathbf{m}]$ , it is therefore sufficient to consider only mappings of



**Figure 4.2** The Pauli weight of operators representing fermionic hopping terms is far simpler to evaluate under the Jordan–Wigner transformation than other mappings. This figure displays the 3D surfaces of the Pauli weights of the hopping terms under the Jordan–Wigner, Bravyi–Kitaev and complete ternary tree transformations. The simplicity of the Jordan–Wigner surface is related to the fact that optimisations of fermionic labellings in Jordan–Wigner transformations correspond to well-defined graph problems.

the form

$$\mathbf{m}' = \mathbf{m}(\sigma) \sim ((\Gamma_{2\sigma(i)}, \Gamma_{2\sigma(i)+1}))_{i=0}^{n-1} \quad \text{for } \sigma \in S_n. \quad (4.23)$$

For the Pauli weight function  $C$ , the optimal fermionic labelling  $\sigma^* \in S_n$  is a permutation that satisfies

$$\sigma^* = \arg \min_{\sigma \in S_n} C \left( \mathbf{m}(\sigma) \cdot \hat{H}_{\text{fermion}} \cdot (\mathbf{m}(\sigma))^\dagger \right). \quad (4.24)$$

Treating the template of a fermion–qubit mapping as a search space of size  $n!$  leads to new optimisation routines for fermion–qubit mappings. Searching an entire template is still classically intractable, but in comparison to searching over the space of all fermion–qubit mappings, it presents simpler optimisation problems which have heuristics [75] and even, in some cases, exact solutions.

The Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}}(\sigma)$  produces the qubit Hamiltonian

$$H_{\text{qubit}}(\sigma) = \mathbf{m}_{\text{JW}}(\sigma) \cdot \hat{H}_{\text{fermion}} \cdot (\mathbf{m}_{\text{JW}}(\sigma))^\dagger \quad (4.25)$$

$$\begin{aligned} &= \underbrace{\frac{1}{2} \sum_{i \in [n]} (c_{ii} + c_{ii}^{ii} + \sum_{j \neq i} c_{ji}^{ij})(\mathbb{1}^{\otimes n} - Z_{\sigma(i)}) + \frac{1}{4} \sum_{(i,j) \in G} (\tilde{c}_{jj}^{ii})(\mathbb{1}^{\otimes n} - Z_{\sigma(i)})(\mathbb{1}^{\otimes n} - Z_{\sigma(j)})}_{\text{local}} \\ &\quad + \underbrace{\frac{1}{2} \sum_{(i,j) \in G} \left( \begin{array}{l} \text{Re}(c_{ij})(X_{\sigma(i)}X_{\sigma(j)} + Y_{\sigma(i)}Y_{\sigma(j)}) \\ + \text{Im}(c_{ij})(X_{\sigma(i)}Y_{\sigma(j)} + Y_{\sigma(i)}X_{\sigma(j)}) \end{array} \right) \left( \bigotimes_{k=\sigma(i)+1}^{\sigma(j)-1} Z_k \right)}_{\text{non-local}} \end{aligned} \quad (4.26)$$

$$= H_{\text{qubit}}^{\text{loc}}(\sigma) + H_{\text{qubit}}^{\text{non-loc}}(\sigma), \quad (4.27)$$

which bears similarity to the qubit Hamiltonian of the canonical Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}}$  in Equation 4.18, save for the action of the permutation  $\sigma$  on the qubit labels. Pauli weight functions of the Hamiltonian in Equation 4.26 thus depend on the difference in the permuted labels of interacting fermionic modes:

$$C(H_{\text{qubit}}(\sigma)) = C(H_{\text{qubit}}^{\text{non-loc}}(\sigma)) + C(H_{\text{qubit}}^{\text{loc}}(\sigma)) \quad (4.28)$$

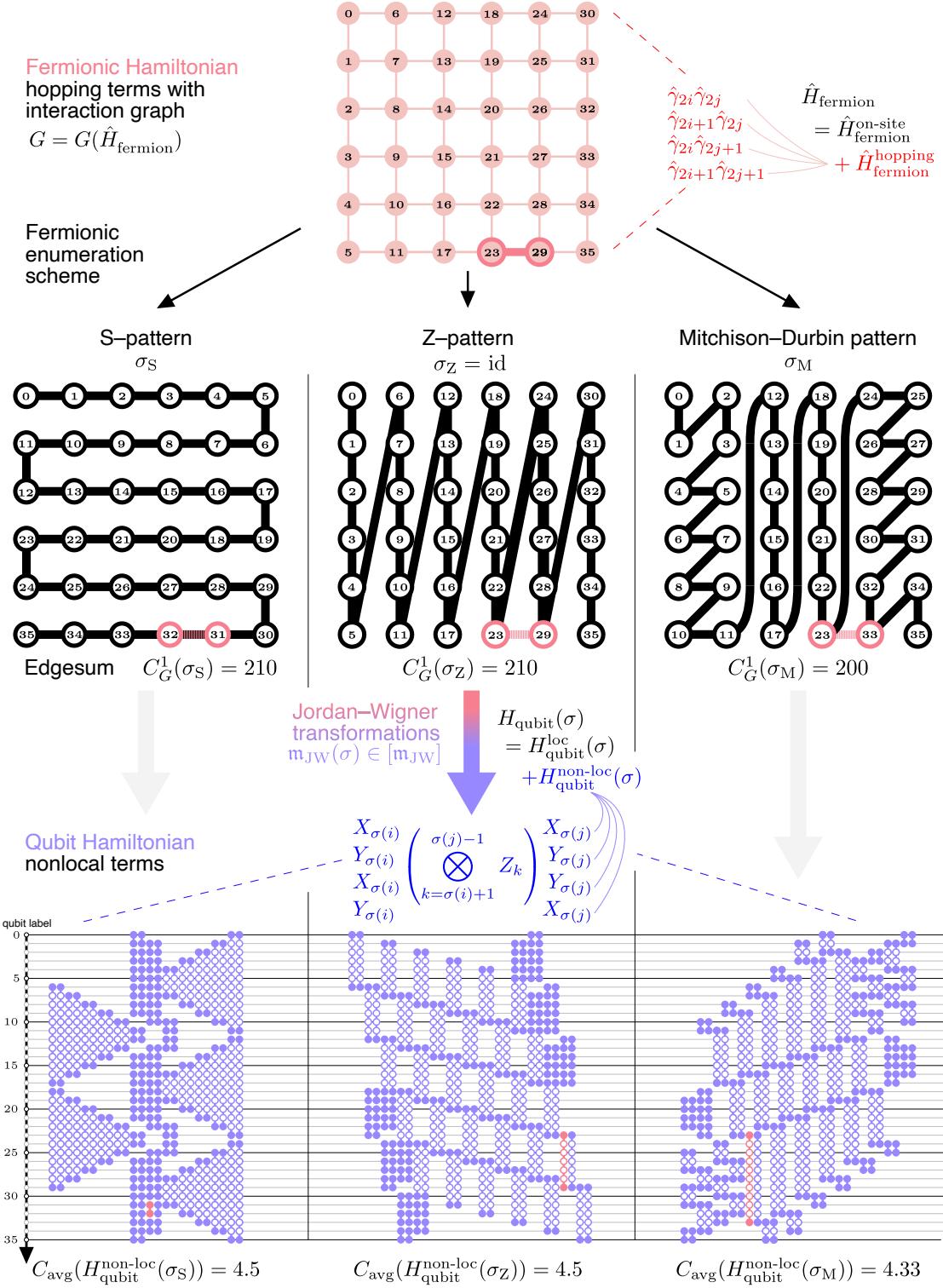
$$= C(\{|\sigma(i) - \sigma(j)|\}_{(i,j) \in G}) + \mathcal{O}(1). \quad (4.29)$$

Throughout this chapter and Chapter 5, permutations that represent vertex relabelling of an  $n$ –vertex graph  $G$  appear in diagrams as paths through  $G$ , starting at the vertex  $i$  with label  $\sigma(i) = 0$  and ending at the vertex  $j$  with label  $\sigma(j) = n$ . It is convenient to define the  $p$ –sum of the vertex labelling scheme  $\sigma \in S_n$  for a labelled  $n$ –vertex graph  $G$  to be

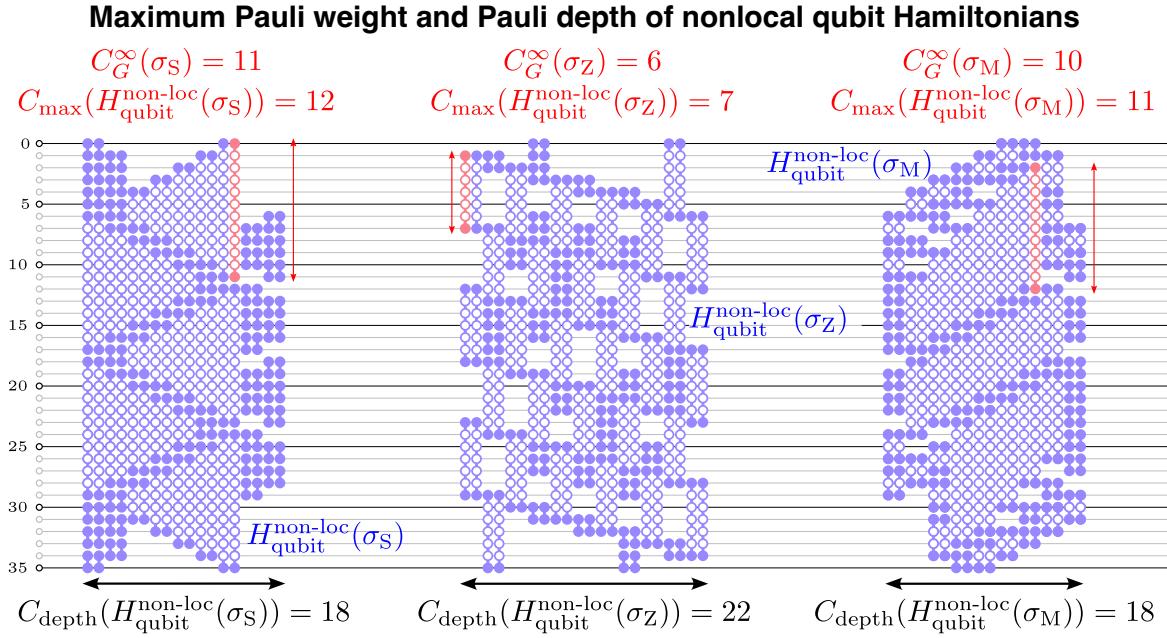
$$C_G^p(\sigma) := \left( \sum_{(i,j) \in G} |\sigma(i) - \sigma(j)|^p \right)^{1/p}. \quad (4.30)$$

For Jordan–Wigner–transformed Hamiltonians, many Pauli cost functions of interest revert to  $p$ –sums with some inspection.

**Example 4.2.2.** (*Pauli cost functions for Jordan–Wigner transformations in terms of  $p$ –sums.*) Let  $\hat{H}_{\text{fermion}} = \hat{H}_{\text{fermion}}^{\text{on-site}} + \hat{H}_{\text{fermion}}^{\text{hopping}}$  be an  $n$ –mode fermionic Hamiltonian of the form in Equation 4.12 with interaction graph  $G = G(\hat{H}_{\text{fermion}})$ . When searching the Jordan–Wigner template  $[\mathbf{m}_{\text{JW}}]$  for optimal fermionic labellings, some cost functions from Example 4.1.3 of the non-local terms  $H_{\text{qubit}}^{\text{non-loc}}(\sigma) = \mathbf{m}_{\text{JW}}(\sigma) \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathbf{m}_{\text{JW}}(\sigma))^\dagger$  simplify to  $p$ –sums of the graph  $G$ :



**Figure 4.3** An illustration depicting how Jordan–Wigner transformations  $m_{\text{JW}}(\sigma)$  for  $\sigma \in S_n$  produce qubit Hamiltonians with non-local terms of differing average Pauli weights. The example uses  $\hat{H}_{\text{fermion}}$  with the interaction graph  $G(\hat{H}_{\text{fermion}})$  of a  $6 \times 6$  square lattice and fermionic labellings  $\sigma_S, \sigma_Z, \sigma_M$ . The figure highlights the term  $\hat{\gamma}_{46}\hat{\gamma}_{58}$  and its representations.



**Figure 4.4** Comparison of the Pauli depths and Maximum Pauli weights of the non-local terms in Jordan–Wigner transformations  $\mathfrak{m}_{\text{JW}}(\sigma)$  of fermions interacting on a  $6 \times 6$  square lattice, for three fermionic labellings:  $\sigma_S, \sigma_Z, \sigma_M \in S_{36}$ .

a) The average Pauli weight (Example a)) relates to the 1–sum, or *edgesum*,

$$C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma)) = \frac{1}{|G|} C_G^1(\sigma) + 1, \quad (4.31)$$

where  $|G|$  is the number of edges in the graph  $G$ . The extra term of 1 in Equation 4.31 incorporates that the weight of a Pauli operator spanning qubits  $i$  to  $j$  is equal to  $|i - j| + 1$ .

Figure 4.3 illustrates non-local qubit Hamiltonians  $H_{\text{qubit}}^{\text{non-loc}}(\sigma)$  arising from applying different mappings within the Jordan–Wigner template  $[\mathfrak{m}_{\text{JW}}]$  to a fermionic system with the underlying connectivity  $G(\hat{H}_{\text{fermion}})$  of a  $6 \times 6$  square lattice. Three different fermionic labellings – the S–pattern  $\sigma_S$ , Z–pattern  $\sigma_Z$ , and Mitchison–Durbin pattern  $\sigma_M$  [97] – feature in the diagram.

Finding fermionic labellings that minimise the average Pauli weight is equivalent to minimising the edgesum  $C_G^1(\sigma)$ . Section 4.2.3 explains that this involves solving NP–complete decision problems, in the general case. In Section 4.2.4, Theorem 3 shows that the Mitchison–Durbin pattern  $\sigma_M$  minimises the edgesum of the square lattice, an improvement of up to 13.9% upon the S– and Z–patterns.

- b) The maximum Pauli weight (Example b)) is

$$C_{\max}(H_{\text{qubit}}^{\text{non-loc}}(\sigma)) = C_G^\infty(\sigma) + 1, \quad (4.32)$$

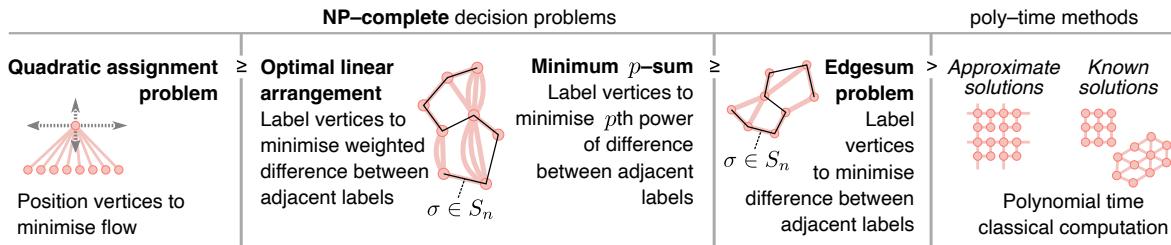
which corresponds to the NP–complete bandwidth problem from Section 4.2.3. If there are  $n = N^2$  fermionic modes and  $G$  is an  $N \times N$  square lattice, the minimum value of the bandwidth  $C_G^\infty$  over all  $\sigma \in S_n$  is  $N$ . Figure 4.4 demonstrates the maximum Pauli weight of a  $6 \times 6$  lattice of fermionic modes for the Jordan–Wigner transformations  $\mathbf{m}_{\text{JW}}(\sigma) \in [\mathbf{m}_{\text{JW}}]$  with  $\sigma = \sigma_S, \sigma_Z, \sigma_M$ , and identifies  $\mathbf{m}_{\text{JW}}(\sigma_Z)$  as a  $C_{\max}$ –optimal mapping in  $[\mathbf{m}_{\text{JW}}]$  for the square lattice fermionic Hamiltonian.

- c) The Pauli measurement depth (Example c)) has no simple expression in terms of the  $p$ –sums of the graph  $G$ . Figure 4.4 demonstrates the variation that arises in Pauli measurement depths of qubit Hamiltonians from changing only the fermionic labelling of the mapping.
- d) *Other properties.* One may also consider other cost functions. For example, if the quantum computer at hand has a restrictive qubit architecture, the aptest Pauli cost functions may prioritise the geometric locality of qubit Hamiltonian terms over Pauli weight. Section 4.4.1 discusses this scenario.

### 4.2.3 Complexity–theoretical preliminaries

This section introduces the notation and complexity theoretic problems that underpin the results of Section 4.2.4. Awareness of these problems is necessary to find optimal fermionic labellings for Jordan–Wigner transformations. Figure 4.5 summarises the hierarchy of relevant graph problems, which have the following common ingredients:

1. A **graph**  $G$  consisting of edges between vertices in  $[n]$ , and
2. a **weight function**  $w : G \rightarrow \mathbb{R}$ , assigning a value  $w(i, j)$  to each edge  $(i, j) \in G$ , and
3. a set  $L$  of possible **locations** for the vertices, and
4. a **distance function**  $d : L \times L \rightarrow \mathbb{R}$  describing the spatial separation between the locations, and
5. a **vertex labelling**  $\sigma : V \rightarrow L$ , which is an injection describing the location of the vertices in  $G$ . If  $L = [n]$ , then the vertex labelling is simply a permutation  $\sigma \in S_n$ .



**Figure 4.5** Complexity-theoretic hierarchy of the  $p$ -sum problem and its variations.

The optimisation problems in this section share the objective of finding a vertex labelling  $\sigma$  that minimises a **cost function**  $C : \sigma \mapsto C(\sigma) \in \mathbb{R}$ , and each is NP-complete as a decision problem. The problems appear in order of descending complexity: subsequent problems are special cases of earlier ones.

### QUADRATIC ASSIGNMENT

**INSTANCE:** Graph  $G$  with vertices  $[n]$ , vertex locations  $L$ , weight function  $w : G \rightarrow \mathbb{R}$ , distance function  $d : L \times L \rightarrow \mathbb{R}$ .

**PROBLEM:** Find a vertex labelling  $\sigma : V \rightarrow L$  that minimises the cost function

$$C(f) = \sum_{(i,j) \in G} w(i,j) \cdot d(\sigma(i), \sigma(j)) . \quad (4.33)$$

*Source: Koopmans and Beckmann [98].*

### OPTIMAL LINEAR ARRANGEMENT

**INSTANCE:** Graph  $G$  with vertices  $[n]$ , weight function  $w : G \rightarrow \mathbb{Z}$ .

**PROBLEM:** Find a vertex labelling  $\sigma \in S_n$  that minimises the cost function

$$C(\sigma) = \sum_{(i,j) \in G} w(i,j) \cdot |\sigma(i) - \sigma(j)| . \quad (4.34)$$

*Source: Garey and Johnson [99].*

### MINIMUM $p$ -SUM

**INSTANCE:** Graph  $G$  with vertices  $[n]$ , integer  $p \in \mathbb{Z}$ .

**PROBLEM:** Find a vertex labelling  $\sigma \in S_n$  that minimises the cost function

$$C(\sigma) := C_G^p(\sigma) = \left( \sum_{(i,j) \in G} |f(i) - f(j)|^p \right)^{1/p}. \quad (4.35)$$

*Source:*

*Mitchison and Durbin [97], Garey and Johnson [100], Juvan and Mohar [101].*

### EDGESUM (or MINIMUM 1-SUM)

**INSTANCE:** Graph  $G$  with vertices  $[n]$ .

**PROBLEM:** Find a vertex labelling  $\sigma \in S_n$  that minimises the cost function

$$C(\sigma) := C_G^1(\sigma) = \sum_{(i,j) \in G} |\sigma(i) - \sigma(j)|. \quad (4.36)$$

*Source: Garey and Johnson [99].*

### BANDWIDTH (or MINIMUM $\infty$ -SUM)

**INSTANCE:** Graph  $G$  with vertices  $[n]$ .

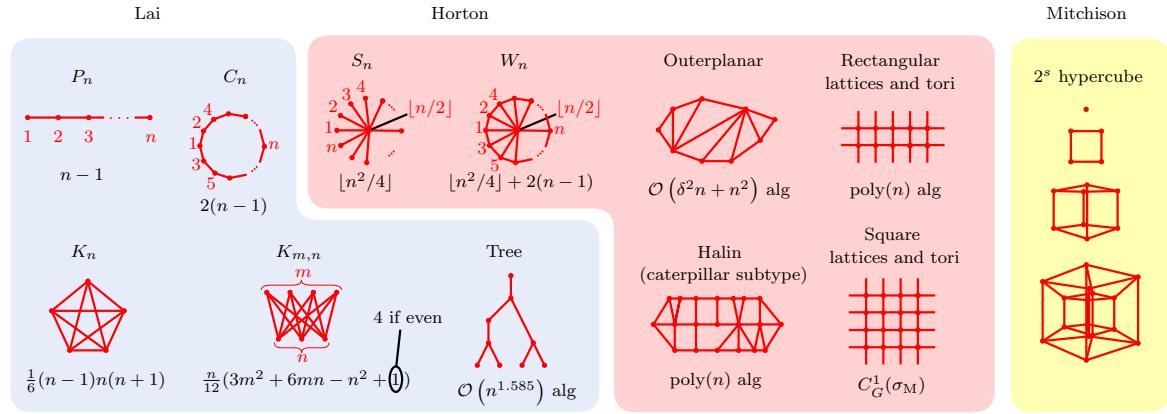
**PROBLEM:** Find a vertex labelling  $\sigma \in S_n$  that minimises the cost function

$$C(\sigma) := C_G^\infty(\sigma) = \max_{(i,j) \in G} |\sigma(i) - \sigma(j)|. \quad (4.37)$$

The optimal linear arrangement problem is a special case of the quadratic assignment problem: the weight function  $w$  restricts to integer values, the vertex labels are  $L = [n]$ , and the distance metric is  $d(i, j) = |i - j|$ .

The minimum  $p$ -sum problem is a special case of the optimal linear arrangement problem, because its weight function is effectively  $w(i, j) = |\sigma(i) - \sigma(j)|^{p-1} \in \mathbb{Z}$ . The decision version of the problem is provably NP-complete for  $p = 1$  [99],  $p = 2$  [104] and as  $p \rightarrow \infty$  [103]. Mitchison and Durbin [97] generalised the minimum  $p$ -sum problem to non-integer values of  $p > 0$ , defining problems that are likely to be at least as hard as the integer- $p$  equivalents.

The decision version of the edgesum problem is NP-complete via a reduction to the simple max-cut problem [99]. Figure 4.6 surveys the graphs with known solutions,



**Figure 4.6** Graph families with known solutions to the edgesum problems. The formula for the minimum edgesum appears beneath each graph, if it exists, or the cost of the best-known classical algorithm for solving that graph’s edgesum problem if not. References for the solution of these graphs appear in Lai [102], Horton [103], and Mitchison’s [97] works as indicated.

as of the date of publication of Horton’s PhD thesis [103] which cites a solution for outerplanar graphs by Frederickson and Hambrusch [105]. Other sources include a survey by Lai [102], and solution for tree graphs by Chung [106]. Study of the edgesum problem leads to novel Jordan–Wigner transformations that minimise the average Pauli weight of square–lattice fermionic Hamiltonians in Section 4.2.4.

The bandwidth problem is the limit of the  $p$ –sum problem as  $p \rightarrow \infty$ . The decision version of the bandwidth problem is NP–complete for arbitrary tree graphs [100]; conversely, Saxe proves that the decision problem as to whether the bandwidth is less than or equal to  $k = \mathcal{O}(1)$  is efficiently solvable [107].

#### 4.2.4 Jordan–Wigner transformations of square–lattice fermionic Hamiltonians with minimum average Pauli weight

Existing proposals for simulating square–lattice fermionic interactions fall into two categories. For a fermionic Hamiltonian with a square lattice interaction graph, the ancilla–qubit–free approach has involved numbering the modes row–by–row in zig-zagging or snake-like patterns [60, 70], which is equivalent to choosing the Jordan–Wigner transformations with fermionic labellings  $\sigma_Z$  and  $\sigma_S \in S_{N^2}$  as in Figure 4.4. The other approach involves embedding the fermionic Hamiltonian in a larger Hilbert space and using an ancilla–qubit mapping, which Chapter 5 discusses in more detail.

Before this work, little attention had gone towards exact optimisation strategies within the ancilla–free framework for fermion–qubit mappings. Concurrent work such

as searching the binary matrices of linear encodings [67] or the tree graphs of ternary tree transformations [73–75] involved computational searches for approximately optimal mappings. Optimising the Jordan–Wigner transformation over fermionic labellings was an unexplored approach, and for the popular example of the square lattice graph, the edgesum solution already existed in classical literature: Graeme Mitchison and Richard Durbin’s model of nerve cell organisation in the brain cortex [97]. Figure 4.3 displays the Mitchison–Durbin pattern  $\sigma_M$ , which is dramatically different to the more conventional Z– and S–patterns. Our main result, Theorem 3, makes use of this curious arrangement to construct a Jordan–Wigner transformation with the minimum average Pauli weight.

**Theorem 3.** (*Jordan–Wigner transformations of square-lattice fermionic Hamiltonians with minimum average Pauli weight.*)

Let  $n = N^2$  for some  $N \in \mathbb{N}$  and let  $\hat{H}_{\text{fermion}}$  be an  $n$ -mode fermionic Hamiltonian of the form in Equation 4.12 with the interaction graph  $G = G(\hat{H}_{\text{fermion}})$  of an  $N \times N$  square lattice. The fermion–qubit mapping  $\mathfrak{m}_{\text{JW}}(\sigma_M)$  yields the minimum average Pauli weight for  $H_{\text{qubit}}(\sigma)$  amongst all Jordan–Wigner transformations in  $[\mathfrak{m}_{\text{JW}}]$ , where the vertex labelling  $\sigma_M$  is the Mitchison–Durbin pattern.

*Proof.* See Section 4.3. □

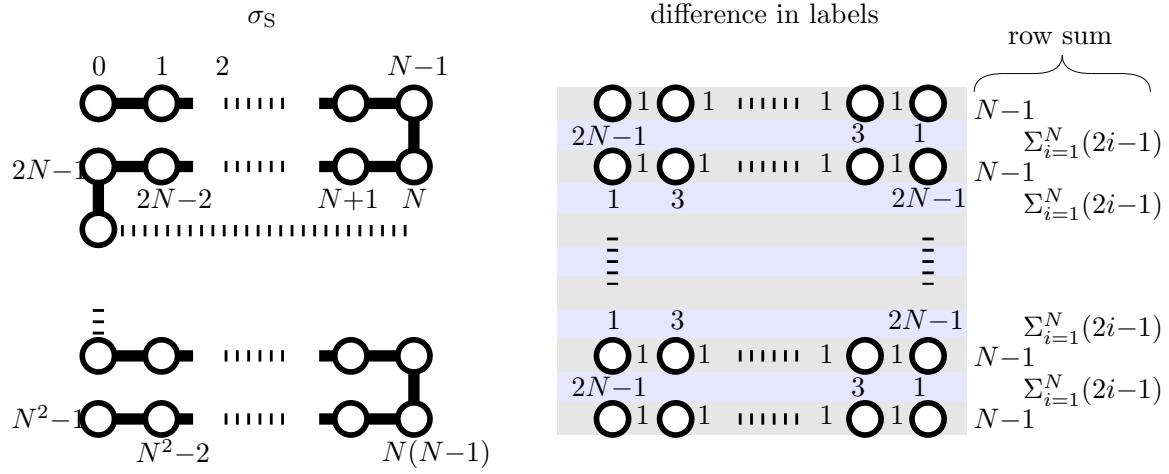
**Remark 4.2.3.** The decision problem involved in finding the vertex labelling that minimises the average Pauli weight of fermionic Hamiltonians with arbitrary interaction graphs is the edgesum problem whih is NP–complete.

Reflecting the literature on the edgesum problem from Section 4.2.3, Remark 4.2.4 details all known scenarios to date where the optimal fermionic labelling is solvable in  $\text{poly}(n)$  time.

**Remark 4.2.4.** (*Optimal vertex labellings for other fermionic interaction graph types.*) If the interaction graph of a fermionic Hamiltonian belongs to any of the graph families in Figure 4.6, then a classical computer can efficiently find Jordan–Wigner transformations of the Hamiltonian with the minimum average Pauli weight.

**Lemma 4.2.5.** If  $G$  is the  $N \times N$  lattice, the S–pattern  $\sigma_S$  vertex labelling has edgesum  $C_G^1(\sigma_S) = N^3 - N$ .

*Proof.* Figure 4.7 displays the S–pattern labelling of  $\sigma_S$  for  $G$  as well as the differences between the vertex labels of adjacent vertices. Regardless of whether  $N$  is odd or even,



**Figure 4.7** Left: S-pattern labelling on the  $N \times N$  square lattice. Right: Differences between vertex labels using the S-pattern labelling, with row totals on the far right.

the edgesum of  $\sigma_S$  is thus

$$C_G^1(\sigma_S) = (N - 1) \times \text{no. of rows} + \sum_{i=1}^N (2i - 1) \times (\text{no. of rows} - 1) \quad (4.38)$$

$$= N^2 - N + (N - 1)(N(N + 1) - N) \quad (4.39)$$

$$= N^2 - N - N^2 + N + N(N^2 - 1) \quad (4.40)$$

$$= N^3 - N. \quad \square$$

The average Pauli weight for the S-pattern on a square lattice is

$$C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_S)) = \frac{C_G^1(\sigma_S)}{|E|} + 1 = \frac{N^3 - N}{2N(N - 1)} + 1 = \frac{1}{2}N + \frac{3}{2}. \quad (4.41)$$

**Corollary 4.2.6.** (*Improvement to the average Pauli weight of qubit representations of square fermionic lattices with optimal fermionic labelling.*)

Using the Mitchison–Durbin pattern  $\sigma_M$  in Jordan–Wigner transformations of fermionic systems with  $N \times N$  square lattice interaction graphs, where  $N \geq 6$ , produces qubit Hamiltonians with lesser average Pauli weights than the Hamiltonians that the Z– and S–patterns produce. As  $N \rightarrow \infty$ , the ratio tends to  $\frac{1}{3}(4 - \sqrt{2}) \approx 0.86$ , so the Mitchison–Durbin pattern offers a  $\sim 13.9\%$  improvement.

