Diagrammatic Design of Ansätze for Quantum Chemistry

Ayman El Amrani

St. John's College

A thesis submitted for the Honour School of Chemistry

Part II 2024

Pour ma mère et mon père.

Acknowledgements

Thank you Thomas Cervoni for your constant motivation and support.
Thank you David Tew and Stefano Gogioso for your patient supervision.
Thank you Razin Shaikh, Boldizsár Poór, Richie Yeung and Harny Wang for always finding the time to answer my questions.
Thank you to my friends and family for supporting me during this unconvential Master's.

Summary

A central challenge in computational quantum chemistry is the accurate simulation of fermionic systems. At the heart of these calculations lies the need to solve the Schrödinger equation to determine the many-electron wavefunction. An exact solution to this problem scales exponentially with the number of electrons. Classical computers struggle to store the increasingly large wavefunctions making this problem computationally intractable in many cases. In contrast, gate-based quantum computing presents a promising solution, offering the potential to represent electronic wavefunctions with polynomially scaling resources [1]. In other words, quantum computers are a natural tool of choice for simulating processes that are inherently quantum [2].

In the last two decades many advancements in quantum computing have been made in both hardware and software bringing us closer to being able to simulate molecular systems. Despite these advancements, we remain in the so-called Noisy Intermediate Scale Quantum (NISQ) era, characterised by challenges such as poor qubit fidelity, low qubit connectivity and limited coherence times. The NISQ era represents a transitional phase in quantum computing, where quantum devices are not yet error-corrected but are still capable of performing computations beyond the reach of classical computers. Overcoming the limitations of the NISQ era is crucial for realising the full potential of quantum computing in various fields, including quantum chemistry and materials science.

The Variational Quantum Eigensolver (VQE) algorithm is a method used to estimate the ground state energy of a molecular Hamiltonian by preparing a trial wavefunction,

calculating its energy, and optimising the wavefunction parameters classically until the energy converges to the best approximation for the ground state energy [3]. It is recognised as a leading algorithm for quantum simulation on NISQ devices due to its reduced resource requirements in terms of qubit count and coherence time [4].

This thesis extends methods developed by Richie Yeung [2] for the preparation and analysis of parametrised quantum circuits, and applies them to ansätze representing fermionic wavefunctions. We are concerned with two main questions on this theme. Firstly, can we use the ZX calculus [cite] to gain insights into the structure of the unitary product ansatz in the context of variational algorithms for quantum chemistry? Secondly, in the context of NISQ devices, can we use these insights to build better ansätze with reduced circuit depth and more efficient resources?

Contents

Chapter 1

ZX Calculus

The ZX calculus is a diagrammatic language for reasoning about quantum processes that has seen a large increase in applications over the past 10 years. It provides a novel perspective on quantum computation and quantum mechanics.

1.1 Diagrams & Generators

By sequentially or horizontally composing the generators known as *Z Spiders* (green) and *X Spiders* (red), we can construct undirected multigraphs known as ZX diagrams [5]. That is, graphs that allow multiple edges between vertices. Since *only connectivity matters* in the ZX calculus, a valid ZX diagram can be deformed as seen fit, provided that the order of inputs and outputs is preserved.

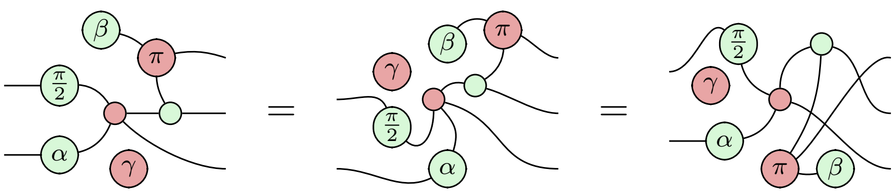
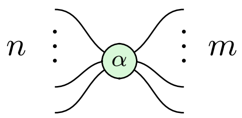


Figure 1.1: Three equivalent ZX diagrams (*only connectivity matters*).

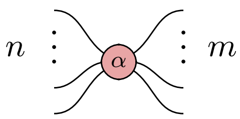
Z Spiders are defined with respect to the Z eigenbasis such that a Z Spider with n inputs and m outputs has the following interpretation as a linear map. Note that in this text, we will interpret the flow of time from left to right.



$$= |0\rangle^{\otimes m} \langle 0|^{\otimes n} + e^{i\alpha} |1\rangle^{\otimes m} \langle 1|^{\otimes n}$$

Figure 1.2: Interpretation of Z Spider as a linear map.

