An Overview of Diagrammatic Notation in Physics

John Selby

April 22, 2014

Abstract

This review looks at the development of new techniques in physics in which diagrams take a central role, both as an intuitive aid in understanding the world but also as a powerful tool for performing calculations. In the first section we look at the historic development of these ideas, in the second we then take a pragmatic approach and show how they can be used to vastly speed up calculations, and then we conclude by trying to look at how and why these various diagrams differ and whether they could be unified. In viewing these different aspects of these diagrammatic techniques we hope to elucidate why one should have an interest in these new ideas both as a practical tool but also for their foundational significance.

1 Historical overview

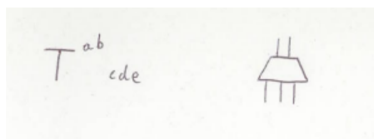
Most uses of diagrams in physics to date have essentially been simplified pictures of the world, or pictures of the world in which we draw on things that we could not normally see. For example diagrams of electric circuits, they are simplified in that they only show how components are connected rather than their actual physical positions in space, but we also often draw on for example external fields which we cannot directly see. Another example would be force diagrams in which we draw objects in some simplified form and draw on arrows to represented forces that we do not "see" normally.

These examples however don't really capture what we mean by the idea of 'diagrammatic notation', the idea of diagrammatic notation is that the diagrams are sufficiently well defined, that there is a consistent set of rewrite rules such that it is possible to say that two diagrams are equivalent in some sense, just by manipulating the diagram itself. In the two above examples if we have two such diagrams we need to use standard mathematical techniques, i.e. calculus, vector addition, basic arithmetic etc. to consider whether two diagrams are equivalent. One could try to argue that the force diagrams actually could be a diagrammatic notation, as one could take the arrows, connect them end-to-end and show that two sets of arrows give the same resulting vector. This would work in principle and I suppose constitute pure diagrammatic reasoning, but this approach lacks the precision that we expect from mathematics.

So far we have discussed examples which are not diagrammatic notation so its about time we gave an example that is (or at least is closer to what we're aiming for). The classic example is the graphical notation developed by Penrose [1]. This was developed to describe tensor calculations in a more visual way to allow for calculations to be performed more easily.

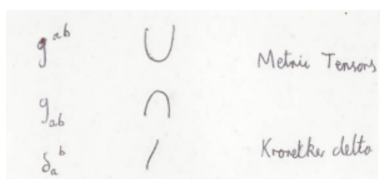
1.1 Penrose graphical notation

Penrose's graphical notation [1] was used to represent tensorial calculations, tensors are represented by some shape and the (abstract) indices of the tensors as lines coming from the shape, lines out of the top would indicate upper indices and lines out of the bottom lower indices.



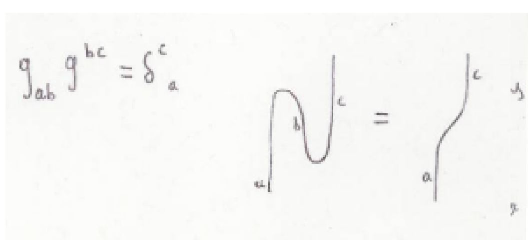
(1)

Particular tensors were given suggestive shapes, for example the metric tensors and the Kronecker delta, to highlight their roles in a calculation.



(2)

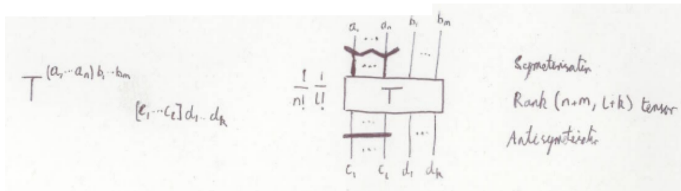
This leads to geometrically intuitive equations such as,



(3)

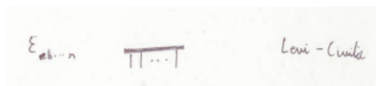
which are not immediately obvious from the standard notation.

Symmetrisation and anti-symmetrisation of indices is given by thick wavy or straight bars across the lines to be (anti)symmetrised,



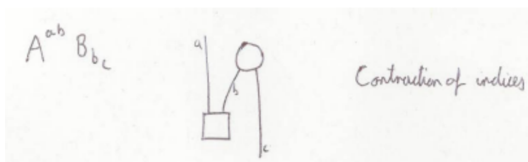
(4)

which leads to an intuitive diagram for the Levi-Civita tensor as,



. (5)

Contraction of indices is given by connecting the wires from one tensor to another,



. (6)

It doesn't really seem like anything has been gained here, all that this has given is a new way to write out a tensorial expression. However it proves to be of great benefit in keeping track of calculations as they become more complicated. In addition diagram 3 shows how by careful choice of "shape" for certain tensors certain identities become intuitive (here as straightening out the line), which in the long run makes things much easier.