*Proof.* On an  $N \times N$  square lattice  $G$ , the Mitchison–Durbin pattern  $\sigma_M$  and the S-pattern  $\sigma_S$  have edgesums

$$C_G^1(\sigma_Z) = N^3 - N, \quad (4.42)$$

$$C_G^1(\sigma_S) = N^3 - N, \quad (4.43)$$

$$C_G^1(\sigma_M) = N^3 - xN^2 + 2x^2N - \frac{2}{3}x^3 + N^2 - xN - 2N + \frac{2}{3}x, \quad (4.44)$$

respectively. The edgesum  $C_G^1(\sigma_Z)$  is straightforward to calculate; Lemma 4.2.5 derives  $C_G^1(\sigma_S)$ , and Section 4.3 contains the derivation of  $C_G^1(\sigma_M)$ . In Equation 4.44, the value of  $x$  is the closest integer to  $N - \frac{1}{2}\sqrt{2N^2 - 2N + \frac{4}{3}}$ . Using Equation 4.31 and the number of hopping terms  $|G| = 2N(N - 1)$ , the average Pauli weights are

$$C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_Z)) = C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_S)) = \frac{N^3 - N}{2N(N - 1)} + 1 = \frac{1}{2}N + \frac{3}{2}, \quad (4.45)$$

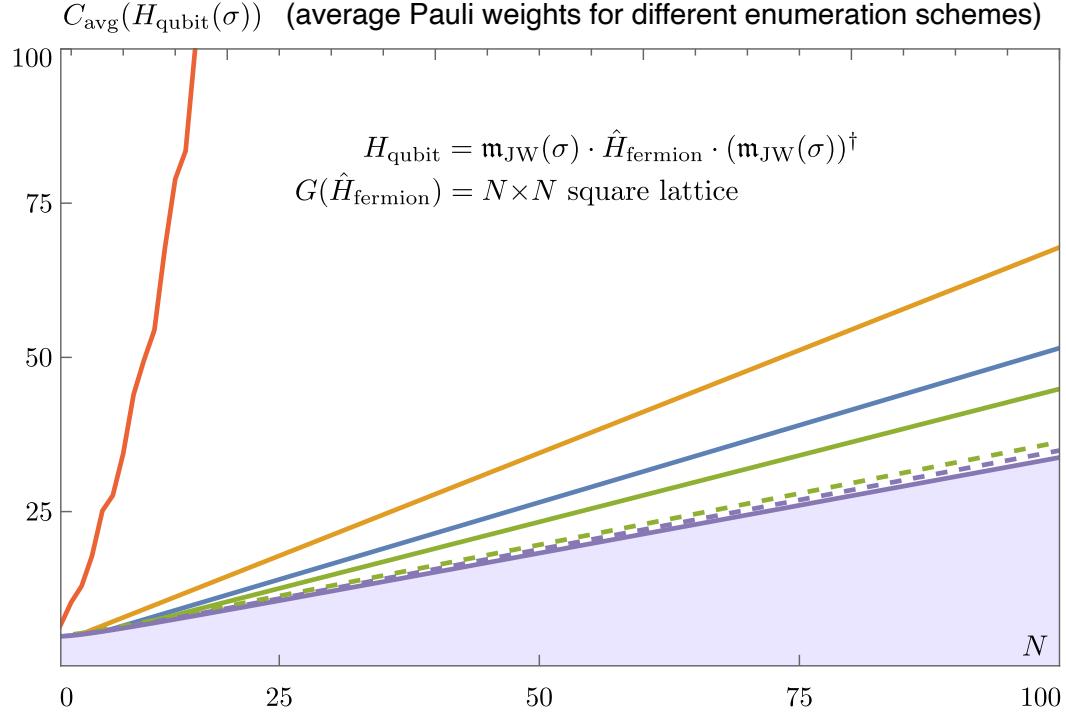
and, for large  $N$ ,

$$C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_M)) \approx \frac{1}{6}(4 - \sqrt{2})N + \frac{1}{12}(20 + \sqrt{2}) = 0.43N + 1.78. \quad (4.46)$$

For small  $N$ , explicit calculation verifies that  $C_G^1(\sigma_S) > C_G^1(\sigma_M)$  for  $N \geq 6$ . The ratio of average Pauli weights is thus  $C_{\text{avg}}(\sigma_M)/C_{\text{avg}}(\sigma_S) = C_{\text{avg}}(\sigma_M)/C_{\text{avg}}(\sigma_Z) \approx \frac{1}{3}(4 - \sqrt{2}) \approx 0.861$ .  $\square$

Corollary 4.2.6 shows that labelling the fermionic modes using the Mitchison–Durbin pattern, rather than the S-pattern as in [60] or the Z-pattern, produces a qubit Hamiltonian with a reduction in average Pauli weight by 13.9%. Our proposal immediately translates to a reduction by the same amount in the number of single-qubit measurements required in, for example, a VQE protocol [54]. Even in the case of simulating small fermionic systems, this method provides a worthwhile advantage. For example, in Figure 4.3 where  $N = 6$ , the ratio of the average Pauli weight using  $\sigma_M$  versus  $\sigma_S$  is  $4.33/4.5 \approx 0.96$ . That is, even for the  $6 \times 6$  lattice, applying Theorem 3 will reduce the number of Pauli measurements by 4%. While modest, this improvement comes free-of-charge.

For  $2 \leq N \leq 100$ , Figure 4.8 shows the resulting average Pauli weights of various fermionic labellings on  $N \times N$  lattices. The Mitchison–Durbin pattern can yield a meaningful reduction: for  $N = 20$ ,  $C_G^1(\sigma_M)/C_G^1(\sigma_S) = \frac{7140}{7980} \approx 89.5\%$ .



#### Chapter 4: $N^2$ –qubit Jordan–Wigner transformations $\mathbf{m}_{\text{JW}}(\sigma)$

- random  $\sigma \in S_{N^2}$
- $\sigma_D$        $C_{\text{avg}}(H_{\text{qubit}}(\sigma_D)) \approx \frac{2}{3}N + \frac{7}{6}$
- $\sigma_S, \sigma_Z$        $C_{\text{avg}}(H_{\text{qubit}}(\sigma_S, \sigma_Z)) = \frac{1}{2}N + \frac{3}{2}$
- $\sigma_M$  ( $\dot{\rho}'$  with  $x \approx 0.29N$ )       $C_{\text{avg}}(H_{\text{qubit}}(\sigma_M)) \approx 0.43N + 1.78$

#### Chapter 5: two–ancilla–qubit Jordan–Wigner transformations $\Pi_{\tilde{\mathcal{H}}} \cdot (L_{P_1} L_{P_0}) \cdot \mathbf{m}_{\text{JW}}(\sigma)$

- $\sigma_M$  ( $\dot{\rho}'$  with  $x \approx 0.29N$ )       $C_{\text{avg}}(\tilde{H}(\sigma_M)) \approx 0.33N + 2.89$
- $\dot{\rho}'$  with  $x = \left\lfloor \frac{N+1}{3} \right\rfloor$        $C_{\text{avg}}(\tilde{H}(\dot{\rho}')) \approx 0.32N + 2.78$
- $\sigma_{M+2}$  ( $\dot{\rho}'$  with  $x \approx 0.40N$ )       $C_{\text{avg}}(\tilde{H}(\sigma_{M+2})) \approx 0.31N + 2.68$

**Figure 4.8** Average Pauli weights  $C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma))$  of Jordan–Wigner transformed hopping terms in an  $N \times N$  square lattice graph with different fermionic labellings  $\sigma \in S_{N^2}$ . The Mitchison–Durbin pattern  $\sigma_M$  provides the minimum average Pauli weight for ancilla–free Jordan–Wigner transformations. The two–ancilla–qubit Jordan–Wigner transformations, which feature in Chapter 5, go beyond this optimality.

### 4.3 Proof of Theorem 3

This section contains a more elaborate version of the original proof [97] that the Mitchison–Durbin pattern  $\sigma_M$  solves the edgesum problem of an  $N \times N$  lattice. Figure 4.9 chronicles the proof, which starts with an arbitrary vertex labelling for the  $N^2$  vertices before performing five sequential modifications (Proposition 2 and Lemmas 4.3.2–4.3.5) which exhaust all possible ways to reduce the edgesum, to give the Mitchison–Durbin pattern  $\sigma_M$ .

The edgesum problem is the minimum- $p$ -sum problem described in Section 4.2.3 with  $p = 1$ . The first part of the proof, Proposition 2, holds for all  $p \geq 1$ ; the rest of the proof concerns only the case  $p = 1$ .

Given an  $n$ -vertex graph  $G$  and a vertex labelling  $\sigma \in S_n$  for  $G$ , the cost function  $C_G^p(\sigma)$  for the minimum  $p$ -sum problem is

$$C_G^p(\sigma) = \left( \sum_{(i,j) \in G} |\sigma(i) - \sigma(j)|^p \right)^{1/p}. \quad (4.47)$$

In the case that  $G$  is an  $N \times N$  square lattice consisting of pairs of horizontally and vertically adjacent vertices, the edgesum has horizontal and vertical contributions:

$$(C_G^p(\sigma))^p = \sum_{\substack{(i,j) \in G; \\ \text{horizontally adjacent}}} |\sigma(i) - \sigma(j)|^p + \sum_{\substack{(i,j) \in G; \\ \text{vertically adjacent}}} |\sigma(i) - \sigma(j)|^p. \quad (4.48)$$

**Definition 4.3.1.** Let  $G$  be the  $N \times N$  square lattice. The vertex labelling  $\sigma \in S_{N^2}$  for  $G$  is *horizontally ordered* if, for any vertex  $i$ ,  $\sigma(j) > \sigma(i)$  for all vertices  $j$  to the right of  $i$ . The vertex labelling  $\sigma$  is *vertically ordered* if  $\sigma(j) > \sigma(i)$  for all vertices  $j$  below  $i$ .

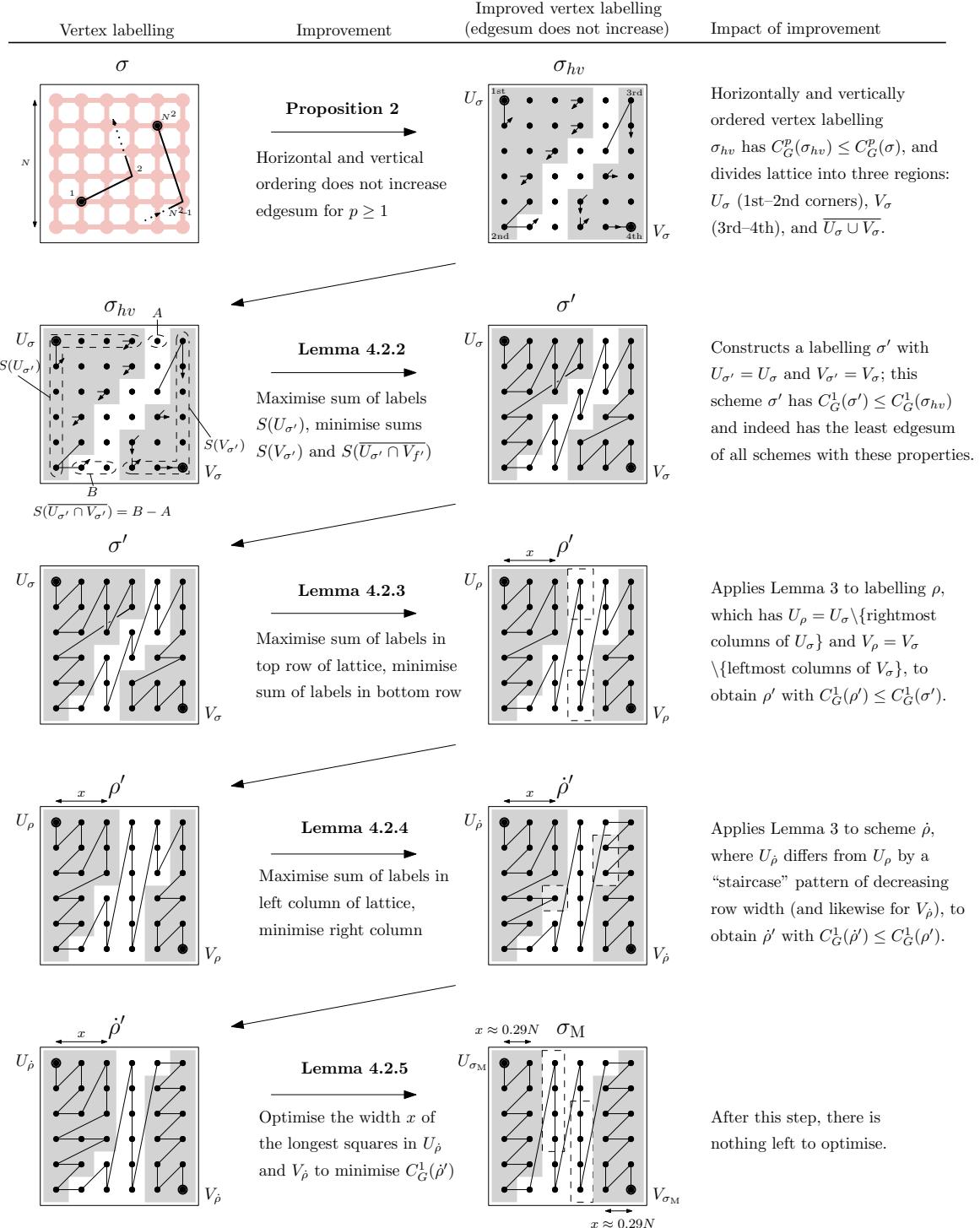
**Proposition 2.** Let  $G$  be the  $N \times N$  square lattice, let  $p \geq 1$ , and let  $\sigma \in S_{N^2}$  be a vertex labelling for  $G$ . Then there exists a horizontally and vertically ordered vertex labelling  $\sigma_{hv}$  such that  $C_G^p(\sigma_{hv}) \leq C_G^p(\sigma)$ .

*Proof.* Construct the vertex labelling  $\sigma_h$  by permuting the vertex labels, under  $\sigma$ , within each row of the lattice so that the labels ascend from left to right. Call  $\sigma_h$  the *horizontal ordering* of  $\sigma$ . Similarly, construct  $\sigma_{hv}$  from  $\sigma_h$  by permuting vertex labels under  $\sigma_h$  within each column to increase from top to bottom. Call  $\sigma_{hv}$  the *vertical ordering* of  $\sigma_h$ . We will prove Proposition 2 via the statements S1 and S2 below:

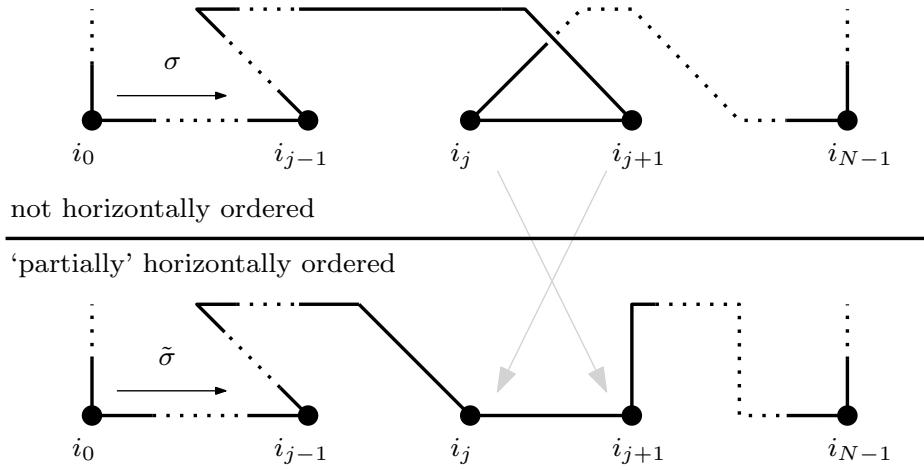
S1 The edgesum of the horizontal ordering of a vertex labelling is at most equal to the edgesum of the original labelling,  $C_G^p(\sigma_h) \leq C_G^p(\sigma)$ .

**Finding a solution to the edgesum problem: constructing the Mitchison–Durbin pattern**

Goal: find vertex labelling  $\sigma \in S_{N^2}$  to minimise  $C_G^1(\sigma) = \sum_{(i,j) \in G} |\sigma(i) - \sigma(j)|$   
 $G = N \times N$  square lattice



**Figure 4.9** Overview of the proof that the Mitchison–Durbin pattern is a solution to the edgesum problem for the square lattice.



$$\sigma(i_0) < \dots < \sigma(i_{j-1}) < \sigma(i_{j+1}) < \sigma(i_j) < \dots < \sigma(i_{N-1})$$

**Figure 4.10** Swapping the labels of vertices  $i_0, \dots, i_{N-1}$  in a row of the  $N \times N$  square lattice as part of the process of constructing a horizontally ordered vertex labelling.

S2 The vertex labelling  $\sigma_{hv}$  is horizontally ordered.

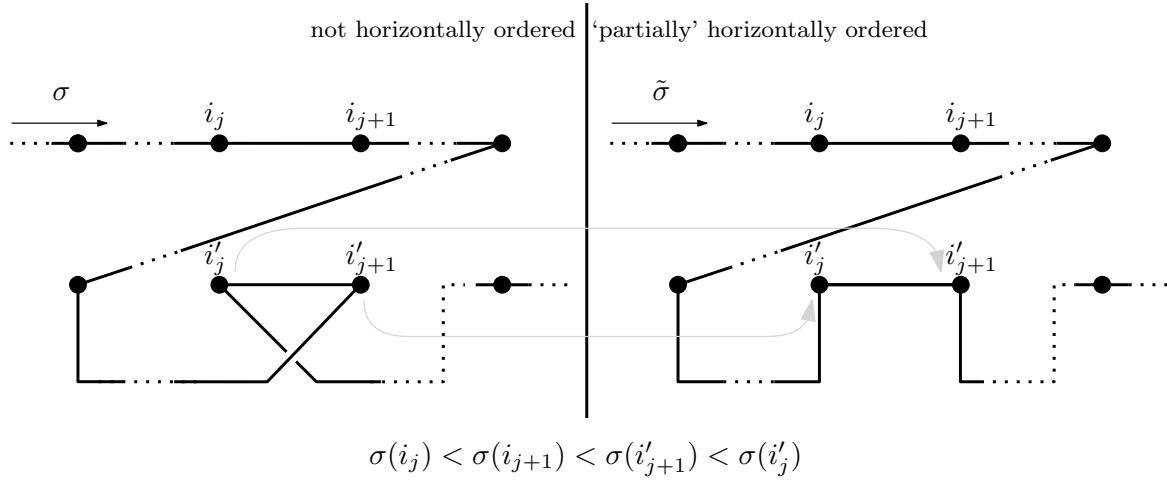
To prove S1, first consider the impact of horizontal ordering on the horizontal contributions to  $C_G^p(\sigma)$  in Equation 4.48.

Claim: The horizontal contributions to  $C_G^p(\sigma_h)$  are less than or equal to the horizontal contributions to  $C_G^p(\sigma)$ .

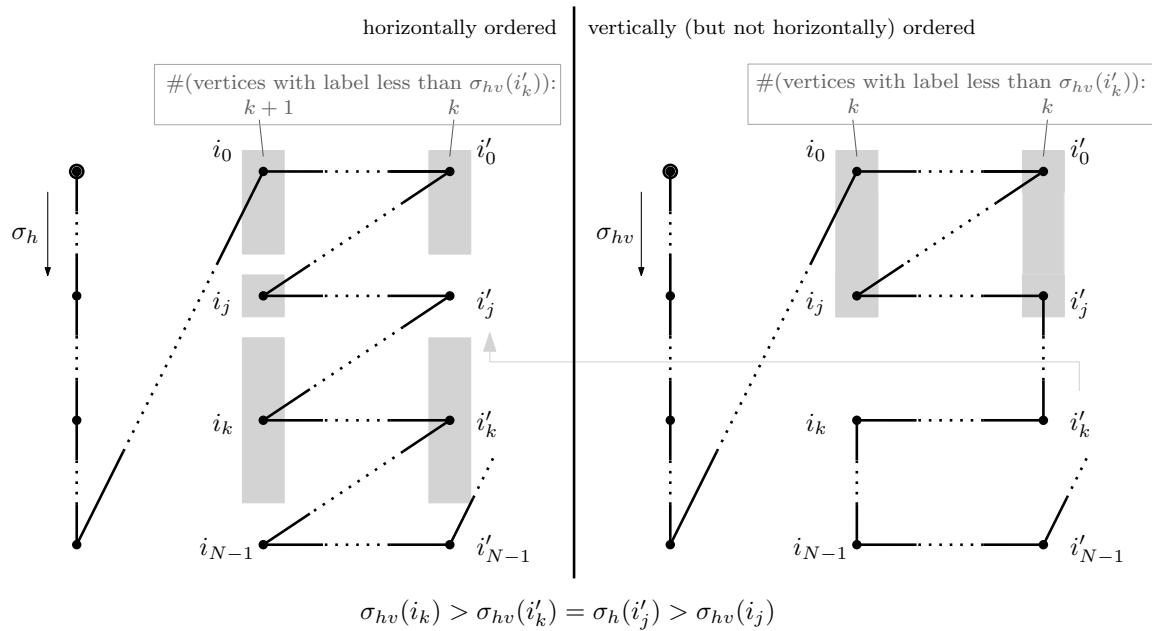
Proof: (See Figure 4.10.) Suppose  $\sigma$  is not horizontally ordered, and so there is a row of the lattice consisting of vertices with consecutive labels  $i_0, i_1, \dots, i_{N-1}$  such that  $\sigma(i_j) > \sigma(i_{j+1})$  for some  $0 < j \leq N - 1$ . The horizontal contributions to  $C_G^p(\sigma)$  from this row are

$$\begin{aligned} & |\sigma(i_0) - \sigma(i_1)|^p + \dots + |\sigma(i_{j-1}) - \sigma(i_j)|^p \\ & \quad + |\sigma(i_j) - \sigma(i_{j+1})|^p \\ & \quad + \dots + |\sigma(i_{N-2}) - \sigma(i_{N-1})|^p. \end{aligned} \tag{4.49}$$

Consider the vertex labelling  $\tilde{\sigma}$  that is identical to  $\sigma$  save that  $\tilde{\sigma}(i_j) = \sigma(i_{j+1})$  and  $\tilde{\sigma}(i_{j+1}) = \sigma(i_j)$ , i.e.  $\tilde{\sigma}$  swaps the labels of vertices  $i_j$  and  $i_{j+1}$  so that  $\tilde{\sigma}(i_j) < \tilde{\sigma}(i_{j+1})$ . Thus,  $\tilde{\sigma}$  is at least 'partially' horizontally ordered, moreso than  $\sigma$ . The horizontal



**Figure 4.11** Swapping vertex labels within a row will affect some of the contributions to the  $p$ -sum of the vertex labelling from vertically adjacent vertices.



**Figure 4.12** Illustration of the contradiction that would arise if a vertical ordering  $\sigma_{hv}$  of a horizontally ordered vertex labelling  $\sigma_h$  were not itself horizontally ordered.

contributions to  $C_G^p(\tilde{\sigma})$  from this row are:

$$\begin{aligned} & |\tilde{\sigma}(i_0) - \tilde{\sigma}(i_1)|^p + \dots + |\tilde{\sigma}(i_{j-1}) - \tilde{\sigma}(i_{j+1})|^p \\ & \quad + |\tilde{\sigma}(i_{j+1}) - \tilde{\sigma}(i_j)|^p \\ & \quad + \dots + |\tilde{\sigma}(i_{N-2}) - \tilde{\sigma}(i_{N-1})|^p. \end{aligned} \quad (4.50)$$

The difference between the horizontal contributions to  $C_G^p(\sigma)$  and to  $C_G^p(\tilde{\sigma})$  is thus

$$|\tilde{\sigma}(i_{j+1}) - \tilde{\sigma}(i_j)|^p + |\tilde{\sigma}(i_{j-1}) - \tilde{\sigma}(i_{j+1})|^p \quad (4.51)$$

$$\begin{aligned} & - |\sigma(i_j) - \sigma(i_{j+1})|^p - |\sigma(i_{j-1}) - \sigma(i_j)|^p \\ & = |\tilde{\sigma}(i_{j-1}) - \tilde{\sigma}(i_{j+1})|^p - |\sigma(i_{j-1}) - \sigma(i_j)|^p \end{aligned} \quad (4.52)$$

$$\begin{aligned} & = |\sigma(i_{j-1}) - \sigma(i_j)|^p - |\sigma(i_{j-1}) - \sigma(i_j)|^p \\ & < 0, \end{aligned} \quad (4.53)$$

since  $p \geq 1$  and  $\sigma(i_{j-1}) < \sigma(i_{j+1}) < \sigma(i_j)$ .

To construct  $\sigma_h$  is to perform as many swaps of the above form (transforming  $\sigma$  into  $\tilde{\sigma}$ ) as is necessary in order to label all  $N$  vertices in each row of the lattice in ascending order. Thus, the horizontal contributions to  $C_G^p(\sigma_h)$  are less than the horizontal contributions to  $C_G^p(\sigma)$ , with equality if and only if  $\sigma = \sigma_h$ . ■

To finish proving S1, consider the impact of horizontal ordering on the vertical contributions to  $C_G^p(\sigma)$  in Equation 4.48.

Claim: The vertical contributions to  $C_G^p(\sigma_h)$  are less than or equal to the vertical contributions to  $C_G^p(\sigma)$ .

Proof: (See Figure 4.11.) Suppose that  $\sigma$  is not horizontally ordered, and that we are in the process of constructing  $\sigma_h$  from  $\sigma$  by re-ordering vertex labels row-by-row. Then there is a row of the lattice consisting of vertices with consecutive labels  $i'_0, i'_1, \dots, i'_{N-1}$  such that  $\sigma(i'_j) > \sigma(i'_{j+1})$  for some  $j$ . Let  $i_j$  and  $i_{j+1}$  be the vertices directly above vertices with labels  $i'_j$  and  $i'_{j+1}$ , respectively; thus  $\sigma(i_j) < \sigma(i_{j+1})$ .

The vertical contributions to  $C_G^p(\sigma)$  from these four vertices is

$$|\sigma(i'_j) - \sigma(i_j)|^p + |\sigma(i'_{j+1}) - \sigma(i_{j+1})|^p. \quad (4.54)$$

Because  $\sigma(i_j) < \sigma(i_{j+1})$  and  $\sigma(i'_j) > \sigma(i'_{j+1})$ , the values  $|\sigma(i_j) - \sigma(i'_j)|^p$  and  $|\sigma(i_{j+1}) - \sigma(i'_{j+1})|^p$  in Equation 4.54 are unequal, with the latter being less than the former.

Consider the vertex labelling  $\tilde{\sigma}$  that differs from  $\sigma$  only by exchanging the labels of vertices  $i'_j$  and  $i'_{j+1}$ , so  $\tilde{\sigma}(i'_j) = \sigma(i'_{j+1})$  and  $\tilde{\sigma}(i'_{j+1}) = \sigma(i'_j)$ . The vertical contributions

to  $C_G^p(\tilde{\sigma})$  from the four vertices is

$$|\sigma(i_j) - \sigma(i'_{j+1})|^p + |\sigma(i_{j+1}) - \sigma(i'_j)|^p. \quad (4.55)$$

Note that for  $p > 1$ , if  $a + b$  is a constant value then  $a^p + b^p$  reduces in value the closer in value  $a$  is to  $b$ . Therefore, the vertical contributions to  $C_G^p(\tilde{\sigma})$  in Equation 4.55 are less than the vertical contributions to  $C_G^p(\sigma)$  in Equation 4.54. If  $p = 1$ , the contributions are equal. This proves the claim, and furthermore S1. ■

To prove S2, consider the horizontally ordered vertex labelling  $\sigma_h$ , as in Figure 4.12. Suppose that  $\sigma_h$  is not vertically ordered, and let  $\sigma_{hv}$  be the vertical ordering of  $\sigma_h$ . It remains to show that  $\sigma_{hv}$  is horizontally ordered. Suppose that  $\sigma_{hv}$  is not horizontally ordered: thus, there exists  $k \in \{0, \dots, N-1\}$  such that the  $k$ th row of the lattice contains vertices  $i_k, i'_k$ , with  $i'_k$  to the right of  $i_k$  and  $\sigma_{hv}(i_k) > \sigma_{hv}(i'_k)$ .

Denote vertices in the same column as  $i'_k$  by  $i'_0, \dots, i'_{N-1}$ , and similarly define the vertices  $i_0, \dots, i_{N-1}$  as those sharing the column of  $i_k$ . Because  $\sigma_{hv}$  is vertically ordered, then  $\sigma_{hv}(i_k) > \sigma_{hv}(i_l)$  and  $\sigma_{hv}(i'_j) > \sigma_{hv}(i'_l)$  for all  $0 \leq l \leq j-1$ . As  $\sigma_{hv}(i_k) > \sigma_{hv}(i'_k)$ , then under  $\sigma_{hv}$ , both the  $k$  vertices above  $i_k$  and the  $k$  vertices above  $i'_k$  have labels less than  $\sigma_{hv}(i'_k)$ . Since all that separates  $\sigma_h$  and  $\sigma_{hv}$  is a permutation of vertex labels that fixes the labels within columns of the lattice, then there should be exactly  $k$  vertices with labels less than  $\sigma_{hv}(i'_k)$  in both the  $i_0, \dots, i_{N-1}$  and  $i'_0, \dots, i'_{N-1}$  columns of the lattice under the vertex labelling  $\sigma_h$  as well.

Because  $\sigma_h$  is horizontally ordered,  $\sigma_h(i_k) < \sigma_h(i'_k)$ . Therefore at least one of  $i_k$  and  $i'_k$  will have a different label under  $\sigma_h$  than under  $\sigma_{hv}$ . Suppose, without loss of generality, that the vertex  $i'_k$  satisfies this: that  $\sigma_h(i'_k) \neq \sigma_{hv}(i'_k)$  and thus there exists some  $j \in \{0, \dots, N-1\}$  such that  $\sigma_h(i'_j) = \sigma_{hv}(i'_k)$ . Under  $\sigma_h$ , there are  $k$  vertices in the  $i'_0, \dots, i'_{N-1}$  column with label less than  $\sigma_{hv}(i'_k)$ , as was the case under  $\sigma_{hv}$ . However, consider the vertices in the  $i_0, \dots, i_{N-1}$  column. Each vertex  $i_l$  to the left of a vertex  $i'_l$  with  $\sigma_h(i'_l) < \sigma_{hv}(i'_k)$  also satisfies  $\sigma_h(i_l) < \sigma_{hv}(i'_k)$  due to horizontal ordering; there are  $k$  of these vertices. Distinct from these vertices, there is also the vertex  $i_j$  to the left of  $i'_j$  which must also have  $\sigma_h(i_j) < \sigma_h(i'_j) = \sigma_{hv}(i'_k)$ . This is the contradiction.

This proves S2 and hence Proposition 2. □

Henceforth, the proof of Theorem 3 only concerns the case  $p = 1$ .

Let  $G$  be the  $N \times N$  square lattice. As a result of Proposition 2, we need only consider horizontally and vertically ordered vertex labellings in the search for the solution to the edgesum problem for  $G$ . Let  $\sigma_{hv}$  be the horizontal and vertical ordering of a vertex labelling  $\sigma$  for  $G$ . Let  $v_{i,j}$  denote the label of the vertex in the  $i$ th row and

$j$ th column of the lattice  $G$ , using matrix index notation so that the top-left vertex has label  $v_{1,1}$ . Then the edgesum of  $\sigma_{hv}$  is

$$C_G^1(\sigma_{hv}) = \sum_{i,j=1}^{N-1} (\sigma_{hv}(v_{i+1,j}) - \sigma_{hv}(v_{i,j}) + \sigma_{hv}(v_{i,j+1}) - \sigma_{hv}(v_{i,j})) \quad (4.56)$$

$$\begin{aligned} &= 2\sigma_{hv}(v_{N,N}) + \sum_{i=2}^{N-1} \sigma_{hv}(v_{i,N}) + \sum_{j=2}^{N-1} \sigma_{hv}(v_{N,j}) - \sum_{i=2}^{N-1} \sigma_{hv}(v_{i,1}) \\ &\quad - \sum_{j=2}^{N-1} \sigma_{hv}(v_{1,j}) - 2\sigma_{hv}(v_{1,1}) \end{aligned} \quad (4.57)$$

$$= \sum_{i \in V_b} \sigma_{hv}(i) + \sum_{i \in V_r} \sigma_{hv}(i) - \left( \sum_{i \in V_l} \sigma_{hv}(i) + \sum_{i \in V_t} \sigma_{hv}(i) \right), \quad (4.58)$$

where  $V_l, V_r, V_t, V_b$  are the vertices in the left column, right column, top row and bottom row of the square lattice, respectively. This result greatly simplifies the calculation of  $C_G^1(\sigma_{hv})$  for the square lattice.

