Spin-networks in the ZX-calculus

Richard D.P. East¹, Pierre Martin-Dussaud², and John Van de Wetering³

¹LIG & Institut Néel, Grenoble, France
²Institute for Gravitation and the Cosmos, The Pennsylvania State University, University Park, Pennsylvania 16802, USA
³University of Oxford, United Kingdom

The ZX-calculus, and the variant we consider in this paper, the ZXH-calculus, are formal diagrammatic languages for qubit quantum computing. In this paper we will show that this language can also be used to describe SU(2) representation theory. To achieve this we first recall the definition of Yutsis diagrams, a standard graphical calculus used in quantum chemistry and quantum gravity which captures the main features of SU(2) representation theory, and we show precisely how Yutsis diagrams embed within Penrose’s binor calculus. We then subsume both of these calculi into the ZXH-calculus. In the process we show how the SU(2) invariance up to a phase of Wigner symbols is trivially provable in the ZXH-calculus. Additionally, we show how we can explicitly diagrammatically calculate $3jm$, $4jm$ and $6j$ symbols. It has long been thought that quantum gravity should be closely aligned to quantum information theory. Our results demonstrate a way in which this connection can be made exact, by writing the spin-networks of loop quantum gravity (LQG) in the ZX-diagrammatic language of quantum computation.

Contents

1 Introduction ................................................................. 2

2 From Yutsis diagrams to spin-networks  .............................................. 3

2.1 SU(2) representation and recoupling theory  ............................. 3

2.2 Yutsis diagrams ........................................................... 7

2.3 Penrose diagrams .......................................................... 11

2.4 Spin-networks ............................................................. 14

3 The ZXH-calculus .......................................................... 15

4 Translating spin-networks into ZXH-diagrams .................................. 20

4.1 Links ................................................................. 21

4.2 Trivalent nodes .......................................................... 27

4.3 The ZXH Spin-network ................................................... 30

Richard D.P. East: rdp.east@gmail.com
Pierre Martin-Dussaud: martindussaud@gmail.com
John Van de Wetering: john@vandewetering.name
1 Introduction

The use of graphical calculi in modern physics is commonplace. The prototypical example are Feynman diagrams, introduced at the end of the 1940s and used for computing transition amplitude in particle physics [1]. In the 1960s, Yutsis et al. developed a graphical calculus for the quantum theory of angular momentum [2], while in the 1970s, Penrose was advocating the use of diagrams to deal with tensors [3]. Modern applications of these tensor networks, include Matrix Product States (MPS) and their higher-dimensional generalisations such as Projected Entangled Pair States (PEPS), which can be used in condensed matter physics to deal with ground states of quantum spin models [4]. A common theme here is that these diagrams offer the ability to reason about complex properties of some physical system without immediate recourse to more primitive calculation, usually in the form of matrix calculation or an extension thereof. The underlying philosophy is to take advantage of the two dimensions of our blackboards to literally draw our calculations, rather than to restrict ourselves to the usual one-dimensional lines of calculations. If done properly, the graphical approach can help us understand the structure of analytical expressions and make computations faster.

Diagrams are intended to capture some relevant aspects of the underlying physical system. The success of this task depends on a few things: universality — whether all the objects in the system can be represented in the language; soundness — whether every truth expressible in the language is true in the underlying system; and completeness — whether every truth of the underlying system is expressible in the language. Of the examples given above, most can then be described as a ‘sound non-universal semantically incomplete representation’.

A relatively recent diagrammatic calculus is the ZX-calculus [5, 6]. Any quantum computation between qubits, or more generally, any linear map between qubits, can be represented as a ZX-diagram, a type of tensor network created from a small number of generating elements. The ZX-calculus is a universal, sound, complete language for qubit linear algebra [7, 8]. The ZX-calculus has found practical use in quantum information, with results in measurement-based quantum computation [9–11], topological quantum computation [12–15], quantum error correction [16, 17] and quantum circuit compilation and optimisation [18–23]. For practical reasons, we will work in this paper with a variation of this calculus called the ZXH-calculus [24].

The ZXH-calculus is a calculus for qubit systems: each wire represents a two-dimensional (i.e. spin-1/2) Hilbert space. However, as was shown in [24], we can combine multiple wires together using a symmetriser in order to represent higher-spins. This connects the
ZXH-calculus to the representation theory of SU(2), and hence to the quantum theory of angular momentum. In this paper, we make this intuition formal and show how to represent Yutsis diagrams and spin-networks as ZXH-diagrams. In particular we show how to describe SU(2)-invariant functions such as the $6j$-symbol and compute them as tensor contractions. We also show how the diagrammatic perspective clarifies the SU(2)-invariance of Wigner symbols, which usually only becomes apparent with fairly opaque calculations.

Although we think this new representation of spin-networks is valuable in its own right, it is worth highlighting the motivating physical application of these results: the theory of loop quantum gravity (LQG). This theory describes quantum states of space as spin-networks, which can be interpreted as Yutsis diagrams. The covariant formulation of LQG (spin-foams) requires similar diagrams, but associated to representations of $SL_2(\mathbb{C})$ instead of SU(2). The hope is that the ZXH-calculus could ultimately provide a way to calculate highly complex quantum gravity computations on quantum computers [25–27].

We present the theory of Yutsis diagrams, Penrose notation and spin-networks in Section 2. Then in Section 3 we give a brief overview of the ZXH-calculus. Our main contributions are presented in Section 4 and we end with some concluding remarks in Section 5.

2 From Yutsis diagrams to spin-networks

What mathematicians call the representation theory of SU(2), physicists call the quantum theory of angular momentum. This latter theory emerged with the very first forays into quantum mechanics as it proved essential in understanding how electrons orbit nuclei. In 1960, Yutsis, Levinson and Vanagas\(^1\) published Mathematical Apparatus of the Theory of Angular Momentum [2], which introduced a graphical calculus that we review in this section.

Soon after this work, Penrose introduced another graphical calculus which worked for general abstract tensor systems [3]. Interestingly, Yutsis diagrams can be embedded into a specific case of Penrose’s calculus, which is called the binor calculus. The binor calculus was used by Penrose to propose a hand-made model of quantum space-time that are called spin-networks [28].

In this section, we present briefly the three frameworks and show how they relate to one another. To make the exposure self-contained, we start with a brief review of SU(2) representation and recoupling theories. For a more detailed overview, see [29].

2.1 SU(2) representation and recoupling theory

SU(2) is the 3-dimensional group of $2 \times 2$ unitary matrices with determinant 1. A unitary representation of SU(2) is a group homomorphism $SU(2) \to U(\mathcal{H})$ where $U(\mathcal{H})$ denotes the set of unitaries on a (finite-dimensional) Hilbert space $\mathcal{H}$. Any such representation can be decomposed as a direct sum of irreducible representations (irreps). It turns out that there is a unique irrep of any integer dimension. We label these irreps by half-integers $j \in \mathbb{N}/2$ dubbed spins. The spin-$j$ irrep has dimension $2j + 1$ and is unique up

---

\(^1\)Their book was originally published in Russian, but an English version was released in 1962. Their name is here translated from Russian, while they are actually Lithuanian, from which language their name is sometimes translated as Jucys, Levinsonas and Vanagas.
to isomorphism. There are several ways to actually write down these irreps. Here he will present one that is particularly useful to us.

The irreps of SU(2). The fundamental irrep (the spin-1/2 representation) SU(2) → U(C^2) is simply the 2 × 2 matrix multiplication of SU(2) over C^2. We can then define the higher-spin irreps by using ‘symmetrised copies’ of this representation. First, define the symmetrisation projection \( S_{2j} \) as the endomorphism over \( (C^2)^\otimes 2j \) such that

\[
S_{2j}(v_1 \otimes \cdots \otimes v_{2j}) = \frac{1}{(2j)!} \sum_{\sigma \in \mathfrak{S}_{2j}} U_\sigma (v_1 \otimes \cdots \otimes v_{2j}).
\]

Here \( \mathfrak{S}_{2j} \) is the \( 2j \)-element permutation group and the \( U_\sigma \) is the permutation unitary acting as

\[
U_\sigma (v_1 \otimes \cdots \otimes v_{2j}) = v_{\sigma(1)} \otimes \cdots \otimes v_{\sigma(2j)}.
\]

The action of \( S \) is to send every vector to a symmetric vector that is invariant under the action of the permutation unitaries. It is straightforward to check that \( S \) is indeed a projector (i.e. self-adjoint and idempotent). It then defines a subspace of \( (C^2)^\otimes 2j \) that we denote by \( \mathcal{H}_j \). This Hilbert-space consists of the symmetric vectors and has dimension \( 2j + 1 \). Indeed, let’s denote the canonical basis of \( C^2 \) by

\[
|0\rangle \overset{\text{def}}{=} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \overset{\text{def}}{=} \begin{pmatrix} 0 \\ 1 \end{pmatrix}.
\]

Then we can write the canonical orthonormal basis of \( \mathcal{H}_j \) as

\[
|j; m\rangle \overset{\text{def}}{=} \sqrt{\frac{(2j)!}{(j + m)!(j - m)!}} S_{2j} \left( |0\rangle \otimes \cdots \otimes |1\rangle \right),
\]

with \( m \in \{-j, -j + 1, \ldots, j - 1, j\} \). Note that here we are using the physicists notation where \( |j; m\rangle \) isn’t a tensor product of two spaces but the characterisation of a single spin state by its intrinsic angular momentum \( j \) (which here just denotes we are working in the Hilbert space \( \mathcal{H}_j \)) and it’s azimuthal component \( m \) (which is labelling the actual basis vector).

As an example, the magnetic basis for spin-\( \frac{3}{2} \) is

\[
\left\{ \begin{pmatrix} 3 \atop 2 \atop \frac{3}{2} \end{pmatrix}, \begin{pmatrix} 3 \atop 2 \atop \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 3 \atop 2 \atop -\frac{1}{2} \end{pmatrix}, \begin{pmatrix} 3 \atop 2 \atop -\frac{3}{2} \end{pmatrix} \right\}.
\]

Written in terms of a qubit basis this is

\[
\left\{ |000\rangle, \frac{1}{\sqrt{3}} (|001\rangle + |010\rangle + |100\rangle), \frac{1}{\sqrt{3}} (|011\rangle + |101\rangle + |110\rangle), |111\rangle \right\}.
\]

Later, when we will be dealing with direct-sums of spin-spaces, the \( j \) component denotes in which Hilbert space the vector is taken to be.

We can define an action of a group element \( U \in SU(2) \) on \( \mathcal{H}_j \) by linearity via

\[
U \cdot (v_1 \otimes \cdots \otimes v_{2j}) = (U v_1) \otimes \cdots \otimes (U v_{2j}).
\]
where \( Uv_j \) is the regular matrix multiplication of the \( 2 \times 2 \) matrix \( U \) with the \( \mathbb{C}^2 \) vector \( v_j \). The corresponding action of a Lie algebra element, \( a \in \text{su}(2) \), is then given as

\[
a \cdot (v_1 \otimes ... \otimes v_{2j}) = \sum_{k=1}^{2j} v_1 \otimes ... \otimes (av_k) \otimes ... \otimes v_{2j}.
\]

(8)

The irreps are the fundamental building blocks from which other representations are built. Indeed any finite-dimensional representation of \( SU(2) \) is completely reducible, i.e. it can be written as a direct sum of irreps. In particular, a tensor product of irreps can be decomposed into a direct sum of irreps, meaning there exists a bijective intertwiner (also known as an equivariant map) that maps the tensor product to a direct sum of irreps. It is the goal of recoupling theory to study the space of such intertwiners.

**Intertwiners.** Given two representations \( V \) and \( W \) of a group \( G \), a linear map \( \iota : V \rightarrow W \) is an intertwiner when it commutes with the group action, meaning that

\[
\iota(g \cdot v) = g \cdot \iota(v)
\]

(9)

for all \( g \in G \) and \( v \in V \). It’s easy to see that the set of intertwiners forms a vector space. We denote this vector space by \( \text{Hom}_G(V,W) \). This space \( \text{Hom}_G(V,W) \) is isomorphic to \( \text{Inv}_G(V \otimes W^*) \), where \( W^* \) is the dual vector space of \( W \) with the dual representation \( (\rho^*(g) = \rho(g^{-1})^T) \), and we define the invariant subspace of any representation on \( \mathcal{H} \) as

\[
\text{Inv}_G(\mathcal{H}) \overset{\text{def}}{=} \{ \psi \in \mathcal{H} | \forall g \in G, g \cdot \psi = \psi \}.
\]

(10)

Equivalently, we can characterise it by

\[
\text{Inv}_G(\mathcal{H}) = \{ \psi \in \mathcal{H} | \forall a \in g, a \cdot \psi = 0 \}.
\]

(11)

Physicists use the term “intertwiner” for any vector in \( \text{Inv}_G(\mathcal{H}) \).