The best evidence for the usefulness of these diagrams is the fact that they have actually been used in a wide range of areas. From classification of classical Lie groups [2], to spin networks in quantum gravity [3] and tensor network states in condensed matter [4].

We see that this notation is useful for dealing with the structure of calculations, where we are interested in how the tensors are connected together. However it does not deal well with the 'internals' of the tensors. We still need a long list of numbers written down somewhere to tell us what most of the tensors are (the exceptions being the special tensors such as the metric), and any contraction of indices will at some point have to resort to standard techniques to calculate them.

This graphical notation whilst being a useful tool in mathematics and physics is fairly limited in it's scope, there is a large part of physics and mathematics that is not just the structure of tensorial calculations. Another major development in graphical notation came about from Category theory, which is anything but limited in scope.

1.2 Category theory

Category theory was originally developed by Saunders MacLane and Samuel Eilenberg [5], in the context of topology but was quickly used to study abstract structures in mathematics and later in computer science [6] and physics [7]. The overarching theme in category theory is to not worry about the the properties of the objects to be studied but to consider only how they interact and relate to each other [8].

A category $\mathcal{C}$ is defined [8] as a collection of 'objects', $|\mathcal{C}|$, together with a set of 'morphisms', $\mathcal{C}(A, B)$, for each pair of objects, $A, B \in |\mathcal{C}|$.

Along with a binary operation,

$$\_ \circ \_ : \mathcal{C}(B, C) \times \mathcal{C}(A, B) \to \mathcal{C}(A, C),$$

which is associative,

$$(f \circ g) \circ h = f \circ (g \circ h),$$

and there exists an identity,

$$I_A \in \mathcal{C}(A, A) \ \forall \ A \in |\mathcal{C}|,$$

such that,

$$f \circ I_A = f = I_B \circ f \ \ \forall \ f \in \mathcal{C}(A, B).$$

From this definition it is not at all clear what this would have to do with diagrammatic notation at all, the first step in that direction is by introducing the idea that the morphisms can be drawn as arrows going between the objects. i.e. if $f \in \mathcal{C}(A, B)$ then this can be written as $f : A \to B$ or even just $A \xrightarrow{f} B$. The binary operation can then be seen just as connecting two of these arrows end to end to create another arrow, e.g. $A \xrightarrow{f} B \xrightarrow{g} C = A \xrightarrow{g \circ f} C$. And the identity is an arrow from an object to itself $A \xrightarrow{I_A} A$.

Larger networks of these arrows can be built up connecting many different objects together to create larger diagrams, a useful property of such a diagram is whether it 'commutes' or not. This means that whichever route we take between two objects in a given diagram will result in the same morphism. Many proofs in category theory boil down to determining whether a given diagram commutes.

Despite diagrammatic notation appearing at this point it is not actually these diagrams that we are primarily interested in. The diagrams we are actually interested in can be seen as an extension of this sort of reasoning to 'monoidal categories', where 'string diagrams'[9] can be used.

A strict monoidal category[10] $\mathcal{M}$ is a category equipped with an operator,

$$\_ \otimes \_ : \mathcal{M} \times \mathcal{M} \to \mathcal{M},$$

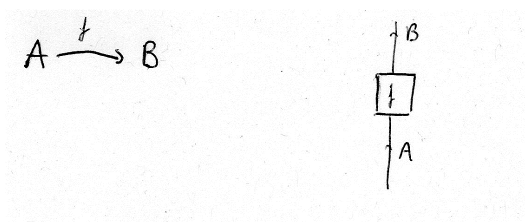
and a unit object $\mathbb{1}$.

Such that $(A \otimes B) \otimes C = A \otimes (B \otimes C)$ and $\mathbb{1} \otimes A = A = A \otimes \mathbb{1}$. Note that in the case where the monoidal category is not strict these equalities become 'natural isomorphisms' subject to certain 'coherence conditions'.

There is a more physical interpretation of monoidal categories in terms of 'process theories'[10] if we have a category where the objects are some physical objects and the morphisms are some physical process that we can do to transform one object into another. In such a category we can imagine that there are two types of composition that we would want. Firstly, we could do one physical process first and then do another to the same object, providing that the objects and transformations match up correctly. This corresponds to the $\circ$ composition of morphisms in a category. The second type of composition corresponds to doing two processes at the same time to a pair of objects, that is the monoidal operation $\otimes$.