Due to horizontal and vertical ordering,  $\sigma_{hv}(v_{1,1}) = 0$  and  $\sigma_{hv}(v_{N,N}) = N^2 - 1$ . Without loss of generality, assume that  $\sigma_{hv}(v_{N,1}) < f_{hv}(v_{1,N})$ . As detailed in Figure 4.9, define  $U_\sigma$  to be the set of vertices with labels between  $\sigma_{hv}(v_{1,1}) = 0$  and  $\sigma_{hv}(v_{N,1})$ , and let  $V_\sigma$  be the set of vertices with labels between  $\sigma_{hv}(v_{N,1})$  and  $\sigma_{hv}(v_{N,N}) = N^2 - 1$ . Define  $S(U_\sigma)$  and  $S(V_\sigma)$  to be the sums of the labels of vertices on the boundary of the lattice in the regions  $U_\sigma$  and  $V_\sigma$ , respectively, and define  $S(\overline{U_\sigma \cap V_\sigma})$  to be the difference between the sum of the labels of vertices in  $\overline{U_\sigma \cap V_\sigma}$  in the bottom of the lattice and the sum of the labels of the vertices in  $\overline{U_\sigma \cap V_\sigma}$  in the top row of the lattice. By Equation 4.58, these quantities completely determine the edgesum:

$$C_G^1(\sigma_{hv}) = S(V_\sigma) + S(\overline{U_\sigma \cap V_\sigma}) - S(U_\sigma). \quad (4.59)$$

The task now is to identify an improved vertex labelling  $\sigma'$  with the least edgesum of all labellings  $\tilde{\sigma}$  with  $U_{\tilde{\sigma}} = U_\sigma$  and  $V_{\tilde{\sigma}} = V_\sigma$ . That is, the task is to find a vertex labelling  $\sigma'$  such that

$$\sigma' = \arg \min_{\substack{\text{labellings } \tilde{\sigma} \\ U_{\tilde{\sigma}} = U_\sigma, V_{\tilde{\sigma}} = V_\sigma}} C_G^1(\tilde{\sigma}) = \arg \min_{\substack{\text{labellings } \tilde{\sigma} \\ U_{\tilde{\sigma}} = U_\sigma, V_{\tilde{\sigma}} = V_\sigma}} \left( S(V_{\tilde{\sigma}}) + S(\overline{U_{\tilde{\sigma}} \cap V_{\tilde{\sigma}}}) - S(U_{\tilde{\sigma}}) \right). \quad (4.60)$$

Note that if  $\sigma$  is such that  $U_\sigma$  and  $V_\sigma$  are the same regions as those yielded by a solution to the edgesum problem, then any  $\sigma'$  satisfying Equation 4.60 is a solution to the edgesum problem. Lemma 4.3.2 details how to construct a candidate for  $\sigma'$  given

$\sigma$ ; using this result, Lemmas 4.3.3–4.3.5 detail how to set  $\sigma$  such that the candidate  $\sigma'$  is a solution to the edgesum problem, completing the proof.

**Lemma 4.3.2.** Let  $G$  be the  $N \times N$  square lattice, with  $v_{i,j}$  the label of the vertex in the  $i$ th row and  $j$ th column of the lattice. Let  $\sigma_{hv}$  be the horizontal and vertical ordering of a vertex labelling  $\sigma$  for  $G$ , dividing the lattice into regions  $U_\sigma$ ,  $V_\sigma$  and  $\overline{U_\sigma \cap V_\sigma}$  as defined above. Then, the steps below construct a horizontally and vertically ordered vertex labelling  $\sigma'$  that satisfies Equation 4.60. Note that while  $U_{\sigma'} = U_\sigma$ ,  $S(U_{\sigma'})$  may not be equal to  $S(U_\sigma)$ , as is the case for  $V_{\sigma'}$  and  $S(V_{\sigma'})$ .

1. To maximise  $S(U_{\sigma'})$ :

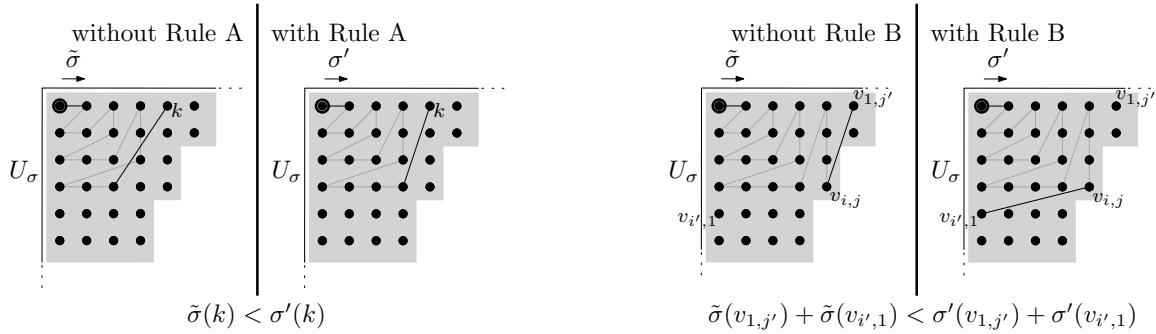
**Rule A:** starting at the beginning of a row  $v_{1,j}$  (resp. column  $v_{i,1}$ ), the vertex labelling  $\sigma'$  proceeds rightwards along the row (resp. downwards along the column) into the interior of  $U_\sigma$  as far as possible without violating horizontal and vertical ordering.

**Rule B:** once the vertex labelling  $\sigma'$  has proceeded along a row or column as far as successive applications of Rule A permits, it must begin afresh at the start of the next-topmost row or the next-rightmost column, whichever is longest.

*Mathematically:* suppose that Rule A terminates at the vertex  $v_{i,j}$ . Let  $i' > i$  and  $j' > j$  denote the indices of the topmost row and rightmost columns, respectively, that are *yet to be enumerated*, i.e. that consist entirely of vertices with labels greater than  $\sigma'(v_{i,j})$ . Then, let the number of vertices that are both in the  $(i')$ th row and also contained within  $U_\sigma$  be  $n_r(i')$ , and similarly let  $n_c(j')$  denote the number of vertices in the  $(j')$ th column that are also contained within  $U_\sigma$ . The following recursive rule describes the process of maximising  $S(U_{\sigma'})$  with Rule A and Rule B:

$$\sigma'(v_{i,j}) + 1 = \begin{cases} \sigma'(v_{i,j+1}) & \text{if } v_{i,j+1} \in U_\sigma \text{ and } \sigma'(v_{i-1,j}) < \sigma'(v_{i,j}), \\ \sigma'(v_{i+1,j}) & \text{if } v_{i-1,j} \in U_\sigma \text{ and } \sigma'(v_{i,j-1}) < \sigma'(v_{i,j}), \\ \sigma'(v_{1,j'}) & \text{else if } n_c(j') \geq n_r(i'), \\ \sigma'(v_{i',1}) & \text{else } n_r(i') > n_c(j'). \end{cases} \quad (4.61)$$

2. To minimise  $S(\overline{U_{\sigma'} \cap V_{\sigma'}})$ : have the vertex labelling  $\sigma'$  begin at the topmost vertex of each column in  $\overline{U_\sigma \cap V_\sigma}$  and fill down to the lowest vertex in that column of  $\overline{U_\sigma \cap V_\sigma}$ .
3. To minimise  $S(V_{\sigma'})$ : perform the inverse procedure for maximising  $S(U_{\sigma'})$ .



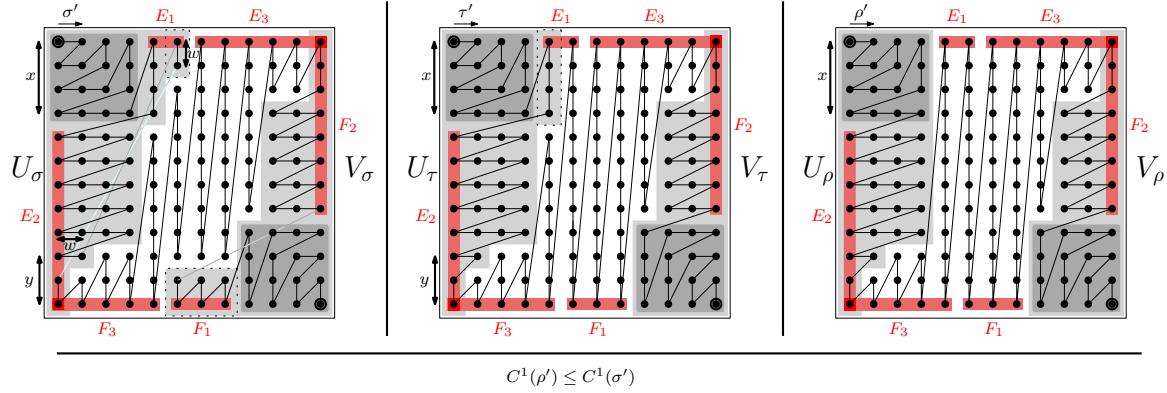
**Figure 4.13** Rules in Lemma 4.3.2 for constructing the vertex labelling  $\sigma'$  so that it has the least edgesum of all vertex labellings  $\tilde{\sigma}$  with  $U_{\tilde{\sigma}} = U_\sigma$  and  $V_{\tilde{\sigma}} = V_\sigma$ .

*Proof.* Begin by noting that maximising  $S(U_{\sigma'})$ , minimising  $S(\overline{U_{\sigma'} \cap V_{\sigma'}})$  and minimising  $S(V_{\sigma'})$  are three independent tasks, thus the Lemma's sequential approach is conceptually valid. Next, with  $\sigma'(v_{1,1}) = 0$ , it must be the case that either  $\sigma'(v_{1,2}) = 1$  or  $\sigma'(v_{2,1}) = 1$ . Without loss of generality, take  $\sigma'(v_{1,2}) = 1$ .

1. To maximise  $S(U_{\sigma'})$ : Rule A and Rule B, as described in Equation 4.61, work in tandem to label vertices in the interior of  $U_\sigma$  with the lowest indices possible. As illustrated in Figure 4.13, this ensures that the vertices on the boundary of the lattice, i.e. the vertices contributing to  $S(U_{\sigma'})$ , have as high an index as possible.
2. To minimise  $S(\overline{U_{\sigma'} \cap V_{\sigma'}})$ , label the vertices in the columns of  $\overline{U_{\sigma'} \cap V_{\sigma'}}$  in ascending order from top-to-bottom. This places labels with the least possible value in the bottom row of the lattice, and labels with the greatest possible value in the top row of the lattice, while preserving horizontal and vertical ordering.
3. As in step 1, a symmetric argument applies to minimising  $S(V_{\sigma'})$ . □

By the definition in Equation 4.60, there must exist a particular shape for regions  $U_\sigma$  and  $V_\sigma$  such that the vertex labelling  $\sigma'$  described in Lemma 4.3.2 is a solution to the edgesum problem. There is an  $x \times x$  square region of vertices in the top-left region of  $U_\sigma$ , where  $x$  is the greatest integer for which the statement “the topmost  $x$  rows of  $U_\sigma$  contain at least  $x$  vertices” is true. Call this region the *largest square of  $U_\sigma$* ; define the *largest square of  $V_\sigma$*  similarly.

**Lemma 4.3.3.** Let  $G$  be the  $N \times N$  square lattice, and let  $\sigma$  be a horizontally and vertically ordered vertex labelling for  $G$ , dividing the lattice into regions  $U_\sigma$ ,  $V_\sigma$  and  $\overline{U_\sigma \cap V_\sigma}$  as defined above. Let  $\sigma'$  be the vertex labelling resulting from applying Lemma 4.3.2 to the labelling  $\sigma$ .



**Figure 4.14** Illustration of Lemma 4.3.3: starting with a vertex labelling  $\sigma'$ , intermediate labelling  $\tau'$ , and final labelling  $\rho'$  demonstrating shapes of  $U_\rho$  and  $V_\rho$  that allow for a lower edgesum.

Now, consider a vertex labelling  $\rho$  that divides the lattice into regions  $U_\rho$ ,  $V_\rho$  and  $\overline{U_\rho \cap V_\rho}$  where  $U_\rho \subseteq U_\sigma$  consists of all the vertices in  $U_\sigma$  except for those to the right of the largest square of  $U_\sigma$ , and where  $V_\rho \subseteq V_\sigma$  consists of all the vertices in  $V_\sigma$  except for those to the left of the largest square of  $V_\sigma$ . The vertices in  $U_\sigma \setminus U_\rho \cup V_\sigma \setminus V_\rho$  are thus in  $\overline{U_\rho \cap V_\rho}$ . Let  $\rho'$  be the labelling resulting from applying Lemma 4.3.2 to  $\rho$ . Then,  $C_G^1(\rho') \leq C_G^1(\sigma')$ .

*Proof.* As in Figure 4.14, define  $E_1$  to be the set containing the topmost vertices in  $U_\sigma$  that are to the right of the largest square in  $U_\sigma$ ,  $E_2$  to be the set containing the leftmost vertices in  $U_\sigma$  beneath the largest square of  $U_\sigma$ , and  $E_3$  to be set containing the topmost vertices of the lattice that are outside of  $U_\sigma$ . Define  $F_1$ ,  $F_2$  and  $F_3$  analogously for the bottommost, rightmost, and outlying vertices along the border of the lattice outside of the largest square in  $V_\sigma$ .

While  $E_1, \dots, F_3$  refer to fixed sets of vertices defined by the vertex labelling  $\sigma$ , let  $E_1(\tau), \dots, F_3(\tau)$  refer to the sums of the labels of those same vertex sets under any vertex labelling  $\tau \in S_{N^2}$ . Using the shorthand  $(E_1 + E_2)(\tau) = E_1(\tau) + E_2(\tau)$ , we have via Equation 4.58,

$$\begin{aligned} C_G^1(\sigma') &= \left( (F_1 + F_2 + F_3) - (E_1 + E_2 + E_3) \right)(\sigma') \\ &\quad + \text{labels under } \sigma' \text{ of boundary vertices of largest squares of } U_\sigma \text{ and } V_\sigma , \end{aligned} \tag{4.62}$$

where  $\sigma'$  is the vertex labelling that arises from applying Lemma 4.3.2 to  $\sigma$ .

Claim: With the description of  $\rho$  and  $\rho'$  from Lemma 4.3.3,

$$(E_1 + E_2 + E_3)(\rho') \geq (E_1 + E_2 + E_3)(\sigma'). \quad (4.63)$$

Proof: Consider a vertex labelling  $\tau$  that has  $U_\tau = U_\sigma$  and  $V_\tau = V_\rho$ , and let  $\tau'$  be the vertex labelling that arises from applying Lemma 4.3.2 to  $\tau$ . Then,  $E_1(\tau') = E_1(\sigma')$  and  $E_2(\tau') = E_2(\sigma')$  as the vertex labels are unchanged. However,  $E_3(\tau') > E_3(\sigma')$ : this is because there are more vertices in  $\overline{U_\tau \cap V_\tau}$  than in  $\overline{U_\sigma \cap V_\sigma}$ .

Next, redefine  $\tau$  such that  $V_\tau = V_\rho$  and  $U_\tau$  is equal to  $U_\sigma$  with its rightmost column removed, as in Figure 4.14. Define  $\tau'$  to be the labelling arising from applying Lemma 4.3.2 to  $\tau$ .

Let  $x$  be the side length of the largest square in  $U_\sigma$ , let  $w$  be the number of vertices in the rightmost column of  $U_\sigma$  and let  $y$  be the number of rows in  $U_\sigma$  with no more than  $w$  vertices.

The labels of the bottom-most  $y$  vertices in  $E_2$  under  $\tau'$  are each  $w$  less than those same vertices under  $\sigma'$ , and so  $E_2(\tau') = E_2(\sigma') - wy$ . Meanwhile,  $E_1(\tau') > E_1(\sigma')$  because of the new label for the topmost vertex of column that was removed from  $U_\sigma$  to make  $U_\tau$ . The label of this vertex increases by at least  $xy$ , and so  $(E_1 + E_2)(\tau') \geq (E_1 + E_2)(\sigma') + xy - wy \geq (E_1 + E_2)(\sigma')$ , as  $xy - wy = (x - w)y \geq 0$  because  $x \geq w$  (with equality iff  $U_\tau = U_\sigma$ ).

Repeating the process of deleting the rightmost column of  $U_\sigma$  will continue to increase  $(E_1 + E_2 + E_3)(\tau')$ . Note that  $E_3$  does not change during the course of this procedure, which terminates when  $\tau = \rho$ , giving

$$(E_1 + E_2 + E_3)(\rho') \geq (E_1 + E_2 + E_3)(\sigma'). \quad (4.64)$$

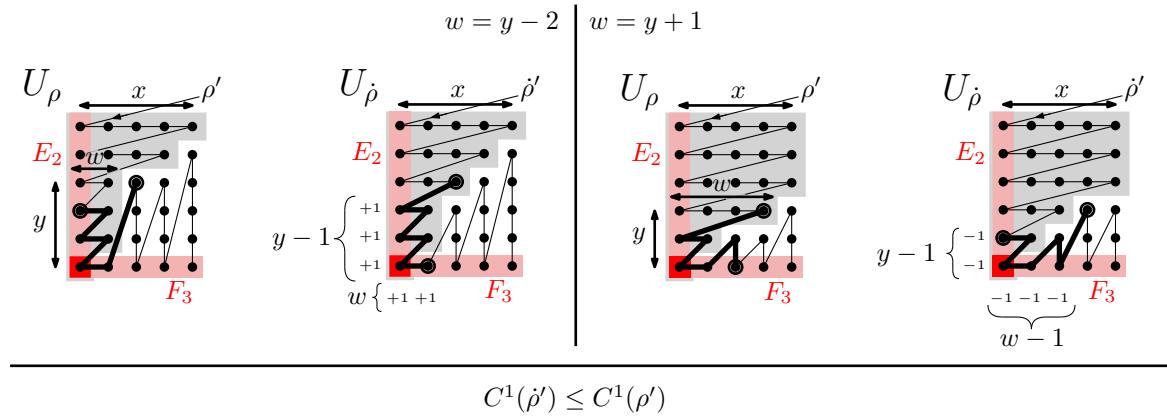
■

By the symmetric of vertex labellings that arise from Lemma 4.3.2, this claim also demonstrates that  $(F_1 + F_2 + F_3)(\rho') \leq (F_1 + F_2 + F_3)(\sigma')$ . Therefore,

$$\left( (F_1 + F_2 + F_3) - (E_1 + E_2 + E_3) \right)(\rho') \leq \left( (F_1 + F_2 + F_3) - (E_1 + E_2 + E_3) \right)(\sigma'), \quad (4.65)$$

which, via Equation 4.62, implies  $C_G^1(\rho') \leq C_G^1(\sigma')$ . □

At this stage, the only part of  $U_\rho$  left to scrutinise is its bottom-left corner: Lemma 4.3.4 details a shape for it to take (and vice-versa for the top-right corner of  $V_\rho$ ) in order to produce a new vertex labelling  $\rho'$  with  $C_G^1(\rho') \leq C_G^1(\rho)$ .



**Figure 4.15** Finding an improvement  $U_{\dot{\rho}}$  to the region  $U_{\rho}$  in Lemma 4.3.4, and hence an improved vertex labelling  $\dot{\rho}'$  which has a lower edgesum than  $\rho'$ . As a visual aid, in both cases  $w = y - 2$  and  $w = y + 1$ , the bolded vertices under scheme  $\rho'$  share the same labels as the bolded vertices under  $\dot{\rho}'$ .

**Lemma 4.3.4.** Let  $G$  be the  $N \times N$  square lattice, and let  $\rho$  be a horizontally and vertically ordered vertex labelling for  $G$  arising from applying Lemma 4.3.3 to some vertex labelling  $\sigma$ . Let  $\rho'$  be the vertex labelling resulting from applying Lemma 4.3.3 to  $\rho$ .

Let  $x$  be the side length of the largest square in  $U_{\rho}$ . Then, consider a new vertex labelling  $\dot{\rho}$  such that  $U_{\dot{\rho}}$  differs from  $U_{\rho}$  by the following: modify  $U_{\rho}$  by imposing a length of  $x$  vertices on all rows down to a height  $x$  above the bottom row, and then give the row at height  $y < x$  a length of  $y$  or  $y - 1$  for all  $y = x - 1, \dots, 1$ . Define  $V_{\dot{\rho}}$  similarly. Apply Lemma 4.3.2 to  $\dot{\rho}$  to obtain  $\dot{\rho}'$ . Then,  $C_G^1(\dot{\rho}') \leq C_G^1(\rho')$ .

*Proof.* Using Equation 4.62,

$$C_G^1(\dot{\rho}') - C_G^1(\rho') = ((F_3 - E_2) + (F_2 - E_3))(\dot{\rho}') - ((F_3 - E_2) + (F_2 - E_3))(\rho') \quad (4.66)$$

$$= \underbrace{(F_3 - E_2)(\dot{\rho}') - (F_3 - E_2)(\rho')}_{\text{involves regions } U_h \text{ and } U_{\dot{\rho}} \text{ only}} + \underbrace{(F_2 - E_3)(\dot{\rho}') + (F_2 - E_3)(\rho')}_{\text{involves regions } V_h \text{ and } V_{\dot{\rho}} \text{ only}} \quad (4.67)$$

as the vertices in  $E_1$  and  $F_1$  have the same labels under  $\rho'$  and  $\dot{\rho}'$ .

First, consider the difference in edgesum due to the difference in shape between  $U_{\dot{\rho}}$  and  $U_{\rho}$ , given in the first line of Equation 4.67. As in Figure 4.15, for  $y = 1, \dots, N - x$

take the  $y$ th row from the bottom of  $U_\rho$ , and let  $w$  be the number of vertices it contains. The vertex labelling  $\dot{\rho}'$  only differs from  $\rho'$  if  $w < y - 1$  or  $w > y$ .

*Case  $w < y - 1$ :* Suppose  $w = y - 1 - k$  for  $k \geq 1$  (Figure 4.15 contains an example with  $y = 4$ ,  $w = 2$ ,  $k = 1$ ). Choose the number of vertices in the  $y$ th row from the bottom of  $U_{\dot{\rho}}$  to be  $y - 1$ . Then the labels of the  $y - 1$  bottom-most vertices in  $E_2$  under  $\dot{\rho}'$  are each greater by  $k$  than the same vertices' labels under  $\rho'$ , i.e.  $E_2(\dot{\rho}') - E_2(\rho') = k(y - 1)$ . Similarly, the labels of the  $w$  leftmost vertices in  $F_3$  under  $\dot{\rho}'$  are each greater by  $k$  than the same vertices' labels under  $\rho'$ , i.e.  $F_3(\dot{\rho}') - F_3(\rho') = kw$ . Thus, the top line of Equation 4.67 is equal to  $k(w - (y - 1)) < 0$ .

*Case  $w > y$ :* Suppose  $w = y + k$  for  $k \geq 1$  (Figure 4.15 contains an example with  $y = 3$ ,  $w = 4$ ,  $k = 1$ ). Choose the number of vertices in the  $y$ th row from the bottom of  $U_{\dot{\rho}}$  to be  $y$ . Then the labels of the  $y - 1$  bottom-most vertices  $E_2$  under  $\dot{\rho}'$  each differ by  $-k$  from the same vertices' labels under  $\rho'$ , i.e.  $E_2(\dot{\rho}') - E_2(\rho') = -k(y - 1)$ . Similarly, the labels of the  $(w - 1)$  leftmost vertices in  $F_3$  each differ by  $-k$  from the same vertices' labels under  $\rho'$ , i.e.  $F_3(\dot{\rho}') - F_3(\rho') = -k(w - 1)$ . Thus, the top line of Equation 4.67 is equal to  $-k((w - 1) - (y - 1)) < 0$ .

Note that, working within the successive restrictions of Lemmas 4.3.2 and 4.3.3, the maximum number of vertices permitted in any row of  $U_{\dot{\rho}}$  is  $x$ . Thus, the rows increase in width from  $y = 1$  by one vertex at a time before reaching the maximum width of  $y = x$  at  $y = x$ .

By symmetry, the same rules work to construct the region  $V_{\dot{\rho}}$  such that  $C_G^1(\dot{\rho}') \leq C_G^1(\rho')$ .  $\square$

Finally, only one degree of freedom remains: the side lengths of the largest squares in  $U_{\dot{\rho}}$  and  $V_{\dot{\rho}}$ . Note that these values only affect the top and bottom lines of Equation 4.67, respectively, and hence if there exists an optimal value  $x$  for the width of  $U_{\dot{\rho}}$ , then it will also be the optimal value for the width of  $V_{\dot{\rho}}$ . Thus, we can take the side lengths of the largest squares in  $U_{\dot{\rho}}$  and  $V_{\dot{\rho}}$  to be the same.

Lemma 4.3.5 expresses the edgesum of the pattern as a function of  $N$  and  $x$ . Corollary 4.3.6 then provides the optimal value for  $x$  and hence the Mitchison–Durbin pattern  $\sigma_M$ , a solution to the edgesum problem.

**Lemma 4.3.5.** Let  $G$  be the  $N \times N$  square lattice, and let  $\dot{\rho}$  be a horizontally and vertically ordered vertex labelling that arises from applying Lemma 4.3.4 to some horizontally and vertically ordered scheme  $\sigma$ .

Let  $x$  be the side length of the largest squares of  $U_{\dot{\rho}}$  and  $V_{\dot{\rho}}$ , and let  $\dot{\rho}'$  be the vertex labelling that results from applying Lemma 4.3.2 to  $\dot{\rho}$ . Then, the edgesum of  $\dot{\rho}$  is

$$C_G^1(\dot{\rho}') = N^3 - xN^2 + 2x^2N - \frac{2}{3}x^3 + N^2 - xN - 2N + \frac{2}{3}x. \quad (4.68)$$

*Proof.* See Section 4.3.1.  $\square$

**Corollary 4.3.6.** For  $N \geq 5$ , the value of  $x$  that minimises  $C_G^1(\dot{\rho}')$  in Equation 4.68 is  $x = N - \frac{1}{2}\sqrt{2N^2 - 2N + \frac{4}{3}} \approx 0.29N$ . Round  $x$  to the nearest integer to obtain the minimum edgesum over all vertex labelling for the  $N \times N$  square lattice.

*Proof.* Treat  $N$  and  $x$  as continuous variables, and use calculus to find that minimum of  $C_G^1(\dot{\rho}')$  in Equation 4.68 occurs when  $x = N - \frac{1}{2}\sqrt{2N^2 - 2N + \frac{4}{3}} \approx 0.29N$ .  $\square$

This completes the proof. The optimal value for  $x$  in Corollary 4.3.6 is a refinement on the value  $x = (1 - \frac{1}{\sqrt{2}})N$  that Mitchison and Durbin give in [97]. Rounding this quantity gives the same integer as  $N - \frac{1}{2}\sqrt{2N^2 - 2N + \frac{4}{3}}$  for most values of  $N$ .

### 4.3.1 Edgesum of the Mitchison–Durbin pattern

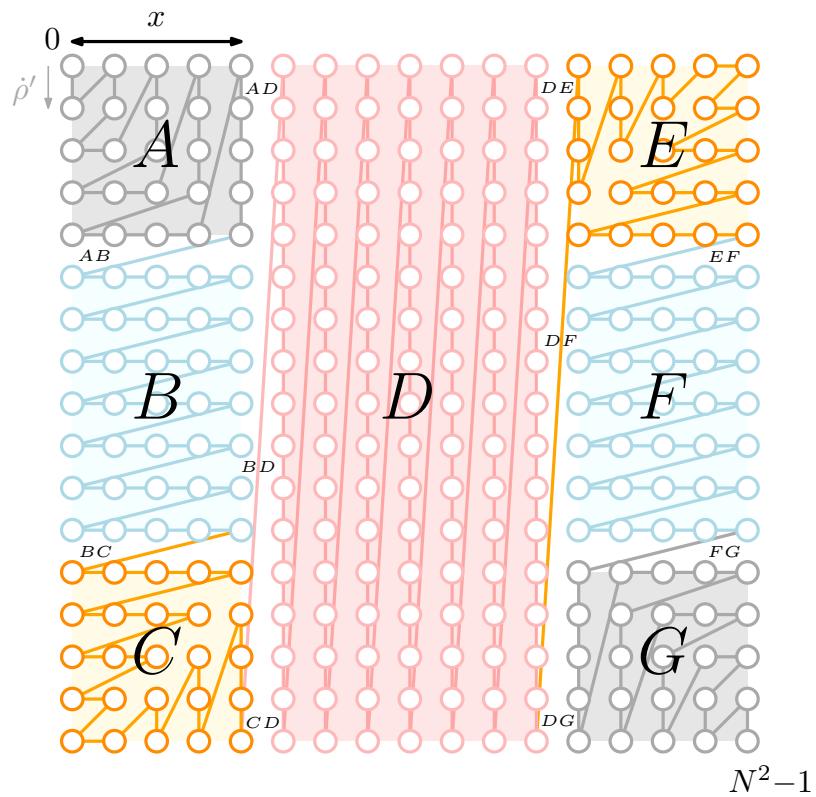
This section calculates the edgesum of the vertex labelling  $\dot{\rho}'$  in Lemma 4.3.5. Divide the vertex labelling on the  $N \times N$  grid up into regions  $A, B, \dots, G$  as in Figure 4.16. Let the label of each region also denote its edgesum. Therefore

$$\begin{aligned} C_G^1(\dot{\rho}') &= A + B + C + D + E + F + G \\ &\quad + AB + AD + BD + BC + CD \\ &\quad + DE + EF + DF + FG + DG, \end{aligned} \quad (4.69)$$

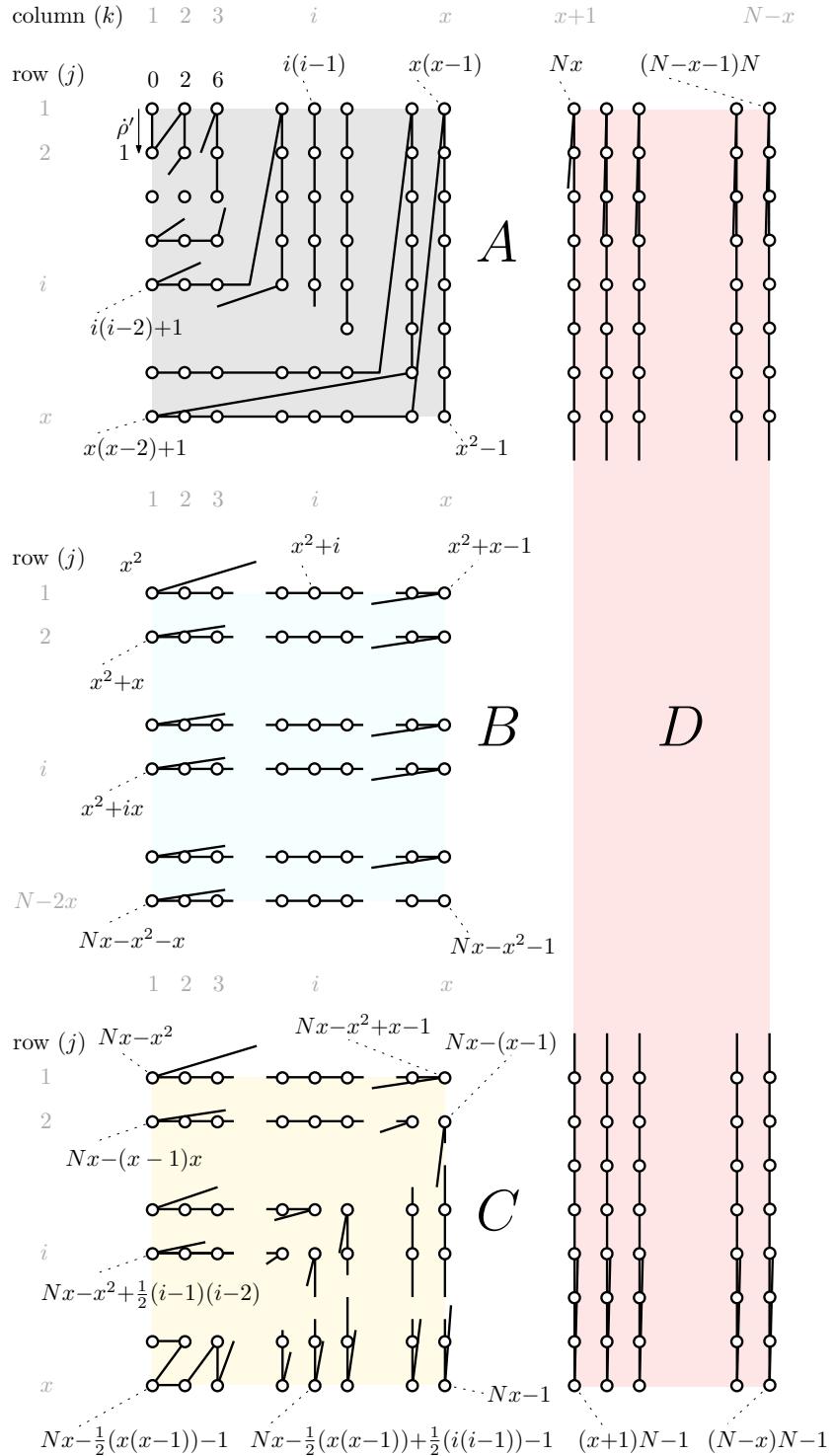
where  $AB, \dots, DG$  denote the sums of the differences between vertex labels across the interfaces between each pair of regions. Due to the symmetry of  $\dot{\rho}'$ , Equation 4.69 becomes

$$C_G^1(\dot{\rho}) = 2(A + B + C) + D + 2(AB + AD + BC + BD + CD). \quad (4.70)$$

We can derive expressions for the contributions of each region to  $C_G^1(\dot{\rho}')$  by observing the patterns in the differences between vertex labels. Figure 4.17 shows the progression of vertex labels in regions  $A$ – $D$  of the square lattice. Within each of these regions, the vertex labelling is horizontally and vertically ordered. Thus, using Equation 4.58 on



**Figure 4.16** Sections of the  $N \times N$  square lattice and a vertex labelling  $\rho'$  that satisfies Lemma 4.3.5.