**Clebsch-Gordan coefficients.** The most fundamental example of an intertwiner is the Clebsch-Gordan intertwiner. Given \( \mathcal{H}_{j_1} \) and \( \mathcal{H}_{j_2} \), their tensor product \( \mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \) can be decomposed into a direct sum of irreps with the following equivalence of representations

\[
\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \cong \bigoplus_{k=|j_1-j_2|}^{j_1+j_2} \mathcal{H}_k.
\]

(12)

There is a historically-preferred bijective intertwiner

\[
C : \mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \rightarrow \bigoplus_{j=|j_1-j_2|}^{j_1+j_2} \mathcal{H}_j,
\]

(13)

which is usually described by the coefficients of its matrix representation when using the canonical bases of the spaces involved. These coefficients are called the Clebsch-Gordan coefficients:

\[
C^{jm}_{j_1m_1j_2m_2} \overset{\text{def}}{=} \langle jm | C | j_1m_1; j_2m_2 \rangle.
\]

(14)

The indices can be read as tensor indices, with \( m \in \{-j,...,j\} \) labeling the elements of a canonical basis of \( \mathcal{H}_j \) (and similarly for \( m_1 \) and \( m_2 \)). For all the coefficients we have
$C_{j_1 m_1 j_2 m_2}^{j m} \in \mathbb{R}$. It turns out that if $m \neq m_1 + m_2$, then also $C_{j_1 m_1 j_2 m_2}^{j m} = 0$ Additionally, by convention we set $C_{j_1 m_1 j_2 m_2}^{j m} = 0$ when the Clebsch-Gordan conditions are not satisfied:

$$j_1 + j_2 + j \in \mathbb{N}$$

$$|j_1 - j_2| \leq j \leq j_1 + j_2.$$  \hspace{1cm} (15)

For all the non-zero coefficients there are analytic expressions, such as

$$C_{j_1 m_1 j_2 m_2}^{j m} = \delta_{m_1+m_2} \sqrt{2j+1} \times \frac{\sqrt{(j+m)!(j-m)!(j_1+j_2)!(j_1-j_2)!(j+j_1-j_2)!}}{(j+j_1+j_2+1)!(j_1+m_1)!(j_1-m_1)!(j_2+m_2)!(j_2-m_2)!} \times \sum_k (-1)^{k+j_2+m_2} (j+j_2+m_1-k)!(j_1-m_1+k)!(j-j_1-j_2-k)! (j+m-k)!(k+j_1-j_2-m)!.$$  \hspace{1cm} (16)

**Wigner’s 3jm-symbol.** The description of the Clebsch-Gordan coefficients has an asymmetry between the Hilbert spaces being tensored, and the Hilbert spaces it is being decomposed into. If we describe the situation in terms of a single state composed of three Hilbert spaces rather than privileging two spaces as inputs of a map to the third the situation becomes more symmetric.

Suppose the Clebsch-Gordan conditions (15) are satisfied (for $j = j_3$), then

$$\dim \text{Inv}_{SU(2)}(\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \otimes \mathcal{H}_{j_3}) = 1.$$  \hspace{1cm} (17)

Hence, there is a unique unit vector in this subspace, up to phase. We can write this vector as

$$|j_1, j_2, j_3 \rangle = \sum_{m_1, m_2, m_3} (\begin{array}{ccc} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{array}) |j_1 m_1; j_2 m_2; j_3 m_3 \rangle,$$  \hspace{1cm} (18)

where

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \overset{\text{def}}{=} \frac{(-1)^{j_1-j_2-m_3}}{\sqrt{2j_3+1}} C_{j_3 m_3}^{j_1 j_2 m_2}. $$  \hspace{1cm} (19)

This collection of coefficients is called Wigner’s 3jm-symbol. These symbols have more symmetries than the Clebsch-Gordan coefficients because they treat the three Hilbert spaces on the same level. This is useful for later graphical representations.

**Wigner’s 4jm-symbol** Instead of recoupling three Hilbert spaces, we can also think about recoupling four Hilbert spaces (which corresponds to decomposing a tensor product of three Hilbert spaces). It can be shown that

$$\text{Inv}_{SU(2)} \left( \bigotimes_{i=1}^{4} \mathcal{H}_{j_i} \right) \cong \mathbb{C}^{j_{\text{max}} - j_{\text{min}}}.$$  \hspace{1cm} (20)

with $j_{\text{min}} \overset{\text{def}}{=} \max(|j_1 - j_2|, |j_3 - j_1|)$ and $j_{\text{max}} \overset{\text{def}}{=} \min(j_1 + j_2, j_3 + j_4)$. Then a possible orthogonal basis is labelled by $j \in \{j_{\text{min}}, j_{\text{min}} + 1, \ldots, j_{\text{max}} - 1, j_{\text{max}}\}$ and denoted by

$$|j_1, j_2, j_3, j_4, j \rangle \overset{\text{def}}{=} \sum_{m_1, m_2, m_3, m_4} (\begin{array}{cccc} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{array}) |j_1, m_1 \rangle \otimes |j_2, m_2 \rangle \otimes \cdots \otimes |j_4, m_4 \rangle$$  \hspace{1cm} (21)
where
\[
\begin{pmatrix}
j_1 & j_2 & j_3 & j_4 \\
m_1 & m_2 & m_3 & m_4 \\
\end{pmatrix}^{(j)} = \sum_{m} (-1)^{j-m} \begin{pmatrix}
j_1 & j_2 & j \\
m_1 & m_2 & m \\
\end{pmatrix} \begin{pmatrix}
 j_3 & j_4 \\
- m_3 & m_4 \\
\end{pmatrix}. \tag{22}
\]
This set of coefficients is called Wigner’s 4jm-symbol. While this definition looks a bit arbitrary, it is actually part of a canonical pattern that becomes visible once we introduce Yutsis and Penrose diagrams in the next sections.

**Wigner’s 6j-symbol.** The partition in equation (22) into \( j_1 \) and \( j_2 \) in the first 3jm-symbol and \( j_3 \) and \( j_4 \) in the second is somewhat arbitrary. Indeed, another basis of (20) can be obtained by exchanging the labels \( j_2 \) and \( j_3 \) in (22). These two possible choices define two alternative bases which are related by a matrix whose coefficients are given by the 6j-symbol:
\[
\begin{pmatrix}
j_1 & j_2 & j_3 \\
j_4 & j_5 & j_6 \\
\end{pmatrix}^{(j)} = \sum_{m_1, \ldots, m_6} (-1)^{\sum_{i=1}^{6} (j_i - m_i)} 
\times \begin{pmatrix}
j_1 & j_2 & j_3 \\
-m_1 & -m_2 & -m_3 \\
\end{pmatrix} \begin{pmatrix}
j_1 & j_5 & j_6 \\
m_1 & -m_5 & m_6 \\
\end{pmatrix} \times \begin{pmatrix}
j_4 & j_2 & j_6 \\
m_4 & m_2 & -m_6 \\
\end{pmatrix} \begin{pmatrix}
j_3 & j_4 & j_5 \\
m_3 & -m_4 & m_5 \\
\end{pmatrix}. \tag{23}
\]
In terms of category theory, the 6j-symbol gives the component of an associator between \( (\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2}) \otimes \mathcal{H}_{j_3} \) and \( \mathcal{H}_{j_1} \otimes (\mathcal{H}_{j_2} \otimes \mathcal{H}_{j_3}) \). It relates different ways to construct the tensor product. It carries structural information about how various irreps relate to one another, without depending on the specific basis chosen within \( \mathcal{H}_j \). As such it is often described as an **invariant**. This invariant object and others that can be defined similarly emerge as core mathematical objects in, for instance, Loop Quantum Gravity.

### 2.2 Yutsis diagrams

The recoupling theory of SU(2) can be represented graphically in a simple manner using **Yutsis diagrams**. There exist various conventions in the literature and we have chosen to stick here to the original conventions introduced by Yutsis in 1960 [2].

**Constructing Yutsis diagrams** The basic generator of Yutsis diagrams is the 3-valent symbol:
\[
\begin{pmatrix}
j_1 & j_2 & j_3 \\
m_1 & m_2 & m_3 \\
\end{pmatrix} = \frac{j_1 m_1}{j_2 m_2} \cdot \frac{j_3 m_3}{j_1 m_1} \cdot \frac{j_2 m_2}{j_3 m_3}. \tag{24}
\]
The signs \(+/−\) on the nodes indicate the direction (anticlockwise/clockwise) in which the spins must be read. To cut down on notation we fix the default orientation to be clockwise (minus).
Only the connectivity of Yutsis diagrams matter, which means that all topological deformations are allowed.

\[
\begin{align*}
j_1 m_1 & \quad j_2 m_2 \quad j_3 m_3 \\
j_{1 m_1} & = j_{2 m_2} \quad j_{3 m_3} \\
j_{1 m_1} & = j_{3 m_3} \quad j_{2 m_2}
\end{align*}
\]

(25)

These symmetries are also known as planar isotopy. These also hold for all other diagrams that we shall construct later.

As for the arrows on the wires, we declare that an ingoing orientation corresponds to negating the magnetic index. For instance

\[
\begin{align*}
j_1 m_1 & \quad j_2 m_2 \quad j_3 m_3 \\
j_1 & = (j_1 - m_1 m_2 m_3).
\end{align*}
\]

(26)

An important special case is when one of the wires has spin 0. The spin-0 strand is then represented with a dashed line (no arrow needed):

\[
\begin{align*}
j_1 m_1 & \quad j_3 m_3 \\
(-1)^j & = \frac{1}{\sqrt{2j_1 + 1}} \delta_{m_1, -m_3} \delta_{j_1, j_3}.
\end{align*}
\]

(27)

In this notation, we can define graphically two basic operations of algebra: multiplication and summation. Multiplication is implemented simply by juxtaposition of diagrams:

\[
\begin{align*}
j_1 m_1 & \quad j_2 m_2 \quad j_3 m_3 \\
j_4 m_4 & \quad j_5 m_5 \quad j_6 m_6 \\
j_1 & = \sum_{m=-j}^{j} \frac{(-1)^j + m}{\sqrt{2j_1 + 1}} \delta_{m_1, -m_3} \delta_{j_1, j_3}.
\end{align*}
\]

(28)

Summation is implemented by connecting wires together. The gluing of wires with the same label jm defines the sum over m (from −j to j), with an additional factor \((-1)^j m\) in the summand, like:

\[
\begin{align*}
j_1 m_1 & \quad j_2 m_2 \quad j_3 m_3 \quad j_4 m_4 \\
j_1 m_1 & = \sum_{m=-j}^{j} (-1)^j m
\end{align*}
\]

(29)

We call wires with both ends attached to a vertex ‘internal’ and the others ‘external’. We only allow wires to be glued together when the direction of the arrows match up, i.e. one
wire has to have an outgoing wire, and one has to have an ingoing wire. We recognise the right-hand side of Eq. (29) as the definition of the $4jm$-symbol:

$$j_{1m_1} j_{2m_2} j_{3m_3} j_{4m_4} = \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(j)}$$  \hfill (30)

Reversing the arrow of an internal wire gives an overall phase:

$$j_{1m_1} j_{2m_2} j_{3m_3} j_{4m_4} = (-1)^{2j} j_{1m_1} j_{2m_2} j_{3m_3} j_{4m_4}.$$  \hfill (31)

For convenience it is also helpful to consider a single wire with an arrow on it as an element of the graphical calculus:

$$j_{1m_1} j_{2m_2} = (-1)^{j_2 - m_2} \delta_{j_1 j_2} \delta_{m_1 m_2} \quad \text{or} \quad j_{m} j_{j} = (-1)^{j-n} \delta_{mn}.$$  \hfill (32)

The phase $(-1)^{j-n}$, which may seem unexpected, is necessary to guarantee that the composition of two single wires is still a single wire.

**Invariant functions.** In a Yutsis diagram, every external wire represents a magnetic tensor index. Such a diagram then represents a tensor with a number of indices equal to the number of external wires. When a Yutsis diagram has no external wires, it encodes a number (instead of a tensor). In this case all magnetic indices are summed over, so that it is only a function of the internal spins $j_i$. We call such expressions invariant functions. Let us give some examples.

The simplest example is the trace of a single wire:

$$j \quad \quad = \sum_m (-1)^{j-m} \quad \quad = \sum_m (-1)^{j-m} (-1)^{j-m} = 2j + 1.$$  \hfill (33)

Just slightly more complicated is the $\Theta$-graph:

$$j_1 j_2 j_3 = (-1)^{j_1+j_2+j_3}.$$  \hfill (34)
The Wigner $6j$-symbol corresponds to quite a canonical diagram:

\[
\begin{pmatrix}
  j_1 & j_2 & j_3 \\
  j_4 & j_5 & j_6
\end{pmatrix} = \frac{j_1}{j_2} \frac{j_4}{j_5} \frac{j_4}{j_6} = (35)
\]

We can define other invariant functions in the same spirit, like the $15j$-symbol:

\[
\begin{pmatrix}
  j_1 & j_2 & j_{11} \\
  j_4 & j_5 & j_{15} \\
  j_7 & j_3 & j_{14} \\
  j_9 & j_6 & j_{13} \\
  j_8 & j_{10} & j_{12}
\end{pmatrix} = \frac{j_1}{j_2} \frac{j_{11}}{j_{12}} \frac{j_{12}}{j_{13}} \frac{j_{13}}{j_{14}} \frac{j_{14}}{j_{15}} = (36)
\]

which is the definition used by [30]. Note that the crossings of wires (over or under) do not have any significance here (i.e. we don’t care about the braiding). In contrast to the $6j$-symbol, there is no consensus on which convention should be used to define the $15j$-symbol. In all cases it corresponds to an invariant function associated to a 3-valent graph with 15 links. There are five topologically distinct such graphs, and hence five different $15j$-symbols [2]. Here we see the power of Yutsis diagrams: it makes huge expressions much more tractable and straightforward to reason about.