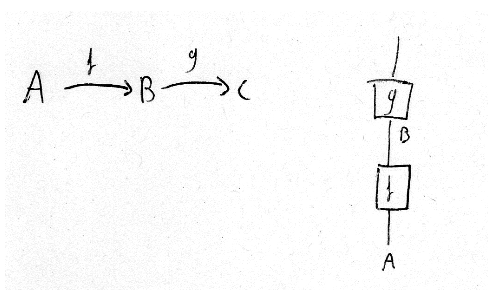
From this viewpoint it is clear that these two products should interact in a particular way, this is the 'interchange law'. In words it says that (for suitable processes f, g, h, i on a suitable pair of objects) "doing (f and g) then (h and i) to a pair of objects" will be the same as "doing (f then h) and (g then i) to the same pair of objects". Or in the monoidal category language $(f \otimes g) \circ (h \otimes i) = (f \circ h) \otimes (g \circ i)$. This can be proved to be true from the definition of a monoidal category.

We now explain the diagrammatic notation for such a process theory, i.e. string diagrams [9]. Like with Penrose graphical notation certain non-obvious equations become utterly trivial when expressed in the new language which gives some suggestion of it's later use. A morphism $A \xrightarrow{f} B$ is notated as,



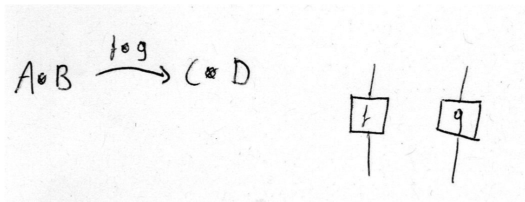
. (7)

composition as,



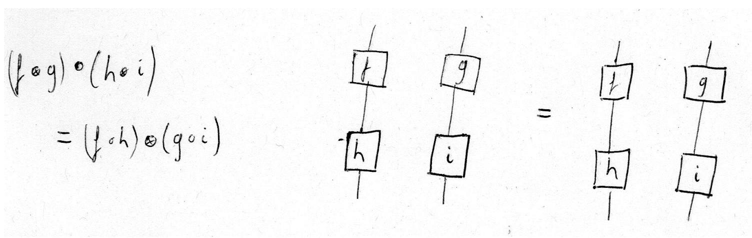
. (8)

monoidal product as,



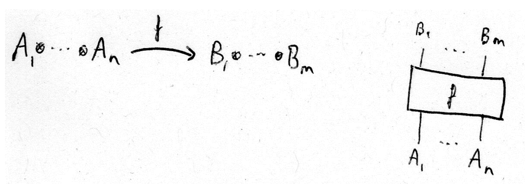
. (9)

and the interchange law becomes tautological,



. (10)

We can generalise this to include processes that take any number of objects to any other number of objects, and so include boxes with arbitrary numbers of inputs and outputs e.g.,



. (11)

Bob Coecke and Samson Abramsky [11] developed this notation for use in quantum mechanics in terms of a (dagger compact symmetric) monoidal category. This is based on the ideas described above but greatly extended, there is not space in this review to go into all of the details but an illustrative example using many of the important features will be shown in the next section.

A good review of the different diagrammatic notation used in category theory is by P. Selinger [12]. It is also worth noting at this point that Penrose's graphical notation can now also be thought of in this language as an example of a string diagram.

There are many examples of diagrammatic notations used now in physics, many of these can be thought of from a categorical perspective but many are not (yet!). In the following sections I hope to introduce at least a few of these but obviously this cannot be an entirely comprehensive review.

2 Diagrams for calculations

In this section we try to motivate the development of these diagrammatic languages from a pragmatic point of view, i.e. that they speed up ones ability to do calculations in physics. An example of this would be Feynman diagrams, they provide a way to perform complicated integrals using a perturbative method in which each term in the perturbation series is “easy” to calculate, in addition it provides a physical intuition of what is happening in the diagrams i.e. a sum of all the possible ways in which particles can interact [13]. It is worth mentioning however that recent developments (for example the amplituhedron [14] in $\mathcal{N} = 4$ SYM) imply that Feynman diagrams may not be the optimal tool, and that perhaps this intuitive picture, in its simplicity, has hidden the real underlying structure (perhaps there's a lesson to be learnt here...).

We now consider two example calculations and show how diagrammatic methods can provide a simpler way of performing calculations in quantum mechanics. We begin with cluster state quantum computing [15] and then consider an example from condensed matter, the toric code [16]. We will compare how standard Dirac notation performs in these two examples against modern diagrammatic notations.