**Figure 4.17** An in-depth look at the vertex labelling  $\rho'$  that arises from Lemma 4.3.5.

regions  $A$  and  $C$  gives:

$$A = \left( \sum_{k=1}^{x-1} (x(x-2) + k) + x^2 - 1 + \sum_{j=1}^x (x(x-1) + j - 1) \right) \quad (4.71)$$

$$- \left( \sum_{k=1}^x (k(k-1)) + \sum_{j=1}^x (j(j-2) + 1) \right) \quad (4.72)$$

$$= \frac{1}{6}(x-1)(6 + x(8x-1)) \quad (4.72)$$

$$= C. \quad (4.73)$$

The other regions' contributions are more straightforward to calculate:

$$B = (N-2x)(x-1) + x^2(N-2x-1) \quad (4.74)$$

$$D = N^3 - 2xN^2 - 2xN + 2x - N. \quad (4.75)$$

The vertex labels in Figure 4.17 also make it simple to calculate the contributions to  $C_G^1(\rho')$  from the interfaces between the regions:

$$AD = \sum_{j=1}^x \left( (x(x-1) + j - 1) - (Nx + j - 1) \right) = Nx^2 - x^3 + x^2 \quad (4.76)$$

$$AB = 1 - 2x + 2x^2 \quad (4.77)$$

$$BD = \frac{1}{2}(N^2 + N - 3xN + xN^2 + 2x^2 - 2x - 2x^2N) \quad (4.78)$$

$$BC = x^2 \quad (4.79)$$

$$CD = 1 - 2x + xN + x^2. \quad (4.80)$$

Substituting Equations 4.72–4.80 into Equation 4.70 gives the result,

$$C_G^1(\rho') = N^3 - xN^2 + 2x^2N - \frac{2}{3}x^3 + N^2 - xN - 2N + \frac{2}{3}x. \quad (4.81)$$

## 4.4 Conclusion

By identifying the inherent role of fermionic labelling in Pauli weight functions of qubit Hamiltonians, this chapter identified novel methods for optimising fermion–qubit mappings. Vertex labelling solutions to well-known graph problems can minimise practical costs of the qubit Hamiltonians of Jordan–Wigner transformations. Theorem 3 illustrates the effect for fermionic Hamiltonians with hopping terms forming a square

lattice of interactions, and identifies a Jordan–Wigner transformation that uses the Mitchison–Durbin pattern to produce qubit Hamiltonians with the minimum possible average Pauli weight. Asymptotically, this is an improvement of 13.9% compared to previous methods using the S–pattern.

Summarising our results, Table 4.1 compares Verstraete and Cirac’s Jordan–Wigner transformation with the S–pattern to our optimal Jordan–Wigner transformation using the fermionic labelling  $\sigma_M$ , and our two–ancilla–qubit mapping, which uses a different fermionic labelling  $\sigma_{M+2}$ . The table includes ancilla–qubit mappings with  $\mathcal{O}(N^2)$  ancilla qubits for comparison; two of which have a visual depiction in Figure 1.4.

**Table 4.1** Fermion–qubit mappings for  $G(\hat{H}_{\text{fermion}}) = N \times N$  square lattice

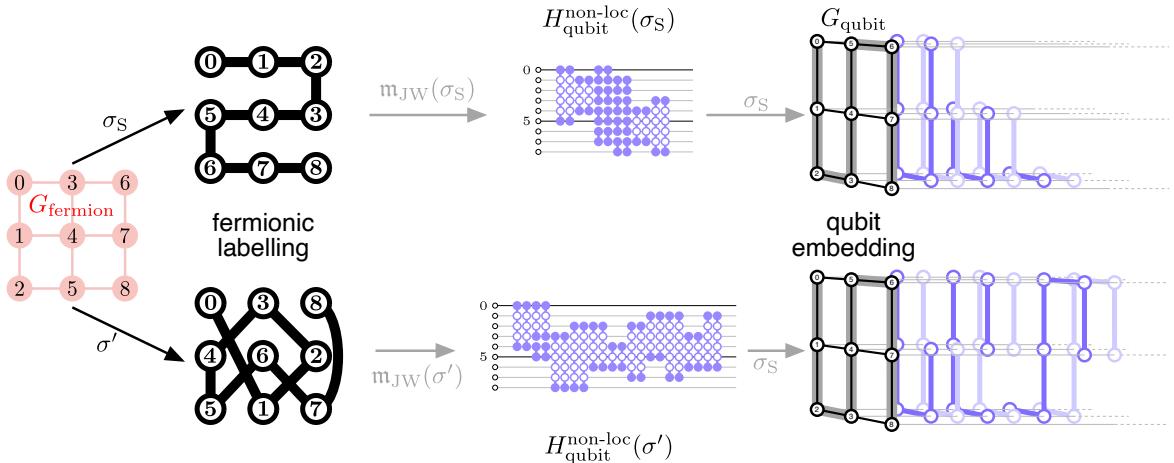
	$\mathfrak{m}_{\text{JW}}(\sigma_S)$ [60]	$\mathfrak{m}_{\text{JW}}(\sigma_M)$	$\Pi_{\tilde{\mathcal{H}}} \cdot L_{P_1} L_{P_0} \cdot \mathfrak{m}(\sigma_{M+2})$	BK superfast [64]	VC [60]	AQM [70]
Cost function	[60]	Chapter 4	Chapter 5	[64]	[60]	[70]
Qubit number	$N^2$	$N^2$	$N^2 + 2$	$2N^2 - 2N$	$2N^2$	$2N^2 - N$
Qubit/mode ratio	1	1	$1 + \frac{2}{N^2}$	$2 - \frac{2}{N}$	2	$2 - \frac{1}{N}$
Average Pauli weight	$\frac{1}{2}N + \frac{3}{2}$	$0.43N + 1.78$	$0.31N + 2.68$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Comparison between different qubit representations of the hopping terms in a fermionic system with an  $N \times N$  square lattice interaction graph. The mapping  $\mathfrak{m}_{\text{JW}}(\sigma_M)$  from Section 4.2.4 gives the minimum average Pauli weight of all  $N^2$ –qubit Jordan–Wigner transformations. The mapping  $\Pi_{\tilde{\mathcal{H}}} \cdot L_{P_1} L_{P_0} \cdot \mathfrak{m}(\sigma_{M+2})$  from Chapter 5 improves upon result while also maintaining a qubit-to-mode ratio of  $1 + \mathcal{O}(\frac{1}{N^2})$ . In comparison, the fermion–qubit mappings BK superfast, VC and AQM use  $\sim 2n^2$  qubits, but produce Hamiltonians with Pauli weight  $\mathcal{O}(1)$ . Which mapping is best ultimately depends on the resources at hand.

#### 4.4.1 Geometric locality on qubit architectures

Most discussions of fermion–qubit mappings assume access to qubits with connectivity constraints [60, 70, 77, 81, 82, 82, 83, 96, 108]. Verstraete and Cirac [60] began this line of enquiry by pursuing an ancilla–qubit mapping to transform a fermionic system to a qubit Hamiltonian with all terms geometrically local on a square lattice. Subsequent works have followed suit, working towards preserving the geometric locality of fermionic systems on qubits, which is possible if the qubit architecture matches the fermionic interaction graph.

While ancilla–free mappings cannot reproduce geometric locality, we briefly consider the problem here. Fermionic labellings influence the distribution of quantum gates



**Figure 4.18** The fermionic labelling  $\sigma \in S_n$  need not follow the underlying connectivity of the fermionic system  $G_{\text{fermion}}$  or the qubit architecture  $G_{\text{qubit}}$ . Embedding qubit Hamiltonians into restricted architectures is a separate problem to the task of choosing a fermionic labelling.

throughout qubit architectures. To see this, consider decomposing the process of representing a fermionic Hamiltonian on a quantum computer into two components: 1) the *labelling* of the fermionic modes, which projects the fermionic connectivity graph  $G_{\text{fermion}} = G(\hat{H}_{\text{fermion}})$  onto a 1D array of qubits, and then 2) the *embedding* of the 1D array of qubits into the qubit architecture  $G_{\text{qubit}}$ . Figure 4.18 visualises the distinction between the two processes.

Verstraete and Cirac’s approach is to introduce the S-pattern to label the fermions, and to use an S-pattern embedding to weave Pauli strings into the qubit lattice, as in the top row of Figure 4.18. Using the same qubit embedding as fermionic labelling is perhaps intuitive if  $G_{\text{fermion}} = G_{\text{qubit}}$ , as in this case, but in this work we emphasise that  $G_{\text{fermion}}$  need not be the same graph as  $G_{\text{qubit}}$ , and that the fermionic labelling is entirely separate to the embedding process. Indeed, as the second row of Figure 4.18 shows, the path of the fermionic labelling need not even follow the connectivity of the qubit architecture at all.

It is possible to conceive of metrics for qubit Hamiltonians that depend on qubit architectures. For example, the qubit routing problem concerns the distribution of Pauli strings on a nonlinear qubit connectivity  $G_{\text{qubit}}$  [109]. The cost function for qubit routing should penalise strings of Pauli gates that spread through  $G_{\text{qubit}}$  sparsely, and reward clustered Pauli strings. This poses a different problem to minimising the average Pauli weight of the qubit Hamiltonian, as the optimal fermionic labelling for the fermion–qubit mapping will need to cluster the Pauli strings of the qubit Hamiltonian while also minimising their weight. We expect that finding optimal fermionic labellings

for cost functions that incorporate qubit architecture is a much more difficult family of problems than simply minimising a cost function of the Pauli weight.

#### 4.4.2 Improving beyond optimality with a constant number of ancilla qubits

In Chapter 5, we provide an  $(N^2 + 2)$ -qubit mapping for fermionic Hamiltonians with  $N \times N$  square-lattice interactions. The strict limit of only two ancilla qubits, regardless of the number of fermionic modes, makes this mapping unlike all others in the literature. In comparison with the results of Theorem 3, the two ancilla qubits allow reduction of the average Pauli weight of the qubit Hamiltonian by 37.9% compared to previous methods using the S-pattern.

# Chapter 5

## Further optimisation with a constant overhead of ancilla qubits

Chapter 4 paved the way for finding optimal Jordan–Wigner transformations in the absence of ancilla qubits. However, there is a large body of work involving ancilla qubits that may appear to sit outside the grasp of our Definition 2.1.7 for fermion–qubit mappings. While capable of producing geometrically local qubit Hamiltonians with average Pauli weight  $\mathcal{O}(1)$  for any  $n$ –mode fermionic system, all ancilla–qubit mappings in the literature come at the cost of  $\mathcal{O}(n)$  additional qubits. In extending our framework to accommodate all ancilla–qubit mappings, this chapter produces a Jordan–Wigner–inspired transformation that straddles the line between ancilla–free and ancilla–qubit mappings to improve upon our prior results.

Section 5.1 expands Definition 2.1.7 to incorporate the theory of ancilla–qubit mappings. By relaxing the unitary definition of fermion–qubit mappings to an isometry, and then imposing a projection into a  $(2^n)$ –dimensional subspace, we show that our definition accommodates the most general definition of ancilla–qubit mappings in the literature. In Section 5.2, we introduce an algorithmic approach to incrementally add ancilla qubits to a mapping in order to reduce the Pauli weight of a specific problem Hamiltonian. Incremental addition of ancillas to the Jordan–Wigner transformation allows Section 5.3 to construct a novel mapping with a constant number of ancilla qubits. Revisiting the square fermionic lattice example, we construct a mapping with only two ancilla qubits. The advantage of optimal fermionic labelling becomes even more pronounced than in Chapter 4, because the two ancilla qubits can target the two most costly groups of qubit terms in the qubit Hamiltonian. Theorem 4 shows that this strategy yields qubit Hamiltonians for square lattice fermionic systems with an

average Pauli weight as much as 27.9% less than the ancilla-free optimal from Chapter 4, and 37.9% less than the naïve labellings that existed before the work in this thesis.

## 5.1 The theory of ancilla–qubit mappings

This section links the most general definition of ancilla–qubit mappings in the literature [83] to our general Definition 2.1.7 for fermion–qubit mappings from Chapter 2. We provide additional details, proving that the literature definition of ancilla–qubit mappings [83] induces a unitary map which is equivalent to a higher–dimensional ancilla–free Pauli–based mapping.

As in Chapter 4, let  $\hat{H}_{\text{fermion}} \in B_h(\mathcal{H}_f^n)$  be an  $n$ –mode fermionic Hamiltonian and suppose that it has the Majorana product decomposition

$$\hat{H}_{\text{fermion}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}, \quad (5.1)$$

where the set of terms  $\Lambda \subset B(\mathcal{H}_f^n)$  consists of products of Majorana operators. Introduce the binary parameters  $m_{\hat{h}}$  and  $M_{\hat{g}, \hat{h}}$  to encode the Hermiticity and commutation relations of the terms of  $\hat{H}_{\text{fermion}}$ :

$$\hat{h}^\dagger = (-1)^{m_{\hat{h}}} \hat{h} \iff m_{\hat{h}} = \begin{cases} 0 & \hat{h}^\dagger = \hat{h} \\ 1 & \hat{h}^\dagger = -\hat{h}, \end{cases} \quad (5.2)$$

$$\hat{g}\hat{h}\hat{g}^{-1}\hat{h}^{-1} = (-1)^{M_{\hat{g}, \hat{h}}} \hat{1} \iff M_{\hat{g}, \hat{h}} = \begin{cases} 0 & [\hat{g}, \hat{h}] = 0 \\ 1 & \{\hat{g}, \hat{h}\} = 0, \end{cases} \quad \text{for all } \hat{g}, \hat{h} \in \Lambda. \quad (5.3)$$

Let  $m > n$  and suppose that  $H_{\text{qubit}} \in B_h(\mathcal{H}_2^{\otimes m})$  is an  $m$ –qubit Hamiltonian with Pauli decomposition  $H_{\text{qubit}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h$ , where, for each distinct  $\hat{h} \in \Lambda$ , the term  $h \in B(\mathcal{H}_2^{\otimes m})$  inherits the commutation and Hermiticity relations of Equation 5.2 and 5.3 from the fermionic Hamiltonian, i.e.

$$h^\dagger = (-1)^{m_{\hat{h}}} h, \quad ghg^{-1}h^{-1} = (-1)^{M_{\hat{g}, \hat{h}}} \hat{1}^{\otimes m} \quad \text{for all } \hat{g}, \hat{h} \in \Lambda. \quad (5.4)$$

Equation 5.4 is the condition that [83] gives in order to define ancilla–qubit mappings. We formalise this definition with Lemma 5.1.1, which uses our general Definition 2.1.7 to justify that there is an isometry between  $\hat{H}_{\text{fermion}}$  and  $H_{\text{qubit}}$ .

**Lemma 5.1.1.** (*Stated but not proved explicitly in [83].*) There exists an isometry  $\mathfrak{l} : \mathcal{H}_f^n \rightarrow \mathcal{H}_2^{\otimes m}$  which maps the fermionic operators  $\hat{h}$  to the qubit operators  $h$ ,

$$\mathfrak{l} : \hat{h} \longmapsto h = \mathfrak{l} \cdot \hat{h} \cdot \mathfrak{l}^\dagger \quad \text{for all } \hat{h} \in \Lambda, \quad (5.5)$$

and hence produces the qubit Hamiltonian via  $H_{\text{qubit}} = \mathfrak{l} \cdot \hat{H}_{\text{fermion}} \cdot \mathfrak{l}^\dagger \in B_h(\mathcal{H}_2^{\otimes m})$ .

*Proof.* Let  $\{\Gamma_i\}_{i=0}^{2n-1} \subset B_h(\mathcal{H}_2^{\otimes m})$  be a set of Hermitian, mutually anticommuting operators satisfying the CARs of Equation 2.13 such that each term of the qubit Hamiltonian  $\{h \mid \hat{h} \in \Lambda\}$  has an expression as a product of the operators in  $\{\Gamma_i\}_{i=0}^{2n-1}$ . By Proposition 1, there exists a unitary matrix  $U_m \in \mathcal{U}_m$  such that  $U_m(\gamma_i \otimes \mathbb{1}^{\otimes(m-n)})U_m^\dagger = \Gamma_i$  for all  $i \in [2n]$ , where  $\{\gamma_i\}_{i=0}^{2n-1}$  are the Majorana representations of the  $n$ -mode Jordan–Wigner transformation. Take the linear map  $\mathfrak{l} := U_m \cdot (\mathfrak{m}_{\text{JW}}|_{\mathcal{H}_f^n}) : \mathcal{H}_f^n \rightarrow \mathcal{H}_2^{\otimes m}$ , where  $\mathfrak{m}_{\text{JW}}|_{\mathcal{H}_f^n}$  is the restriction of the  $m$ -node Jordan–Wigner transformation to the  $n$ -mode Hilbert space  $\mathcal{H}_f^n$ ; the map  $\mathfrak{l}$  is, by definition, an isometry satisfying Equation 5.5.  $\square$

The spectrum of  $H_{\text{qubit}}$  thus contains the spectrum of  $\hat{H}_{\text{fermion}}$ , and a subset of eigenvectors of  $H_{\text{qubit}}$  are isometrically equivalent to the eigenvectors of  $\hat{H}_{\text{fermion}}$ .

The higher dimension of  $H_{\text{qubit}}$  gives the qubit Hamiltonian spectral properties that are extraneous to the fermionic system of  $\hat{H}_{\text{fermion}}$ . The relevant  $2^n$ -dimensional information exists in the subspace  $\widetilde{\mathcal{H}} < \mathcal{H}_2^{\otimes m}$  of the  $m$ -qubit quantum register that preserves the algebraic dependence of the operators in  $\Lambda$ :

$$\begin{aligned} \widetilde{\mathcal{H}} := \left\{ |\tilde{\psi}\rangle \in \mathcal{H}_2^{\otimes m} \mid \left( \prod_{\hat{h} \in \Lambda'} h \right) |\tilde{\psi}\rangle = \pm |\tilde{\psi}\rangle \right. \\ \left. \text{for all } \Lambda' \subset \Lambda \text{ for which } \prod_{\hat{h} \in \Lambda'} \hat{h} = \pm \hat{1} \right\}. \end{aligned} \quad (5.6)$$

**Definition 5.1.2.** (*General ancilla–qubit mapping definition, inspired by [83].*)

An *ancilla–qubit mapping*  $\Pi_{\widetilde{\mathcal{H}}} \cdot \mathfrak{l}$  is a unitary operator that combines the process of applying an isometry  $\mathfrak{l} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes m})$  to an  $n$ -mode fermionic Hamiltonian  $\hat{H}_{\text{fermion}}$  and applying the projection  $\Pi_{\widetilde{\mathcal{H}}}$  from the  $(2^m)$ -dimensional Hilbert space  $\mathcal{H}_2^{\otimes m}$  into the  $(2^n)$ -dimensional subspace  $\widetilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes m}$ . The  $n$ -mode fermionic Hamiltonian  $\hat{H}_{\text{fermion}}$  and  $\widetilde{H} = H_{\text{qubit}}|_{\widetilde{\mathcal{H}}}$  are unitarily equivalent. Thus, ancilla–qubit mappings are useful to fermionic simulation in much the same way as the ancilla–free mappings of Definition 2.1.7, if there is access to a sufficient number of extra qubits.

**Definition 5.1.3.** (*The stabilisers of ancilla–qubit mappings.*) The operators that define the relevant subspace  $\tilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes m}$  for an ancilla–qubit mapping  $\Pi_{\tilde{\mathcal{H}}} \cdot \mathfrak{l}$ ,

$$\left\{ \left( \prod_{\hat{h} \in \Lambda'} h \right) \in B(\mathcal{H}_2^{\otimes m}) \quad \middle| \quad \Lambda' \subset \Lambda \text{ such that } \prod_{\hat{h} \in \Lambda'} \hat{h} = \hat{1} \right\}, \quad (5.7)$$

are the *stabilisers* of the ancilla–qubit mapping.

As an example of the requirement to restrict higher–dimensional qubit Hamiltonians to a stabiliser subspace, the anti–Hermitian fermionic operators  $\hat{h}_1 = \hat{\gamma}_0 \hat{\gamma}_1$ ,  $\hat{h}_2 = \hat{\gamma}_1 \hat{\gamma}_2$ ,  $\hat{h}_3 = \hat{\gamma}_2 \hat{\gamma}_0$  mutually anticommute and multiply to  $\hat{h}_1 \hat{h}_2 \hat{h}_3 = \hat{1}$ . The anti–Hermitian qubit operators  $h_1 = iX_0$ ,  $h_2 = iZ_0 X_1$ ,  $h_3 = iZ_0 Z_1 X_2$  observe the same commutation relations, and thus induce an isometric map  $\mathfrak{l} : \hat{h}_i \mapsto h_i$  from  $\mathcal{H}_f^2$  to  $\mathcal{H}_2^{\otimes 3}$ , but the product of the operators is  $h_1 h_2 h_3 = iY_0 Y_1 X_2$  and not  $\mathbb{1}^{\otimes 3}$ . Without restricting, then, the isometry  $\mathfrak{l}$  would unfaithfully represent the fermionic operator  $\hat{h}_1 \hat{h}_2 \hat{h}_3$  as a nontrivial qubit operation.

## 5.2 Incremental addition of ancilla qubits

This section develops a process for designing ancilla–qubit mappings, formalising and generalising the auxiliary qubit method of [70]. The ancilla–qubit mappings that are abundant in the literature, e.g. [59, 60, 77, 82, 83, 110], have the conceptual starting point of geometrically local Hamiltonians, leaving the fermion–qubit mapping implicit as Section 1.3.1 described. In contrast, the proposal in this section starts with an explicit fermion–qubit mapping of the form in Definition 2.1.7, and then incrementally adds ancilla qubits one at a time to reduce the Pauli weight of the most non-local terms of a target qubit Hamiltonian.

### 5.2.1 The first ancilla qubit

Suppose that the  $n$ –mode Pauli–based mapping  $\mathfrak{m}$  produces the qubit Hamiltonian  $H_{\text{qubit}} = \mathfrak{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathfrak{m}^\dagger \in B_h(\mathcal{H}_2^{\otimes n})$  from the fermionic Hamiltonian  $\hat{H}_{\text{fermion}} \in B_h(\mathcal{H}_f^n)$ . Let  $\Lambda \subset B(\mathcal{H}_f^n)$  be the terms of the Majorana product decomposition  $\hat{H}_{\text{fermion}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}$ , and let the qubit Hamiltonian have Pauli decomposition

$$H_{\text{qubit}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h. \quad (5.8)$$

This section uses a single ancilla qubit to reduce the average Pauli weight of the qubit Hamiltonian. Let  $P \in \mathcal{P}_n$  be a Hermitian  $n$ -qubit Pauli operator, and let  $L_P \in B(\mathcal{H}_2^{\otimes n}, \mathcal{H}_2^{\otimes(n+1)})$  be a linear operator that conditionally multiplies the terms of  $H_{\text{qubit}}$  by  $P$ , depending on whether doing so reduces the Pauli weight, acting on the ancilla qubit via:

$$L_P : h \longmapsto h' := L_P \cdot h \cdot (L_P)^\dagger = \begin{cases} hP \otimes (\kappa_h)_n & W(hP) < W(h) \\ h \otimes (\kappa_h)_n & W(h) \leq W(hP). \end{cases} \quad (5.9)$$

In Equation 5.9, allocation of the operators  $\kappa_h \in \{\mathbb{1}, X, Y, Z\}$  must ensure that the terms of the  $(n + 1)$ -qubit Hamiltonian

$$H'_{\text{qubit}} = L_P \cdot H_{\text{qubit}} \cdot (L_P)^\dagger = (L_P \cdot \mathfrak{m}) \cdot \hat{H}_{\text{fermion}} \cdot (L_P \cdot \mathfrak{m})^\dagger = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h' \quad (5.10)$$

satisfy the same Hermiticity and commutation relations as the terms of  $\hat{H}_{\text{fermion}}$ . That is, the terms must meet the conditions

$$(h')^\dagger = (-1)^{m_{\hat{h}}} h' \quad (5.11)$$

$$(g')^{-1}(h')^{-1}g'h' = (-1)^{M_{\hat{h}, \hat{g}}} \mathbb{1}^{\otimes(n+1)}. \quad (5.12)$$

to ensure, via Lemma 5.1.1, that the map  $\mathfrak{l} = L_P \cdot \mathfrak{m} : \mathcal{H}_f^n \rightarrow \mathcal{H}_2^{\otimes(n+1)}$  is an isometric operator of the form in Equation 5.5.

The isometry requirements in Equations 5.11 and 5.12 dictate valid assignments of  $\kappa_h$  for each term  $h$  of the qubit Hamiltonian. Choosing single-qubit Hermitian Pauli operators for each  $\kappa_h$  automatically satisfies the Hermiticity requirement in Equation 5.11. Consider for each term  $h$  whether or not  $L_P$  reduces the Pauli weight of the term, i.e. whether  $h' = hP \otimes \kappa_g$  or  $h' = h \otimes \kappa_g$ , and apply the requirement of Equation 5.12:

**1** If  $g' = g \otimes \kappa_g$  and  $h' = h \otimes \kappa_h$ , then

$$(g')^{-1}(h')^{-1}gh = g^{-1}h^{-1}gh \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n \quad (5.13)$$

$$= (-1)^{M_{\hat{g}, \hat{h}}} \mathbb{1}^{\otimes n} \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n, \quad (5.14)$$

which implies that  $[\kappa_g, \kappa_h] = 0$ .

**2** If  $g' = gP \otimes \kappa_g$  and  $h' = h \otimes \kappa_h$ , then, using  $h^{-1} = h^\dagger$  to express  $[h^{-1}, P^{-1}] = [h, P]$  and  $\{h^{-1}, P^{-1}\} = \{h, P\}$ ,

$$(g')^{-1}(h')^{-1}gh \quad (5.15)$$

$$= g^{-1}P^{-1}g^{-1}hgP \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n \quad (5.16)$$

$$= \begin{cases} (-1)^{M_{g,h}} \otimes \mathbb{1}^{\otimes n} \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n & [h, P] = 0 \\ -(-1)^{M_{g,h}} \otimes \mathbb{1}^{\otimes n} \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n & \{h, P\} = 0, \end{cases} \quad (5.17)$$

which implies that  $\{\kappa_g, \kappa_h\} = 0$  if  $\{g, P\} = 0$ , or that  $[\kappa_g, \kappa_h] = 0$  if  $[g, P] = 0$ .

**3** If  $g' = gP \otimes \kappa_g$  and  $h' = hP \otimes \kappa_h$ , then

$$(g')^{-1}(h')^{-1}gh = P^{-1}g^{-1}P^{-1}h^{-1}gPhP \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n \quad (5.18)$$

$$= \begin{cases} (-1)^{M_{g,h}} \otimes \mathbb{1}^{\otimes n} \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n & [g, P] = [h, P] = 0 \\ & \text{or} \\ & \{g, P\} = \{h, P\} = 0 \\ -(-1)^{M_{g,h}} \otimes \mathbb{1}^{\otimes n} \otimes (\kappa_g^{-1}\kappa_h^{-1}\kappa_g\kappa_h)_n & [g, P] = \{h, P\} = 0 \\ & \text{or} \\ & \{g, P\} = [h, P] = 0, \end{cases} \quad (5.19)$$

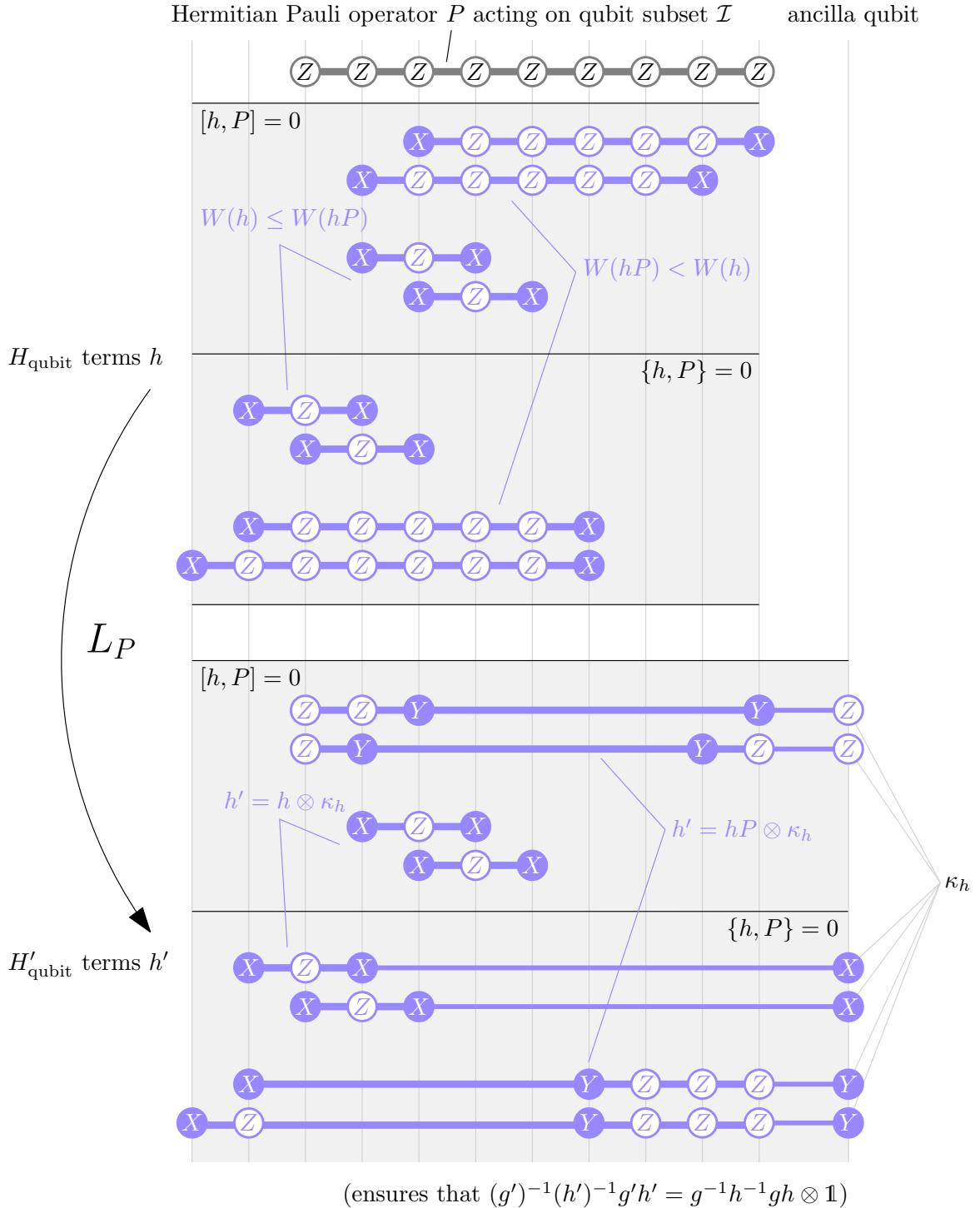
which implies that  $[\kappa_g, \kappa_h] = 0$  if exactly one of  $\{g, P\}$  and  $\{h, P\}$  is equal to 0, and that  $\{\kappa_g, \kappa_h\} = 0$  otherwise.