**Matrix representation.** Generic Yutsis diagrams have external wires, i.e. they are tensors (and not scalars). They can thus be represented as matrices, but we must fix a convention to be able to read them appropriately.

1. The inbound arrows are treated as inputs, i.e. columns of the matrix. Dually, outbound arrows are outputs and correspond to rows.

2. Among all the inbound (resp. outbound) arrows, we must decide their ordering. We fix that the first one is the top left and the next ones follow with clockwise orientation.

3. Within each Hilbert space, we must decide the ordering of the canonical basis. We follow the natural ordering of the qubit basis. For $j = 1/2$, we have

\[
\left| \frac{1}{2}, \frac{1}{2} \right\rangle = |0\rangle \quad \quad \left| \frac{1}{2}, -\frac{1}{2} \right\rangle = |1\rangle
\]

Thus in general, it goes from $|j, j\rangle$ to $|j, -j\rangle$, i.e. in decreasing order of magnetic index $m$. 

---

10
For instance, consider the tensor
\[
\left( \begin{array}{ccc}
\frac{1}{2} & \frac{1}{2} & 1 \\
-m_1 & -m_2 & m_3
\end{array} \right)
\]
\[\text{(38)}\]
Its matrix representation is
\[
\frac{1}{2} \frac{1}{2} 1 \quad \rightarrow \quad \begin{pmatrix}
\frac{-1}{\sqrt{3}} & 0 & 0 \\
0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\
0 & 0 & 0 & -\frac{1}{\sqrt{3}}
\end{pmatrix}
\]
\[\text{(39)}\]
The top left component corresponds to \( m_1 = 1/2, \ m_2 = 1/2 \) and \( m_3 = 1 \). The second column corresponds to \( m_1 = 1/2 \) and \( m_2 = -1/2 \). If the arrow on \( j_3 \) is reverted, we get instead the tensor
\[
\left( \begin{array}{ccc}
\frac{1}{2} & \frac{1}{2} & 1 \\
-m_1 & -m_2 & m_3
\end{array} \right)
\]
\[\text{(40)}\]
with the matrix representation
\[
\frac{1}{2} \frac{1}{2} 1 \quad \rightarrow \quad \begin{pmatrix}
0 & 0 & -\frac{1}{\sqrt{3}} \\
0 & \frac{1}{\sqrt{6}} & 0 \\
0 & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & 0 & 0
\end{pmatrix}
\]
\[\text{(41)}\]
The first \(-\frac{1}{\sqrt{3}}\) corresponds to \( m_1 = 1/2, m_2 = 1/2 \) and \( m_3 = -1 \).

### 2.3 Penrose diagrams

In Section 2.1, we saw that the spin-\( j \) irrep can be built from the symmetrisation of \( 2j \) copies of \( \mathbb{C}^2 \). This suggests another graphical calculus that goes under the name of \textit{Penrose binor calculus}. An introduction can be found in [31]. Here we will cover the basics.

In this calculus, the identity over \( \mathbb{C}^2 \) is a single strand
\[
\begin{array}{c}
\approx |0\rangle\langle 0| + |1\rangle\langle 1| .
\end{array}
\]
\[\text{(42)}\]
The free legs carry implicit labels of copies of \( \mathbb{C}^2 \). There is also a duality between up and down. The cap stands for
\[
\approx i \langle 01 | - i \langle 10 | .
\]
\[\text{(43)}\]
The cup is
\[
\approx i |01 \rangle - i |10 \rangle .
\]
\[\text{(44)}\]
With these definitions, the diagrams are then well-behaved under deformation:
\[
\approx \quad = \quad 
\]
\[\text{(45)}\]
Finally, the crossing is the regular swap, but with a global minus sign:
\[
= - |00\rangle\langle 00| - |10\rangle\langle 01| - |01\rangle\langle 10| - |11\rangle\langle 11|.
\]
\[\text{(46)}\]
These rules guarantee planar isotopy, i.e. diagrams can be continuously deformed while preserving their interpretation as linear maps.
**Fundamental equations.** Binor calculus has two core equations that can be deduced from the definitions above.

\[ \bigcirc = -2 \]  \hspace{1cm} (47)

\[ + \bigotimes + \bigotimes = 0 \]  \hspace{1cm} (48)

The value of the loop in the first equation secretly gives the “dimension” of the tensor calculus. Here it is \(-2\): the ”2” explains the name ”binor” and the minus sign explains the title of the original article ”negative dimensional tensors” by Penrose [3]. The second equation is known under the name of ”skein relation” or ”binor identity”.

**Anti-symmetriser.** To connect to Yutsis graphical calculus, we must build the spin-\( j \) irrep. This is done by anti-symmetrising the strands:

\[ j \rightarrow \overset{\text{def}}{\longrightarrow} \begin{array}{c} 2j \vspace{.25cm} \\
\vdots \\
0 \ldots 1 \\
\end{array} \overset{\text{def}}{=} \frac{1}{(2j)!} \sum_{\sigma \in S_{2j}} (-1)^{|\sigma|} \begin{array}{c} \ldots \\
\sigma \\
\end{array} \]  \hspace{1cm} (49)

where \(|\sigma|\) is the parity of \(\sigma\) and the \(\sigma\)-labelled box represents the corresponding permutation of the \(2j\) strands from one side to the other. Although this looks like an anti-symmetrisation (since we have the \((-1)^{|\sigma|}\) term), the operator is actually a projector from \(\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2\) to \(S_{2j}(\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2)\), because of the minus sign in (46). So we have

\[ \begin{array}{c} 2j \\
\vdots \\
0 \ldots 1 \\
\end{array} = S_{2j}(\begin{array}{c} 0 \ldots 1 \\
\vdots \\
0 \ldots 1 \\
\end{array}) = \sqrt{\frac{(j - m)! (j + m)!}{(2j)!}} |j; m\rangle. \]  \hspace{1cm} (50)

Taking the trace of the symmetrised space (i.e. making a loop) gives the value

\[ j \bigcirc = (-1)^{2j}(2j + 1), \]  \hspace{1cm} (51)

which is the dimension of the space up to a phase.

**Invariant tensors.** Now, let us draw in binor calculus, the analogue of the 3-valent vertex, i.e. the \(3jm\)-symbol. The essential property to be noticed is the invariance of the cup under the action of SU(2), i.e. for any \(u \in \text{SU}(2)\):

\[ \begin{array}{c} u \\
\vdots \\
u \\
\end{array} \equiv i u \cdot (|01\rangle - |10\rangle) = i (|01\rangle - |10\rangle) \equiv \]  \hspace{1cm} (52)
Then the diagram depicts a vector that belongs to $\text{Inv}_{\text{SU}(2)}(\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \otimes \mathcal{H}_{j_3})$, so it is proportional to $|j_1, j_2, j_3\rangle$. This should be interpreted as a kind of “railroad switch”, where the fundamental wires within the three symmetrised bundles redistribute between themselves. Because we are dealing with symmetrised spaces we only care about how many wires go from each bundle to the other bundle. It turns out that there is then actually only one way in which to connect the wires, when it is not impossible. The Clebsch-Gordan conditions (15) precisely state when such a recoupling is possible.

Plugging the vectors of eq. (50), one gets

\[
N(j_1, j_2, j_3)e^{i\phi(j_1,j_2,j_3)} \begin{pmatrix} j_1 \\ m_1 \\ j_2 \\ m_2 \\ j_3 \\ m_3 \end{pmatrix} 
\times \sqrt{\frac{(j_1 - m_1)!(j_1 + m_1)!(j_2 - m_2)!(j_2 + m_2)!(j_3 - m_3)!(j_3 + m_3)!}{(2j_1)!(2j_2)!(2j_3)!}}. \quad (54)
\]

The functions $N$ and $e^{i\phi}$ can be determined by computing a special case for an easy choice of $m_i$, like $m_1 = j_1, m_2 = j_3 - j_1$ and $m_3 = -j_3$. Then, the diagram is

\[
\begin{align*}
&= i^{j_1 + j_2 + j_3} (-j_1 + j_2 + j_3)!(j_1 + j_2 - j_3)! \\
&\times \frac{1}{(2j_2)!}. \quad (55)
\end{align*}
\]
while using eq. (16), the RHS gives

\[ N(j_1, j_2, j_3) e^{i\phi(j_1, j_2, j_3)} \sqrt{\frac{(-j_1 + j_2 + j_3)!(j_1 + j_2 - j_3)!}{(2j_2)!}} \times (-1)^{j_1 - j_2 + j_3} \sqrt{\frac{(2j_1)!(2j_3)!}{(j_1 + j_2 + j_3 + 1)!(j_1 - j_2 + j_3)!}} \]

Thus,

\[ \sqrt{\frac{(j_1 + j_2 + j_3 + 1)!(j_1 - j_2 + j_3)!(j_1 - j_2 + j_3)!}{(2j_1)!(2j_2)!(2j_3)!}} \]

and

\[ e^{i\phi(j_1, j_2, j_3)} = (-1)^{j_1 - j_2 + j_3}. \]

All the invariant functions that were introduced in the previous section can now be translated to binor diagrams. We can then use the two core equations (47) and (48), which are sufficient to simplify the diagrams and compute their actual value. We will often refer to the value \( N(j_1, j_2, j_3) \) as the binor coefficient.

### 2.4 Spin-networks

Spin-networks were originally conceived by Penrose as a means of deriving continuous space-time from graphs coloured by discrete spins [28]. They were initially formulated in terms of the binor calculus. In 1995, Rovelli and Smolin discovered a generalisation of the original spin-networks and used them as a label for quantum states of space [32].

A 3-valent spin-network is an open graph where each node has 3 associated links and where each link is associated with a spin \( j_i \in \mathbb{N}/2 \), such that the three spins of the links coming together in a node satisfy the Clebsch-Gordan conditions. An example of a 3-valent spin-network is shown in Figure 1. Such a diagram represents a tensor in binor calculus. It can also be interpreted as a Yutsis diagram, provided that arrows are assigned to the links.

Similarly, a 4-valent spin-network is an open graph where each node has 4 associated links and where each link \( l \) is labelled by a spin \( j_l \) such that for each set of four spins \( j_1, j_2, j_3 \) and \( j_4 \) coming together in a node we have:

\[ \text{Inv}_{SU(2)}(H_{j_1} \otimes H_{j_2} \otimes H_{j_3} \otimes H_{j_4}) \neq 0. \]

This definition easily generalises to higher valencies, and in the case of 3-valent spin-networks we get back the definition above. In the 4-valent case, the invariant subspace can be more than one-dimensional. This dimension can be computed graphically by the number of ways to connect the four links at a node when viewed as a binor diagram.

---

2Mathematicians usually say *edge* or *arrow*, but not “link”, which has another meaning in knot theory. The terminology of LQG keeps “edge” for spin-foams, and uses “link” for spin-networks.
3 The ZXH-calculus

The ZX-calculus is a universal, sound and complete graphical language for qubit systems. It has found use in quantum computation and quantum foundations, for instance in the fields of quantum circuit optimisation, verification and simulation, quantum error correction, and measurement-based quantum computation \cite{9-11, 22, 23}. The advantage of the ZX-calculus is that it allows a graphical representation of (even non-unitary) linear maps, without resorting to dealing with the matrices directly. This allows one to do computations graphically.

There are several variations on the ZX-calculus that differ in which generators they view as fundamental. One of these variations is the ZH-calculus \cite{33, 34}. Since these variations are all universal, they can each be translated into one another. For this reason, we will use generators from both the ZX- and ZH-calculus. We call the resulting graphical language the \textit{ZXH-calculus} \cite{24}.

Let us give a brief overview of the ZX- and ZH-calculi. For an in-depth review see Ref. \cite{35} and for a book-length introduction see Ref. \cite{36}.

\textbf{The ZX-calculus.} The ZX-calculus is a diagrammatic language similar to quantum circuit notation \cite{5, 6}. A ZX-diagram consists of \textit{wires} and \textit{spiders}. Wires entering the diagram from the left are \textit{inputs}; wires exiting to the right are \textit{outputs}. A wire represents a \textit{qubit}, i.e. a two-dimensional Hilbert space, i.e. a spin-$1/2$ degree-of-freedom. Given two diagrams we can compose them by joining the outputs of the first to the inputs of the second, or form their tensor product by stacking the two diagrams.
Spiders are linear operations which can have any number of input or output wires. There are two varieties: Z-spiders depicted as green dots and X-spiders depicted as red dots, each of which can be labelled by a phase \( \alpha \in \mathbb{R} \):

\[
\begin{align*}
\text{Z-spiders:} & \quad |0\cdots0\rangle|0\cdots0\rangle + e^{i\alpha}|1\cdots1\rangle|1\cdots1\rangle \\
\text{X-spiders:} & \quad |+\cdots+\rangle|+\cdots+\rangle + e^{i\alpha}|−\cdots−\rangle|−\cdots−\rangle
\end{align*}
\]

(60) (61)

ZX-diagrams are constructed iteratively from these spiders by composition and juxtaposition. As a special case, diagrams with no inputs represent (unnormalised) state preparations, while diagrams with no open wires represent complex scalars.