2.1 Cluster state quantum computing

Cluster state quantum computing [15], is a way of performing a quantum computation which has three stages. In the first a ‘cluster state’ is prepared. This is achieved by first initialising a number of qubits in the state $|+\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle$ and then applying a number of $CZ = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11|$ unitary operators to some pairs of these qubits. Any state formed in this way is a cluster state, but we will use a particular example here, a ‘linear cluster of four qubits’. We will label these qubits by $i \in 1, 2, 3, 4$.

The second stage involves measuring qubits in the bases $|\pm\alpha\rangle := (1/\sqrt{2})(|0\rangle \pm e^{i\alpha}|1\rangle)$, where a different value of the phase α is picked for each qubit. This “simulates” the unitary evolution process in standard quantum computing, this is what our example calculation will show.

The final stage is to apply some Pauli corrections to counteract for getting the ‘wrong’ measurement outcomes, this can be combined into the final measurement step in the computational basis.

In this example we show a standard result, that is that given a linear cluster where the first qubit is prepared in state $|\psi\rangle$ it is possible by performing three measurements to project the fourth qubit into the state $HU|\psi\rangle$ where the three angles correspond to the angles in an Euler decomposition of the unitary U.

Firstly using Dirac notation, we begin with the state,

$$|\psi_1 +_2 +_3 +_4\rangle,$$

apply the CZ unitaries,

$$CZ_{12}CZ_{23}CZ_{34}|\psi_1 +_2 +_3 +_4\rangle,$$

and then measure the first three qubits, assuming we get the correct (i.e. $+\alpha$ not $-\alpha$) outcomes, then (up to normalisation) the state of qubit 4 at the end of this process will be,

$$|\phi_4\rangle = \langle\alpha_1\beta_2\gamma_3|CZ_{12}CZ_{23}CZ_{34}|\psi_1 +_2 +_3 +_4\rangle.$$

We have suppressed identity unitaries in the above for simplicity but the meaning should be clear (for example $CZ_{23} := \mathbb{1}_1 \otimes CZ_{23} \otimes \mathbb{1}_4$). We define $|\psi\rangle := A|0\rangle + B|1\rangle$.

Now for the long and tedious calculation,

$$\begin{aligned}
|\phi_4\rangle &= \langle\alpha_1\beta_2\gamma_3|CZ_{12}CZ_{23}CZ_{34}|\psi_1 +_2 +_3 +_4\rangle \\
&= \langle\alpha_1\beta_2|CZ_{12}|\psi_1\rangle\langle\gamma_3|CZ_{23}CZ_{34}|+_2 +_3 +_4\rangle \\
&= \frac{1}{2}(((\langle 0| + e^{-i\alpha}\langle 1|)_1(\langle 0| + e^{-i\beta}|1\rangle)_2 CZ_{12}(A|0\rangle + B|1\rangle)\langle\gamma_3|CZ_{23}CZ_{34}|+_2 +_3 +_4\rangle \\
&= \frac{1}{2}((A + Be^{-i\alpha})\langle 0_2| + e^{-i\beta}(A - Be^{-i\alpha})\langle 1_2|)\langle\gamma_3|CZ_{23}CZ_{34}|+_2 +_3 +_4\rangle \\
&= \frac{1}{2}((A + Be^{-i\alpha})\langle 0_2| + e^{-i\beta}(A - Be^{-i\alpha})\langle 1_2|)\langle\gamma_3|CZ_{23}|+_2\rangle CZ_{34}|+_3 +_4\rangle \\
&= \frac{1}{4}((A + Be^{-i\alpha})\langle 0_2| + e^{-i\beta}(A - Be^{-i\alpha})\langle 1_2|)(\langle 0| + e^{-i\gamma}\langle 1|)CZ_{23}(|0\rangle + |1\rangle)CZ_{34}|+_3 +_4\rangle \\
&= \frac{1}{4}((A + Be^{-i\alpha} + e^{-i\beta}(A - Be^{-i\alpha}))\langle 0_3| + e^{-i\gamma}(A + Be^{-i\alpha} - e^{-i\beta}(A - Be^{-i\alpha}))\langle 1_3|)CZ_{34}|+_3 +_4\rangle \\
&= \frac{1}{8}((A + Be^{-i\alpha} + e^{-i\beta}(A - Be^{-i\alpha}))\langle 0_3| + e^{-i\gamma}(A + Be^{-i\alpha} - e^{-i\beta}(A - Be^{-i\alpha}))\langle 1_3|)(|00\rangle + |01\rangle + |10 \\
&= \frac{1}{4\sqrt{2}}((A + Be^{-i\alpha} + e^{-i\beta}(A - Be^{-i\alpha}))|+_4\rangle + e^{-i\gamma}(A + Be^{-i\alpha} - e^{-i\beta}(A - Be^{-i\alpha}))\langle -_4|)
\end{aligned}$$