Figure 5.1 illustrates the following satisfying assignment for the operators  $\kappa_h$ , which ensure that the modified terms  $h' = h \otimes \kappa_h$  uphold conditions **1–3**:

$$h' = \begin{cases} hP \otimes Z_n & [h, P] = 0 \quad \text{and} \quad W(hP) < W(h) \\ h \otimes X_n & \{h, P\} = 0 \quad \text{and} \quad W(P) \leq W(hP) \\ hP \otimes Y_n & \{h, P\} = 0 \quad \text{and} \quad W(hP) < W(h) \\ h \otimes \mathbb{1}_n & [h, P] = 0 \quad \text{and} \quad W(P) \leq W(hP). \end{cases} \quad (5.20)$$

It remains to identify the subspace  $\widetilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes(n+1)}$  within which  $H'_{\text{qubit}}$  is unitarily equivalent to  $\hat{H}_{\text{fermion}}$ . As there is only one ancilla qubit, there is only one stabiliser to find: Lemma 5.2.1 achieves the task for mappings that yield Equation 5.20.

**Lemma 5.2.1.** The stabiliser of any single–ancilla–qubit mapping that maps  $h \mapsto h'$  according to Equation 5.20 is  $P \otimes Z_n$ . That is, for any subset  $\Lambda' \subset \Lambda$  of the terms of  $\hat{H}_{\text{fermion}}$  that multiplies to the fermionic identity, the product of the corresponding



**Figure 5.1** The operator  $L_P$  reduces the weight of terms in a qubit Hamiltonian by conditionally multiplying each term by a Hermitian Pauli operator  $P$  if doing so would reduce the Pauli weight of each term. To ensure that the modified terms satisfy the same commutation relations amongst as before, it is also necessary for many of the terms to act with a corrective single-qubit operator  $\kappa_h$  on the ancilla.

terms in  $H'_{\text{qubit}}$  is either the identity or  $P \otimes Z_n$ :

$$\prod_{\hat{h} \in \Lambda'} \hat{h} = \pm \hat{1} \quad \xrightarrow{\text{apply } \mathfrak{m}} \quad \prod_{\hat{h} \in \Lambda'} h = \pm \mathbb{1}^{\otimes n} \quad \xrightarrow{\text{apply } L_P} \quad \prod_{\hat{h} \in \Lambda'} h' = \begin{cases} \pm \mathbb{1}^{\otimes(n+1)} \\ \pm P \otimes Z_n \end{cases} \quad (5.21)$$

*Proof.* Let  $\Lambda = \{\hat{h}_1, \hat{h}_2, \dots, \hat{h}_k\}$  and suppose, without loss of generality, that  $\prod_{i=1}^k \hat{h}_i = \hat{1}$ . Then the product of  $n$ -qubit Pauli operators is  $\prod_{i=1}^k h_i = \mathbb{1}^{\otimes n}$ , where  $h_i = \mathfrak{m} \cdot \hat{h}_i \cdot \mathfrak{m}^\dagger$ . Let  $h'_i = L_P \cdot h_i \cdot (L_P)^\dagger$ . Another way of writing Equation 5.20 is

$$h'_i = \begin{cases} (h_i \otimes \mathbb{1}_n) (P \otimes Z_n) & [h_i, P] = 0 \quad \text{and} \quad W(h_i P) < W(h) \\ (h_i \otimes X) (\mathbb{1}^{\otimes n} \otimes \mathbb{1}_n) & \{h_i, P\} = 0 \quad \text{and} \quad W(P) \leq W(h_i P) \\ (h_i \otimes X) \otimes (P \otimes Z_n) & \{h_i, P\} = 0 \quad \text{and} \quad W(h_i P) < W(h) \\ (h_i \otimes \mathbb{1}_n) \otimes (\mathbb{1}^{\otimes n} \otimes \mathbb{1}_n) & [h_i, P] = 0 \quad \text{and} \quad W(P) \leq W(h_i P). \end{cases} \quad (5.22)$$

From Equation 5.22, it is apparent that the only circumstance in which Equation 5.21 does not hold is if an odd number of the operators in  $\Lambda'$  anticommute with  $P$ ; that is,

$$\prod_{i=1}^k h'_i = \begin{cases} \mathbb{1}^{\otimes(n+1)} & \{h_i, P\} = 0 \text{ for an even number of } \hat{h}_i \in \Lambda', \text{ and} \\ & [h_i, P] = 0 \text{ for an even number of } \hat{h}_i \in \Lambda' \\ P \otimes Z_n & \{h_i, P\} = 0 \text{ for an even number of } \hat{h}_i \in \Lambda', \text{ and} \\ & [h_i, P] = 0 \text{ for an odd number of } \hat{h}_i \in \Lambda' \\ \text{otherwise} & \{h_i, P\} = 0 \text{ for an odd number of } \hat{h}_i \in \Lambda'. \end{cases} \quad (5.23)$$

The following claim thus completes the proof.

Claim: In the set of fermionic operators  $\{\hat{h}_1, \hat{h}_2, \dots, \hat{h}_k\}$  with  $\prod_{i=1}^k \hat{h}_i = \hat{1}$ , an even number of the qubit operators  $h_i$  satisfy  $\{h_i, P\} = 0$ .

Proof: Note that one operator is proportional to the product of the others, i.e.

$$\prod_{i=1}^k \hat{h}_i = \hat{1} \implies \hat{h}_1 = \pm \prod_{i=2}^k \hat{h}_i \xrightarrow{\text{apply } \mathfrak{m}} h_1 = \pm \prod_{i=2}^k h_i. \quad (5.24)$$

If  $[h_1, P] = 0$ , then  $[\prod_{i=2}^k h_i, P] = 0$  and so an even number of the operators  $\{h_i\}_{i=2}^k$  anticommute with  $P$ . Otherwise, if  $\{h_1, P\} = 0$ , then  $\{\prod_{i=2}^k h_i, P\} = 0$  and so an odd number of the operators  $\{h_i\}_{i=2}^k$  anticommute with  $P$ . Either way, the total number of operators in  $\Lambda'$  that anticommute with  $P$  is even. ■ □

Lemma 5.2.1 shows that the Hilbert space within which  $H'_{\text{qubit}}$  is unitarily equivalent to  $\hat{H}_{\text{fermion}}$  is the stabiliser space of  $P \otimes Z_n$ ,

$$\widetilde{\mathcal{H}} = \left\{ |\tilde{\psi}\rangle \in \mathcal{H}_2^{\otimes(n+1)} \mid (P \otimes Z_n) |\tilde{\psi}\rangle = |\tilde{\psi}\rangle \right\}. \quad (5.25)$$

The precise formula for the single–ancilla–qubit mapping is thus the unitary operator

$$(\Pi_{\widetilde{\mathcal{H}}} \cdot L_P \cdot \mathbf{m}) : \mathcal{H}_f^n \longrightarrow \widetilde{\mathcal{H}}, \quad (5.26)$$

where  $\Pi_{\widetilde{\mathcal{H}}}$  is the projection from  $\mathcal{H}_2^{\otimes(n+1)}$  onto the stabiliser space of  $P \otimes Z_n$ . The final  $(n+1)$ –qubit Hamiltonian is equal to

$$\widetilde{H} = (\Pi_{\widetilde{\mathcal{H}}} \cdot L_P \cdot \mathbf{m}) \cdot \hat{H}_{\text{fermion}} \cdot (\Pi_{\widetilde{\mathcal{H}}} \cdot L_P \cdot \mathbf{m})^\dagger \quad (5.27)$$

$$= H_{\text{qubit}}|_{\widetilde{\mathcal{H}}} \quad (5.28)$$

$$= \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \tilde{h}, \quad (5.29)$$

and replicates the properties of  $\hat{H}_{\text{fermion}}$  with terms  $\tilde{h}$  that have a lower Pauli weight than both  $H_{\text{qubit}}$  and  $H'_{\text{qubit}}$ .

At this point the reader may be concerned that the new-and-improved Hamiltonian  $\widetilde{H}$  might not preserve the subspace  $\widetilde{\mathcal{H}}$  of interest. If true, this would invalidate any proposal to simulate  $\hat{H}_{\text{fermion}}$  with the single–ancilla–qubit mapping, as the action of  $\widetilde{H}$  could evolve the system out of the subspace  $\widetilde{\mathcal{H}}$ , invalidating the conditions for unitary equivalence between the fermionic and qubit Hamiltonians. As a sanity check, notice from Equation 5.20 that  $[h', P] = 0$  for all terms of  $H'_{\text{qubit}}$ , and so  $\widetilde{H}$  does indeed fix the subspace  $\widetilde{\mathcal{H}}$ .

### 5.2.2 The second ancilla qubit

Consider repeating the process of Section 5.2.1 with a second ancilla qubit: let  $Q \in \mathcal{P}_n$  be a Hermitian  $n$ –qubit Pauli operator and let  $L_Q$  be a linear map that acts on the terms of the  $(n+1)$ –qubit Hamiltonian  $H'_{\text{qubit}}$  along with a second ancilla qubit via

$$L_Q : h' \longmapsto h'' := L_Q L_P \cdot h \cdot (L_Q L_P)^\dagger \quad (5.30)$$

$$= \begin{cases} h'(Q \otimes \mathbb{1}_n) \otimes (\kappa_{h'})_{n+1} & W(h'Q) < W(h') \\ h' \otimes (\kappa_{h'})_{n+1} & W(h') \leq W(h'Q). \end{cases} \quad (5.31)$$

Ensuring that  $(L_Q L_P) \cdot \mathbf{m}$  is an isometric operator of the form in Equation 5.5 requires ensuring that the terms of the twice-modified Hamiltonian

$$H''_{\text{qubit}} = L_Q \cdot H'_{\text{qubit}} \cdot (L_Q)^\dagger = (L_Q L_P) \cdot H_{\text{qubit}} \cdot (L_Q L_P)^\dagger = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h'' \quad (5.32)$$

satisfy the Hermiticity and commutation relations of  $\hat{H}_{\text{fermion}}$ , i.e.

$$(h'')^\dagger = (-1)^{m_{\hat{h}}} h'' \quad (5.33)$$

$$(g'')^{-1}(h'')^{-1} g'' h'' = (-1)^{M_{\hat{h}, \hat{g}}} \mathbb{1}^{\otimes(n+2)}. \quad (5.34)$$

The requirements **1–3** reappear for the isometry of  $\mathbf{l} = (L_Q L_P) \cdot \mathbf{m}$ , pertaining in this instance to the operators  $\kappa_{h'}$  and  $Q$ , which has a satisfying assignment much like Equation 5.20. However, unlike in the single-ancilla case, there is a fourth scenario:

**4** If  $h'' = (hP \otimes (\kappa_h)_n \otimes \mathbb{1}_{n+1})$  and  $g'' = (gQ \otimes \mathbb{1}_n \otimes (\kappa_{g'})_{n+2})$ , then

$$(g'')^{-1}(h'')^{-1} g'' h'' = Ph^\dagger Qg^\dagger hPgQ \otimes \mathbb{1}_n \otimes \mathbb{1}_{n+1} \quad (5.35)$$

$$= \begin{cases} P(Qh^\dagger - [Q, h^\dagger])g^\dagger hPgQ \otimes \mathbb{1}^{\otimes 2} & [h^\dagger, Q] = 0 \\ P(\{h^\dagger, Q\} - Qh^\dagger)g^\dagger hPgQ \otimes \mathbb{1}^{\otimes 2} & \{h^\dagger, Q\} = 0 \end{cases} \quad (5.36)$$

$$= PQh^\dagger g^\dagger hgPQ \otimes \mathbb{1}^{\otimes 2} \quad (5.37)$$

$$= (-1)^{M_{\hat{h}, \hat{g}}} PQPQ \otimes \mathbb{1}^{\otimes 2}, \quad (5.38)$$

which satisfies Equation 5.34 if and only if  $[P, Q] = 0$ . Therefore, an additional requirement for the  $(n+2)$ -qubit mapping is that the new Pauli operator  $Q$  must commute with the previous Pauli operator  $P$ .

The second step is to find the Hilbert space  $\widetilde{\mathcal{H}}$  within which  $H''_{\text{qubit}}$  restricts to a Hamiltonian  $\widetilde{H}$  that is unitarily equivalent to  $\hat{H}_{\text{fermion}}$ . Following the same logic as Lemma 5.2.1, the intersection of the stabiliser states of  $(P \otimes Z_n \otimes \mathbb{1}_{n+1})$  and  $(Q \otimes \mathbb{1}_n \otimes Z_{n+1})$  is the valid subspace. The commutation of the Pauli operators  $P$  and  $Q$  guarantees its existence:

$$\widetilde{\mathcal{H}} = \left\{ |\tilde{\psi}\rangle \in \mathcal{H}_2^{\otimes(n+2)} \mid (P \otimes Z_n \otimes \mathbb{1}_{n+1}) |\tilde{\psi}\rangle = (Q \otimes \mathbb{1}_n \otimes Z_{n+1}) |\tilde{\psi}\rangle = |\tilde{\psi}\rangle \right\}. \quad (5.39)$$

The ancilla–qubit mapping  $\Pi_{\tilde{\mathcal{H}}} \cdot L_Q L_P \cdot \mathbf{m}$  produces the  $(n+2)$ –qubit Hamiltonian

$$\widetilde{H} = (\Pi_{\tilde{\mathcal{H}}} \cdot L_Q L_P \cdot \mathbf{m}) \cdot \hat{H}_{\text{fermion}} \cdot (\Pi_{\tilde{\mathcal{H}}} \cdot L_Q L_P \cdot \mathbf{m})^\dagger \quad (5.40)$$

$$= H''_{\text{qubit}}|_{\tilde{\mathcal{H}}} \quad (5.41)$$

$$= \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \tilde{h}, \quad (5.42)$$

which has terms  $\tilde{h}$  with a lower Pauli weight than  $H_{\text{qubit}}$ ,  $H'_{\text{qubit}}$  and  $H''_{\text{qubit}}$ .

### 5.2.3 Subsequent ancilla qubits

Iteratively applying the procedure of Section 5.2.1 introduces more ancilla qubits in a bid to further reduce the Pauli weight of qubit Hamiltonian terms. After  $r$  iterations, the result is an  $(n+r)$ –qubit Hamiltonian

$$H_{\text{qubit}}^{(r)} = \left( \prod_{i=0}^{r-1} L_{P_i} \cdot \mathbf{m} \right) \cdot \hat{H}_{\text{fermion}} \cdot \left( \prod_{i=0}^{r-1} L_{P_i} \cdot \mathbf{m} \right)^\dagger = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \tilde{h}^{(r)}, \quad (5.43)$$

where the set  $\{P_i\}_{i=0}^{r-1}$  consists of Hermitian  $n$ –qubit Pauli operators, and  $L_{P_j}$  acts on the terms of the  $(n+j)$ –qubit Hamiltonian

$$H_{\text{qubit}}^{(j)} = \left( \prod_{i=0}^{j-1} L_{P_i} \cdot \mathbf{m} \right) \cdot \hat{H}_{\text{fermion}} \cdot \left( \prod_{i=0}^{j-1} L_{P_i} \cdot \mathbf{m} \right)^\dagger = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h^{(j)} \quad (5.44)$$

to produce the  $(n+j+1)$ –qubit Hamiltonian  $H_{\text{qubit}}^{(j+1)}$  with terms  $h^{(j+1)}$  via

$$L_{P_j} : h^{(j)} \longmapsto h^{(j+1)} = \begin{cases} h^{(j)}(P_j \otimes \mathbf{1}^{\otimes(j-1)}) \otimes (\kappa_{h^{(j)}})_{n+j} & W(h^{(j)} P_j) < W(h^{(j)}) \\ h^{(j)} \otimes (\kappa_{h^{(j)}})_{n+j} & W(h^{(j)}) \leq W(h^{(j)} P_j). \end{cases} \quad (5.45)$$

Conditions **1–3** apply individually to each ancilla qubit, resulting in assignments for  $\kappa_{h^{(j)}}$  of the form in Equation 5.20. Extrapolating condition **4**, the set of Pauli operators  $\{P_i\}_{i=0}^{r-1}$  must also be mutually commuting in order to ensure that  $\mathbf{l} = \left( \prod_{i=0}^{r-1} L_{P_i} \cdot \mathbf{m} \right)$  is an isometry of the form in Equation 5.5.

The subspace  $\widetilde{\mathcal{H}} < \mathcal{H}_2^{\otimes(n+r)}$  of the  $(n+r)$ –qubit register within which  $H_{\text{qubit}}^{(r)}$  shares the spectrum of  $\hat{H}_{\text{fermion}}$  consists of the simultaneous stabiliser states of the commuting

Pauli operators  $\{P_i \otimes Z_{n+i}\}_{i=0}^{r-1}$ :

$$\widetilde{\mathcal{H}} = \left\{ |\tilde{\psi}\rangle \in \mathcal{H}_2^{\otimes(n+r)} \mid (P_i \otimes Z_{n+i}) |\tilde{\psi}\rangle = |\tilde{\psi}\rangle \text{ for all } i \in \{1, \dots, r\} \right\}. \quad (5.46)$$

The  $r$ -ancilla–qubit mapping is

$$\left( \Pi_{\widetilde{\mathcal{H}}} \cdot \prod_{i=0}^{r-1} L_{P_i} \cdot \mathbf{m} \right) : \mathcal{H}_f^n \longrightarrow \widetilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes(n+r)}, \quad (5.47)$$

and the modified  $(n+r)$ -qubit Hamiltonian is

$$\widetilde{H} = \left( \Pi_{\widetilde{\mathcal{H}}} \cdot \prod_{i=0}^{r-1} L_{P_i} \cdot \mathbf{m} \right) \hat{H}_{\text{fermion}} \cdot \left( \Pi_{\widetilde{\mathcal{H}}} \cdot \prod_{i=1}^r L_{P_i} \cdot \mathbf{m} \right)^\dagger \quad (5.48)$$

$$= H_{\text{qubit}}^{(r)} |_{\widetilde{\mathcal{H}}} \quad (5.49)$$

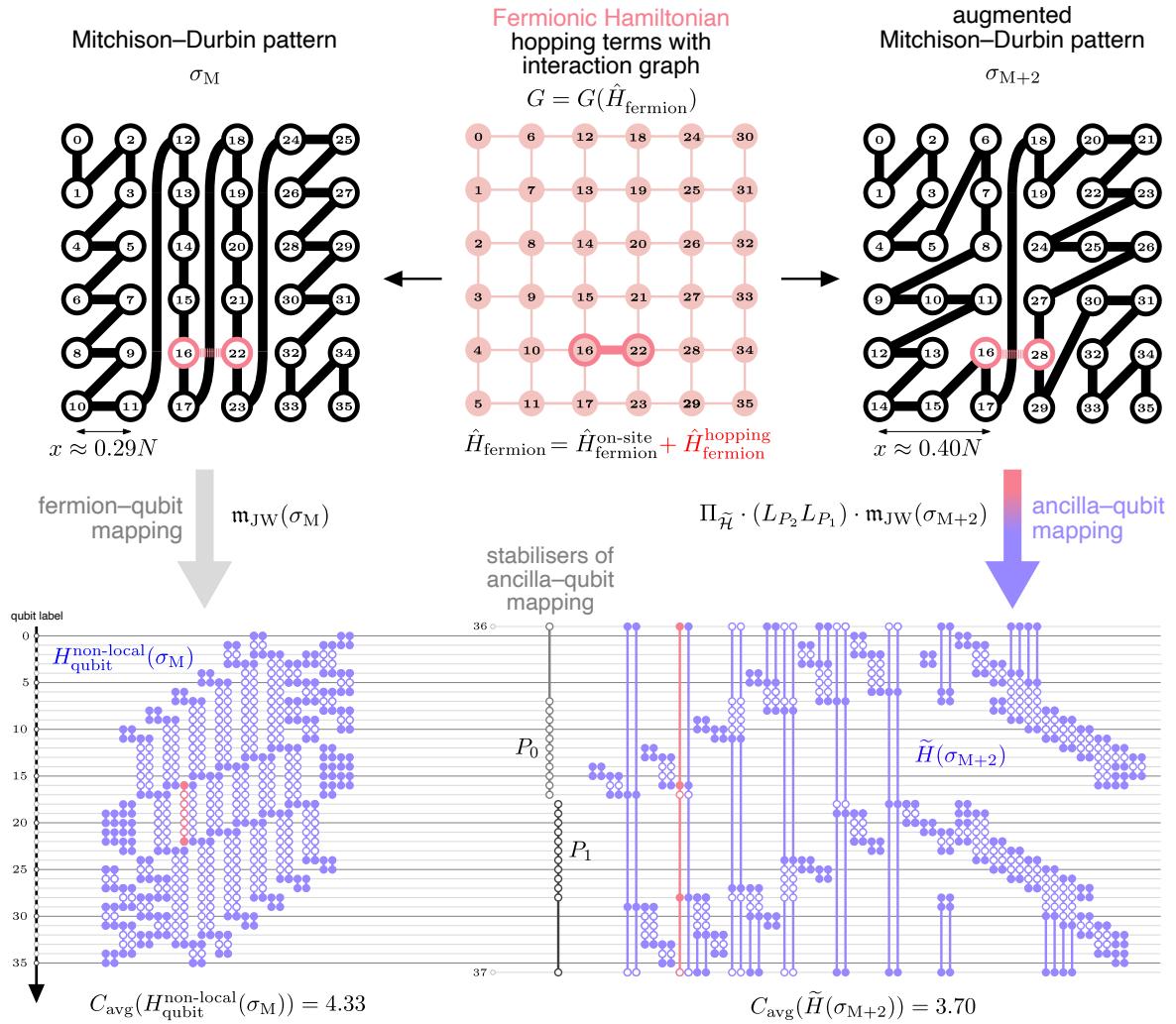
$$= \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \tilde{h}. \quad (5.50)$$

## 5.3 Beyond optimality with only two ancilla qubits

A feature that all ancilla–qubit mappings in the literature share is the use of a number of ancilla qubits  $(m-n) = \mathcal{O}(n)$ , which scales with the system size. The method in Section 5.2 allows a different approach, however. In this section, we devise patchwork solutions to the limitations of the Jordan–Wigner transformation in Chapter 4, producing new qubit–efficient mappings by applying a hard cut-off of  $(m-n) = \mathcal{O}(1)$  ancilla qubits. For fermionic systems with hopping terms forming square lattice interactions, we demonstrate how just two ancilla qubits can allow for a further reduction of the average Pauli weight of qubit Hamiltonians via the Jordan–Wigner transformation to 37.9% below that of the S–pattern, in comparison to the optimal ancilla–free advantage of 13.9% from the Mitchison–Durbin pattern in Chapter 4. Figure 5.2 displays our two–ancilla–qubit mapping alongside the optimal mapping from Chapter 4, demonstrating a qubit Hamiltonian with ‘beyond–optimal’ average Pauli weight.

### 5.3.1 Ancilla–qubit Jordan–Wigner transformations

Under the  $n$ -mode Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}}(\sigma)$  with fermionic labelling  $\sigma \in S_n$ , the hopping terms  $\hat{\gamma}_{2i}\hat{\gamma}_{2j}$ ,  $\hat{\gamma}_{2i+1}\hat{\gamma}_{2j+1}$ ,  $\hat{\gamma}_{2i}\hat{\gamma}_{2j+1}$  and  $\hat{\gamma}_{2i+1}\hat{\gamma}_{2j}$  of the fermionic Hamiltonian in Equation 4.12 map to sums of Pauli operators made non-local by the Pauli operators  $\bigotimes_{k=\sigma(i)+1}^{\sigma(j)-1} Z_k$ , as Equation 4.26 and Figure 4.2 demonstrate. Long



**Figure 5.2** Comparison between Jordan–Wigner representations of the hopping terms in a  $6 \times 6$  fermionic square lattice. The 36-mode Jordan–Wigner transformation that uses the Mitchison–Durbin fermionic labelling  $m_{\text{JW}}(\sigma_M)$  produces the Hamiltonian  $H_{\text{qubit}}^{\text{non-local}}(\sigma_M)$  with optimal average Pauli weight  $C_{\text{avg}} = 4.33$  for all 36-qubit Jordan–Wigner transformations. The ancilla–qubit mapping  $\Pi_{\tilde{H}} \cdot (L_{P_1} L_{P_2}) \cdot m_{\text{JW}}(\sigma_{M+2})$  makes judicious use of stabilisers  $P_1$  and  $P_2$  to simplify weighty terms, producing the Hamiltonian  $\tilde{H}(\sigma_{M+2})$  with average Pauli weight  $C_{\text{avg}} = 3.70$ .

strings of Pauli– $Z$  operators may be common to many terms in the qubit Hamiltonian, and lead to a natural application of the ancilla–qubit mapping protocol of Section 5.2. Each successive ancilla qubit allows cancellation of long strings of  $Z$  operators from terms with high Pauli weight, at the occasional expense of corrective Paulis on the ancilla qubits to preserve the crucial unitary property of the ancilla–qubit mapping.

Our procedure for extending a Jordan–Wigner transformation with  $r$  ancilla qubits to reduce the Pauli weight of a target Hamiltonian  $H_{\text{qubit}}^{\text{non-loc}}(\sigma) = \mathbf{m}_{\text{JW}}(\sigma) \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathbf{m}_{\text{JW}}(\sigma))^{\dagger}$ , for some fermionic labelling  $\sigma \in S_n$ , proceeds as follows.

Choose a set of *stabilisers*  $\{P_k = Z_{S_k} \mid S_k \subseteq [n]\}_{k=0}^{r-1}$ . The most effective choice of stabilisers is one that will cancel the most  $Z$ –gates of the terms in  $H_{\text{qubit}}^{\text{non-loc}}(\sigma)$ , using as few ancilla qubits  $r$  as possible. By definition, all of the stabilisers  $Z_{S_k}$  commute with each other, satisfying condition 4 of isometry.

As it stands, the unmodified  $n$ –qubit Hamiltonian is

$$H_{\text{qubit}}^{\text{non-loc}}(\sigma) = \mathbf{m}_{\text{JW}}(\sigma) \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathbf{m}_{\text{JW}})^{\dagger} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h \quad (5.51)$$

$$= \frac{1}{2} \sum_{(i,j) \in G} \left( \begin{array}{l} \text{Re}(c_{ij})(X_{\sigma(i)}X_{\sigma(j)} + Y_{\sigma(i)}Y_{\sigma(j)}) \\ + \text{Im}(c_{ij})(X_{\sigma(i)}Y_{\sigma(j)} + Y_{\sigma(i)}X_{\sigma(j)}) \end{array} \right) \left( \bigotimes_{k=\sigma(i)+1}^{\sigma(j)-1} Z_k \right) \quad (5.52)$$

$$= \frac{1}{2} \sum_{\substack{(i,j) \in G \\ A,B \in \{X,Y\}}} \alpha_{ij} A_{\sigma(i)} B_{\sigma(j)} Z_{[\sigma(i),\sigma(j)]}, \quad (5.53)$$

where Equation 5.53 introduces shorthand notation for ease of use: the coefficients  $\alpha_{ij}$  are either  $\text{Re}(c_{ij})$  or  $\text{Im}(c_{ij})$  and the subscript in the expression  $Z_{[\sigma(i),\sigma(j)]}$  indicates the action of  $Z$  operators on all qubits with labels in the set  $\{\min\{\sigma(i),\sigma(j)\} + 1, \dots, \max\{\sigma(i),\sigma(j)\} - 1\}$ . The Pauli weight of a qubit Hamiltonian term  $h$  is

$$W(h) = W(A_{\sigma(i)} B_{\sigma(j)} Z_{[\sigma(i),\sigma(j)]}) = |\sigma(i) - \sigma(j)| + 1. \quad (5.54)$$

Apply the map  $L_{P_0}$  to each term of the Hamiltonian, producing:

$$L_{P_0} : h \longmapsto h' , \quad \text{where } h = A_{\sigma(i)} B_{\sigma(j)} Z_{[\sigma(i)+1, \sigma(j)-1]} \quad \text{and} \quad (5.55)$$

$$h' = \begin{cases} \overbrace{A'_{\sigma(i)} B'_{\sigma(j)} Z_{[\sigma(i)+1, \sigma(j)-1]}}^{hP_0} \triangle (\mathcal{S}_1 \setminus \{\sigma(i), \sigma(j)\}) \otimes \overbrace{\begin{cases} Z_{N^2} & [h, P_0] = 0 \\ Y_{N^2} & \{h, P_0\} = 0 \end{cases}}^{\kappa_h} \\ \qquad \qquad \qquad \text{if } W(hP_0) < W(h) , \\ \overbrace{A_{\sigma(i)} B_{\sigma(j)} Z_{[\sigma(i)+1, \sigma(j)-1]}}^h \otimes \overbrace{\begin{cases} \mathbb{1}_{N^2} & [h, P] = 0 \\ X_{N^2} & \{h, P\} = 0 \end{cases}}^{\kappa_h} \\ \qquad \qquad \qquad \text{if } W(h) \leq W(hP_0) , \end{cases} \quad (5.56)$$

with the placeholder labels  $A', B'$  denoting operators in  $\{\pm X, \pm Y\} \cup \{\pm iX, \pm iY\}$  and where the symbol  $\triangle$  denotes the symmetric difference. The weight of the new term is

$$W(h') = \min\{W(hP_0) + 1, W(h) + W(\kappa_h)\} \quad (5.57)$$

$$= \min \left\{ \begin{array}{l} W(\overbrace{A'_{\sigma(i)} B'_{\sigma(j)} Z_{[\sigma(i)+1, \sigma(j)-1]}}^{hP_0} \triangle (\mathcal{S}_1 \setminus \{\sigma(i), \sigma(j)\})) + 1, \\ W(\underbrace{A_{\sigma(i)} B_{\sigma(j)} Z_{[\sigma(i)+1, \sigma(j)-1]}}_h) + (|\{\sigma(i), \sigma(j)\} \cap \mathcal{S}_1| \mod 2) \end{array} \right\} \quad (5.58)$$

$$= \min \left\{ \begin{array}{l} |\mathcal{S}_1 \setminus \{\sigma(i), \sigma(j)\} \triangle [\sigma(i) + 1, \sigma(j) - 1]| + 3, \\ \underbrace{|\sigma(i) - \sigma(j)| + 1}_W(h) + \underbrace{(|\{\sigma(i), \sigma(j)\} \cap \mathcal{S}_1| \mod 2)}_{0 \text{ or } 1} \end{array} \right\} \quad (5.59)$$

$$\leq W(h) . \quad (5.60)$$

Following in the footsteps of Section 5.2.3, let  $H_{\text{qubit}}^{(j+1)}$  be the result of applying  $L_{P_j}$  to the  $(n+j)$ -qubit Hamiltonian  $(H_{\text{qubit}}^{\text{non-loc}})^{(j)}(\sigma)$ , appending a  $j$ th ancilla qubit. After applying  $r$  such modifications  $L_{P_0}, L_{P_1}, \dots, L_{P_{r-1}}$  and restricting the qubit register to the  $(2^{n-r})$ -dimensional space  $\widetilde{\mathcal{H}}$  of simultaneous stabiliser states of the operators  $\{P_i \otimes Z_{n+i}\}_{i=0}^{r-1}$ , the result is a modified  $(n+r)$ -qubit Hamiltonian  $\widetilde{H}(\sigma) = (H_{\text{qubit}}^{\text{non-loc}})^{(r)}(\sigma)|_{\widetilde{\mathcal{H}}}$  with terms that can be significantly lower in Pauli weight than in the original  $n$ -qubit Hamiltonian  $H_{\text{qubit}}^{\text{non-loc}}(\sigma)$ . The ancilla-qubit mapping itself is the

unitary map

$$\left( \Pi_{\tilde{\mathcal{H}}} \cdot \prod_{i=1}^r L_{P_i} \cdot \mathfrak{m}_{\text{JW}}(\sigma) \right) : \mathcal{H}_f^n \longrightarrow \tilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes(n+2)}, \quad (5.61)$$

which we call an *r*-ancilla-qubit Jordan–Wigner transformation.