Similarly, X Spiders, which are defined with respect to the X eigenbasis, are interpreted as the following linear map.



$$= |+\rangle^{\otimes m} \langle +|^{\otimes n} + e^{i\alpha} |-\rangle^{\otimes m} \langle -|^{\otimes n}$$

Figure 1.3: Interpretation of X Spider as a linear map.

We can recover the $|0\rangle$ eigenstate using an X Spider that has a phase of zero, or the $|1\rangle$ eigenstate using an X Spider that has a phase of π.

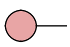
 $= |+\rangle + |-\rangle = \sqrt{2}\,|0\rangle$

Figure 1.4: $|0\rangle$ eigenstate

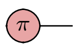
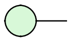
 $= |+\rangle - |-\rangle = \sqrt{2}\,|1\rangle$

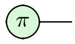
Figure 1.5: $|1\rangle$ eigenstate

Likewise, we have the $|+\rangle$ and $|-\rangle$ basis states from the corresponding Z Spider



$= |0\rangle + |1\rangle = \sqrt{2}\,|+\rangle$

Figure 1.6: $|+\rangle$ eigenstate

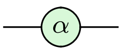


$= |0\rangle - |1\rangle = \sqrt{2}\,|-\rangle$

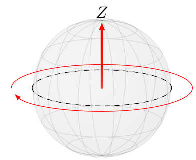
Figure 1.7: $|-\rangle$ eigenstate

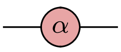
Whilst we obtain the correct states, we obtain the wrong scalar factor. For the remainder of this thesis, we will ignore global non-zero scalar factors. Hence, equal signs should be interpreted as 'equal up to a global phase'.

Single qubit rotations in the Z basis are represented by a Z Spider with a single input and a single output. Arbitrary rotations in the X basis are represented by the corresponding X spider. We can view these as rotations of the Bloch sphere.



$$= |0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \rightarrow$$





$$= |+\rangle\langle +| + e^{i\alpha}|-\rangle\langle -| = \frac{1}{2}\begin{pmatrix} 1+e^{i\alpha} & 1-e^{i\alpha} \\ 1-e^{i\alpha} & 1+e^{i\alpha} \end{pmatrix} \rightarrow$$

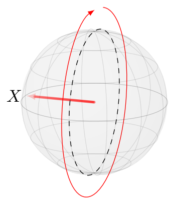
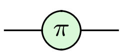
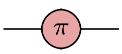


Figure 1.8: Arbitrary single qubit rotations in the Z and X bases.

We can recover the Pauli Z and Pauli X matrices by setting the angle $\alpha = \pi$.



$$= |0\rangle\langle 0| + e^{i\pi}|1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

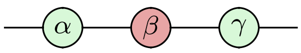


$$= |+\rangle\langle +| + e^{i\pi}|-\rangle\langle -| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Figure 1.9: Pauli Z and X gates in the ZX calculus.

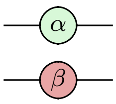
Composition

To calculate the matrix of a ZX diagram consisting of sequentially composed spiders, we take the matrix product. Note that the order of operation of matrix multiplication is the reverse as in the ZX diagram as we have defined it.



$$= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\gamma} \end{pmatrix} \begin{pmatrix} 1+e^{i\beta} & 1-e^{i\beta} \\ 1-e^{i\beta} & 1+e^{i\beta} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

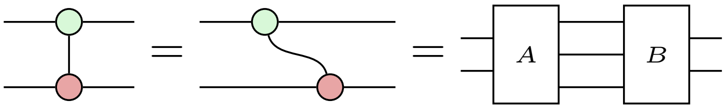
Alternatively, we could have chosen to compose the spiders in parallel, resulting in the tensor product.



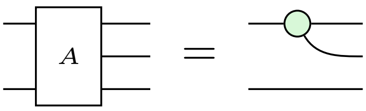
$$= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \otimes \begin{pmatrix} 1+e^{i\beta} & 1-e^{i\beta} \\ 1-e^{i\beta} & 1+e^{i\beta} \end{pmatrix}$$

Deriving the CNOT Gate

The CNOT gate in the ZX calculus is represented by a Z spider (control qubit) and an X spider (target qubit). We can arbitrarily deform the diagram and decompose it into matrix and tensor products as follows.

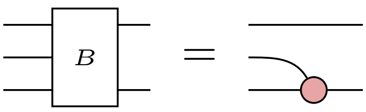


We can calculate matrix A, consisting of a single-input and two-output Z Spider (4×2 matrix) and an empty wire (identity matrix), by taking the tensor product.



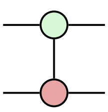
$$= \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Similarly, to calculate the matrix B, we take the following tensor product.



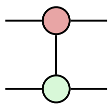
$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

We can then calculate the CNOT matrix by taking the matrix product of matrix A and matrix B as follows.