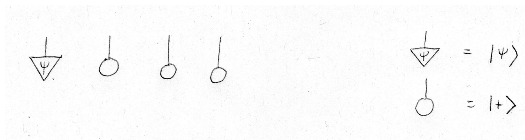
$$\begin{aligned}
H|\phi_4\rangle &= \frac{1}{4\sqrt{2}}(A((1 + e^{-i\beta})|0_4\rangle + (1 - e^{-i\beta})e^{-i\gamma}|1_4\rangle) + B((1 - e^{-i\beta})|0_4\rangle + (1 + e^{-i\beta})e^{-i\gamma}|1_4\rangle)) && (21) \\
&= U(A|0\rangle + B|1\rangle) && (22) \\
& && (23)
\end{aligned}$$

$$|\phi_4\rangle = HU|\psi\rangle$$

The above is an outline of the calculation, I got bored at the end and skipped a lot of it but the idea is there, its not difficult but not very fun either. The same calculation could be done with matrices but that would be even more tedious and doesn't highlight the physics of what is happening.

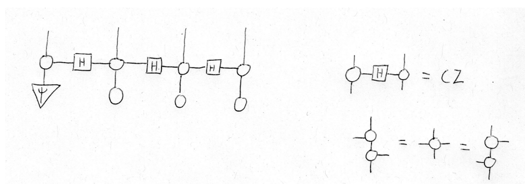
We next show how the calculation can be done using the notation of Coecke and Abramsky [17], we explicitly use this below and show what graphical identity is used at each step. Once one is relatively familiar with these the calculation can essentially be done without writing anything down apart from the initial picture of the calculation.

Again we initialise the state as $|\psi + ++\rangle$, this is drawn as,



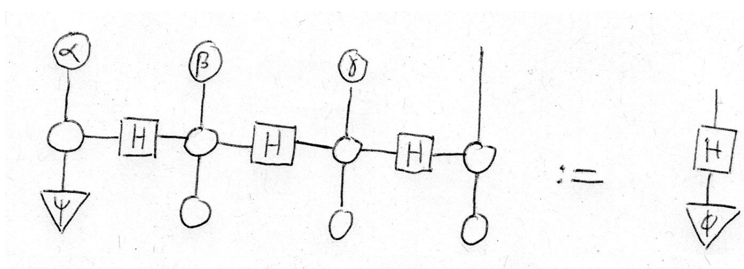
. (24)

apply the CZ operations, (there has actually been a graphical rule used already in this stage, the 'spider rule' which we will come back to later),



. (25)

measure the first three qubits,

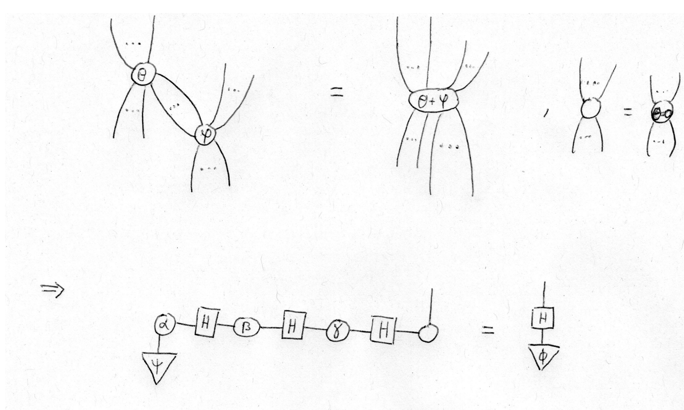


. (26)

.

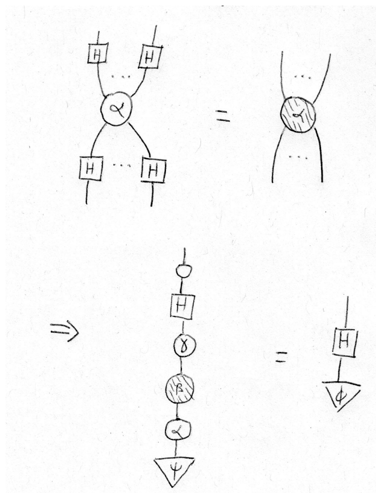
This is the initial diagram that one would write down, it is just a graphical description of the protocol being carried out.