As a demonstration, let us write down some simple state preparations and unitaries in the ZX-calculus:

\[
\begin{align*}
\text{Z-spiders:} & \quad |0\rangle + |1\rangle = \sqrt{2}|+\rangle \\
\text{X-spiders:} & \quad |0\rangle|0\rangle + e^{i\alpha}|1\rangle|1\rangle = Z_\alpha \\
& \quad |+\rangle|+\rangle + e^{i\alpha}|--\rangle|--\rangle = X_\alpha
\end{align*}
\]

(62) (63) (64) (65)

Note that, while (64) and (65) have a label \( \alpha \), we have not given a label to the state preparations (62) and (63). By convention, a spider without a label is taken to have a label \( \alpha = 0 \). When we take \( \alpha = \pi \) in (64) and (65) we get Pauli matrices:

\[
\begin{align*}
\text{Z-spiders:} & \quad |0\rangle + |1\rangle = Z \\
\text{X-spiders:} & \quad |0\rangle|0\rangle + e^{i\pi}|1\rangle|1\rangle = X
\end{align*}
\]

(66)

By composing spiders we can make more complicated linear maps, such as the CNOT gate:

\[
\begin{align*}
\text{CNOT:} & \quad = \frac{1}{\sqrt{2}} \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix} \propto \text{CNOT}
\end{align*}
\]

(67)

We can treat a ZX-diagram as a graphical depiction of a tensor network, similar in style to the work of Penrose (see Section 2.3). In this interpretation, a wire between two spiders denotes a tensor contraction. As tensors, Z- and X-spiders can be written as follows:

\[
\begin{align*}
\text{Z-spiders:} & \quad (\begin{array}{c}
\alpha \\
i_1
\end{array})_{i_1\cdots i_m} = \begin{cases}
1 & \text{if } i_1 = \cdots = i_m = j_1 = \cdots = j_n = 0 \\
e^{i\alpha} & \text{if } i_1 = \cdots = i_m = j_1 = \cdots = j_n = 1 \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

(68)

\[
\begin{align*}
\text{X-spiders:} & \quad (\begin{array}{c}
\alpha \\
i_1
\end{array})_{i_1\cdots i_m} = (\frac{1}{\sqrt{2}})^{n+m} \cdot \begin{cases}
1 + e^{i\alpha} & \text{if } \bigoplus_\alpha i_\alpha \oplus \bigoplus_\beta j_\beta = 0 \\
1 - e^{i\alpha} & \text{if } \bigoplus_\alpha i_\alpha \oplus \bigoplus_\beta j_\beta = 1
\end{cases}
\end{align*}
\]

(69)

where \( i_\alpha, j_\beta \) range over \( \{0, 1\} \) and \( \oplus \) is addition modulo 2.

\(^3\text{Note that if you are reading this document in monochrome or otherwise have difficulty distinguishing green and red, Z-spiders will appear lightly-shaded and X-spiders darkly-shaded.}\)
ZX-diagrams have a number of symmetries that make them easy to work with. In particular, we can treat a ZX-diagram as an undirected (multi-)graph\(^4\), so that we can move the vertices around in the plane, bending, unbending, crossing, and uncrossing wires, as long as the connectivity and the order of the inputs and outputs is maintained. These deformations of the diagram do not affect the linear map it represents. Indeed, the reader might have noticed that in the CNOT diagram (67) we drew a vertical wire without explaining whether this denotes an input or an output from the Z- and X-spider. We are warranted in drawing it this way because:

\[
\begin{array}{c}
\tau = \\
\end{array}
\]

Besides these topological symmetries, ZX-diagrams have a set of rewrite rules associated to them, collectively referred to as the ZX-calculus. See Figure 2 for a set of these rules. As a small demonstration of these rewrite rules, let us prove diagrammatically that the CNOT diagram (67) indeed acts like the CNOT. The computational basis states are given by the following diagrams.

\[
|00\rangle = \frac{1}{2} \quad |01\rangle = \frac{1}{2} \quad |10\rangle = \frac{1}{2} \quad |11\rangle = \frac{1}{2}
\]

Then we can check that the diagram has the correct action on these basis states:

\[
\begin{array}{c}
\tau = \\
\end{array}
\]

\^[4]A multi-graph is a graph where there can be multiple edges between the same vertices.
The **ZH-calculus.** ZX-diagrams were introduced over a decade ago [5] and have proven useful for reasoning about Clifford computation and single-qubit phase rotation gates [12, 23, 37]. It is however harder to reason about certain logical constructions, in particular the AND operation $|x⟩ \otimes |y⟩ \mapsto |x \cdot y⟩$. For instance, the only way to represent a CCNOT gate (also commonly known as the Toffoli gate) in the ZX-calculus is to expand it into Clifford and phase gates. In 2018, a new graphical calculus was introduced to remedy this problem: the **ZH-calculus** [33]. This calculus adds another generator to the ZX-calculus that allows for a compact representation of an AND gate. This new generator is the \(H\)-box:

$$m \left\{ \begin{array}{c} \uparrow \\ \downarrow \\ \downarrow \end{array} \right\} n \overset{\text{def}}{=} \sum a^{i_1 \ldots i_m j_1 \ldots j_n} |i_1 \ldots j_n⟩ ⟨i_1 \ldots j_n| \quad (73)$$

with \(a \in \mathbb{C}\) and the sum in this equation going over all \(i_1,\ldots,i_m,j_1,\ldots,j_n \in \{0,1\}\) so that an H-box represents a matrix where all entries are equal to 1, except for the bottom right element, which is \(a\). As a tensor we can write

$$(a)_{i_1 \ldots i_m}^{j_1 \ldots j_n} = \begin{cases} a & \text{if } i_1 = \ldots = i_m = j_1 = \ldots = j_n = 1 \\ 1 & \text{otherwise} \end{cases} \quad (74)$$

Whereas for spiders we only draw the phase on the spider when it is nonzero, for H-boxes we only draw the label when it is not equal to \(-1\). This is because the 1-input, 1-output H-box with a phase of \(-1\) corresponds to the familiar Hadamard gate (up to a global scalar):

$$= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (75)$$

Note that in this paper we only need H-boxes labelled by \(-1\). We give the general definition for completeness’ sake. We have the following relations among the three generators, Z-spiders, X-spiders and H-boxes:

$$m \left\{ \begin{array}{c} \uparrow \\ \uparrow \end{array} \right\} n = \sqrt{2}^{n+m} \quad m \left\{ \begin{array}{c} \uparrow \\ \downarrow \end{array} \right\} n \quad (76)$$

$$m \left\{ \begin{array}{c} \downarrow \\ \downarrow \end{array} \right\} n = \sqrt{2}^{n+m} \quad m \left\{ \begin{array}{c} \downarrow \\ \uparrow \end{array} \right\} n \quad (77)$$

$$\alpha \overset{\text{def}}{=} \frac{\alpha}{\sqrt{2}} \quad (78)$$

Note that it is also possible to represent H-boxes of higher arity, using just Z- and X-spiders, but this is quite involved and not necessary for our purposes [38].

In addition to the rules of the ZX-calculus of Figure 2 and the relations among the generators (76)–(79) we have some further rules involving H-boxes: see Figure 3.

We present in Appendix A a condensed overview of all the rewrite rules and relations we have introduced so far.

An H-box with zero input and output wires that is labelled by \(a\) is equal to the scalar \(a\). This means we can always translate the scalars in the hybrid notation of Figures 2
Figure 3: The rules of the ZH-calculus. These rules hold for all $a, b \in \mathbb{C}$. Note ‘...’ should be read as ‘0 or more’. The right-hand side of (in) and (hb) and the left-hand side of (m) contain fully connected bipartite graphs. In this paper we will only need the rules in the left column. The rest are shown for completeness. The rule names stand for (hh)-cancellation, remove wire, (ex)plode, (ab)sorb, (hf)use, (hc)opy, (m)ultiplication, (av)erage and (in)roduction (as it introduces additional wires to the H-box on the left-hand side).

and 3 into a ZH-diagram. For instance, the self-inverseness of the Hadamard gate can be represented as follows:

\[ \begin{array}{c}
\text{(hh)} \\
\text{= 2}
\end{array} \]

ZH-diagrams are universal, meaning that any linear map between complex vector spaces of dimension $2^n$ can be represented as a ZH-diagram. Furthermore, the ZH-calculus is complete, meaning that if two diagrams represent the same linear map, then we can find a sequence of rewrites from Figures 2 and 3 and equations (76)–(79) that transforms one diagram into the other [33, 34]. However, in general, such a sequence of rewrites will involve diagrams of size exponential in the number of inputs and outputs\(^5\). The key to working with ZH-diagrams efficiently is then to find good heuristics for simplifying diagrams.

H-boxes allow us to straightforwardly represent controlled-phase gates. For instance, a CCZ($\theta$) gate, i.e. a gate that maps the computational basis state $|xyz\rangle$ to $e^{i\theta xyz} |xyz\rangle$ is given by:

\[ e^{i\theta} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta}
\end{pmatrix} \]

\[ (81) \]

\(^5\)As otherwise we could establish efficient classical simulation of quantum computation, among other unlikely consequences such as $P=NP$. 

19
As a special case of (81) we also have the standard controlled-Z (CZ) gate:

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & -1
\end{pmatrix}
\]  

(82)

As another variation on these diagrams, we have the following diagram that we will use different iterations on throughout this paper:

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

(83)

i.e. this linear map throws away a |11⟩ input, but otherwise acts as the identity.

As another variation on (81) we can represent the CCNOT gate as follows:

\[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

(84)

The ZXH-calculus. Those familiar with the ZX-calculus or the ZH-calculus might have noticed that they have conflicting definitions of the X-spider and the 2-ary H-box, resulting in different scalar factors of \(\sqrt{2}\). In this paper we use the conventions also used in PyZX [39] in order to aid in our calculations. This means that our Z- and X-spider are defined as is usual in the ZX-calculus. However, most literature on the ZX-calculus also includes a yellow box to represent the Hadamard gate. In our case, we use the convention of the ZX-calculus that such a box represents an unnormalised Hadamard gate (cf. (75)). Hence, certain scalar factors will be different than is usual in the literature on the ZX-calculus. Conversely, our H-box and Z-spider match the definition used in the ZH-calculus, but our X-spider does not match the corresponding definition in the ZH-calculus, and is off by certain factors of \(\sqrt{2}\). It is unfortunately not possible to have a fully satisfactory convention when it comes to scalar factors in the ZX/ZH-calculus, and choices have to be made about where scalar corrections appear (more on this can be found here [40]). In order to prevent confusion about these clashing scalar conventions, we will refer to our version of the ZX and ZH calculus as the ZXH-calculus throughout the paper.

4 Translating spin-networks into ZXH-diagrams

As the ZXH-calculus is a universal language for spin-1/2 systems, it does not come as a surprise that it can be used to express the quantum theory of angular momentum. However, the precise way to do this is not obvious, and takes quite some work. In this section we present this translation of the theory of angular momentum to the ZXH-calculus. First we introduce the building blocks, i.e. the links and the 3-valent nodes. Then we show how to connect them and compute invariant recoupling functions.
4.1 Links

Spin-1/2. The simplest Yutsis diagrams have only spin-1/2 links. From equation (32), we see that these links can be written as:

\[
\frac{1}{2} \left| \begin{array}{c}
0 \\
1
\end{array} \right| = \left( \begin{array}{cc}
1 & 0 \\
0 & -1
\end{array} \right) = |0\rangle\langle 0| - |1\rangle\langle 1| = -\frac{\pi}{2}.
\] (85)

Spin-1 links [24] In Section 2.3, we have already seen that the spin-j Yutsis links can be expressed as the symmetrisation of \(2j\) fundamental Penrose wires. So to represent higher spins in the ZXH-calculus, we just need to express the symmetriser of Eq. (1) as a ZXH-diagram.

A general method to do this is to make use of the controlled-SWAP gate. The symmetriser \(S_n\) is defined as a superposition of permutation unitaries \(U_\sigma\), with \(\sigma \in S_n\). Each \(U_\sigma\) can straightforwardly be written as a ZXH-diagram by just permuting the wires, but the superposition requires a controlled permutation operator. This approach to the diagrammatic creation of spin-spaces was first outlined in [24].

For the spin-1 link, it turns out to be sufficient to have controlled SWAP operators that have the control qubit post-selected into \(\langle + |\). This can be conveniently represented as a ZXH-diagram as follows:

We will refer to the right diagram here as a CSWAP in what follows. The top wire in this diagram is the control wire which determines whether the SWAP fires. It is post-selected onto the \(|+\rangle\) state in order to get rid of the control wire. By inputting a computational basis state we can verify that it indeed performs the maps required.

First, when the input is \(|0\rangle\):

\[
\begin{align*}
\left( \begin{array}{cc}
1 & 0 \\
0 & 0
\end{array} \right) &= \sqrt{2} \\
\left( \begin{array}{cc}
0 & 1 \\
1 & 0
\end{array} \right) &= \sqrt{2}
\end{align*}
\]

(87)

It might be surprising that the Yutsis diagram on the LHS looks asymmetric while the ZXH-diagram looks symmetric. The underlying matrix is indeed symmetric, but the arrow is here useful to abide by the convention and to write the Yutsis diagram as a matrix (see Matrix representation on page 10). On the ZXH diagram, this information is implicit and globally carried by the polarisation left/right of the plane.
And second, when the input is $|1\rangle$:

\[
\begin{align*}
\text{id} & = 2\sqrt{2} \\
\text{ho} & = 4 \\
\text{ab} & = 4 \\
\text{hh} & = 1
\end{align*}
\]

Now, to write the symmetrisation projector on $n = 2$ wires we need an equal superposition of the identity permutation and the SWAP. We accomplish this by setting the control of (86) to $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$:

\[
1 \quad = |00\rangle\langle 00| + |11\rangle\langle 11| - \frac{1}{2} (|01\rangle + |10\rangle) (\langle 01| + \langle 10|) = \frac{1}{\sqrt{2}}
\]

Here the Z $\pi$ rotation on each of the wires is necessary to add the $-1$ phase that is present in the Yutsis diagram. Note that these phases are not a part of the definition of the symmetriser, but are just a convention present in Yutsis diagrams. We will see such phases appear when representing higher-spin Yutsis wires as well.