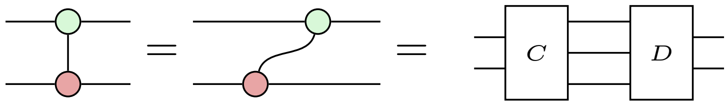
$$= \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \right] \left[\begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right] \simeq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Had we chosen to make the first qubit the target and the second qubit the control, we would have obtained the following.



$$= \left[\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0\end{pmatrix} \otimes \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\right]\left[\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \otimes \begin{pmatrix}1 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 1\end{pmatrix}\right] \simeq \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

Since *only connectivity matters*, we could have equivalently calculated the matrix of the CNOT gate by deforming the diagram as follows.



Hadamard Generator

All quantum gates are unitary transformations. Therefore, up to a global phase, an arbitrary single qubit rotation U can be viewed as a rotation of the Bloch sphere about some axis. We can decompose the unitary U using Euler angles to represent the rotation as three successive rotations [5].

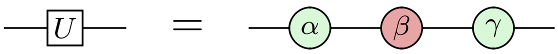


Figure 1.10: Arbitrary single-qubit rotation.

Recall that the Hadamard gate H switches between the $|0\rangle/|1\rangle$ and $|+\rangle/|-\rangle$ bases. That is, it corresponds to a rotation of the Bloch sphere by π radians about the line bisecting the X and Z axes. By choosing $\alpha = \beta = \gamma = \frac{\pi}{2}$, we obtain the Hadamard gate up to a global phase of $e^{-i\frac{\pi}{4}}$. We define the Hadamard generator below.

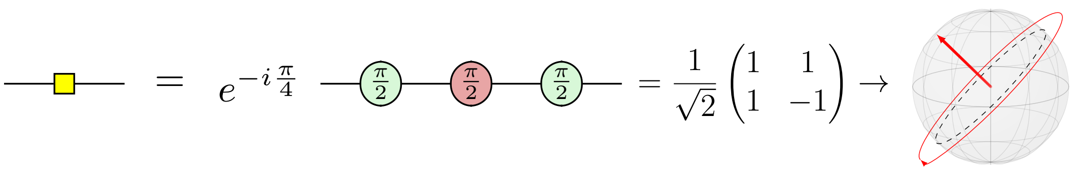


Figure 1.11: Hadamard generator in the ZX calculus.

There are many equivalent ways of decomposing the Hadamard gate using Euler angles. Note that the rightmost representations need no scalar corrections.

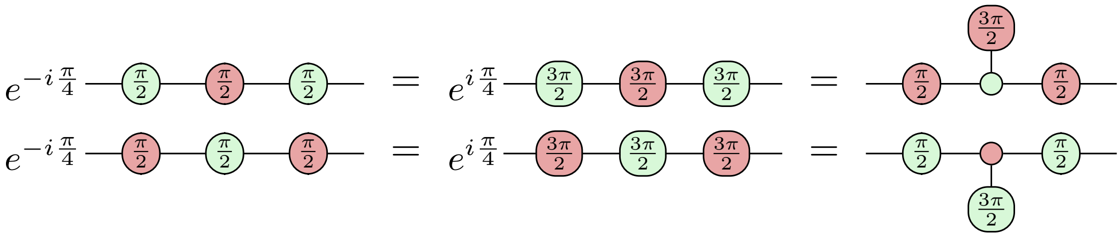
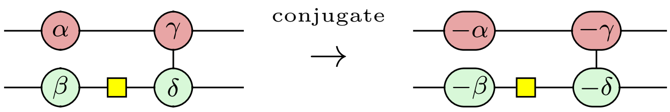


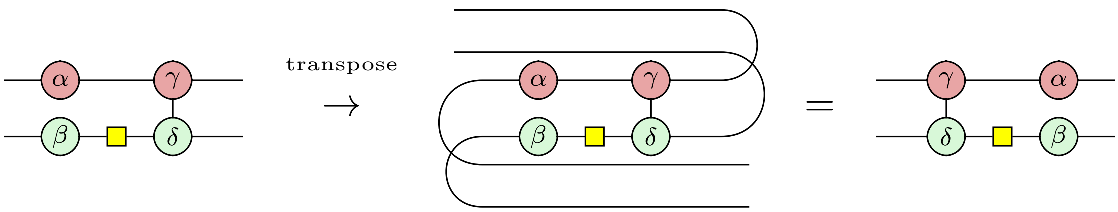
Figure 1.12: Equivalent definitions of the Hadamard generator.