Next we start using some basic rewrite rules that let us put this diagram into a simpler form, firstly the 'spider rule'



. (27)

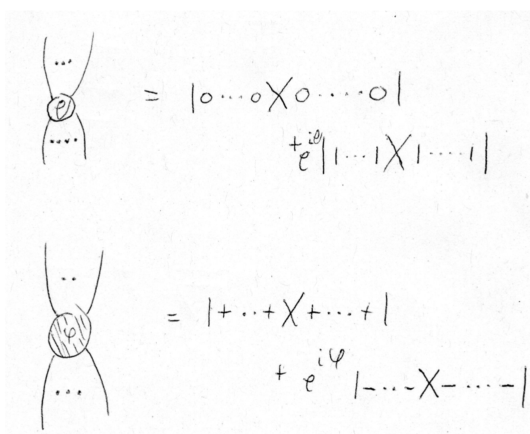
and then the 'colour change rule'



. (28)

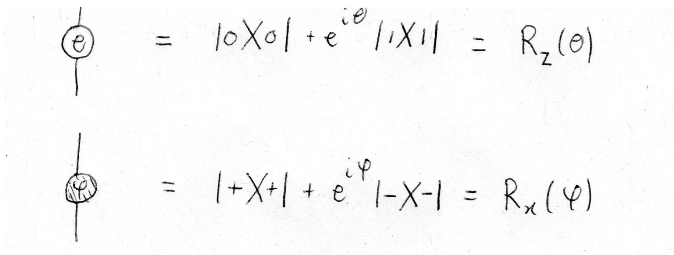
.

At this point it is worth mentioning what these 'spiders' actually are, they can be thought of either in terms of category theory, or more simply as a way of keeping track of Dirac notation, as Penrose's notation was for keeping track of tensors. Taking this approach the spiders can be defined as,



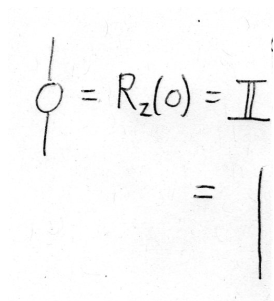
. (29)

the relevant spiders in our diagram can now be seen to be,



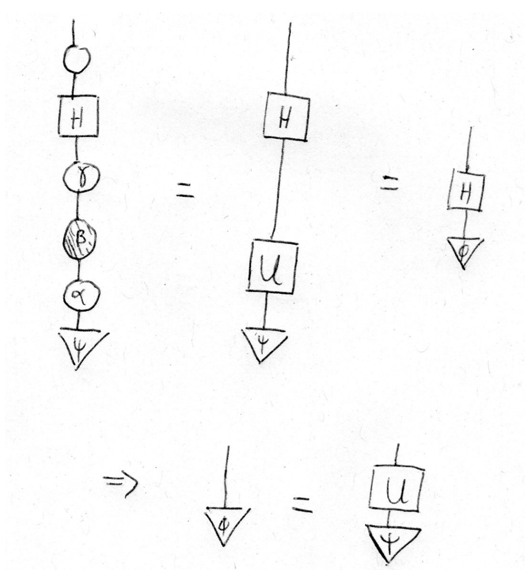
. (30)

,



. (31)

and so we can see that we have $R_z(\gamma)R_x(\beta)R_z(\alpha) = U(\alpha, \beta, \gamma)$ where this corresponds to an Euler decomposition of an arbitrary single qubit gate, U. Using these we can then finally write our diagram as,



. (32)

,

and see that we have obtained the result we were looking for.

There are other methods that can be used to solve this problem, the one most commonly used by people working on cluster states is ***check this*** a combination of cluster state diagrams (graph states), circuit diagrams and Dirac notation or alternatively using the stabiliser formalism provides a concise way of performing these calculations. The brute force Dirac calculation is probably not really used by anyone, particularly for more complicated situations.

Hardy [18] provides another graphical notation which looks fairly similar to circuit diagrams or string diagrams. This notation is very useful for Hardy's reformulation of quantum theory, in particular on providing an operationalist perspective on quantum theory, but they are difficult to use for practical calculations.

2.2 Toric code

The Toric code can be thought of as the ground state of a Hamiltonian of a square lattice. The Hamiltonian is,

$$H = -\sum_v A_v - \sum_p B_p,$$

where v labels the vertices of the lattice and p the plaquette, (i.e. the squares). $A_v := \prod_{i\in v} X_i$ and $B_p := \prod_{i\in p} Z_i$, where $i \in v$ means that the qubit i is attached to vertex v and $i \in p$ means that the qubit is part of the plaquette p.