Describing the application of the remaining maps  $L_{P_2}, L_{P_3}, \dots, L_{P_r}$  in the style of Equation 5.55 is cumbersome without a tangible example at hand: the next section returns to the square-lattice fermionic systems of Chapter 4 to improve upon the results.

### 5.3.2 Beyond the ancilla-free optimality of Jordan–Wigner transformations for square-lattice fermionic systems

In this section, we derive a two-ancilla-qubit Jordan–Wigner transformation which, upon application to fermionic systems with square-lattice interactions, yields qubit Hamiltonians with average Pauli weights up to 27.9% less than the ancilla-free optimum from Section 4.2.4, and up to 37.9% less than the standard ancilla-free choices of the  $S$ - and  $Z$ -patterns. Figure 4.8 shows the average Pauli weights of the two-ancilla qubit Hamiltonian  $\tilde{H}(\sigma_{M+2})$  and related Hamiltonians  $\tilde{H}(\rho')$  that serve as intermediate steps on the path to this result.

Let  $\hat{H}_{\text{fermion}}$  be an  $N^2$ -mode fermionic Hamiltonian with interaction graph  $G = G(\hat{H}_{\text{fermion}})$  equal to the  $N \times N$  square lattice. As the quartic terms become local under any Jordan–Wigner transformation, consider only the hopping terms,

$$\hat{H}_{\text{fermion}}^{\text{hopping}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}, \quad (5.62)$$

where  $\Lambda$  contains the terms in the Majorana product decomposition of the fermionic Hamiltonian.

What follows is a description of the ancilla-qubit mapping

$$\Pi_{\tilde{\mathcal{H}}} \cdot (L_{P_1} L_{P_0}) \cdot \mathfrak{m}_{\text{JW}}(\rho') : \mathcal{H}_f^{N^2} \longrightarrow \tilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes(n+2)}, \quad (5.63)$$

that produces a qubit Hamiltonian with average Pauli weight below the ancilla-free minimum of Section 4.2.4.

- 1. Applying  $\mathfrak{m}_{\text{JW}}(\rho')$ .** Apply the  $N^2$ -qubit Jordan–Wigner transformation  $\mathfrak{m}_{\text{JW}}(\rho')$ , where  $\rho'$  is the Mitchison–Durbin pattern with a variable value for  $x$ , the side-

length of the distinctive square in the corners of the pattern, as in Figure 4.16:

$$H_{\text{qubit}}^{\text{non-loc}}(\dot{\rho}') = \mathbf{m}_{\text{JW}}(\dot{\rho}') \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathbf{m}_{\text{JW}}(\dot{\rho}'))^\dagger = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h. \quad (5.64)$$

In the default Mitchison–Durbin pattern  $\sigma_M$ , the value is  $x \approx 0.29N$ ; Section 4.3 only proves that this pattern minimises the average Pauli weight of non-local qubit Hamiltonians under Jordan–Wigner transformations with *no* ancilla qubits. Since we are permitting ancillas, treat  $x \in \{1, 2, \dots, \lfloor N/2 \rfloor\}$  as a variable, its optimal value currently unknown.

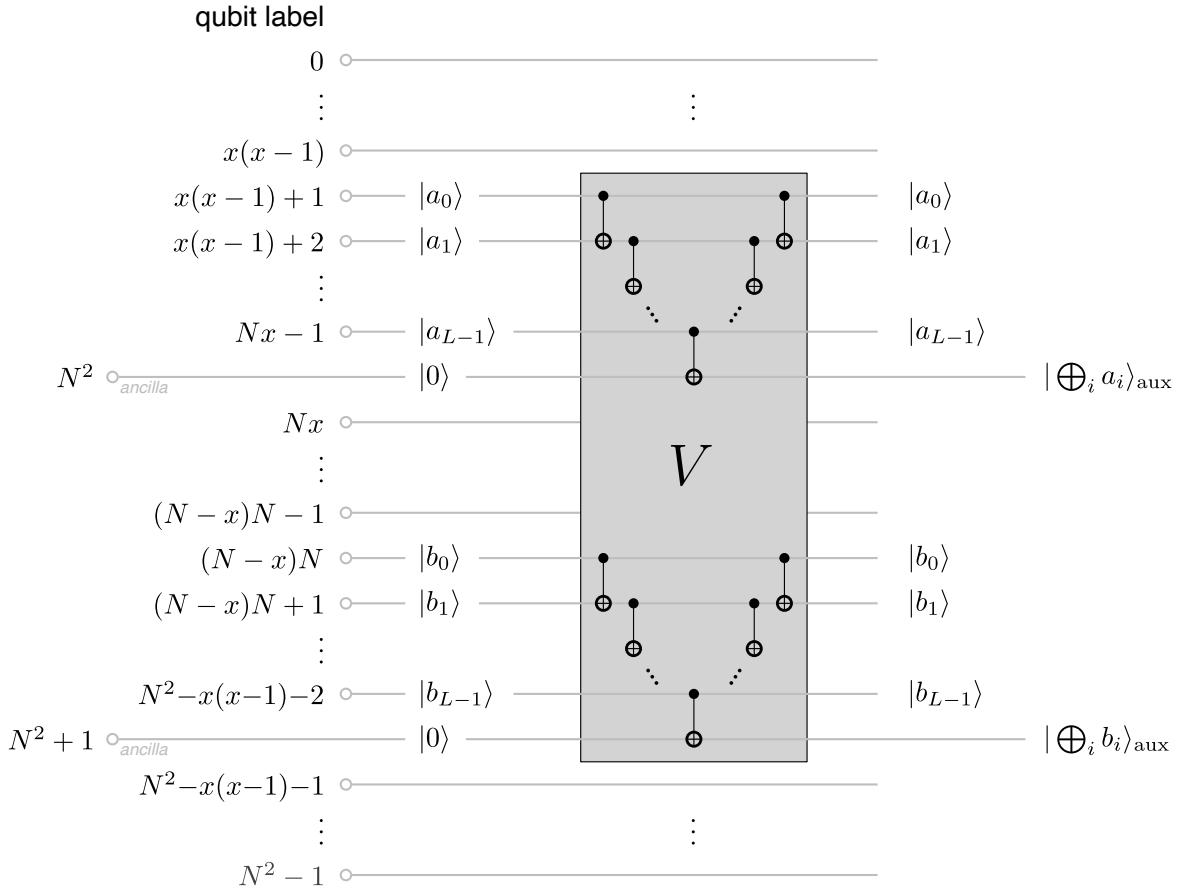
Inspection of the fermionic labelling scheme  $\dot{\rho}'$  reveals the ideal choice of Pauli operators  $P_0, P_1$  for a two–ancilla–qubit mapping. The edgesum  $C_G^1(\dot{\rho}')$ , with formula in Equation 4.81, has two costly contributions: the  $\mathcal{O}(x^2N)$ –valued difference between adjacent vertices in the  $AD$ – and  $DG$ –interfaces in Figure 4.16. We will construct a two–ancilla–qubit mapping to target long strings of  $Z$  operators arising from these contributions of these edges to  $C_G^1(\dot{\rho}')$ , hence reducing the average Pauli weight of the modified qubit Hamiltonian below the ancilla–free optimal of  $C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_M))$ .

- 2. Applying  $L_{P_1}L_{P_0}$ .** Introduce two ancilla qubits and apply the maps  $L_{P_0}$  and  $L_{P_1}$ . In this case, we choose

$$P_0 = \left( \bigotimes_{i=x(x-1)+1}^{xN-1} Z_i \right), \quad P_1 = \left( \bigotimes_{i=(N-x)N}^{N^2-x^2+x-2} Z_i \right). \quad (5.65)$$

The intervals  $[x(x-1)+1, xN-1]$  and  $[(N-x)N, N^2-x^2+x-2]$  bridge the gaps between the outermost vertices on the  $AD$ – and  $DG$ –interfaces, respectively, which is clear from Figure 4.17. Figure 5.4 illustrates examples of the two Pauli operators for  $N = 17$  and differing values of  $x$ , and how the maps  $L_{P_0}$  and  $L_{P_1}$  modify the qubit Hamiltonian terms.

- 3. Applying  $\Pi_{\widetilde{\mathcal{H}}}$ .** Let  $\widetilde{\mathcal{H}}$  be the space of simultaneous stabiliser states of  $P_0 \otimes Z_{N^2} \otimes \mathbb{1}_{N^2+1}$  and  $P_1 \otimes \mathbb{1}_{N^2} \otimes Z_{N^2+1}$ . To implement  $\Pi_{\mathcal{H}}$ , it is sufficient to prepare the  $(N^2 + 2)$ –qubit register in a state of  $\widetilde{\mathcal{H}}$ . Define the unitary mapping  $V$  to be a cascade of controlled–NOT operations that stores the net parity of the qubits appearing in the support of  $P_i$  in the phase of the  $i$ th ancilla qubit, i.e. the qubit with label  $(n+i)$ . Figure 5.3 shows the circuit for  $V$ .



**Figure 5.3** The circuit for  $V$ , which maps any  $(N^2)$ -qubit state  $|\psi\rangle$  to an  $(N^2 + 2)$ -qubit simultaneous stabiliser state of  $P_0$  and  $P_1$ , with both ancilla qubits starting in state  $|0\rangle$ .

Applying  $V$  to any state  $|\psi\rangle \in \mathcal{H}_2^{\otimes N^2}$  of the original  $N^2$ -qubit system produces a state in  $\widetilde{\mathcal{H}}$ ,

$$|\tilde{\psi}\rangle = V(|\psi\rangle \otimes |0\rangle^{\otimes 2}) \implies (P_i \otimes Z_{n+i}) |\tilde{\psi}\rangle = |\tilde{\psi}\rangle \quad \text{for } i \in \{0, 1\}. \quad (5.66)$$

In practice, to implement  $\Pi_{\widetilde{\mathcal{H}}}$  is to apply  $V$  to the  $(N^2)$ -qubit register with both ancilla qubits in the state  $|0\rangle$ .

The two-ancilla-qubit mapping in Equation 5.63 produces the Hamiltonian

$$\widetilde{H}(\rho') = (\Pi_{\widetilde{\mathcal{H}}} \cdot L_{P_1} L_{P_2} \cdot \mathfrak{m}(\rho')) \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\Pi_{\widetilde{\mathcal{H}}} \cdot L_{P_1} L_{P_2} \cdot \mathfrak{m}(\rho'))^\dagger \quad (5.67)$$

$$= \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \tilde{h}, \quad (5.68)$$

which has terms  $\tilde{h}$  with Pauli weight less than the terms  $h$  of  $H_{\text{qubit}}^{\text{non-loc}}(\rho')$ . In showing how the choice of  $x$  in the fermionic labelling  $\rho'$  affects the reduction in average Pauli weight, Theorem 4 find a new optimal value for  $x$  and quantifies the ability of constant–ancilla–qubit mappings to go beyond the optimality of the Mitchison–Durbin pattern.

**Theorem 4.** (*A fermionic labelling that goes beyond ancilla-free optimality.*) Let  $\sigma_{M+2}$  be the fermionic labelling that results from choosing  $x$  to be the value that minimises the average Pauli weight of the qubit Hamiltonian  $\tilde{H}(\rho')$ . Then, the optimal value is  $x \approx 0.40N$ , and the resulting average Pauli weight is

$$C_{\text{avg}}(\tilde{H}(\sigma_{M+2})) \approx 0.31N + 2.68. \quad (5.69)$$

Comparing to the results from Chapter 4, the result compares to the  $S$ –pattern  $\sigma_S$  and Mitchison–Durbin pattern  $\sigma_M$  favourably, yielding the following reductions in average Pauli weight:

$$1 - \frac{C_{\text{avg}}(\tilde{H}(\sigma_{M+2}))}{C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_S))} \approx 37.9\% \quad (5.70)$$

$$1 - \frac{C_{\text{avg}}(\tilde{H}(\sigma_{M+2}))}{C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_M))} \approx 27.9\%. \quad (5.71)$$

*Proof.* Throughout this proof, we refer to fermionic Hamiltonian terms with the symbol  $\hat{h}$ . The image of these terms under  $\mathbf{m}_{\text{JW}}(\rho')$  are the terms of the unmodified qubit Hamiltonian, which we denote by  $h$ . After applying  $L_{P_0}$ ,  $L_{P_1}$  and the projection  $\Pi_{\tilde{\mathcal{H}}}$  to the qubit Hamiltonian, the result is the modified qubit Hamiltonian  $\tilde{H}$  with terms that we denote by  $\tilde{h}$ .

Define the *total* Pauli weight of all terms in the modified Hamiltonian  $\tilde{H}(\rho')$  to be the sum of the Pauli weights of all terms in  $\tilde{H}(\rho')$ . Figure 5.4 highlights the vertices that correspond to the support of the Pauli operators  $P_i$  and the edges of the square lattice that correspond to hopping terms  $\tilde{h}$  for which  $\tilde{h} = hP_i \otimes \kappa_{h(i)}$  or  $\tilde{h} = h \otimes X_{N^2+i}$ . All other hopping terms  $\tilde{h}$  map via  $\mathbf{m}_{\text{JW}}(\rho')$  to terms  $h$  which become  $\tilde{h} = h \otimes \mathbb{1}^{\otimes 2}$  upon modification by  $L_{P_0}$ ,  $L_{P_1}$  and  $\Pi_{\tilde{\mathcal{H}}}$ . Noting that nearly all of the highlighted edges exist at the interfaces of the regions  $A$ – $G$  of the lattice, using the labels in Figure 4.16

and 5.4, one way to count the total Pauli weight of  $\widetilde{H}(\dot{\rho}')$  is

$$\frac{1}{4} \times \text{Total Pauli weight of } \widetilde{H}(\dot{\rho}') \quad (5.72)$$

$$\begin{aligned} &\leq \overbrace{\frac{1}{4} \times \text{Total Pauli weight of } H_{\text{qubit}}^{\text{non-loc}}(\dot{\rho}') \quad (1)} \\ &\quad + 2 \times \overbrace{\frac{1}{4} \left( \text{difference } W(\tilde{h}) - W(h) \text{ in Pauli weights} \right.} \\ &\quad \left. \text{of terms } \hat{h} \text{ in the } AD-, BD-, CD- \text{ and } AB-\text{interfaces} \right) \quad (2) \end{aligned} \quad (5.73)$$

The expression in Equation 5.72 observes that the total Pauli weight arising from terms along the  $DG-$ ,  $DF-$ ,  $DE-$  and  $FG-$ interfaces is equal to that of the  $AD-$ ,  $BD-$ ,  $CD-$  and  $AB-$ interfaces, respectively, due to the symmetry of  $\dot{\rho}'$ . The persistent factors of  $\frac{1}{4}$  account for the fact that each edge of the lattice  $G$  contributes one hopping term  $\hat{h}$  to the fermionic Hamiltonian, but four non-local terms  $h = \alpha_{ij} A_{\dot{\rho}'(i)} B_{\dot{\rho}'(j)} Z_{[\dot{\rho}'(i)+1, \dot{\rho}'(j)+1]}$  of equal weight in the qubit Hamiltonian under a Jordan–Wigner transformation: one for each combination of  $A, B \in \{X, Y\}$ .

The expression in Equation 5.72 is also an upper bound for the total Pauli weight, because if  $x$  is large enough then it is possible that a small number of internal edges in the regions  $C$  and  $E$  may also correspond to non-local terms that reduce in Pauli weight with the application of  $L_{P_0}$  and  $L_{P_1}$ , as the leftmost diagram in Figure 5.4 demonstrates. Although this reduces the average Pauli weight of the qubit Hamiltonian even further, its effect is negligible compared to the savings across the interfaces for which Equation 5.72 does account. We omit this detail for simplicity.

For term (1) in Equation 5.72, there are four hopping terms for each of the  $2N(N-1)$  edges in the  $N \times N$  square lattice graph  $G$ . Using Equation 4.31 that the total Pauli weight of the square lattice qubit Hamiltonian  $H_{\text{qubit}}^{\text{non-loc}}(\dot{\rho}')$  is  $4(C_G^1(\dot{\rho}') + 2N(N-1))$ , and hence (1) is  $C_G^1(\dot{\rho}') + 2N(N-1)$ .

For term (2) in Equation 5.72, the calculation is more involved.

Consider the Hamiltonian terms of  $\widetilde{H}(\dot{\rho}')$  arising from fermionic modes in the  $AD$ -interface. Denote by  $\hat{h}_{(\dot{\rho}'(a), \dot{\rho}'(b))}^{AD}$  the hopping term between fermionic modes with labels  $\dot{\rho}'(a) \in A$  and  $\dot{\rho}'(b) \in D$ , where  $a, b \in [N^2]$  and with the superscript ‘ $AD$ ’ present to emphasise the lattice region. From inspection of Figure 4.17, the hopping terms across the  $j$ th row of the  $AD$ -interface have the form  $\hat{h}_{(x(x-1)+(j-1), Nx+(j-1))}^{AD}$ , for  $j = 1, 2, \dots, x$ . The first row maps to terms  $\tilde{h}_{(x(x-1), Nx)} = A'_{x(x-1)} B'_{Nx} \otimes Z_{N^2}$  in

qubit Hamiltonian  $H_{\text{qubit}}^{\text{non-loc}}(\dot{\rho}') = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} h$  with terms  $h = \left( \bigotimes_{k=\sigma(i)+1}^{\sigma(j)-1} Z_k \right) A_{\sigma(i)} X_{\sigma(j)}$

$\dot{\rho}'$  (small  $x$ )

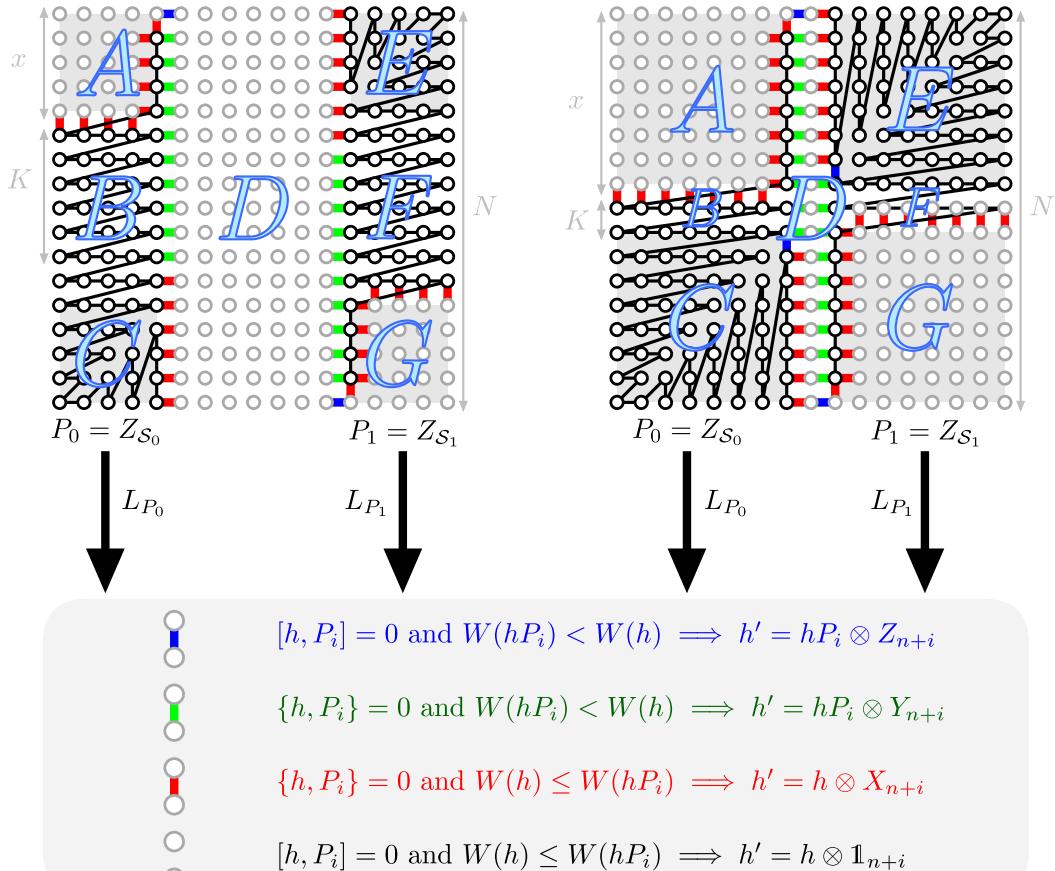
$$x \leq \left\lfloor \frac{N+1}{3} \right\rfloor \implies K = \left\lfloor \frac{Nx - x^2}{2x - 1} \right\rfloor$$

e.g.  $N = 17, x = 5, K = 6$

$\dot{\rho}'$  (large  $x$ )

$$x > \left\lfloor \frac{N+1}{3} \right\rfloor \implies K = N - 2x + 1$$

e.g.  $N = 17, x = 8, K = 2$



$$\text{adjusted qubit Hamiltonian } \tilde{H}(\dot{\rho}') = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \tilde{h}$$

**Figure 5.4** Pauli operators  $P_0, P_1$  and instructions for applying the maps  $L_{P_0}$  and  $L_{P_1}$  as part of the two-ancilla-qubit Jordan-Wigner transformation with fermionic labelling  $\dot{\rho}' \in S_{N^2}$ . The result is a qubit Hamiltonian with significantly reduced average Pauli weight which is equivalent to the initial qubit Hamiltonian within the space of stabiliser states of  $P_0 \otimes Z_n$  and  $P_1 \otimes Z_{n+1}$ . Setting  $x \approx 0.40N$  in the fermionic labelling  $\dot{\rho}'$  yields the minimum average Pauli weight.

the modified qubit Hamiltonian, which consist of the two matrices  $A', B' \in \{X, Y\}$  and a single  $Z$  matrix on the ancilla qubit. For all other  $j = 2, 3, \dots, x$ , the terms  $\tilde{h}_{(x(x-1)+(j-1), Nx+(j-1))}^{AD}$  consist of  $A'_{x(x-1)+(j-1)}, B'_{Nx+(j-1)} \in \{X, Y\}$ , another  $(2j-3)$  Pauli  $Z$  matrices and a  $Y$  matrix on the ancilla qubit. Thus, the contribution to the total Pauli weight of  $\tilde{H}(\dot{\rho}')$  from the  $AD$ -interface is

$$\frac{1}{4} \times \widetilde{AD} := \sum_{j=1}^x W(\tilde{h}_{(x(x-1)+j, Nx+j)}^{AD}) = 3 + \sum_{j=2}^x 2j = 1 + x + x^2. \quad (5.74)$$

Denote hopping terms across the  $BD$ -interface and the first row of the  $CD$ -interface by the expression  $\hat{h}_{\dot{\rho}'(a), \dot{\rho}'(b)}^{BD}$ , where  $\dot{\rho}'(a) \in \{B\}$  or the first row of  $C$ , and  $\dot{\rho}'(b) \in D$ . The hopping term corresponding to the  $j$ th row has the form  $\hat{h}_{(x^2+jx-1, Nx+x+j-1)}^{BD}$ , for  $j = 1, 2, \dots, N - 2x + 1$ . Figure 5.4 depicts how the first  $K$  rows in the  $BD$ -interface correspond to terms of the form  $\tilde{h}_{\dot{\rho}'(a), \dot{\rho}'(b)}^{BD} = h_{\dot{\rho}'(a), \dot{\rho}'(b)}^{BD} P_0 \otimes Y_{N^2}$  in the modified qubit Hamiltonian, for some integer  $K \in \{0, 1, \dots, N - 2x + 1\}$ , and how the first row of the  $CD$ -interface may also map to terms of this form. Using the labels in Figure 4.17, these terms consist of  $(2+j)x + j - 3$  Pauli  $Z$  gates, the operators  $A'_{x^2+jx-1}$  and  $B'_{Nx+j-1}$  and a  $Y$  matrix on the ancilla. The remaining hopping terms across the  $BD$ -interface and the first row of the  $CD$ -interface, for  $j \in \{K+1, \dots, N - 2x + 1\}$ , map to terms of the form  $\tilde{h}_{(x^2+jx-1, Nx+x+j-1)}^{BD} = h_{(x^2+jx-1, Nx+x+j-1)}^{BD} \otimes X_{N^2}$ . The contribution of these terms to the total Pauli weight of  $\tilde{H}(\dot{\rho}')$  is thus

$$\frac{1}{4} \times \widetilde{BD} := \sum_{j=1}^{N-2x+1} W(\tilde{h}_{(x^2+jx-1, Nx+x+j-1)}^{BD}) \quad (5.75)$$

$$= \sum_{j=1}^K ((2+j)x + j) \quad (5.76)$$

$$\begin{aligned} &+ \sum_{j=K+1}^{N-2x+1} \left( (Nx + x + j - 1) - (x^2 + jx - 1) + 2 \right) \\ &= \frac{1}{2}(K + K^2 + 5Kx + K^2x) \end{aligned} \quad (5.77)$$

$$+ \frac{1}{2}(K + 2x - N - 1)(2x + Kx - Nx - K - N - 6).$$

The value  $K$  depends on the value of  $x$ , as Figure 5.4 illustrates. From a naïve inspection of Equation 5.76, the  $j$ th row of the  $BD$ -interface hopping terms correspond to unmodified Hamiltonian terms  $h$  with  $W(hP_0) \leq W(h)$ , such that  $\tilde{h} = hP_0 \otimes Y_{N^2}$ ,

as long as

$$2(x-1) + (j-1)x + 3 < (Nx + x + j - 1) - (x^2 + jx - 1) + 1 \quad (5.78)$$

$$j < \frac{Nx - x^2}{2x - 1}. \quad (5.79)$$

This might tempt us to assume  $K = \lfloor (Nx - x^2)/(2x - 1) \rfloor$ . However, there are  $N - 2x + 1$  rows in the  $BD$ -interface and first row of the  $CD$ -interface, so the expression in Equation 5.76 is only valid if

$$K = \min \left\{ \left\lfloor \frac{Nx - x^2}{2x - 1} \right\rfloor, N - 2x + 1 \right\} = \begin{cases} N - 2x + 1 & x > \left\lfloor \frac{N+1}{3} \right\rfloor \\ \left\lfloor \frac{Nx - x^2}{2x - 1} \right\rfloor & 1 \leq x \leq \left\lfloor \frac{N+1}{3} \right\rfloor, \end{cases} \quad (5.80)$$

omitting some analysis in the derivation of the right-hand-side of Equation 5.80 for brevity. Figure 5.4 shows possible scenarios with  $N = 17$  and  $x$  above or below the threshold value of  $\lfloor (N+1)/3 \rfloor = 6$ .

Denote the terms across all but the first row of the  $CD$ -interface by  $\hat{h}_{(\rho'(a), \rho'(b))}^{CD}$ . The  $j$ th row of the  $CD$ -interface has the form  $\hat{h}_{(Nx+j-x-1, (x+1)N-x+j-1)}^{CD}$  for  $j = 2, 3, \dots, x$ , and corresponds to terms of the form  $\tilde{h}_{(Nx+j-x-1, (x+1)N-x+j-1)}^{CD}$  which are equal to  $h_{(Nx+j-x-1, (x+1)N-x+j-1)}^{CD} \otimes X_{N^2}$ , consisting of  $N(x+1) - Nx - 1$  Pauli  $Z$  matrices, the operators  $A'_{Nx+j-x-1}$  and  $B'_{(x+1)N-x-1}$  and an  $X$  matrix on the ancilla qubit. The contribution to the total Pauli weight of  $\widetilde{H}(\rho')$  from all but the first row of the  $CD$ -interface is

$$\frac{1}{4} \times \widetilde{CD} := \sum_{j=2}^x W(\tilde{h}_{(Nx+j-x-1, (x+1)N-x+j-1)}^{CD}) = \sum_{j=2}^x (N(x+1) - Nx + 2) \quad (5.81)$$

$$= (x-1)(2-Nx+N(1+x)). \quad (5.82)$$

Denote the terms across the  $AB$ -interface by  $\hat{h}_{(\rho'(a), \rho'(b))}^{AB}$ . Note that the  $j$ th edge across the  $AB$ -interface corresponds to the fermionic term  $\hat{h}_{(x(x-2)+j, x^2+j-1)}^{AB}$  for  $j = 1, 2, \dots, x-1$ , which maps to terms of the form  $\tilde{h}_{(x(x-2)+j, x^2+j-1)}^{AB} = h_{(x(x-2)+j, x^2+j-1)}^{AB} \otimes X_{N^2}$  consisting of  $2(x-1)$  Pauli  $Z$  matrices, the operators  $A_{x(x-2)+j}$  and  $B_{x^2+j-1}$ , and an  $X$  matrix on the ancilla qubit. This holds for all but the rightmost edge, which corresponds to terms of the form  $\tilde{h}_{(x^2-1, x^2+x-1)}^{AB} = h_{(x^2-1, x^2+x-1)}^{AB} \otimes \mathbb{1}^{\otimes 2}$  that consist of  $(x-1)$  Pauli  $Z$  matrices along with the operators  $A_{x^2-1}$  and  $B_{x^2+x-1}$ . The contribution

to the total Pauli weight of  $\widetilde{H}(\dot{\rho}')$  from the  $AB$ -interface is thus

$$\frac{1}{4} \times \widetilde{AB} := \sum_{j=1}^{x-1} W(\tilde{h}_{(x(x-2)+j, x^2+j-1)}^{AB}) + W(\tilde{h}_{(x^2-1, x^2+x-1)}^{AB}) \quad (5.83)$$

$$= \sum_{j=1}^{x-1} (2(x-1) + 3) + (x+1) \quad (5.84)$$

$$= 2x^2. \quad (5.85)$$

We can now finally substitute these values into Equation 5.72 and express the total Pauli weight of  $\widetilde{H}(\dot{\rho}')$ .

$$\frac{\text{Total Pauli weight of } \widetilde{H}(\dot{\rho}')}{4} \leq \overbrace{C_G^1(\dot{\rho}') + 2N(N-1)}^{(1)} \quad (5.86)$$

$$+ 2 \times \frac{1}{4} \left( (\widetilde{AD} + \widetilde{BD} + \widetilde{CD} + \widetilde{AB}) - (AD + BD + CD + AB) \right) \underbrace{\phantom{+ 2 \times \frac{1}{4} \left( (\widetilde{AD} + \widetilde{BD} + \widetilde{CD} + \widetilde{AB}) - (AD + BD + CD + AB) \right)}}_{(2)}$$

$$= N^3 - N^2(x-5) + 2K^2x - N(2+x) \quad (5.87)$$

$$+ \frac{4}{3}x(x^2+2) + 2K(x(x-N+2)-5),$$

yielding

$$C_{\text{avg}}(\widetilde{H}(\dot{\rho}')) \leq \frac{\text{Total Pauli weight of } \widetilde{H}(\dot{\rho}')}{2N(N-1)} \quad (5.88)$$

$$= \frac{1}{6N(N-1)} \left( 3N(N(N+5)-2) + 6K^2x + (8-3N(N+1))x + 4x^3 + 6K(x(x-N+2)-2) \right) \quad (5.89)$$

To find the value of  $x$  that minimises  $C_{\text{avg}}(\widetilde{H}(\dot{\rho}'))$ , it is important to bear in mind that there are two regimes: the ‘small  $x$ ’ regime, where  $x \leq \lfloor (N+1)/3 \rfloor$  and  $K = \lfloor (Nx - x^2)/(2x-1) \rfloor$ , and the ‘large  $x$ ’ regime, where  $x > \lfloor (N+1)/3 \rfloor$  and  $K = N - 2x + 1$ . First, as an example, try the value for  $x \approx 0.29N$  from the original Mitchison–Durbin pattern  $\sigma_M$ . The average Pauli weight of the resulting Hamiltonian,

$$C_{\text{avg}}(\widetilde{H}(\sigma_M)) = C_{\text{avg}}(\widetilde{H}(\dot{\rho}'|_{x \approx 0.29N})) \approx 0.33N + 2.89, \quad (5.90)$$

is a demonstrable reduction from the ancilla-free alternative  $C_{\text{avg}}(H_{\text{qubit}}^{\text{non-loc}}(\sigma_M))$ , as Figure 4.8 shows.