**Higher-spin links** We construct the symmetriser for higher-spins using induction: if we have the symmetriser $S_n$, then the way we get $S_{n+1}$ is to compose $S_n$ with a coherent superposition of the identity and the SWAP gates from the $(n+1)$th qubit to every other qubit. We call this operator $T_n$:

\[
T_n \overset{\text{def}}{=} \text{id} + \text{SWAP}_{1,n+1} + \text{SWAP}_{2,n+1} + \cdots + \text{SWAP}_{n,n+1}.
\]

We construct this superposition as a ZXH-diagram by writing CSWAP gates from the $(n+1)$th qubit to each other qubit and then connecting all the control wires in such a way that at most one CSWAP ‘fires’ at the same time. Let us show a few examples and then explain the general procedure.

For $n = 3$:

\[
\frac{3}{2} \quad = \frac{1}{3\sqrt{2}}
\]

For $n = 4$:

\[
2 \quad = \frac{1}{48}
\]
And for $n = 5$:

\[
\pi = \frac{1}{7680}
\]

Note that each larger diagram contains the smaller diagram. To see how to generalise the construction to higher spins, there are two non-trivial parts that need to be understood: the global scalar of the diagram and the “crown” (the subdiagram above the circuit that controls the superposition of CSWAP gates).

The crown. The function of the crown is to produce a superposition of states that triggers one or none of the CSWAP gates, i.e. to produce the state

\[
|0\ldots0\rangle + |10\ldots0\rangle + \ldots |0\ldots01\rangle.
\]

This is the superposition of $n$ states of the canonical basis. For instance, for $n = 3$

\[
|0\rangle + |10\rangle + |01\rangle = 2(|00\rangle + |10\rangle + |01\rangle).
\]

By taking $k$ adjacent single-wire Z-spiders we get the equal superposition of all $2^k$ elements of the canonical basis

\[
|0\ldots0\rangle + |0\ldots01\rangle + |0\ldots010\rangle + |0\ldots011\rangle + \ldots + |1\ldots1\rangle.
\]

From this state, we can obtain the state we need (94) by projecting the ‘unwanted’ states to 0. This requires $2^k \geq n$. We choose the minimal value of $k$ that satisfies this inequality, i.e. $k = \lceil \log_2 n \rceil$. Then the crown is constructed as follows:

1. Add $k$ Z-spiders on top of the crown.
2. Add $2^k - 1$ groups of $k$ X-spiders, each with phases 0 or $\pi$, such that each group of spiders has a different sequence of phases 0 and $\pi$. Connect each group with the row above.
3. Add $2^k - 1$ H-boxes. Connect each H-box to the $k$ X-spiders of a single group.
4. To $n$ of the H-boxes, connect a Hadamard, i.e. an arity-2 H-box, and connect each of these to the control-wire of one of the CSWAPs. This connectivity makes it so that only when every input to the H-box is an $X(\pi)$ state (i.e. $|1\rangle$) will the output be a $|1\rangle$ as well, which triggers the swap. In all other cases they become a plain single wire red spider which will not trigger the swap and leave it as an identity:

\[
\frac{1}{\sqrt{2}} = 2|0\rangle
\]

\[
\frac{1}{\sqrt{2}} = 2|1\rangle.
\]
5. To the remaining $2^k - 1 - n$ H-boxes, connect a 0-phase Z-spider. This construction throws away the unnecessary states in the superposition: when all the inputs to the H-box are $|1\rangle$’s (i.e. X-spiders with π phase), the diagram evaluates to zero:

\[
\begin{align*}
\text{\includegraphics{diagram1.png}} &= \sqrt{2} \sqrt{\frac{\pi}{\sqrt{5}}} = 0 \\
\text{\includegraphics{diagram2.png}} &= \sqrt{2} \frac{\sqrt{\frac{\pi}{\sqrt{5}}}}{\sqrt{\sqrt{2}}} = \sqrt{2}^{\pi/7} \tag{98}
\end{align*}
\]

Let’s demonstrate this procedure with an example. The simplest case for which this construction works is for $n = 3$. We then set $k = 2$. Unfusing a single-arity Z-spider, we get a state $\left(|0\rangle + |1\rangle\right) \otimes \left(|0\rangle + |1\rangle\right)$ on top. Expanding this we get a sum of four diagrams, where the Z-spiders are replaced by spiders representing respectively $|00\rangle$, $|10\rangle$, $|01\rangle$, $|11\rangle$:

\[
\begin{align*}
\text{\includegraphics{diagram3.png}} &= \text{\includegraphics{diagram4.png}} + \text{\includegraphics{diagram5.png}} + \text{\includegraphics{diagram6.png}} + \text{\includegraphics{diagram7.png}} \\
\text{\includegraphics{diagram8.png}} &= \frac{1}{\sqrt{\sqrt{2}}} \text{\includegraphics{diagram9.png}} + \frac{1}{\sqrt{\sqrt{2}}} \text{\includegraphics{diagram10.png}} + \frac{1}{\sqrt{\sqrt{2}}} \text{\includegraphics{diagram11.png}} + \frac{1}{\sqrt{\sqrt{2}}} \text{\includegraphics{diagram12.png}} \\
\text{\includegraphics{diagram13.png}} &= \sqrt{2} \frac{\sqrt{\frac{\pi}{\sqrt{5}}}}{\sqrt{\sqrt{2}}} \tag{99}
\end{align*}
\]

The scalar. In the lower-spin examples (91)–(93), we could see that each diagram required a global scalar factor in order to get the correct normalisation. Let us describe how to calculate this scalar. We describe how to get the diagram for the unnormalised symmetrisation projector. We then get the actual scalar we need by dividing by $\frac{1}{n!}$, which gives us the correctly normalised symmetrisation projector. There are two contributions to the scalar:

1. Each CSWAP diagram (86) is ‘too small’ by a factor of $\frac{1}{\sqrt{2}}$ so that we need to correct by a $\sqrt{2}$ for each CSWAP in the diagram. There are $\sum_{i=2}^{n} (i - 1) = \frac{n(n-1)}{2}$ such gates in our diagram for the $n$-wire symmetriser.

2. When $n \geq 3$ we need to introduce corrections for the H-boxes in the crown. There is a special case for the $n = 3$ symmetriser, where the H-box is too large by 2, and requires a correction of $\frac{1}{2}$. Then for $n \geq 4$, every pair of H-box+Hadamard or H-box+Z-spider is too large by a factor of 2. In the part of the crown for $T_n$ (c.f. (90)) there are $2^k - 1$ such pairs, and thus the correction is $\sum_{i=4}^{n} (2^{[\log_2 i]} - 1)$ for the entire crown needed for building $S_n$. 

---

24
Combining this, we see that the scalar correction $\lambda_n$ for the normalised $n$-wire symmetriser is
\[
\lambda_n = \frac{2^{\frac{n(n-1)}{4}}}{n!} \cdot \left( \frac{1}{2} \right)^\beta \sum_{i=4}^n (2^{\lceil \log_2 i \rceil} - 1)
\]
where $\beta = 0$ if $n < 3$ else it is 1, which accounts for the absence of the 2 from the idiosyncratic $n = 3$ symmetriser for $n = 2$ crown. It should be noted that while this quite complicated analytic formula makes the connection between wire number and scalar concrete, intuitively it just means that we are multiplying by $\sqrt{2}$ for each CSWAP, dividing by 2 for each multi-leg H-box seen in a crown, and then all divided by $n!$ where $n$ is the number of wires.

**Alternative crowns** It has been brought to our attention late in the manuscript drafting that an alternative construction of the crown can be obtained by using the general principle of having a Z-spider for each qubit and connecting each pair by an H-box attached to a single wire Z-spider. For example, for three qubits this construction gives:

\[
\text{(101)}
\]

For four qubits we have:

\[
\text{(102)}
\]

This construction generalises in the obvious manner. Note that these diagrams are only equal up to a non-zero scalar to the state we need.

There is also a third different way to construct the crown. It turns out that our crowns can be related to the *W state* which features in the ZW-calculus, a related graphical calculus [41, 42]. To see this, recall that the W state is given by:
\[
|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle)
\]

(103)

This is close to being the state we are interested in. Indeed, if we contract one of the qubits with a $\langle + |$ effect we would get $\frac{1}{\sqrt{2}\sqrt{3}} (|00\rangle + |10\rangle + |01\rangle)$. In [35] it was shown the W state can be written as
\[
\text{(104)}
\]

and so we can generate the crowns we want (again up to a scalar factor) by taking W-states (and their multi-qubit generalisations) and appending a green spider to one of the external legs to give the necessary post-selection.

\[7\text{For this observation we are indebted to Aleks Kissinger.}\]

\[8\text{For this observation we are indebted to Quanlong Wang.}\]
**Symmetriser shorthand**  It will be practical to adopt a diagrammatic shorthand for our symmetrised wires. We will draw a line perpendicular through a series of wires to indicate the existence of a symmetriser over these wires rather than depict the ZKH-diagram for it explicitly. We do this when the structure of the symmetriser isn’t the component of interest. For instance we may write:

\[
\begin{align*}
1 \quad & \quad = \quad \frac{1}{\sqrt{2}} \quad \quad \quad \text{or} \quad \quad \quad \quad \frac{5}{2} \quad & \quad = \quad \frac{1}{7680} \\
\end{align*}
\]

(105)

Note that this is just shorthand, and that we still consider it equal to the diagrammatic construction we introduced above.

**Connecting wires.** There are two ways to compose Yutsis diagrams: either by juxtaposition, or by connecting wires. Juxtaposition simply corresponds to multiplication, so that this corresponds to juxtaposition of ZKH-diagrams as well. Connecting two wires is a little more involved to describe in terms of ZKH-diagrams.

Recall that the connection of two wires in a Yutsis diagram corresponds to a summation over a magnetic index \( m \) together with a phase, see Eq. (29). To understand how this is realised in ZKH, consider that when we connect two spin-\( j \) wires we should again get a spin-\( j \) wire:

\[
\sum_{k} (-1)^{j-k} \left( \begin{array}{c} m \\ k \\ n \\ m \end{array} \right) = \left( \begin{array}{c} m \\ k \\ n \\ m \end{array} \right)
\]

(106)

For spin-1/2, it is easy to realise that this is achieved with an intermediate green pi:

\[
\begin{array}{c}
\pi \\
\pi \\
\pi \\
\pi
\end{array} = \pi
\]

(107)

For general spin-\( j \), the connection is performed by the intermediate diagram

\[
\begin{array}{c}
\frac{1}{\lambda_n} \\
\pi \\
\pi \\
\pi
\end{array}
\]

(108)

where \( \lambda_n \) is the normalisation factor of the symmetriser diagram (100). Thus, we have

\[
\frac{\lambda_n^2}{\lambda_n} \begin{array}{c}
\pi \\
\pi \\
\pi \\
\pi
\end{array} = \lambda_n \begin{array}{c}
\pi \\
\pi \\
\pi \\
\pi
\end{array}
\]

(109)

This rule can be explained as follows. We want to sum the magnetic index \( m_3 \) from \(-j_3\) to \( j_3\). This amounts to applying the operator \( \sum_{m=-j_3}^{j_3} |j_3 m_3\rangle \langle j_3 m_3| \), which is the identity operator on the symmetric subspace, i.e. the symmetriser. However, the summation also requires a factor \( (-1)^{j_3-n_3} \). This factor corresponds to a \(-1\) phase being applied if there are an odd number of \(|1\rangle\)'s in the symmetric basis element. This phase is implemented by applying a \( Z(\pi) \) rotation to each of the wires corresponding to the summed over spin \( j_3 \) (recalling that a \( Z(\pi) \) rotation is \(|0\rangle - |1\rangle\)). The proportionality factor is determined straightforwardly.
From this rule, we can compute for instance the loop:
\[
\begin{array}{c}
\includegraphics[width=1cm]{loop.png}
\end{array}
= 2 = \begin{array}{c}
\includegraphics[width=1cm]{loop.png}
\end{array}.
\tag{110}
\]

### 4.2 Trivalent nodes

We have now seen how the links of Yutsis diagrams can be described as ZXH-diagrams. Let’s now look at the vertices, though note that these only match up to a complex phase when a group action is applied.