Often it is a hard problem to take a given Hamiltonian and determine it's ground state, in this case it is relatively simple as all of the terms in the Hamiltonian commute. However for a large lattice it is difficult to write out the state as it will contain a large number of terms and so there have been recent developments in condensed matter to find efficient representations of states such as these.

These efficient representations are called 'tensor network states (TNS)' where different shaped networks are used for different situations such as 'matrix product states (MPS)'[19], 'Projected Entangled Pairs (PEPS)'[19], 'Multi-scale Entanglement Renormalisation Ansatz (MERA)'[4].

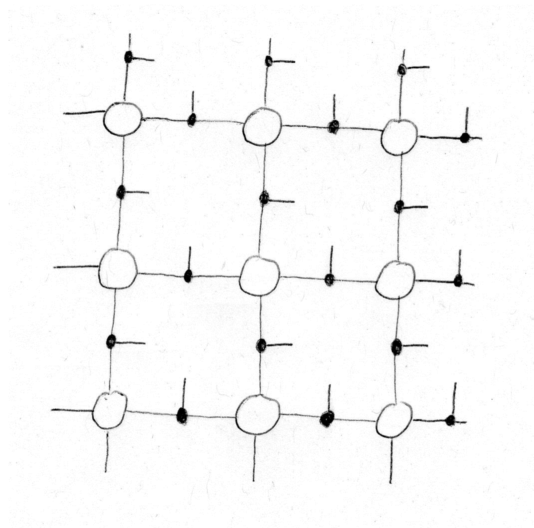
A generic quantum state can be written as,

$$|\psi\rangle = \sum_{i_1 \ldots i_n} \psi_{i_1 \ldots i_n} |i_1 ... i_n\rangle$$

, the coefficients $\psi_{i_1 \ldots i_n}$ can be thought of as a tensor. The basic idea of TNSs is that this tensor which contains 2^n complex coefficients, can for many interesting physical situations, be more efficiently represented by considering a network of smaller tensors.

Graphically this network of tensors can be written using Penrose graphical notation. Where lines connecting tensors are just virtual summation indices and unconnected lines correspond to the qubits. One could also interpret these diagrams in terms of string diagrams, where the connecting lines would be some quantum system being transferred between two quantum processes/interactions, unconnected lines would form the state once all interactions had occurred.

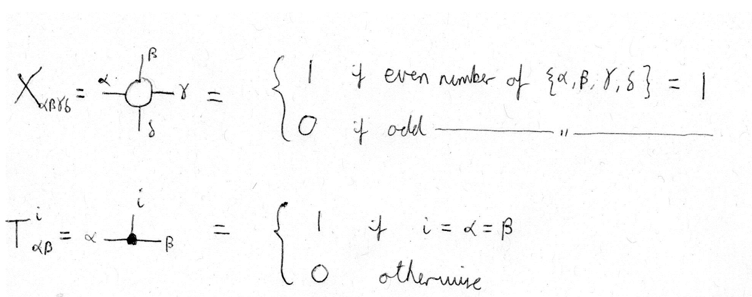
The Toric code as an efficient description as a PEPs tensor network[19],



(33)

.

Here we see that there are two types of tensor in our network, these are,



(34)

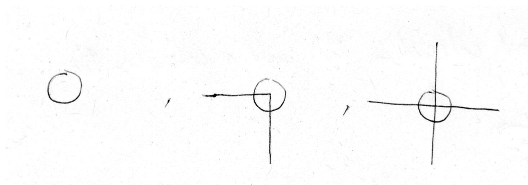
the X tensors are entirely virtual as all of their indices are contracted, the T tensors have a disconnected index which corresponds to the physical state of a qubit living at that point in the lattice. We use Greek indices for the virtual indices and roman letters for the state index.

Using this tensor network we can easily determine for any given configuration of qubits (i.e. state in the computational basis) what the amplitude for that state is.

By looking at this tensor network carefully one can see another interpretation of this toric code state, that is as a 'string net condensate'[20].

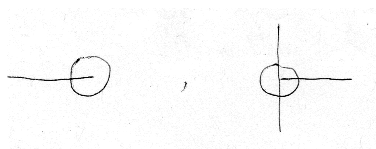
First note that a valid description of a configuration for the qubits can be graphically written by colouring in the qubits in state $|1\rangle$ and doing nothing to those in state $|0\rangle$. Equally valid would be to colour in the edge of the lattice that the $|1\rangle$ is on.