However, at the threshold of the small and large  $x$  regimes, the choice  $x = \lfloor (N+1)/3 \rfloor$  actually yields an improvement over the formerly-optimal Mitchison–Durbin pattern  $\sigma_M$ . The average Pauli weight in this instance is

$$C_{\text{avg}}\left(\widetilde{H}\left(\dot{\rho}'\Big|_{x=\lfloor \frac{N+1}{3} \rfloor}\right)\right) \approx 0.32N + 2.78. \quad (5.91)$$

But there is a better value yet for  $x$ : in the large  $x$  regime, an approximately optimal value for  $x$  to minimise the total Pauli weight is

$$x = \frac{1}{24} (21 + 3N + \sqrt{3}\sqrt{15N^2 - 18N - 53}) \approx \frac{1}{24}(3 + 3\sqrt{5})N \approx 0.40N, \quad (5.92)$$

which is the result of treating  $x$  and  $N$  as continuous variables and minimising Equation 5.89 with  $K = N - 2x + 1$ . Define the vertex labelling  $\sigma_{M+2}$  to be  $\dot{\rho}'$  with  $x$  equal to the rounded value of that in Equation 5.92, i.e.  $x \approx 0.40N$ . The resulting average Pauli weight is

$$C_{\text{avg}}(\widetilde{H}(\sigma_{M+2})) = C_{\text{avg}}\left(\widetilde{H}(\dot{\rho}'|_{x \approx 0.40N})\right) \approx 0.31N + 2.68. \quad (5.93)$$

This proves Theorem 4, but the analysis does not include the cost of implementing  $\Pi_{\mathcal{H}}$  via the unitary operation  $V$ . It is possible to implement  $V$  using fewer gates than the circuit in Figure 5.3 at the cost of more ancilla qubits [111]. However, even adding the cost of the circuit in Figure 5.3, which is  $4(L-1)+2 = 4((Nx-1)-(x(x-1)+1)-1)+2 = 4Nx - x^2 + \mathcal{O}(x)$  CNOTs, to Equation 5.86 has negligible effects on the calculations in this section. The value  $x \approx 0.40N$  in Equation 5.92 still minimises the average Pauli weight.  $\square$

## 5.4 Conclusion

While this chapter involved purely analytical results, the intention of this work is to set the stage for developing constant–ancilla mappings with computational techniques. Section 5.3 restricted the application of our technique in three ways: to the Jordan–Wigner transformation  $\mathbf{m}_{\text{JW}}$  and its fermionic relabellings; to fermionic systems with the interaction graph  $G(\hat{H}_{\text{fermion}})$  of a square lattice only; and to using only two ancilla qubits. These restrictions allowed us to devise, effectively by hand, an improvement upon the ancilla-free techniques of Chapter 4. The structure of the incremental ancilla–

qubit mapping technique of Section 5.3.1 is to aid in the formulation of constant–ancilla Jordan–Wigner transformations for quadratic assignment problems, for any fermionic interaction graph and any number of qubits. Computational results that we have obtained thus far have confirmed the results of Theorem 4, which concerns two–ancilla–qubit Jordan–Wigner transformations for fermionic square lattice systems. It is worth making a separate comment that the task of expanding the model to include mappings other than the Jordan–Wigner transformation introduces other difficulties.

### 5.4.1 Constant ancilla mappings that are not Jordan–Wigner transformations

We hypothesise that the constant–ancilla–qubit method is not as straightforward when applied to general mappings that are not the Jordan–Wigner transformation. This is because other mappings represent fermionic hopping terms as Pauli strings  $\hat{\gamma}_{2i}\hat{\gamma}_{2j} \mapsto X_{\mathcal{I}_1}Y_{\mathcal{I}_2}Z_{\mathcal{I}_3}$  for integer subsets  $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3 \subset [n]$  with a less obvious structure. For example, while in the Jordan–Wigner transformation the hopping terms map to qubit operators with non-local strings  $(\otimes_{k=i+1}^{j-1} Z_k)$ , if  $\mathfrak{m}$  is an  $n$ –mode ancilla qubit mapping with  $\mathfrak{m} = U_{\mathfrak{m}} \cdot \mathfrak{m}_{\text{JW}}$ , then by Lemma 2.4.5 these same terms map to  $U_{\mathfrak{m}}(Z_{i+1} \dots Z_{j-1})U_{\mathfrak{m}}^\dagger = Z_{\mathcal{I}}$  for some integer subset  $\mathcal{I} \subset [n]$  which no longer depends on the distance between  $i$  and  $j$ . Hence in the new mapping,  $Z_{\mathcal{I}_3}$  may not even be the non-local part of the hopping term. Terms may also include multiple  $X$ ,  $Y$ ,  $Z$  matrices, as the intervals  $\mathcal{I}_1$ ,  $\mathcal{I}_2$  and  $\mathcal{I}_3$  could overlap.

In applying the method from Section 5.1 with a non–Jordan–Wigner transformation  $\mathfrak{m}$  as the base mapping, we predict that it would be computationally hard to devise an effective set of commuting, Hermitian Pauli strings  $\{P_i\}_{i=1}^r$  that target the non-local components of a qubit Hamiltonian. The reason this was straightforward for the Jordan–Wigner case was because we knew *ab initio* that the problematic parts of the representations of the hopping terms would consist only of  $Z$  matrices. Nevertheless, we see this as an important line of enquiry: Figure 4.2 demonstrates the well–known fact that mappings other than the Jordan–Wigner transformation are capable of producing qubit Hamiltonians with a significantly lower Pauli weight.

# Chapter 6

## Conclusion

This thesis presented a concerted effort to define, classify and optimise fermion–qubit mappings, with each stage building upon the last to provide new results and perspectives. A primary motivating factor of our work has been the need for a definition of fermion–qubit mappings that encapsulates the evolving literature completely. The work took place during a resurgence of interest in fermion–qubit mappings: in particular, multiple works on ternary tree transformations and ancilla–qubit mappings emerged in tandem with our investigations.

Chapter 2 started with the completely general Definition 2.1.7 for fermion–qubit mappings as unitary transformations between fermionic and qubit Hilbert spaces. With this definition in hand, we explored the separate paradigms of Pauli–based mappings and the hierarchy of classical, affine and linear encodings of the Fock basis. Theorem 1 demonstrates an immediate application of our definition by identifying affine encodings as Pauli–based mappings and deriving the formulae for their Majorana representations.

Chapter 3 used the definition and language of fermion–qubit mappings from Chapter 2 to build a classification system for Pauli–based mappings. Theorem 2 used this classification to prove that product–preserving ternary tree transformations are equivalent to linear encodings of the Fock basis. The equivalence classes of our classification separate Pauli–based mappings into templates, stripping away the labelling symmetries that are inconsequential to most of the cost functions in the literature. In identifying the equivalence between two disparate areas of interest in the field – product–preserving ternary tree transformations and linear encodings – we hope to contribute a perspective that could unify the contemporary approaches to customising architecture–aware fermion–qubit mappings.

Chapter 4 put fermion–qubit mappings to use producing qubit representations of fermionic Hamiltonians, using the definition of Chapter 2 and the template classification

in Chapter 3. The precise notation we built throughout the thesis allowed us to identify that optimisation problems for fermion–qubit mappings for practical cost functions, such as the average and maximum Pauli weights of the qubit Hamiltonian, rely only on the fermionic labelling. In the case of the Jordan–Wigner transformation, the optimisation problems translate to  $p$ –sum problems from graph theory. Theorem 3 took advantage of this connection to provide optimal Jordan–Wigner transformations for fermionic interactions in a square lattice pattern, yielding qubit Hamiltonians with average Pauli weights 13.9% below previous attempts. A novel feature of this optimisation is that the improvements come free-of-charge: by simply relabelling the fermionic modes, we can produce qubit Hamiltonians that are more efficient to simulate on quantum technology, without requiring additional qubits or non–Pauli Majorana representations.

Continuing the theme of optimisation, Chapter 5 introduced a new approach to ancilla–qubit mappings to enhance the results of Chapter 4. Beginning by extending Definition 2.1.7 to include ancilla–qubit mappings, we proposed a new approach of incrementally adding ancilla qubits in a bid to reduce the Pauli weight of the terms in a target qubit Hamiltonian. Theorem 4 exemplifies this new perspective with a mapping that yields qubit Hamiltonians with average Pauli weights 37.9% below previous attempts, at the expense of only 2 ancilla qubits regardless of the size of the target fermionic lattice.

In summary, this thesis provides a comprehensive framework for defining, classifying, and optimizing fermion–qubit mappings, addressing key mathematical and notational gaps in the field during a period of renewed interest. By establishing a robust definition, we have been able to relate mappings that arose through different notational paradigms. The classification system and optimisation techniques that we have developed offer new directions for designing cost–effective qubit Hamiltonians in fermionic simulation. As the field continues to evolve, our constant–ancilla mapping technique is ripe for computational optimisation in simulations of fermionic systems beyond the square lattice connectivity. Future research could also devise cost functions that tailor mappings to restricted qubit architectures. Regardless of the direction that research takes, maintaining a precise mathematical definition like the one in this thesis is crucial to ensure the clarity, consistency and reproducibility in the search for the next fermion–qubit mapping.

# References

- [1] M. Chiew, B. Harrison, and S. Strelchuk. Ternary tree transformations are equivalent to linear encodings of the Fock basis. *arXiv eprint arxiv:2412.07578*, Dec 2024.
- [2] B. Harrison, M. Chiew, J. Necaise, A. Projansky, S. Strelchuk, and J. D. Whitfield. A Sierpinski Triangle Fermion-to-Qubit Transform. *arXiv eprint arXiv:2409.04348*, 2024. URL <https://arxiv.org/abs/2409.04348>.
- [3] M. Chiew and S. Strelchuk. Discovering optimal fermion-qubit mappings through algorithmic enumeration. *Quantum*, 7:1145, October 2023. ISSN 2521-327X. doi: 10.22331/q-2023-10-18-1145. URL <https://doi.org/10.22331/q-2023-10-18-1145>.
- [4] W. Pauli. Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren. *Zeitschrift für Physik*, 31(1): 765–783, February 1925. doi: 10.1007/BF02980631.
- [5] J. D. Whitfield, Jacob Biamonte, and A. Aspuru-Guzik. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics*, 109(5): 735–750, 2011. doi: 10.1080/00268976.2011.552441. URL <https://doi.org/10.1080/00268976.2011.552441>.
- [6] A. Messiah. *Quantum Mechanics*. Number v. 2 in Quantum Mechanics. Elsevier Science, 1961. ISBN 978-0-7204-0045-8. URL [https://books.google.com.au/books?id=VR93vUk8d\\_8C](https://books.google.com.au/books?id=VR93vUk8d_8C).
- [7] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.
- [8] S. Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, 1996. doi: 10.1126/science.273.5278.1073. URL <https://www.science.org/doi/abs/10.1126/science.273.5278.1073>.
- [9] E. Schrödinger. Quantisierung als Eigenwertproblem. *Annalen der Physik*, 384(4):361–376, 1926. doi: <https://doi.org/10.1002/andp.19263840404>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19263840404>.
- [10] T. Helgaker, P. Jorgensen, and J. Olsen. *Molecular Electronic-Structure Theory*. Wiley, 2014. ISBN 978-1-119-01955-8. URL <https://books.google.com.au/books?id=lNVLBAAAQBAJ>.

- [11] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White. Towards quantum chemistry on a quantum computer. *Nature Chemistry*, 2(2):106–111, February 2010. ISSN 1755-4349. doi: 10.1038/nchem.483. URL <https://www.nature.com/articles/nchem.483>. Publisher: Nature Publishing Group.
- [12] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A*, 92:062318, Dec 2015. doi: 10.1103/PhysRevA.92.062318. URL <https://link.aps.org/doi/10.1103/PhysRevA.92.062318>.
- [13] P. Kl. Barkoutsos, N. Moll, P. W. J. Staar, P. Mueller, A. Fuhrer, S. Filipp, M. Troyer, and I. Tavernelli. Fermionic Hamiltonians for quantum simulations: a general reduction scheme, 2017. URL <https://arxiv.org/abs/1706.03637>.
- [14] P. M. Chaikin and T. C. Lubensky. *Principles of Condensed Matter Physics*. Cambridge University Press, 1995.
- [15] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, M. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik. Quantum Chemistry in the Age of Quantum Computing. *Chemical Reviews*, 119(19):10856–10915, 2019. doi: 10.1021/acs.chemrev.8b00803. URL <https://doi.org/10.1021/acs.chemrev.8b00803>. PMID: 31469277.
- [16] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan. Quantum Algorithms for Quantum Chemistry and Quantum Materials Science. *Chemical Reviews*, 120(22):12685–12717, 2020. doi: 10.1021/acs.chemrev.9b00829. URL <https://doi.org/10.1021/acs.chemrev.9b00829>. PMID: 33090772.
- [17] N. Friis, A. R. Lee, and D. E. Bruschi. Fermionic-mode entanglement in quantum information. *Phys. Rev. A*, 87:022338, Feb 2013. doi: 10.1103/PhysRevA.87.022338. URL <https://link.aps.org/doi/10.1103/PhysRevA.87.022338>.
- [18] S. Manfred. Renormalization in condensed matter: Fermionic systems – from mathematics to materials. *Nuclear Physics B*, 941:868–899, 2019. ISSN 0550-3213. doi: <https://doi.org/10.1016/j.nuclphysb.2018.07.004>. URL <https://www.sciencedirect.com/science/article/pii/S0550321318301901>.
- [19] C. V. Kraus. *A quantum information perspective of fermionic quantum many-body systems*. PhD thesis, Nov 2009.
- [20] W. Marciano and H. Pagels. Quantum chromodynamics. *Physics Reports*, 36(3):137–276, 1978. ISSN 0370-1573. doi: [https://doi.org/10.1016/0370-1573\(78\)90208-9](https://doi.org/10.1016/0370-1573(78)90208-9). URL <https://www.sciencedirect.com/science/article/pii/0370157378902089>.
- [21] G. M. D’Ariano, F. Manessi, P. Perinotti, and A. Tosini. The Feynman problem and fermionic entanglement: Fermionic theory versus qubit theory. *International Journal of Modern Physics A*, 29(17):1430025, 2014. doi: 10.1142/S0217751X14300257. URL <https://doi.org/10.1142/S0217751X14300257>.

- [22] A. Hamed Moosavian and S. Jordan. Faster quantum algorithm to simulate fermionic quantum field theory. *Phys. Rev. A*, 98:012332, Jul 2018. doi: 10.1103/PhysRevA.98.012332. URL <https://link.aps.org/doi/10.1103/PhysRevA.98.012332>.
- [23] B. Chakraborty, M. Honda, T. Izubuchi, Y. Kikuchi, and A. Tomiya. Digital quantum simulation of the Schwinger model with topological term via adiabatic state preparation. *arXiv preprint arXiv:2001.00485*, 2020.
- [24] M. Born and R. Oppenheimer. Zur Quantentheorie der Moleküle. *Annalen der Physik*, 389(20):457–484, 1927. doi: <https://doi.org/10.1002/andp.19273892002>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19273892002>.
- [25] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, February 2016. ISSN 1367-2630. doi: 10.1088/1367-2630/18/2/023023. URL <https://iopscience.iop.org/article/10.1088/1367-2630/18/2/023023>.
- [26] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush. Improved techniques for preparing eigenstates of fermionic Hamiltonians. *npj Quantum Information*, 4(1):1–7, May 2018. ISSN 2056-6387. doi: 10.1038/s41534-018-0071-5. URL <https://www.nature.com/articles/s41534-018-0071-5>. Publisher: Nature Publishing Group.
- [27] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Phys. Rev.*, 136: B864–B871, Nov 1964. doi: 10.1103/PhysRev.136.B864. URL <https://link.aps.org/doi/10.1103/PhysRev.136.B864>.
- [28] E. S. Kryachko and E. V. Ludeña. Density functional theory: Foundations reviewed. *Physics Reports*, 544(2):123–239, 2014. ISSN 0370-1573. doi: <https://doi.org/10.1016/j.physrep.2014.06.002>. URL <https://www.sciencedirect.com/science/article/pii/S0370157314002075>. Density functional theory: Foundations reviewed.
- [29] J. Hubbard. Electron correlations in narrow energy bands III. An improved solution. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 281(1386):401–419, September 1964. ISSN 0080-4630, 2053-9169. doi: 10.1098/rspa.1964.0190. URL <https://royalsocietypublishing.org/doi/10.1098/rspa.1964.0190>.
- [30] M. C. Gutzwiller. Effect of Correlation on the Ferromagnetism of Transition Metals. *Phys. Rev. Lett.*, 10:159–162, Mar 1963. doi: 10.1103/PhysRevLett.10.159. URL <https://link.aps.org/doi/10.1103/PhysRevLett.10.159>.
- [31] J. Kanamori. Electron Correlation and Ferromagnetism of Transition Metals. *Progress of Theoretical Physics*, 30(3):275–289, 09 1963. ISSN 0033-068X. doi: 10.1143/PTP.30.275. URL <https://doi.org/10.1143/PTP.30.275>.
- [32] H. F. Trotter. On the Product of Semi-Groups of Operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959. ISSN 00029939, 10886826. URL <http://www.jstor.org/stable/2033649>.

- [33] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A*, 146(6):319–323, 1990. ISSN 0375-9601. doi: [https://doi.org/10.1016/0375-9601\(90\)90962-N](https://doi.org/10.1016/0375-9601(90)90962-N). URL <https://www.sciencedirect.com/science/article/pii/037596019090962N>.
- [34] A. Y. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, May 2002. ISBN 978-0-8218-3229-5 978-1-4704-1800-7. doi: 10.1090/gsm/047. URL <http://www.ams.org/gsm/047>.
- [35] J. Kempe, A. Kitaev, and O. Regev. The Complexity of the Local Hamiltonian Problem. *SIAM Journal on Computing*, 35(5):1070–1097, January 2006. ISSN 0097-5397. doi: 10.1137/S0097539704445226. URL <https://pubs.siam.org/doi/10.1137/S0097539704445226>. Publisher: Society for Industrial and Applied Mathematics.
- [36] J. D. Watson and T. S. Cubitt. Computational Complexity of the Ground State Energy Density Problem. *arXiv eprint arXiv:2107.05060*, 2021. URL <https://arxiv.org/abs/2107.05060>.
- [37] C. Moler and C. Van Loan. Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. *SIAM Review*, 45(1):3–49, 2003. doi: 10.1137/S00361445024180. URL <https://doi.org/10.1137/S00361445024180>.
- [38] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [39] D. Gottesman. The Heisenberg Representation of Quantum Computers. *arXiv preprint arXiv:quant-ph/9807006*, July 1998. URL <http://arxiv.org/abs/quant-ph/9807006>.
- [40] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, pages 212–219, Philadelphia, Pennsylvania, United States, 1996. ACM Press. ISBN 978-0-89791-785-8. doi: 10.1145/237814.237866. URL <http://portal.acm.org/citation.cfm?doid=237814.237866>.
- [41] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *arXiv preprint arxiv:quant-ph:0005055*, May 2000. doi: 10.1090/comm/305/05215.
- [42] Y. Shi. Both Toffoli and Controlled-NOT need little help to do universal quantum computation. *arXiv eprint arXiv:quant-ph/0205115*, 2002. URL <https://arxiv.org/abs/quant-ph/0205115>.
- [43] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998. doi: 10.1098/rspa.1998.0164. URL <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1998.0164>.

- [44] D. Aharonov. A Simple Proof that Toffoli and Hadamard are Quantum Universal. *arXiv eprint arXiv:quant-ph/0301040*, 2003. URL <https://arxiv.org/abs/quant-ph/0301040>.
- [45] F. Arute, K. Arya, R. Babbush, D. Bacon, J. Bardin, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1666-5. URL <https://www.nature.com/articles/s41586-019-1666-5>. Publisher: Nature Publishing Group.
- [46] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin. A quantum processor based on coherent transport of entangled atom arrays. *Nature*, 604(7906):451–456, April 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04592-6. URL <https://www.nature.com/articles/s41586-022-04592-6>. Publisher: Nature Publishing Group.
- [47] M. De Smet, Y. Matsumoto, A.-M. J. Zwerver, L. Tryputen, S. L. de Snoo, S. V. Amitonov, A. Sammak, N. Samkharadze, Ö. Güll, R. N. M. Wasserman, M. Rimbach-Russ, G. Scappucci, and L. M. K. Vandersypen. High-fidelity single-spin shuttling in silicon. *arXiv eprint arXiv:2406.07267*, 2024. URL <https://arxiv.org/abs/2406.07267>.
- [48] C. D. Hill, E. Peretz, S. J. Hile, M. G. House, M. Fuechsle, S. Rogge, M. Y. Simmons, and L. C. L. Hollenberg. A surface code quantum computer in silicon. *Science Advances*, 1(9):e1500707, 2015. doi: 10.1126/sciadv.1500707. URL <https://www.science.org/doi/abs/10.1126/sciadv.1500707>.
- [49] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, et al. A Race-Track Trapped-Ion Quantum Processor. *Phys. Rev. X*, 13:041052, Dec 2023. doi: 10.1103/PhysRevX.13.041052. URL <https://link.aps.org/doi/10.1103/PhysRevX.13.041052>.
- [50] P. J. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, et al. Scalable Quantum Simulation of Molecular Energies. *Phys. Rev. X*, 6:031007, Jul 2016. doi: 10.1103/PhysRevX.6.031007. URL <https://link.aps.org/doi/10.1103/PhysRevX.6.031007>.
- [51] T. Hensgens, T. Fujita, L. Janssen, Xiao Li, C. J. Van Diepen, C. Reichl, W. Wegscheider, S. Das Sarma, and L. M. K. Vandersypen. Quantum simulation of a Fermi–Hubbard model using a semiconductor quantum dot array. *Nature*, 548(7665):70–73, August 2017. ISSN 1476-4687. doi: 10.1038/nature23022. URL <https://www.nature.com/articles/nature23022>. Publisher: Nature Publishing Group.
- [52] A. Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. *arXiv preprint arXiv:quant-ph/9511026*, 1995.
- [53] A. Peruzzo, J. McClean, P. Shadbolt, M. H. Yung, X. Zhou, P. Love, A. Aspuru-Guzik, and J. O’Brien. A variational eigenvalue solver on a quantum processor. *Nature communications*, 5, 04 2013. doi: 10.1038/ncomms5213.

- [54] J. R. McClean, R. Babbush, P. J. Love, and A. Aspuru-Guzik. Exploiting Locality in Quantum Computation for Quantum Chemistry. *The Journal of Physical Chemistry Letters*, 5(24):4368–4380, 2014. doi: 10.1021/jz501649m. URL <https://doi.org/10.1021/jz501649m>. PMID: 26273989.
- [55] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, March 2007. ISSN 1432-0916. doi: 10.1007/s00220-006-0150-x. URL <https://doi.org/10.1007/s00220-006-0150-x>.
- [56] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, Mar 2015. doi: 10.1103/PhysRevLett.114.090502. URL <https://link.aps.org/doi/10.1103/PhysRevLett.114.090502>.
- [57] E. Farhi, J. Goldstone, and S. Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv eprint arXiv:1411.4028*, 2014. URL <https://arxiv.org/abs/1411.4028>.
- [58] D. S. Abrams and S. Lloyd. Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors. *Physical Review Letters*, 83(24):5162–5165, December 1999. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.83.5162. URL <https://link.aps.org/doi/10.1103/PhysRevLett.83.5162>.
- [59] K. Setia, S. Bravyi, A. Mezzacapo, and J. D. Whitfield. Superfast encodings for fermionic quantum simulation. *Physical Review Research*, 1(3):033033, 2019.
- [60] F. Verstraete and J. I. Cirac. Mapping local hamiltonians of fermions to local hamiltonians of spins. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09012, sep 2005. doi: 10.1088/1742-5468/2005/09/P09012. URL <https://dx.doi.org/10.1088/1742-5468/2005/09/P09012>.
- [61] P. Jordan and E. Wigner. About the Pauli exclusion principle. *Zeitschrift für Physik*, 47(9-10):631–651, September 1928. ISSN 1434-6001, 1434-601X. doi: 10.1007/BF01331938. URL <http://link.springer.com/10.1007/BF01331938>.
- [62] E. Lieb, T. Schultz, and D. Mattis. Two soluble models of an antiferromagnetic chain. *Annals of Physics*, 16(3):407–466, 1961. ISSN 0003-4916. doi: [https://doi.org/10.1016/0003-4916\(61\)90115-4](https://doi.org/10.1016/0003-4916(61)90115-4). URL <https://www.sciencedirect.com/science/article/pii/0003491661901154>.
- [63] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. Quantum algorithms for fermionic simulations. *Physical Review A*, 64(2):022319, 2001.
- [64] S. Bravyi and A. Y. Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.
- [65] B. Harrison, J. Necaise, A. Projansky, and J. D. Whitfield. A Sierpinski Triangle Data Structure for Efficient Array Value Update and Prefix Sum Calculation. *arXiv eprint arXiv:2403.03990*, 2024. URL <https://arxiv.org/abs/2403.03990>.

- [66] J. T. Seeley, M. J. Richard, and P. J. Love. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, December 2012. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.4768229. URL <https://pubs.aip.org/jcp/article/137/22/224109/194931/The-Bravyi-Kitaev-transformation-for-quantum>.
- [67] Q. Wang, Z.-P. Cian, M. Li, I. L. Markov, and Y. Nam. Ever more optimized simulations of fermionic systems on a quantum computer. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2023. doi: 10.1109/DAC56929.2023.10247693.
- [68] Q. Wang, M. Li, C. Monroe, and Y. Nam. Resource-Optimized Fermionic Local-Hamiltonian Simulation on Quantum Computer for Quantum Chemistry. *Quantum*, 5:509, July 2021. ISSN 2521-327X. doi: 10.22331/q-2021-07-26-509. URL <http://arxiv.org/abs/2004.04151>.
- [69] M. Steudtner and S. Wehner. Fermion-to-qubit mappings with varying resource requirements for quantum simulation. *New Journal of Physics*, 20(6):063010, jun 2018. doi: 10.1088/1367-2630/aac54f. URL <https://dx.doi.org/10.1088/1367-2630/aac54f>.
- [70] M. Steudtner and S. Wehner. Quantum codes for quantum simulation of fermions on a square lattice of qubits. *Phys. Rev. A*, 99:022308, Feb 2019. doi: 10.1103/PhysRevA.99.022308. URL <https://link.aps.org/doi/10.1103/PhysRevA.99.022308>.
- [71] A. Y. Vlasov. Clifford Algebras, Spin Groups and Qubit Trees. *Quanta*, 11(1): 97–114, December 2022. ISSN 1314-7374. doi: 10.12743/quanta.v11i1.199. URL <http://dx.doi.org/10.12743/quanta.v11i1.199>.
- [72] Z. Jiang, A. Kalev, W. Mruczkiewicz, and H. Neven. Optimal fermion-to-qubit mapping via ternary trees with applications to reduced quantum states learning. *Quantum*, 4:276, June 2020. ISSN 2521-327X. doi: 10.22331/q-2020-06-04-276. URL <https://quantum-journal.org/papers/q-2020-06-04-276/>.
- [73] A. Miller, Z. Zimborás, S. Knecht, S. Maniscalco, and G. García-Pérez. Bonsai Algorithm: Grow Your Own Fermion-to-Qubit Mappings. *PRX Quantum*, 4: 030314, Aug 2023. doi: 10.1103/PRXQuantum.4.030314. URL <https://link.aps.org/doi/10.1103/PRXQuantum.4.030314>.
- [74] A. Miller, A. Glos, and Z. Zimborás. Treespilation: Architecture- and State-Optimised Fermion-to-Qubit Mappings. *arXiv eprint arXiv:2403.03992*, 2024.
- [75] T. Parella-Dilmé, K. Kottmann, L. Zambrano, L. Mortimer, J. S. Kottmann, and A. Acín. Reducing Entanglement with Physically Inspired Fermion-To-Qubit Mappings. *PRX Quantum*, 5:030333, Aug 2024. doi: 10.1103/PRXQuantum.5.030333. URL <https://link.aps.org/doi/10.1103/PRXQuantum.5.030333>.
- [76] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. G. Fowler, A. Megrant, E. Jeffrey, et al. Digital quantum simulation of fermionic models with a superconducting circuit. *Nature Communications*, 6(1):7654, July 2015. ISSN 2041-1723.

- doi: 10.1038/ncomms8654. URL <https://www.nature.com/articles/ncomms8654>. Publisher: Nature Publishing Group.
- [77] Y.-A. Chen, A. Kapustin, and Đ. Radičević. Exact bosonization in two spatial dimensions and a new class of lattice gauge theories. *Annals of Physics*, 393: 234–253, 2018. ISSN 0003-4916. doi: <https://doi.org/10.1016/j.aop.2018.03.024>. URL <https://www.sciencedirect.com/science/article/pii/S0003491618300873>.
  - [78] Y.-A. Chen and A. Kapustin. Bosonization in three spatial dimensions and a 2-form gauge theory. *Phys. Rev. B*, 100:245127, Dec 2019. doi: 10.1103/PhysRevB.100.245127. URL <https://link.aps.org/doi/10.1103/PhysRevB.100.245127>.
  - [79] Y.-A. Chen. Exact bosonization in arbitrary dimensions. *Phys. Rev. Res.*, 2: 033527, Sep 2020. doi: 10.1103/PhysRevResearch.2.033527. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.2.033527>.
  - [80] Y.-A. Chen and Y. Xu. Equivalence between fermion-to-qubit mappings in two spatial dimensions. *PRX Quantum*, 4:010326, Mar 2023. doi: 10.1103/PRXQuantum.4.010326. URL <https://link.aps.org/doi/10.1103/PRXQuantum.4.010326>.
  - [81] Z. Jiang, J. McClean, R. Babbush, and H. Neven. Majorana loop stabilizer codes for error mitigation in fermionic quantum simulations. *Physical Review Applied*, 12(6):064041, 2019.
  - [82] C. Derby, J. Klassen, J. Bausch, and T. Cubitt. Compact fermion to qubit mappings. *Phys. Rev. B*, 104:035118, Jul 2021. doi: 10.1103/PhysRevB.104.035118. URL <https://link.aps.org/doi/10.1103/PhysRevB.104.035118>.
  - [83] R. W. Chien and J. Klassen. Optimizing fermionic encodings for both hamiltonian and hardware. *arXiv eprint arXiv:2210.05652*, 2022. URL <https://arxiv.org/abs/2210.05652>.
  - [84] R. Nigmatullin, K. Hemery, K. Ghanem, S. Moses, D. Gresh, P. Siegfried, M. Mills, T. Gatterman, N. Hewitt, E. Granet, and H. Dreyer. Experimental demonstration of break-even for the compact fermionic encoding. *arXiv eprint arXiv:2409.06789*, 2024. URL <https://arxiv.org/abs/2409.06789>.
  - [85] C. Cade, L. Mineh, A. Montanaro, and S. Stanisic. Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Physical Review B*, 102(23):235122, December 2020. ISSN 2469-9950, 2469-9969. doi: 10.1103/PhysRevB.102.235122. URL <http://arxiv.org/abs/1912.06007>. arXiv:1912.06007 [quant-ph].
  - [86] T. Weaving, A. Ralli, P. J. Love, S. Succi, and P. V. Coveney. Contextual Subspace Variational Quantum Eigensolver Calculation of the Dissociation Curve of Molecular Nitrogen on a Superconducting Quantum Computer. *arXiv preprint arXiv:2312.04392*, 2023. doi: 10.48550/arXiv.2312.04392.