To start with, we look at the simplest possible 3-valent vertices:

\[
\begin{align*}
\left(\begin{array}{c}
\includegraphics[width=1.5cm]{vertex1.png}
\end{array}\right) &= \frac{1}{\sqrt{2}} (|10\rangle - |01\rangle) = \frac{1}{\sqrt{2}} \pi \\
\left(\begin{array}{c}
\includegraphics[width=1.5cm]{vertex2.png}
\end{array}\right) &= \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle) = \frac{1}{\sqrt{2}} \pi \\
\left(\begin{array}{c}
\includegraphics[width=1.5cm]{vertex3.png}
\end{array}\right) &= \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle) = \frac{1}{\sqrt{2}} \pi \\
\left(\begin{array}{c}
\includegraphics[width=1.5cm]{vertex4.png}
\end{array}\right) &= \frac{1}{\sqrt{2}} (-|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \pi \pi \pi
\end{align*}
\tag{111-114}
\]

For higher-spin vertices we look to equation (54) which shows how Penrose and Yutsis trivalent-nodes are related. The translation of the 3-valent vertex from Yutsis to the ZXH-calculus is very similar, but we must be careful about the normalisation and about the cups and caps that connect the wires. We have

\[
\begin{array}{c}
\includegraphics[width=3cm]{higher_spin_vertex.png}
\end{array}
\tag{115}
\]

with \( N \) the binor calculus normalisation given by (57) and the \( \lambda_{2j_i} \) the normalisation corrections given by equation (100) (one for every link). The three links are connected with the caps of equation (111). Note that the differing ordering of the Z and X spiders here originates from the cyclic ordering of the Yutsis wires.
Reversing the arrows. As shown in Eq. (26), reversing the direction of a wire in a Yutsis diagram corresponds to mapping the $|j;m\rangle$ state to $|j;-m\rangle$. Recall that in our symmetrised representation of spins that the $|j;m\rangle$ state corresponds to the symmetrised computational basis state where there are $j-m$ $|1\rangle$’s. Hence, the mapping $|j;m\rangle \mapsto |j;-m\rangle$ is implemented by interchanging $|0\rangle$ and $|1\rangle$ on each of the component spin-1/2 wires, i.e. by doing a NOT gate on all the qubits. This means that in the ZXH-diagram, to reverse an edge, we need to apply an $X(\pi)$ rotation to the external wires of each wire involved in the spin and that these rotations must come before the $Z(\pi)$ rotations on this wire.

$\begin{align*}
j_1 & \quad j_2 & \quad j_3 \\
\uparrow & \quad \uparrow & \quad \downarrow \\
\ldots & \quad \ldots & \\
2j_1 & \quad 2j_2 & \quad 2j_3
\end{align*}$

$\begin{align*}
j_1 & \quad j_2 & \quad j_3 \\
\uparrow & \quad \uparrow & \quad \downarrow \\
\ldots & \quad \ldots & \\
2j_1 & \quad 2j_2 & \quad 2j_3
\end{align*}$

Note that equations (111) and (114) are special cases of this.

4jm symbol. Now that we know how to write down trivalent nodes and wires, constructing 4jm-symbols is straightforward. By applying the rules for connecting wires, we find

$\begin{align*}
j_1 & \quad j_2 & \quad j_3 & \quad j_4 & \quad j_5 \\
\downarrow & \quad \downarrow & \quad \downarrow & \quad \downarrow & \\
\ldots & \quad \ldots & \quad \ldots & \\
2j_1 & \quad 2j_2 & \quad 2j_3 & \quad 2j_4 & \quad 2j_5
\end{align*}$

$\begin{align*}
j_1 & \quad j_2 & \quad j_3 & \quad j_4 & \quad j_5 \\
\downarrow & \quad \downarrow & \quad \downarrow & \quad \downarrow & \\
\ldots & \quad \ldots & \quad \ldots & \\
2j_1 & \quad 2j_2 & \quad 2j_3 & \quad 2j_4 & \quad 2j_5
\end{align*}$

Diagrammatic invariance. The 3jm-symbol defines a tensor in the SU(2)-invariant subspace of $\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \otimes \mathcal{H}_{j_3}$. This invariance can be checked concretely up to phase by applying an arbitrary $U \in SU(2)$ to each of the external wires and seeing that it can be removed. Using Euler decompositions, we can write any such $U$ as

$U = e^{-\frac{\alpha}{2}Z}e^{-\frac{i\alpha}{2}X}e^{-\frac{\alpha}{2}Z} = e^{-\frac{i}{2}(\alpha+\beta+\gamma)}$.

for some $\alpha$, $\beta$ and $\gamma$. Recall, from Eq. (7) that the action of SU(2) on higher spins is just applying the unitary to all the spin-1/2 space separately. The invariance condition hence

$^9$The Euler decomposition is often defined with the Pauli $Y$ instead of $Z$, but using $Z$ is a more natural choice in ZX-calculus.
becomes:

\[ e^{-i(j_1+j_2+j_3)(\alpha+\beta+\gamma)} \]

This can be shown to follow from the following equations:

1. \[ \begin{array}{c}
\begin{array}{c}
\alpha \\
\beta \\
\gamma
\end{array}
\end{array} = \begin{array}{c}
\begin{array}{c}
\text{...} \\
\text{...} \\
\text{...}
\end{array}
\end{array} \]

(120)

2. \[ \begin{array}{c}
\begin{array}{c}
\alpha \\
\beta \\
\gamma
\end{array} = e^{i\beta} \begin{array}{c}
\begin{array}{c}
\pi \\
\text{...} \\
\text{...}
\end{array}
\end{array} \]

(121)

3. \[ \begin{array}{c}
\begin{array}{c}
\alpha \\
\beta \\
\gamma
\end{array} = e^{i(\alpha+\gamma)} \begin{array}{c}
\begin{array}{c}
\text{...} \\
\text{...} \\
\text{...}
\end{array}
\end{array} \]

(122)

The proof of (120) can be easily worked out for a spin-1 wire with some arbitrary single qubit operator \( g \):

\[
\begin{array}{c}
\begin{array}{c}
g \\
\text{...} \\
\text{...} \\
\text{...}
\end{array}
\end{array} = \begin{array}{c}
\begin{array}{c}
\sqrt{\frac{1}{2}} \begin{array}{c}
\begin{array}{c}
g \\
\text{...} \\
\text{...} \\
\text{...}
\end{array}
\end{array} \end{array}
\end{array} = \begin{array}{c}
\begin{array}{c}
\sqrt{\frac{1}{2}} \begin{array}{c}
\begin{array}{c}
g \\
\text{...} \\
\text{...} \\
\text{...}
\end{array}
\end{array} \end{array} \end{array} \]

(123)

The equality for higher spins is proved analogously.

When we reverse the direction of a wire, we can similarly check that invariance is preserved. But instead of applying \( U \), we need to apply

\[ XUX = e^{-\frac{i}{2}(\alpha+\beta+\gamma)\text{sgn}(-\alpha,\beta,-\gamma)} \]

(124)

Note that this is not the adjoint of \( U \), which might be seen as surprising as it is often the case in graphical calculi that reversing the diagram amounts to dualising the space, i.e. switching from \( U \) to \( U^\dagger \). However, the inversion of an arrow as defined in the Yutsis calculus does not quite perform this transformation. Instead, it only flips the basis states \(|j;m\rangle \) (see Eq. (26)), which as we’ve seen corresponds to applying the NOT gate \( X \) to all the spin-1/2 wires in the symmetrised space.
With these considerations, the invariance condition for the $3jm$-symbol with the direction of the leftmost input changed reads:

$$e^{-i[(j_1 + j_2 + j_3)\beta + (j_2 + j_3 - j_1)(\alpha + \gamma)]} =$$

(125)

4.3 The ZXH Spin-network

As seen in Section 2.4, $3jm$-symbols can be used to specify intertwiners of spin-networks. Having translated Yutsis diagrams into the ZXH-calculus, it then becomes possible to write spin-networks as ZXH-diagrams by interconnecting compatible $3jm$-symbols; see Figure 4.

We can see in this figure that the arrangements of the $\pi$-phases, particularly the $Z$ phases associated to the symmetrisers, are removable (given some sign changes). It is interesting to note just how many of these can actually be removed, showing the redundancy in the original definition of the $3$-vertices.

4.4 Examples

Now that we have the tools to describe arbitrary Yutsis diagrams as ZXH-diagrams, let us give some examples. We used Sagemath to check correctness of the values of the Yutsis diagrams, and PyZX [39] to compute the values of the ZXH-diagrams, in order to verify that our constructions are correct.

The computations can be found in the appendix B.
4.4.1  $3jm$-symbol

**Example 1.** Let us consider the following 3-vertex:

\[
\begin{array}{ccc}
\frac{1}{2} & \frac{1}{2} & 1 \\
\end{array} = \begin{pmatrix}
\frac{-1}{\sqrt{3}} & 0 & 0 \\
0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\
0 & 0 & 0 \\
\end{pmatrix}
\] (126)

Following our convention for the matrix representation (see page 10), we get a $4 \times 3$ matrix as output because we input two spin-1/2 wires which means we have a $2 \cdot 2 = 4$-dimensional input, while we output a spin-1 wire, which is 3-dimensional.

From our construction described above, this should correspond to the following ZXH-diagram:

\[
\sqrt{2}^{-1} \cdot \sqrt{\frac{\pi}{3}} \begin{array}{c}
\pi \\
\pi \\
\pi \\
\end{array} = \begin{pmatrix}
\frac{-1}{\sqrt{3}} & 0 & 0 & 0 \\
0 & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & 0 \\
0 & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & 0 \\
0 & 0 & 0 & -\frac{1}{\sqrt{3}} \\
\end{pmatrix}
\] (127)

The SWAPs on the inputs here are required to produce the correct matrix in PyZX, because of the convention on the input ordering.

As we are working with a ZXH-diagram, its qubit matrix will always have the shape $2^n \times 2^m$, where $n$ is the number of inputs, and $m$ the number of outputs. In order to compare eq. (127) with the matrix of Eq. (126), we must project the codomain of this matrix to the symmetric subspace. This means multiplying the matrix of Eq. (127) on the left with

\[
P_1 = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\] (128)

This matrix implements the relations

\[
|1;1\rangle = |00\rangle \\
|1;0\rangle = \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle) \\
|1;-1\rangle = |11\rangle \\
\] (129)

And then we see that the matrices indeed match. Without the matrix $P_1$, it would still be true that the LHS of (126) and (127) correspond to the same linear map. The purpose of $P_1$ is to match the representation bases and thus obtain the exact same matrix.

**Example 2.** Now, let’s consider the 3-valent vertex

\[
\begin{array}{ccc}
1 & 1 & 1 \\
\end{array} \quad (130)
\]
It can be computed with the following ZXH diagram:

\[
M = \sqrt{2^{-3}} \sqrt{\frac{1}{3}}
\]

PyZX outputs

\[
M = \frac{1}{2\sqrt{3}} \begin{pmatrix}
0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0
\end{pmatrix}
\]

To obtain the corresponding Yutsis diagram, it is necessary to project the domains and codomains to the symmetric subspaces. The projector on the codomain is the same as (128). For the domain, the projector must send a tensor product of four qubits to two spin-1 spaces. This can be written explicitly as the following matrix:

\[
P_{1,1} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & \sqrt{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2} & \sqrt{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \sqrt{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \sqrt{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \sqrt{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

In general, to move from the spin-basis of Yutsis diagrams to the qubit-basis of ZXH-diagrams we make use of the canonical relationship given in equation (4) which we can use to construct the required basis-transformation matrix.

We then finally get:

\[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1
\end{pmatrix} = P_{1} M P_{1,1} = \begin{pmatrix}
0 & \sqrt{\frac{1}{6}} & 0 & \sqrt{\frac{1}{6}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\sqrt{\frac{1}{6}} & 0 & 0 & 0 & \sqrt{\frac{1}{6}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{\frac{1}{6}} & 0 & -\sqrt{\frac{1}{6}} & 0 & 0 & 0
\end{pmatrix}
\]

We can check this using standard Sagemath computation.
**Example 3.** Instead of considering the entire matrix of coefficients, let us give a different perspective, by directly calculating a single specific coefficient. In particular, let’s consider the coefficient sending $|\frac{1}{2}; -\frac{1}{2}\rangle \otimes |\frac{1}{2}; -\frac{1}{2}\rangle \rightarrow |1; -1\rangle$. The explicit calculation is then:

\[
\left(\begin{array}{cccc}
\frac{1}{2} & \frac{1}{2} & 1 & -1 \\
\frac{1}{2} & \frac{1}{2} & -1 & 1
\end{array}\right)_{3jm} = \\
\left(\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, 1, -1\right)

= \sqrt{2^{-4}} \sqrt{2^{-1}} \sqrt{\frac{T}{3}} = -\sqrt{\frac{T}{3}} \tag{135}
\]

There are a couple of things to note about this. First, the left-hand side here denotes a particular Wigner coefficient (and not a matrix). Second, the conventions for $3jm$ symbols dictate that we need to add a minus sign to the magnetic indices, since some of our input wires are inbound on the node in the corresponding Yutsis diagram (cf. Eq. (26)). Third, we are inputting $X(\pi)$ states here, because these correspond to $|1\rangle$ and we have $|1/2; 1/2\rangle = |1\rangle$ (and similarly we are post-selecting for $|11\rangle = |1; -1\rangle$).

Finally, we’ve written the scalar of the diagram here as a product of terms in order to indicate their separate origins. The correction of $\sqrt{2^{-4}}$ is because the plugged $X(\pi)$ states are equal to $\sqrt{2}|1\rangle$. The correction of $\sqrt{2^{-1}}$ is $\lambda_2$ and the $\sqrt{\frac{T}{3}}$ factor is the binor calculus coefficient $N(\frac{1}{2}, \frac{1}{2}, 1)$ (cf. Eq. (57)).