Conjugate, Transpose and Adjoint

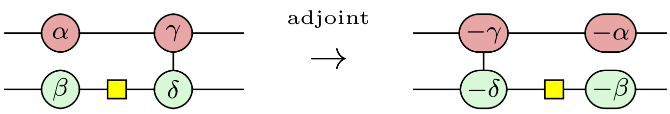
We can find the conjugate of a ZX diagram by simply negating the phases of all spiders in the diagram, $\alpha \to -\alpha, \beta \to -\beta, \ldots$.



Intuitively, we can find the transpose of a ZX diagram by turning all inputs into outputs and all outputs into inputs.



It is then a simple matter to find the Hermitian adjoint of a ZX diagram by first finding its conjugate, then its transpose.



1.2 Rewrite Rules

This section introduces the various rewrite rules that come equipped with the ZX calculus. These rules extend the ZX calculus from notation into a language.

Spider Fusion

The most fundamental rule of the ZX calculus is the *spider fusion* rule [5]. It states that two spiders connected by one or more wires fuse if they are the same colour. It is the generalisation of adding the phases of successive rotations of the Bloch sphere. Since we interpret the phases α and β as $e^{i\alpha}$ and $e^{i\beta}$, it follows that the phase $\alpha + \beta$ is modulo 2π.

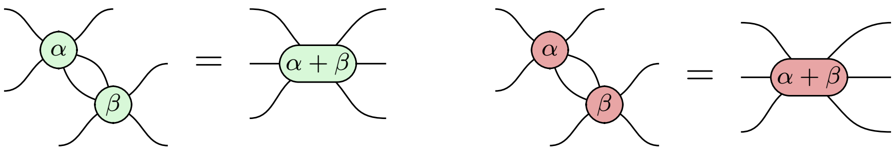


Figure 1.13: Spider fusion rule for Z spiders (left) and X spiders (right).

Using this rule we can identify useful commutation relations. Z rotations commute through CNOT controls and X rotations commute through CNOT targets.

Identity Removal

The *identity removal* rule states that any two-legged spider with no phase ($\alpha = 0$) is equivalent to an empty wire since a rotation by 0 radians is the same as no rotation.

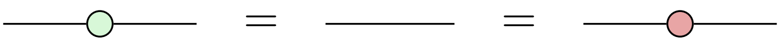
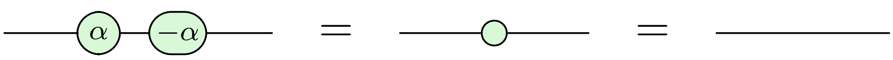


Figure 1.14: Identity removal rule.

Combining this with the spider fusion rule, we see that two successive rotations with opposite phases is equivalent to an empty wire.



Bialgebra Rule

Unlike the previous rules we have introduced, the *bialgebra rule* takes some time to understand intuitively. It is nethertheless important in many derivations.

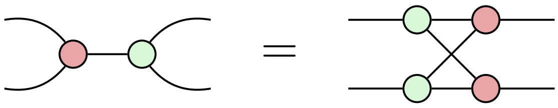
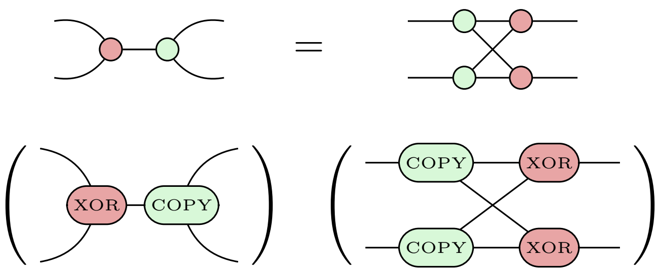


Figure 1.15: Bialgebra rule.

We assume that the left part of the bialgebra rule (three-legged X spider) behaves like the classical XOR gate with the computational basis states whilst the right part (three-legged Z spider) behaves like the classical COPY gate. The latter is known as the π copy rule (see appendix).

By considering the natural commutation relation of the classical XOR and COPY gates as motivation, it is clear that XORing the bits then copying them is indeed the same as copying then XORing.



Appendices

Bibliography

[1] Burton, H. G. A., Marti-Dafcik, D., Tew, D. P. & Wales, D. J. Exact electronic states with shallow quantum circuits from global optimisation. *npj Quantum Information* **9** (2023).

[2] Yeung, R. Diagrammatic design and study of ansätze for quantum machine learning (2020). `2011.11073`.

[3] McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics* **18**, 023023 (2016).

[4] Kirby, W. M. & Love, P. J. Variational quantum eigensolvers for sparse hamiltonians. *Phys. Rev. Lett. 127, 110503 (2021)* **127**, 110503 (2020). `2012.07171`.

[5] van de Wetering, J. Zx-calculus for the working quantum computer scientist (2020). `2012.13966`.