- [87] H. G. A. Burton. Accurate and gate-efficient quantum Ansätze for electronic states without adaptive optimization. *Phys. Rev. Res.*, 6:023300, Jun 2024. doi: 10.1103/PhysRevResearch.6.023300. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.6.023300>.
- [88] J. Dereziński and C. Gérard. *Mathematics of Quantization and Quantum Fields*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, Cambridge, 2023. ISBN 978-1-00-929082-1. doi: 10.1017/9781009290876. URL <https://www.cambridge.org/core/books/mathematics-of-quantization-and-quantum-fields/3F2652F5759A09E8165EE08E3F91CC35>.
- [89] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme. Tapering off qubits to simulate fermionic Hamiltonians, 2017. URL <https://arxiv.org/abs/1701.08213>.
- [90] R. Sarkar and E. Berg. On sets of maximally commuting and anticommuting Pauli operators. *Research in the Mathematical Sciences*, 8, 03 2021. doi: 10.1007/s40687-020-00244-1.
- [91] J. Dehaene and B. De Moor. The Clifford group, stabilizer states, and linear and quadratic operations over GF(2). *Physical Review A*, 68(4):042318, October 2003. ISSN 1050-2947, 1094-1622. doi: 10.1103/PhysRevA.68.042318. URL <http://arxiv.org/abs/quant-ph/0304125>. arXiv:quant-ph/0304125.
- [92] D. Picozzi and J. Tennyson. Symmetry-adapted encodings for qubit number reduction by point-group and other Boolean symmetries. *Quantum Science and Technology*, 8(3):035026, jun 2023. doi: 10.1088/2058-9565/acd86c. URL <https://dx.doi.org/10.1088/2058-9565/acd86c>.
- [93] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, November 2004. ISSN 1050-2947, 1094-1622. doi: 10.1103/PhysRevA.70.052328. URL <https://link.aps.org/doi/10.1103/PhysRevA.70.052328>.
- [94] C. McLauchlan and B. Béri. A new twist on the Majorana surface code: Bosonic and fermionic defects for fault-tolerant quantum computation. *Quantum*, 8:1400, July 2024. ISSN 2521-327X. doi: 10.22331/q-2024-07-10-1400. URL <https://doi.org/10.22331/q-2024-07-10-1400>.
- [95] B. Harrison, J. Necaise, A. Projansky, and J. D. Whitfield. A Sierpinski Triangle Data Structure for Efficient Array Value Update and Prefix Sum Calculation, March 2024. URL [http://arxiv.org/abs/2403.03990](https://arxiv.org/abs/2403.03990).
- [96] J. Bringewatt and Z. Davoudi. Parallelization techniques for quantum simulation of fermionic systems. *Quantum*, 7:975, April 2023. ISSN 2521-327X. doi: 10.22331/q-2023-04-13-975. URL <https://doi.org/10.22331/q-2023-04-13-975>.
- [97] G. Mitchison and R. Durbin. Optimal numberings of an  $N \times N$  array. *SIAM Journal on Algebraic Discrete Methods*, 7(4):571–582, 1986.

- [98] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pages 53–76, 1957.
- [99] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, 1974.
- [100] M. R. Garey, R. L. Graham, D. S. Johnson, and D. E. Knuth. Complexity results for bandwidth minimization. *SIAM Journal on Applied Mathematics*, 34(3):477–495, 1978.
- [101] M. Juvan and B. Mohar. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics*, 36(2):153–168, 1992. ISSN 0166-218X. doi: [https://doi.org/10.1016/0166-218X\(92\)90229-4](https://doi.org/10.1016/0166-218X(92)90229-4). URL <https://www.sciencedirect.com/science/article/pii/0166218X92902294>.
- [102] Y.-L. Lai and K. Williams. A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs. *Journal of graph theory*, 31(2):75–94, 1999.
- [103] S. B. Horton. *The optimal linear arrangement problem: algorithms and approximation*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, 1997.
- [104] A. George and A. Pothen. An analysis of spectral envelope reduction via quadratic assignment problems. *SIAM Journal on Matrix Analysis and Applications*, 18(3):706–732, 1997. doi: 10.1137/S089547989427470X. URL <https://doi.org/10.1137/S089547989427470X>.
- [105] G. N. Frederickson and S. E. Hambrusch. Planar linear arrangements of outerplanar graphs. *IEEE transactions on Circuits and Systems*, 35(3):323–333, 1988.
- [106] F.-R. K. Chung. On optimal linear arrangements of trees. *Computers & mathematics with applications*, 10(1):43–60, 1984.
- [107] J. B. Saxe. Dynamic-Programming Algorithms for Recognizing Small-Bandwidth Graphs in Polynomial Time. *SIAM Journal on Algebraic Discrete Methods*, 1(4):363–369, 1980. doi: 10.1137/0601042. URL <https://doi.org/10.1137/0601042>.
- [108] J. D. Whitfield, V. Havlíček, and M. Troyer. Local spin operators for fermion simulations. *Phys. Rev. A*, 94:030301, Sep 2016. doi: 10.1103/PhysRevA.94.030301. URL <https://link.aps.org/doi/10.1103/PhysRevA.94.030301>.
- [109] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah. On the Qubit Routing Problem. In Wim van Dam and Laura Mancinska, editors, *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, volume 135 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:32, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-112-2. doi: 10.4230/LIPIcs.TQC.2019.5. URL <http://drops.dagstuhl.de/opus/volltexte/2019/10397>.

- [110] M. G. Algaba, P. V. Sriluckshmy, M. Leib, and D. Šimkovic IV. Low-depth simulations of fermionic systems on square-grid quantum hardware. *Quantum*, 8:1327, April 2024. ISSN 2521-327X. doi: 10.22331/q-2024-04-30-1327. URL <https://doi.org/10.22331/q-2024-04-30-1327>.
- [111] J. Jiang and X. Sun and S.-H. Teng and B. Wu and K. Wu and J. Zhang. Optimal Space-Depth Trade-Off of CNOT Circuits in Quantum Logic Synthesis. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 213–229, 2019. doi: 10.1137/1.9781611975994.13. URL <https://epubs.siam.org/doi/abs/10.1137/1.9781611975994.13>.



# **Appendix A**

## **Glossary**

The following tables contain a glossary of the notation in this thesis for ease of use.

<i>symbol</i>	<i>object type</i>	<i>description</i>
<b>Chapter 1</b>		
$B_{(\mathrm{h})}(\mathcal{H})$	operator algebra	The set of bounded, linear (Hermitian) operators on the Hilbert space $\mathcal{H}$ .
$\mathcal{H}_{\mathrm{f}}^n$	fermionic state space	The $2^n$ -dimensional state space of an $n$ -mode fermionic system.
$ \Omega_{\mathrm{vac}}\rangle$	$\in \mathcal{H}_{\mathrm{f}}^n$	The fermionic vacuum state, representing the wavefunction of an $n$ -mode fermionic system with no fermions present.
$ \mathbf{f}\rangle$	$\in \mathcal{H}_{\mathrm{f}}^n$	Fock basis state corresponding to the occupation number vector $\mathbf{f} = (f_0, f_1, \dots, f_{n-1}) \in \mathbb{Z}_2^n$ .
$\mathbf{1}_i$	$\in \mathbb{Z}_2^n$	The $n$ -bit string consisting of zeroes except in the $i$ th place.
$\hat{a}_i^{(\dagger)}$	$\in B(\mathcal{H}_{\mathrm{f}}^n)$	Annihilation (creation) operator for a fermion in the $i$ th mode, for $i \in [n]$ . Collectively, the operators satisfy the canonical anticommutation relations $\{\hat{a}_i, \hat{a}_j^\dagger\} = \delta_{ij}\hat{1}$ , $\{\hat{a}_i, \hat{a}_j\} = 0$ .
$\hat{\gamma}_{2i}, \hat{\gamma}_{2i+1}$	$\in B(\mathcal{H}_{\mathrm{f}}^n)$	Majorana operators defined via $\hat{\gamma}_{2i} = \hat{a}_i + \hat{a}_i^\dagger$ , $\hat{\gamma}_{2i+1} = -i(\hat{a}_i - \hat{a}_i^\dagger)$ or equivalently $\hat{a}_i = \frac{1}{2}(\hat{\gamma}_{2i} + i\hat{\gamma}_{2i+1})$ for $i \in [n]$ . The Majorana operators satisfy the canonical anticommutation relations $\hat{\gamma}_i^\dagger = \hat{\gamma}_i$ , $\{\hat{\gamma}_i, \hat{\gamma}_j\} = 2\delta_{ij}\hat{1}$ for $i, j \in [2n]$ .
$\hat{H}_{\mathrm{fermion}}$	$\in B_{\mathrm{h}}(\mathcal{H}_{\mathrm{f}}^n)$	Second-quantised fermionic Hamiltonian. Two useful decompositions include $\hat{H}_{\mathrm{fermion}} = \sum_{i,j \in [n]} (c_{ij}) \hat{a}_i^\dagger \hat{a}_j + \sum_{i,j,k,l} (c_{kl}^{ij}) \hat{a}_i^\dagger \hat{a}_j \hat{a}_k^\dagger \hat{a}_l$ and the Majorana product decomposition $\hat{H}_{\mathrm{fermion}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}$ where $\Lambda$ is a subset of Majorana products.
$\mathcal{H}_2$	$\sim \mathbb{C}_2$	Two-dimensional complex Hilbert space with standard basis $\mathfrak{C}_1 = \{ 0\rangle,  1\rangle\}$ , which describes the state of a single qubit or fermionic mode.
$\mathcal{H}_2^{\otimes m}$	$\sim \mathbb{C}_2^{\otimes n}$	The $2^m$ -dimensional state space of an $m$ -qubit system.

---

<i>symbol</i>	<i>object type</i>	<i>description</i>
$\mathfrak{C}_m$	$\subset \mathcal{H}_2^{\otimes m}$	The phaseless computational basis for $\mathcal{H}_2^{\otimes m}$ : $\mathfrak{C}_m = \{ 00\dots 0\rangle,  10\dots 0\rangle, \dots,  11\dots 1\rangle\}$ .
$\mathcal{U}_m$	subgroup of $2^m \times 2^m$ matrices	The $m$ -qubit unitary matrices, which satisfy $U^\dagger U = \mathbb{1}^{\otimes m}$ for all $U \in \mathcal{U}_m$ .
$X, Y, Z$	$\in \mathcal{U}_1$	The single-qubit Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ , $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ , $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ .
$\mathcal{P}_m$	subgroup of $\mathcal{U}_m$	The $m$ -qubit Pauli group $\langle X, Y, Z \rangle^{\otimes m}$ . We call the elements of $\mathcal{P}_m$ Pauli strings.
$\mathcal{C}_m$	subgroup of $\mathcal{U}_m$	The $m$ -qubit Clifford group, all $2^m \times 2^m$ unitary matrices that normalise the Pauli group: $\mathcal{C}_m = \{C \in \mathcal{U}_m \mid C\mathcal{P}_m C^\dagger = \mathcal{P}_m\}$ .
$C$	$: \mathcal{U}_m \rightarrow \mathbb{R}$ (function)	A cost function that models the resource requirements of $m$ -qubit unitary operations according to a relevant metric, e.g. fundamental gate count, $T$ -gate count, query count. We extend the domain to Hermitian operators $C : B_h(\mathcal{H}_2^{\otimes m}) \rightarrow \mathbb{R}$ via applying $C$ to the set of terms in a unitary decomposition of the Hermitian operator.
$W$	$: \mathcal{P}_m \rightarrow \mathbb{R}$ (function)	The weight of a Pauli operator, which is the number of qubits upon which it operates nontrivially.

---

<i>symbol</i>	<i>object type</i>	<i>description</i>
<b>Chapter 2</b>		
$\mathcal{U}(\mathcal{H}_1, \mathcal{H}_2)$	unitary map	A unitary (i.e. inner–product preserving) map between Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ .
$\mathfrak{m}$	$\in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	An $n$ –mode fermion–qubit mapping.
$\Gamma_{2i}, \Gamma_{2i+1}$	$\in \mathcal{U}_n$	The representations of the Majorana operators $\hat{\gamma}_{2i}, \hat{\gamma}_{2i+1}$ under a fermion–qubit mapping $\mathfrak{m}$ for $i \in [n]$ . We say that $\mathfrak{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))$ , as an ordered sequence of pairs of Majorana representations identifies the mapping $\mathfrak{m}$ up to a complex phase via $\Gamma_i = \mathfrak{m} \cdot \hat{\gamma}_i \cdot \mathfrak{m}^\dagger$ for all $i \in [2n]$ . As anticommuting, Hermitian, $n$ –qubit unitary operators, the Majorana representations satisfy $\{\Gamma_i, \Gamma_j\} = 2\delta_{ij}\mathbb{1}^{\otimes n}$ , $\Gamma_i^\dagger = \Gamma_i$ .
$ \mathbf{f}_m\rangle$	$\in \mathcal{H}_2^{\otimes n}$	The representations of the Fock basis states $\{ \mathbf{f}\rangle \mid \mathbf{f} \in \mathbb{Z}_2^n\}$ under a fermion–qubit mapping $\mathfrak{m}$ . The qubit Fock basis uniquely identifies the mapping $\mathfrak{m}$ via $ \mathbf{f}_m\rangle = \mathfrak{m} \mathbf{f}\rangle$ . The state $ \mathbf{f}_m\rangle$ is a $(-1)^{f_i}$ –eigenstate of $-i\Gamma_{2i}\Gamma_{2i+1}$ for all $i \in [2n]$ .
$\mathfrak{m}_{\text{JW}}$	$\in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	The $n$ –mode Jordan–Wigner transformation.
$\gamma_{2i}, \gamma_{2i+1}$	$\in \mathcal{P}_n$	The $n$ –qubit Pauli operators $\gamma_{2i} = (\bigotimes_{k=0}^{i-1} Z_k)X_i$ , $\gamma_{2i+1} = (\bigotimes_{k=0}^{i-1} Z_k)Y_i$ for $i \in [n]$ , which are the representations of the Majorana operators $\hat{\gamma}_i$ under the Jordan–Wigner transformation $\mathfrak{m}_{\text{JW}}$ .
$U_m$ ( $C_m$ )	$\in \mathcal{U}_n$ ( $\mathcal{C}_n$ )	The unique unitary (Clifford) matrix with the property $U_m$ ( $C_m$ ) : $ \mathbf{f}\rangle \mapsto  \mathbf{f}_m\rangle$ for all $\mathbf{f} \in \mathbb{Z}_2^n$ , where $\mathfrak{m}$ is an $n$ –mode (Pauli–based) fermion–qubit mapping. The matrix satisfies $U_m \gamma_i U_m^\dagger = \Gamma_i$ (or $C_m \gamma_i C_m^\dagger = \Gamma_i$ in the case that $\Gamma_i \in \mathcal{P}_n$ ) for all $i \in [2n]$ .
$H_{\text{qubit}}$	$\in B_h(\mathcal{H}_2^{\otimes n})$	The $n$ –qubit Hamiltonian that results from applying a fermion–qubit mapping $\mathfrak{m}$ to an $n$ –mode fermionic Hamiltonian $\hat{H}_{\text{fermion}}$ , via $H_{\text{qubit}} = \mathfrak{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathfrak{m}^\dagger$ .

<i>symbol</i>	<i>object type</i>	<i>description</i>
“Pauli–based mapping”	$\mathfrak{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathfrak{m}$ for which the Majorana representations $\Gamma_i \in \mathcal{P}_n$ are Pauli operators.
$\tilde{\mathcal{P}}_n$	$\subset \mathcal{P}_n$	The unsigned Pauli operators $\{\mathbb{1}, X, Y, Z\}^{\otimes n}$ . Any Pauli operators is of the form $\pm P$ or $\pm iP$ , for some $P \in \mathcal{P}_n$ .
$\mathcal{G}_{(\max)}$	$\subset \mathcal{P}_n$	A (maximally) anticommuting set of $n$ –qubit Pauli operators $\mathcal{G} = \{\Gamma_i\}_{i=0}^{k-1}$ , where the elements satisfy $\{\Gamma_i, \Gamma_j\} = 0$ . Maximally anticommuting sets of Pauli operators have no anticommuting supersets, and reach a maximum size of $2n+1$ elements.
$\prod \mathcal{G}$	$\in \mathcal{P}_n$	The product $\prod_{\Gamma \in \mathcal{G}} \Gamma$ of elements in an anticommuting set of $n$ –qubit Pauli operators $\mathcal{G}$ . If $ \mathcal{G}  = 2n$ , then $\prod \mathcal{G}$ is (up to a phase of $\pm 1$ or $\pm i$ ) the only other element of $\mathcal{P}_n$ to anticommute with each of the elements of $\mathcal{G}$ .
$T$	graph	An $n$ –vertex ternary tree, which is a labelled tree graph with $n$ vertices, each of which has at most three children.
$\tilde{\mathcal{G}}_T$	$\subset \tilde{\mathcal{P}}_n$	The maximally anticommuting set $\tilde{\mathcal{G}}_T$ of unsigned Pauli operators that arises from converting the $(2n+1)$ root-to-leaf paths of an $n$ –vertex ternary tree $T$ into Pauli strings.
“ $T$ –based mapping”	$\mathfrak{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathfrak{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ for which $\pm \Gamma_i \in \tilde{\mathcal{G}}_T$ for some $n$ –vertex ternary tree $T$ .
“ternary tree transformation”	$\mathfrak{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathfrak{m}$ for which there exists some ternary tree $T$ such that $\mathfrak{m}$ is a $T$ –based mapping.
$e^{i\theta} \mathfrak{C}_n, \pm \mathfrak{C}_n$	$\subset \mathcal{H}_2^{\otimes n}$	Computational bases with phase offsets for $\theta \in [0, 2\pi]$ : $e^{i\theta} \mathfrak{C}_n = \{e^{i\theta}  00\dots 0\rangle, \dots, e^{i\theta}  11\dots 1\rangle\}$ and $\pm \mathfrak{C}_n = \mathfrak{C}_n \cup (-1) \mathfrak{C}_n$ .

<i>symbol</i>	<i>object type</i>	<i>description</i>
“product-preserving mapping”	$\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathbf{m}$ with Fock states $ \mathbf{f}_m\rangle$ that are product states. A sufficient condition is for $ \mathbf{0}_m\rangle$ to be a product state.
“classical encoding of the Fock basis”	$\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathbf{m}$ for which $ \mathbf{f}_m\rangle \in \mathfrak{C}_n$ for all $\mathbf{f} \in \mathbb{Z}_2^n$ .
“affine encoding of the Fock basis”	$\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathbf{m}$ for which $ \mathbf{f}_m\rangle =  G(\mathbf{f} \oplus \mathbf{b})\rangle$ for all $\mathbf{f} \in \mathbb{Z}_2^n$ , where $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ and $\mathbf{b} \in \mathbb{Z}_2^n$ . Affine encodings are the Pauli–based classical encodings ( <i>Theorem 1</i> ).
“linear encoding of the Fock basis”	$\mathbf{m} \in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	A fermion–qubit mapping $\mathbf{m}$ for which $ \mathbf{f}_m\rangle =  G\mathbf{f}\rangle$ for all $\mathbf{f} \in \mathbb{Z}_2^n$ , where $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ .
$[C]$	$2n \times (2n + 1)$ tableau	The stabiliser tableau of the Clifford operator $C \in \mathcal{C}_n$ . The first $n$ columns are the $2n$ -bit symplectic representations of (unsigned) $CX_i C^\dagger$ , and the second $n$ columns of (unsigned) $CZ_i C^\dagger$ , for $i \in [n]$ . The final column stores the signs of the operators $CX_i C^\dagger$ in its first $n$ entries, and of $CZ_i C^\dagger$ in its final $n$ entries. Any Clifford operator of the form $Ce^{i\theta}$ for $\theta \in [0, 2\pi)$ shares the same tableau.
$C_G$	$\in \mathcal{C}_n$	The unique Clifford operator $C_G \in \mathcal{C}_n$ to implement the basis transformation $ \mathbf{f}\rangle \mapsto  G\mathbf{f}\rangle$ , where $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ . Its stabiliser tableau is $[C_G] = \left[ \begin{array}{cc c} G & 0 & 0 \\ 0 & (G^{-1})^\top & 0 \end{array} \right].$
$U(i), P(i),$ $F(i), R(i)$	$\subset [n]$	The update, parity, flip and remainder sets of the integer $i \in [n]$ for a specific invertible binary matrix $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ .

<i>symbol</i>	<i>object type</i>	<i>description</i>
<b>Chapter 3</b>		
$[\mathfrak{m}]$	equivalence class on the set of Pauli-based mappings	The template for a Pauli-based mapping $\mathfrak{m}$ . The template $[\mathfrak{m}]$ consists of all mappings with identical Majorana representations to $\mathfrak{m}$ up to the labelling of qubits, local measurement bases and fermions, and up to the signs and ordering of Majorana representations within pairs.
$\mathfrak{m}(T)$	fermion–qubit mapping	The unique $T$ -based mapping that linearly encodes the Fock basis of the form $\mathfrak{m}(T) \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ .
$C_T$	$\in \mathcal{C}_n$	The unique Clifford operator $C_T \in \mathcal{C}_n$ such that $C_T \gamma_i C_T^\dagger = \Gamma_i$ , where $\mathfrak{m}(T) \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ .
$\{\tilde{\Gamma}_i\}_{i=0}^{2n}$	$\subset \tilde{\mathcal{P}}_n$	A specific ordering of the elements in $\tilde{\mathcal{G}}_T$ .
$\#y$	$\tilde{\mathcal{P}}_n \rightarrow [n]$ (function)	A function that counts the number of $Y$ operators in unsigned Pauli strings.
$\{\hat{\Gamma}_i\}_{i=0}^{2n-1}$	$\subset \{\mathbb{1}, X, -iY, Z\}^{\otimes n}$	A set of anticommuting Pauli strings $\{\hat{\Gamma}_i, \hat{\Gamma}_j\} = 0$ , where $\hat{\Gamma}_i = (-1)^{\#y(\tilde{\Gamma}_i)} \tilde{\Gamma}_i$ .
$T_S$	tree graph	The ternary subtree of $T$ on the vertices with labels $S \subseteq [n]$ .
$G_T$	$\in \mathrm{GL}_n(\mathbb{Z}_2)$	The unique invertible binary matrix $G \in \mathrm{GL}_n(\mathbb{Z}_2)$ such that $ G_T \mathbf{f}\rangle =  \mathbf{f}_{\mathfrak{m}(T)}\rangle$ for all $\mathbf{f} \in \mathbb{Z}_2^n$ .

<i>symbol</i>	<i>object type</i>	<i>description</i>
<b>Chapter 4</b>		
$\sum_{\hat{h} \in \Lambda} c_{\hat{h}} h$	$\in B_h(\mathcal{H}_2^{\otimes n})$	The Pauli decomposition of a qubit Hamiltonian $H_{\text{qubit}} = \mathbf{m} \cdot \hat{H}_{\text{fermion}} \cdot \mathbf{m}^\dagger$ that arises from applying a Pauli-based mapping $\mathbf{m}$ to a fermionic Hamiltonian $\hat{H}_{\text{fermion}} = \sum_{\hat{h} \in \Lambda} c_{\hat{h}} \hat{h}$ .
$C_{\text{avg}}$	$: B_h(\mathcal{H}_2^{\otimes n}) \rightarrow \mathbb{R}$ (function)	A cost function that gives the average weight of the terms in the Pauli decomposition of a qubit Hamiltonian.
$C_{\text{max}}$	$: B_h(\mathcal{H}_2^{\otimes n}) \rightarrow \mathbb{R}$ (function)	A cost function that gives the maximum weight of any one term in the Pauli decomposition of a qubit Hamiltonian.
$C_{\text{depth}}$	$: B_h(\mathcal{H}_2^{\otimes n}) \rightarrow \mathbb{R}$ (function)	A cost function that gives minimum number of non-overlapping subsets of the terms in the Pauli decomposition of a qubit Hamiltonian.
$G(\hat{H}_{\text{fermion}})$	$n$ -vertex graph	A graph consisting of edges $(i, j)$ between vertices $i, j \in [n]$ if and only if the terms $\hat{\gamma}_{2i}\hat{\gamma}_{2j}$ , $\hat{\gamma}_{2i+1}\hat{\gamma}_{2j}$ , $\hat{\gamma}_{2i}\hat{\gamma}_{2j+1}$ or $\hat{\gamma}_{2i+1}\hat{\gamma}_{2j+1}$ appear in the $n$ -mode fermionic Hamiltonian $\hat{H}_{\text{fermion}}$ .
$\hat{H}_{\text{fermion}}^{\text{on-site}}$	$\in B_h(\mathcal{H}_f^n)$	The terms in the $n$ -mode fermionic Hamiltonian $\hat{H}_{\text{fermion}}$ that are not fermionic hopping terms.
$\hat{H}_{\text{fermion}}^{\text{hopping}}$	$\in B_h(\mathcal{H}_f^n)$	The hopping terms of the $n$ -mode fermionic Hamiltonian $\hat{H}_{\text{fermion}}$ .
$H_{\text{qubit}}^{\text{loc}}$	$\in B_h(\mathcal{H}_2^{\otimes n})$	The $n$ -qubit Hamiltonian $H_{\text{qubit}}^{\text{loc}} = \mathbf{m}_{\text{JW}} \cdot \hat{H}_{\text{fermion}}^{\text{on-site}} \cdot (\mathbf{m}_{\text{JW}})^\dagger$ , which consists of terms that are local, acting nontrivially on two qubits or less.
$H_{\text{qubit}}^{\text{non-loc}}$	$\in B_h(\mathcal{H}_2^{\otimes n})$	The $n$ -qubit Hamiltonian $H_{\text{qubit}}^{\text{non-loc}} = \mathbf{m}_{\text{JW}} \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathbf{m}_{\text{JW}})^\dagger$ , which consists of terms that are non-local, acting nontrivially on anywhere between two and $n$ qubits. Hopping terms between modes $i$ and $j$ in $\hat{H}_{\text{fermion}}^{\text{hopping}}$ map to non-local Pauli operators with weight $( i - j  + 1)$ in $H_{\text{qubit}}^{\text{non-loc}}$ .

---

<i>symbol</i>	<i>object type</i>	<i>description</i>
$\mathfrak{m}(\sigma)$	$\in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes n})$	The $n$ -mode fermion–qubit mapping that results from applying the permutation $\sigma \in S_n$ to relabel fermionic modes. If $\mathfrak{m} \sim ((\Gamma_{2i}, \Gamma_{2i+1}))_{i=0}^{n-1}$ , then $\mathfrak{m}(\sigma) \sim ((\Gamma_{2\sigma(i)}, \Gamma_{2\sigma(i)+1}))_{i=0}^{n-1}$ .
$H_{\text{qubit}}^{\text{non-loc}}(\sigma)$	$\in B_h(\mathcal{H}_2^{\otimes n})$	The $n$ -qubit Hamiltonian $H_{\text{qubit}}^{\text{non-loc}} = \mathfrak{m}_{\text{JW}}(\sigma) \cdot \hat{H}_{\text{fermion}}^{\text{hopping}} \cdot (\mathfrak{m}_{\text{JW}}(\sigma))^{\dagger}$ , which consists of terms that are non-local, acting nontrivially on anywhere between two and $n$ qubits. Hopping terms between modes $i$ and $j$ in $\hat{H}_{\text{fermion}}^{\text{hopping}}$ map to non-local Pauli operators with weight $( \sigma(i) - \sigma(j)  + 1)$ in $H_{\text{qubit}}^{\text{non-loc}}(\sigma)$ .
$C_G^p(\sigma)$	$: S_n \rightarrow \mathbb{N}$ (function)	The $p$ -sum $C_G^p(\sigma) = (\sum_{(i,j) \in G}  \sigma(i) - \sigma(j) ^p)^{1/p}$ of the $n$ -vertex graph $G$ after relabelling the vertices $i \mapsto \sigma(i)$ for some permutation $\sigma \in S_n$ .
$\sigma_Z, \sigma_S$	$\in S_{N^2}$	Permutations of the vertex labels in an $N \times N$ square lattice graph, so that the path taken by ascending vertex labels follows either the Z– or S– patterns through the graph, respectively.
$\sigma_M$	$\in S_{N^2}$	The permutation of the vertex labels in an $N \times N$ square lattice graph $G$ that effects the Mitchison–Durbin pattern, which minimises the 1-sum (or “edge-sum”) $C_G^1(\sigma)$ for all $\sigma \in S_{N^2}$ .

---

<i>symbol</i>	<i>object type</i>	<i>description</i>
<b>Chapter 5</b>		
$\mathfrak{l}$	$\in \mathcal{U}(\mathcal{H}_f^n, \mathcal{H}_2^{\otimes m})$ (isometry)	An isometry between $n$ -mode and $m$ -qubit Hilbert spaces, for $m > n$ .
$\tilde{\mathcal{H}}$	$\leq \mathcal{H}_2^{\otimes m}$	A $(2^n)$ -dimensional subspace of $\mathcal{H}_2^{\otimes m}$ spanned by states $ \tilde{\psi}\rangle \in \mathcal{H}_2^{\otimes m}$ that satisfy $(\prod_{\hat{h} \in \Lambda} h)  \tilde{\psi}\rangle = \pm  \tilde{\psi}\rangle$ for each subset of Majorana products $\Lambda' \subset \Lambda$ satisfying $\prod_{\hat{h} \in \Lambda'} \hat{h} = \pm \hat{1}$ , where $\Lambda$ is the set of terms in the Majorana product decomposition of an $n$ -mode fermionic Hamiltonian $\hat{H}_{\text{fermion}}$ and $h = \mathfrak{l} \cdot \hat{h} \cdot \mathfrak{l}$ for each $\hat{h} \in \Lambda$ .
$\Pi_{\tilde{\mathcal{H}}}$	$\in B(\mathcal{H}_2^{\otimes m}, \tilde{\mathcal{H}})$ (projector)	The projector from $\mathcal{H}_2^{\otimes m}$ into $\tilde{\mathcal{H}}$ , where $\tilde{\mathcal{H}}$ is a $(2^n)$ -dimensional subspace of $\mathcal{H}_2^{\otimes m}$ .
$\Pi_{\tilde{\mathcal{H}}} \cdot \mathfrak{l}$	$\in \mathcal{U}(\mathcal{H}_2^{\otimes m}, \tilde{\mathcal{H}})$	An ancilla–qubit mapping, a unitary map from $\mathcal{H}_f^n$ to $\tilde{\mathcal{H}} \leq \mathcal{H}_2^{\otimes m}$ .
$L_P$	$\in B(\mathcal{H}_2^{\otimes n}, \mathcal{H}_2^{\otimes(n+1)})$	A linear map that acts on the terms $\{h\}_{\hat{h} \in \Lambda}$ in the Pauli decomposition of a qubit Hamiltonian via $L_P : h \mapsto h' = hP \otimes \kappa_h$ or $h \otimes \kappa_h$ , whichever has less Pauli weight, where $\kappa_h$ ensures that the terms $\{h'\}_{\hat{h} \in \Lambda}$ obey the same Hermiticity and commutation relations as the original terms.
$\Pi_{\tilde{\mathcal{H}}} \cdot (\prod_{k=0}^{r-1} L_{P_k}) \cdot \mathfrak{m}_{\text{JW}}(\sigma)$		An $r$ -ancilla–qubit Jordan–Wigner transformation, using fermionic labelling $\sigma \in S_n$ .
$\rho'$	$\in S_{N^2}$	The permutation of vertex labels in an $N \times N$ square lattice graph $G$ that effects the Mitchison–Durbin pattern with variable side-length $x$ of the top-left square.
$\sigma_{M+2}$	$\in S_{N^2}$	The permutation of vertex labels in an $N \times N$ square lattice that effects a Mitchison–Durbin pattern $\rho'$ with $x \approx 0.40N$ . This fermionic labelling minimises the average Pauli weight of the qubit Hamiltonian arising from two-ancilla–qubit Jordan–Wigner transformations of square-lattice fermionic interactions.