Evaluating the value of the diagram itself is straightforward:

\[
(c) = -1 \\
\text{(ex)} = -\frac{1}{\sqrt{2}} \\
\text{(f)} = -\frac{1}{\sqrt{2}} \tag{136}
\]

Multiplying this with the scalar correction of $\sqrt{2^{-4}} \sqrt{2^{-1}} \sqrt{\frac{T}{3}}$, we arrive at the final correct answer of $-1/\sqrt{3}$.

**4.4.2 4jm-symbol**

Now let’s consider a slightly more complex diagram: the Wigner 4jm-symbol (29).

**Example 4.** The simplest non-trivial 4jm-symbol diagram corresponds to four spin-1/2s joining together. There are then two possible ways to connect the wires (in other words,
the space of intertwiners is 2-dimensional). First, where the internal wire is spin-1:

\[
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array} = \begin{pmatrix}
\frac{1}{3} & 0 & 0 & 0 \\
0 & -\frac{1}{6} & -\frac{1}{6} & 0 \\
0 & -\frac{1}{6} & -\frac{1}{6} & 0 \\
0 & 0 & 0 & \frac{1}{3}
\end{pmatrix}
\]

(137)

\[
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array} = \sqrt{2}^{-1} \sqrt{\frac{T^2}{3}}
\]

(138)

Note that unlike the previous 3jm-symbols we considered, here the matrices exactly correspond to each other, without requiring projectors, as we are mapping from spin-$\frac{3}{2}$ spaces to spin-$\frac{1}{2}$ spaces.

The other possible 4jm-symbol has the intertwiner be spin-0:

\[
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
0 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
0 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
0 \\
\end{array} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & -\frac{1}{3} & \frac{1}{3} & 0 \\
0 & \frac{1}{3} & -\frac{1}{3} & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

(139)

As the spin is zero there are no wires between the individual 3jm-symbols, and furthermore these 3jms reduce to the special cases of the cup and cap given in (111) and (112).

**Example 5.** Now let’s consider a 4jm-symbol with some larger spins:

\[
\begin{array}{c}
1 \\
\downarrow \\
1 \\
\end{array}
\begin{array}{c}
1 \\
\downarrow \\
\frac{1}{2} \\
\end{array}
\begin{array}{c}
\frac{1}{2} \\
\downarrow \\
1 \\
\end{array} (140)
\]

The corresponding matrix can be computed with the following ZXH-diagram:

\[
M = \sqrt{\frac{T^3}{2}} \sqrt{\frac{T}{3}} \sqrt{\frac{T}{3}}
\]

(141)

PyZX outputs

\[
M = \frac{1}{6} \begin{pmatrix}
0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0
\end{pmatrix}
\]

(142)
To obtain the final correct matrix, we only need to project the domain into the spin-basis and so we will have

\[
\begin{bmatrix}
1 & 1 & \frac{1}{2} & \frac{1}{2} \\
1 & 1 & \frac{1}{2} & \frac{1}{2} \\
1 & 1 & \frac{1}{2} & \frac{1}{2} \\
1 & 1 & \frac{1}{2} & \frac{1}{2} \\
\end{bmatrix} = MP_{1,1}
\]

(143)

Example 6. As before, we can also directly calculate specific coefficients of the 4jm-symbol:

\[
\begin{pmatrix}
1 & 1 & \frac{1}{2} & \frac{1}{2} \\
-1 & 0 & \frac{1}{2} & \frac{1}{2} \\
\end{pmatrix}^{(1)} = \begin{bmatrix}
1, 1 & 1, 0 & \frac{1}{2}, \frac{1}{2} & \frac{1}{2}, \frac{1}{2} \\
\end{bmatrix}
\]

(144)

\[
= \sqrt{\frac{7}{2}} \sqrt{\frac{3}{2}} \sqrt{\frac{3}{3}} \sqrt{\frac{3}{3}}
\]

(145)

\[
= \frac{\sqrt{2}}{6}
\]

(146)

The scalar factor \(\sqrt{\frac{7}{2}}\) is \(\lambda_3\), the \(\sqrt{\frac{3}{2}}\) is to rescale the inputs/outputs to basis elements (remembering that \(|j = 1; m = 0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\) so that it only requires one \(\sqrt{2}\) to be rescaled unlike the pairs of \(|1\rangle\) input/output states) and the \(\sqrt{\frac{7}{3}}\)'s are the factors from the binor calculus \(N(1, 1, 1)\) and \(N(1/2, 1/2, 1)\). The diagram itself evaluates to \(\sqrt{2^7}\):
Although this calculation looks lengthy, it can efficiently and automatically be done by PyZX.

4.4.3 6j-symbol

Let’s now consider an invariant function, specifically the Wigner 6j symbol

\[
\begin{pmatrix}
  j_1 & j_2 & j_3 \\
  j_4 & j_5 & j_6
\end{pmatrix}
\]

As shown in Eq. (35), it is composed of four interlinked 3jm-symbols or equivalently as two 4jm-symbols. We will take the latter as a diagrammatic starting point and can therefore state explicitly that:

\[
\begin{pmatrix}
  j_1 & j_2 & j_3 \\
  j_4 & j_5 & j_6
\end{pmatrix} = C
\]

Where

\[
C = \frac{\lambda_{2j_1} \lambda_{2j_2} \lambda_{2j_3} \lambda_{2j_4} \lambda_{2j_5} \lambda_{2j_6}}{N(j_1, j_2, j_3) N(j_1, j_6, j_5) N(j_2, j_6, j_4) N(j_3, j_4, j_5)}
\]
Example 7. Let’s work out a concrete example.

\[
\begin{array}{c}
\{ 2 \ 1 \ 1 \\
1 \ 1 \ 1
\end{array}
\] = C

with
\[
C = \frac{1}{48} \left( \frac{1}{\sqrt{2}} \right)^5 \left( \frac{2!2!2!}{4!1!1!1!} \right)^2 \left( \frac{4!2!2!}{5!2!2!0!} \right)^2.
\]

\(C\) captures the scalar corrections for the symmetrisers \(\lambda\) and by the four normalisations from the binor calculus \(N\). The diagram evaluates to \(480\sqrt{2}\) (as calculated in \(\text{PyZX}\)). We get the final answer

\[
\begin{array}{c}
\{ 2 \ 1 \ 1 \\
1 \ 1 \ 1
\end{array}
\] = \(\frac{1}{6}\).

Example 8. Let’s consider now the following larger \(6j\)-symbol:

\[
\begin{array}{c}
\{ 2 \ 2 \ 2 \\
1 \ 1 \ 1
\end{array}
\] = C

with
\[
C = \left( \frac{1}{48} \right)^3 \left( \frac{1}{\sqrt{2}} \right)^3 \left( \frac{4!4!4!}{7!2!2!2!} \right)^2 \left( \frac{4!2!2!}{5!2!2!0!} \right)^2.
\]

The value of the diagram is \(645120\sqrt{2}\) as calculated via tensor contraction in \(\text{PyZX}\). The final value of the \(6j\)-symbol is hence

\[
\begin{array}{c}
\{ 2 \ 2 \ 2 \\
1 \ 1 \ 1
\end{array}
\] = \(\frac{\sqrt{21}}{30}\).
4.5 Higher Wigner invariants and the question of quantum computation

With the example of the $6j$-symbol in hand it is straightforward to see how to generalise this to higher-order invariant functions: just glue together the $3jm$-symbols in the manner outlined above. There is a caveat however as the contraction of a tensor network is a classically difficult problem ($\#P$-hard, and hence in particular, NP-hard). If we were to simply consider these diagrams as tensors and contract them, the cost of doing so is largely dependent on contraction optimisation methods and there isn’t a priori any guarantee that it offers computational advantages over other approaches. Though there is likely to be simplifications possible by using the ZXH-calculus rewrite rules, which would be beneficial, there is also another approach we can consider. The field of quantum computing may offer a route to practical calculations for more complicated objects such as the spin-foam vertex amplitudes, a model of which is given by 4-valent spin networks, which are in general difficult to calculate classically. What we present here then serves as a path to take SU(2) spin-networks and encode them directly into a framework that can be viewed as qubits and operators on them.

A qubit representation however is not in itself a guarantee that we can actually run it on a quantum computer. To write the diagram as something we can execute on a quantum computer, we need to represent it as a quantum circuit. In this problem of ‘circuit extraction’ the aim is to take a problem instance and describe it in terms of input states and output measurements with unitaries in the middle [23]. Whether the problem can be solved efficiently by a quantum algorithm then depends on whether the extracted circuit, including the input state and measurement effects, are efficiently preparable by a quantum computer. If, as with our CSWAPS, there are ‘post-selected measurements’ (cf. (86))—measurements that must give a certain result when the algorithm is run—then during execution of the algorithm it is possible we fail to get the values we require and then must rerun the algorithm. This means each additional post-selected measurement will exponentially reduce our chance of a successful run. Secondly, the input states of the algorithm may themselves be complex to create, even while the unitary operation on the input is simple. For these reasons it is still an open question how to make use of quantum computers for calculations of SU(2) invariant symbols. We remain optimistic however that solutions can be identified given the developments seen in the broader literature, for instance [25, 26], and [27]. This latter one is particularly interesting as they take 4-valent intertwiners (fixed at dimension 2) and attempt to determine spin-network transition amplitudes based off these on IBM superconducting devices.

5 Conclusion

In this paper, we have showed how to represent Yutsis diagrams and Penrose spin-networks as ZXH-diagrams. This translation allows us to diagrammatically prove properties like SU(2)-invariance, as well as to calculate the values of some Wigner symbols. On a conceptual level, this reveals the link between spin-networks and quantum information on qubits.

There are a number of open questions and areas of future work. First, we can study the usage of the ZXH-calculus for Loop Quantum Gravity, by representing geometrical observables (area and volume) acting on the spin-networks. These observables provide physical meaning to the spin-networks as quantum states of space. This would require the development of a diagrammatic description of the representation of the Lie algebra $\mathfrak{su}(2)$.
in the ZXH-calculus.

A second direction of interest would be the development of diagrams adapted to spin-foam models. The latter can be thought of as an extension of spin-networks, meant to describe quantum states of space-time (and not only space). Spin-foams require the computation of the so-called vertex amplitude, which is usually computationally hard. To do this in the ZXH-calculus would require us to define edges labelled by $SL(2; \mathbb{C})$ representations and $SL(2; \mathbb{C})$ Wigner symbols. It is known that the principal unitary representations of this group decompose into a direct sum of SU(2) irreps. This fact points towards an extension of the diagrams here developed.

A third direction of research would be to extend the translation described here to include the world of condensed matter physics. It has been shown that the ZXH-calculus can be used to describe condensed matter systems [24], where the pictorial tensor networks used in the domain of condensed matter are extended to full-blown diagrammatic calculations. The interaction between spin-networks and Ising models has also been discussed in the literature [43, 44] and indeed tackling these models with the ZXH-calculus is a task that sits nicely on the intersection of the work seen here and in [24].

Finally, there is the question of whether our description of spin-networks in terms of the ZXH-calculus allows one to translate spin-network calculations into calculations runnable on a quantum computer. Due to the non-unitary nature of some of the operators used in constructing our spin-networks, and the use of post-selected measurements, this is not a straightforward question. However, we have here forged a link between spin-networks and the qubit world, as the ZX-calculus is usually considered to be a tool to discuss the creation, implementation, and optimisation of quantum algorithms [21, 35, 45]. It may also be possible to find a more computationally efficient version of the symmetriser or that we somehow obviate their use by means of working with already symmetrised spaces, such as spin systems. A natural and concrete extension would also be to try to subsume the work seen in [25, 27] into the ZXH-calculus to see if their techniques can be extended and ours informed by their circuit structure. In this way we could see how quantum computers could be brought to bear on the calculations that are of interest to those who study spin-networks, like in LQG, or spin-recoupling more generally which is of interest to quantum chemistry.

The quantum theory of angular momentum was a major milestone in the development of quantum physics in the last century, ranging from quantum chemistry to quantum gravity. In this paper we have seen that we can use the diagrammatic language of the ZXH-calculus to study this area as well. This connects the fundamental theory of angular momentum and spin-networks to the language of qubits and quantum computation.

Acknowledgments

RDPE would like to acknowledge financial support from the “Investissements d’avenir” (ANR-15-IDEX02) program of the French National Research Agency and discussions with N. de Beaudrap, A. Kissinger, Q. Wang and M. Christodoulou. PMD thanks Alexandra Elbakyan for her help to access the scientific literature. JvdW acknowledge that most of the work for this article was supported by an NWO Rubicon personal fellowship. Additionally, JvdW acknowledges that this project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101018390. PMD acknowledges that this publication was made
possible through the support of the ID# 61466 grant from the John Templeton Foundation, as part of the “Quantum Information Structure of Spacetime (QISS)’ project (qiss.fr). The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation or the European Research Agency.

A Overview of graphical rewrite rules

\[ m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n = \sqrt{2^{n+m}} \quad m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n \]

\[ \alpha = e^{i\theta} \]

\[ m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n = \sqrt{2^{n+m}} \quad m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n \]

\[ e^{i\theta} = \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \]

\[ m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n = m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n \quad \left( \sqrt{2} \right)^{(n-1)(m-1)} \]

\[ \left( \pi \right) = e^{i\pi} \quad \left( \pi \right) = e^{i\pi} \quad \left( \pi \right) = e^{i\pi} \quad \left( \pi \right) = e^{i\pi} \quad \left( \pi \right) = e^{i\pi} \]

\[ \left( \text{id} \right) = \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \]

\[ \left( \text{ho} \right) = \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \]

\[ m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n = 2 \quad m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n \]

\[ \left( \text{hf} \right) = \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \]

\[ \left( \text{hc} \right) = \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \quad \frac{n}{\sqrt{2}} \quad \frac{m}{\sqrt{2}} \]

\[ m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n = \frac{1}{\sqrt{2}}^{n-1} \quad m \left\{ \begin{array}{c}
\alpha \\
\vdots
\end{array} \right\} n \]

\[ \left( \text{ab} \right) = \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \]

\[ \left( \text{ex} \right) = \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \]
### B Sage verification code

**Sagemath** is a free and open source alternative to proprietary computation softwares like Mathematica or Maple.

The 3jm symbols were verified via

```python
def vertex_ooi(j1,j2,j3):
    M = matrix([[
        (-1)**(j3-m3)*wigner_3j(j1,j2,j3,m1,m2,-m3)
        for m1 in srange(-j1,j1+1)
        for m2 in srange(-j2,j2+1)
        for m3 in srange(-j3,j3+1)
    ]])
    return M

def vertex_ooo(j1,j2,j3):
    M = matrix([[
        wigner_3j(j1,j2,j3,-1*(m1),-1*(m2),-1*(m3))
        for m3 in srange(-j3,j3+1)
        for m1 in srange(-j1,j1+1)
        for m2 in srange(-j2,j2+1)
    ]])
    return M
```

The 4jm symbols were verified via

```python
def wigner_4jm(j1,j2,j3,j4,m1,m2,m3,m4,j):
    return sum((-1)**(j-m)*wigner_3j(j1,j2,j3,m1,m2,m)*wigner_3j(j1,j5,j6,m1,-m5,m6)*wigner_3j(j4,j2,j6,m4,m2,-m6)*wigner_3j(j3,j4,j5,m3,-m4,m5)
        for m in srange(-j,j+1))

def vertex_iioo(j1,j2,j3,j4,j):
    M = matrix([[
        wigner_4jm(j1,j2,j3,j4,-1*(-m1),-1*(-m2),-1*(m3),-1*(m4),j)
        for m1 in srange(-j1,j1+1)
        for m2 in srange(-j2,j2+1)
        for m3 in srange(-j3,j3+1)
        for m4 in srange(-j4,j4+1)
    ]])
    return M
```

The 6jm symbols were verified via

```python
def sixJ_symbol(j1,j2,j3,j4,j5,j6):
    return sum(sum(sum(sum(sum(sum((-1)**(j1+j2+j3+j4+j5+j6-m1-m2-m3-m4-m5-m6)
        *wigner_3j(j1,j2,j3,-m1,-m2,-m3)*wigner_3j(j1,j5,j6,m1,-m5,m6)
        *wigner_3j(j4,j2,j6,m4,m2,-m6)*wigner_3j(j3,j4,j5,m3,-m4,m5)
        for m1 in srange(-j1,j1+1))
        for m2 in srange(-j2,j2+1))
        for m3 in srange(-j3,j3+1))
        for m4 in srange(-j4,j4+1))
        for m5 in srange(-j5,j5+1))
        for m6 in srange(-j6,j6+1))
```

```
for m2 in srange(-j2,j2+1))
for m3 in srange(-j3,j3+1))
for m4 in srange(-j4,j4+1))
for m5 in srange(-j5,j5+1))
for m6 in srange(-j6,j6+1))

References
[1] R. P. Feynman. Space-Time Approach to Quantum Electrodynamics. 76(6):769–789, 1949. DOI: 10.1103/PhysRev.76.769.
[2] A. P. Yutsis, I. B. Levinson, and V. V. Vanagas. Mathematical Apparatus of the Theory of Angular Momentum. Israel Program for Scientific Translations, 1962.
[3] Roger Penrose. Applications of Negative Dimensional Tensors. In D. J. A. Welsh, editor, Combinatorial Mathematics and Its Applications. Academic Press, 1971.
[4] Jacob Biamonte and Ville Bergholm. Tensor Networks in a Nutshell. arXiv:1708.00006.
[5] Bob Coecke and Ross Duncan. Interacting quantum observables. In International Colloquium on Automata, Languages, and Programming, pages 298–310. Springer, 2008. DOI: 10.1007/978-3-540-70583-3_25.
[6] Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. New Journal of Physics, 13(4):043016, 2011. DOI: 10.1088/1367-2630/13/4/043016.
[7] Renaud Vilmart. A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics. In 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), pages 1–10, 2019. DOI: 10.1109/LICS.2019.8785765.
[8] Kang Feng Ng and Quanlong Wang. A universal completion of the zx-calculus. arXiv preprint arXiv:1706.09877, 2017.
[9] Ross Duncan and Simon Perdrix. Rewriting measurement-based quantum computations with generalised flow. In International Colloquium on Automata, Languages, and Programming, pages 285–296. Springer, 2010. DOI: 10.1007/978-3-642-14162-1_24.
[10] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski, and John van de Wetering. There and back again: A circuit extraction tale. Quantum, 5:421, 3 2021. ISSN 2521-327X. DOI: 10.22331/q-2021-03-25-421. arXiv: 2003.01664.
[11] Aleks Kissinger and John van de Wetering. Universal MBQC with generalised parity-phase interactions and Pauli measurements. Quantum, 3, 2019. DOI: 10.22331/q-2019-04-26-134.
[12] Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery. Quantum, 4, 2020. DOI: 10.22331/q-2020-01-09-218.
[13] Craig Gidney and Austin G. Fowler. Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $|T\rangle$ transformation. Quantum, 3:135, 4 2019. ISSN 2521-327X. DOI: 10.22331/q-2019-04-30-135.
[14] Michael Hanks, Marta P. Estarellas, William J. Munro, and Kae Nemoto. Effective Compression of Quantum Braided Circuits Aided by ZX-Calculus. Physical Review X, 10:041030, 2020. DOI: 10.1103/PhysRevX.10.041030.
[15] Niel de Beaudrap, Ross Duncan, Dominic Horsman, and Simon Perdrix. Pauli Fusion: a Computational Model to Realise Quantum Transformations from ZX Terms. In Bob Coecke and Matthew Leifer, editors, Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019, volume 318 of Electronic Proceedings in Theoretical Computer Science, pages 85–105. Open Publishing Association, 2020. DOI: 10.4204/EPTCS.318.6. arXiv: 1904.12817. 2

[16] Nicholas Chancellor, Aleks Kissinger, Joschka Roffe, Stefan Zohren, and Dominic Horsman. Graphical Structures for Design and Verification of Quantum Error Correction. 2016. arXiv: 1611.08012. 2

[17] Liam Garvie and Ross Duncan. Verifying the Smallest Interesting Colour Code with Quantomatic. In Bob Coecke and Aleks Kissinger, editors, Proceedings 14th International Conference on Quantum Physics and Logic, Nijmegen, The Netherlands, 3-7 July 2017, volume 266 of Electronic Proceedings in Theoretical Computer Science, pages 147–163. Open Publishing Association, 2018. DOI: 10.4204/EPTCS.266.10. arXiv: 1706.02717. 2

[18] John van de Wetering. Constructing quantum circuits with global gates. New Journal of Physics, 23(4):043015, 4 2021. DOI: 10.1088/1367-2630/abf1b3. arXiv: 2012.09061. 2

[19] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons, and Seyon Sivarajah. Phase gadget synthesis for shallow circuits. In Bob Coecke and Matthew Leifer, editors, Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019, volume 318 of Electronic Proceedings in Theoretical Computer Science, pages 213–228. Open Publishing Association, 2020. DOI: 10.4204/EPTCS.318.13. arXiv: 1906.01734.

[20] Alexander Cowtan, Will Simmons, and Ross Duncan. A Generic Compilation Strategy for the Unitary Coupled Cluster Ansatz. 2020. arXiv: 2007.10515.

[21] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Fast and effective techniques for T-count reduction via spider nest identities. 2020. arXiv: 2004.05164. 39

[22] Aleks Kissinger and John van de Wetering. Reducing the number of non-Clifford gates in quantum circuits. Physical Review A, Vol.102-2, 102:022406, 8 2020. DOI: 10.1103/PhysRevA.102.022406. 15

[23] Ross Duncan, Aleks Kissinger, Simon Pedrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. Quantum, 4:279, 6 2020. ISSN 2521-327X. DOI: 10.22331/q-2020-06-04-279. 2, 15, 18, 38

[24] Richard DP East, John van de Wetering, Nicholas Chancellor, and Adolfo G Grushin. Aklt-states as zx-diagrams: diagrammatic reasoning for quantum states. 2020. arXiv: 2012.01219. 2, 15, 21, 39

[25] Jakub Mielczarek. Spin foam vertex amplitudes on quantum computer—preliminary results. Universe, 5(8):179, 2019. arXiv: 1810.07100. 3, 38, 39

[26] Lior Cohen, Anthony J Brady, Zichang Huang, Hongguang Liu, Dongxue Qu, Jonathan P Dowling, and Muxin Han. Efficient simulation of loop quantum gravity: A scalable linear-optical approach. Physical Review Letters, 126(2):020501, 2021. arXiv: 2003.0344. 38

[27] Grzegorz Czelusta and Jakub Mielczarek. Quantum simulations of a qubit of space. Physical Review D, 103(4):046001, 2021. arXiv: 2003.13124. 3, 38, 39

[28] Roger Penrose. Angular momentum: An approach to combinatorial spacetime. In T Bastin, editor, Quantum Theory and Beyond, pages 151–180. Cambridge University Press, 1971. 3, 14
[29] Pierre Martin-Dussaud. A Primer of Group Theory for Loop Quantum Gravity and Spin-foams. *Gen Relativ Gravit.*, 51(9):110, September 2019. ISSN 0001-7701, 1572-9532. DOI: 10.1007/s10714-019-2583-5. arXiv: 1902.08439.

[30] Giorgio Sarno, Simone Speziale, and Gabriele V. Stagno. 2-vertex Lorentzian spin foam amplitudes for dipole transitions. *Gen Relativ Gravit.*, 50(4):43. ISSN 0001-7701. DOI: 10.1007/s10714-018-2360-x. arXiv: 1801.03771.

[31] Seth A Major. A spin network primer. *American Journal of Physics*, 67(11):972–980, 1999. arXiv: gr-qc/9905020.

[32] Carlo Rovelli and Lee Smolin. Spin networks and quantum gravity. *Physical Review D*, 52(10):5743–5759, 1995. ISSN 05562821. DOI: 10.1103/PhysRevD.52.5743. arXiv: gr-qc/9709028v2.

[33] Miriam Backens and Aleks Kissinger. ZH: A complete graphical calculus for quantum computations involving classical non-linearity. 2018. arXiv: 1805.02175.

[34] Miriam Backens, Aleks Kissinger, Hector Miller-Bakewell, John van de Wetering, and Sal Wolffs. Completeness of the ZH-calculus. *arXiv preprint arXiv:2103.06610*, 2021.

[35] John van de Wetering. ZX-calculus for the working quantum computer scientist. 2020. arXiv: 2012.13966.

[36] B. Coecke and A. Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. DOI: 10.1017/9781316219317.

[37] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014. DOI: 10.1088/1367-2630/16/9/093021. arXiv: 1307.7025.

[38] Stach Kuijpers, John van de Wetering, and Aleks Kissinger. Graphical fourier theory and the cost of quantum addition. 2019. arXiv: 1904.07551.

[39] Aleks Kissinger and John van de Wetering. Pyzx: Large scale automated diagrammatic reasoning. In Bob Coecke and Matthew Leifer, editors, *Proceedings 16th International Conference on Quantum Physics and Logic*, Chapman University, Orange, CA, USA., 10-14 June 2019, volume 318 of *Electronic Proceedings in Theoretical Computer Science*, pages 229–241. Open Publishing Association, 2020. DOI: 10.4204/EPTCS.318.14. arXiv: 1904.04735.

[40] Niel de Beaudrap. Well-tempered ZX and ZH calculi. 2020. arXiv: 2006.02557.

[41] Bob Coecke and Aleks Kissinger. The compositional structure of multipartite quantum entanglement. In *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 297–308. Springer, 2010. DOI: 10.1007/978-3-642-14162-1_25.

[42] Amar Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 573–584. IEEE, 2015. DOI: 10.1109/LICS.2015.59.

[43] Alexandre Feller and Etera R Livine. Ising spin network states for loop quantum gravity: a toy model for phase transitions. *Classical and Quantum Gravity*, 33(6): 065005, 2016. arXiv: 1509.05297.

[44] Valentin Bonzom, Francesco Costantino, and Etera R Livine. Duality between spin networks and the 2d ising model. *Communications in Mathematical Physics*, 344(2): 531–579, 2016. arXiv: 1504.02822.

[45] Titouan Carette, Yohann D’Anello, and Simon Perdrix. Quantum Algorithms and Oracles with the Scalable ZX-calculus. In Chris Heunen and Miriam Backens, editors, *Proceedings 18th International Conference on Quantum Physics and Logic*, Gdansk,