If we do this then we can note that the X tensors will only be non-zero when there are an even number of these coloured in edges going into them. e.g.



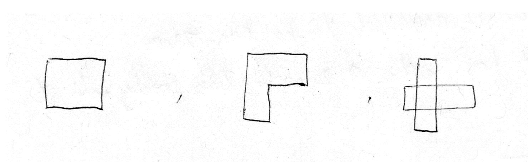
(35)

else they will give zero, e.g.,



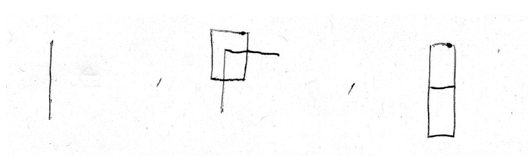
(36)

This means that over the lattice, we see that the coloured in lines can never end, as that would give an X tensor with either one of three coloured in lines. For example, configurations like,



(37)

are allowed, whereas,



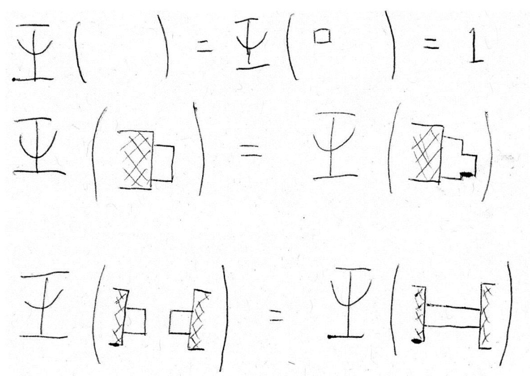
(38)

would give zero and so are forbidden.

The state described by this PEPs can then be seen to be an equal weighted superposition of all states which can be drawn as any number of closed strings. This is what is meant by a string net condensate.

In fact once we have this insight then this statement is sufficient to fully define the state, Wen [20] goes further than this though and shows that this global patten can be determined by a set of local graphical rewrite rules, 'dancing rules'. That is a set of operations that we can do to the state that leave it invariant. There is another view of the toric code in quantum information in terms of stabilisers which is essentially the same as this viewpoint.

The rules are,



(39)

this set of rules is sufficient to fully specify the state. This is a much more concise way of talking about the toric code then either the Dirac notation or the TNS description. Other sets of rules can be considered as well which would lead to different types of states in condensed matter such as, spin liquids, fractional quantum hall states, superfluids and many others.

So we see here that not only does a diagrammatic viewpoint lead to a much more efficient description of the physics, it also gives physical insight that can lead to generalisations and new discoveries.

2.3 Where is this calculatory "power" come from?

It is interesting to try to consider why it is that diagrams seem to have this ability to both simplify calculations as well as to provide physical insight into the world. One partial explanation may be that by using diagrams we are utilising a second dimension in our calculations, in standard mathematics are equations are strings of symbols in 1D, whereas when we move to using diagrams we utilise the full 2D plane of the paper we're writing on. If we then consider what we try to do in 1D mathematics it appears to be a compression of this 2D mathematics onto 1D. For example in process theories when we have both parallel and sequential composition we need to introduce $\circ$ and $\otimes$ to take this into account and then impose some axioms about how they interact. From this viewpoint the 2D mathematics is far more natural, and the compression to 1D hides the true nature of what is going on.

Yet despite the naturality of these diagrams, in all of the examples we are using above we are always at some level resorting to use of 1D mathematics. In terms of Penrose graphical notation it's built on the foundations of tensors and linear algebra, and the string diagrams are built on category theory, all of which are defined in 1D notation. Even in Hardy's notation which is built on the ideas of diagrams and composition, he provides a translation to 1D mathematics to persuade people of the rigour of his calculations.

Mathematics fundamentally can be viewed in terms of formal systems, an axiomatic way to determine the 'truth' of a set of symbols [21]. There is no a priori reason that this necessitates the symbols to be arranged in one dimensional strings. So we suggest that resorting to one dimensional mathematics is not inevitable and should be something we should work towards avoiding as diagrammatic reasoning appears to be a far more natural approach.

Another illustrative analogy that can be drawn is to computer science. In computer science everything at its most fundamental is based on long strings of binary, but no programmer would ever try to write a large program using this language. Instead there are many layers of abstraction between the machine code up to the programming language that is actually used. Similarly, we can view matrix calculations in quantum mechanics as the equivalent of binary and see first Dirac notation and then string diagrams built as 'higher level' languages on top of this. This analogy may be particularly pertinent to quantum computing if large scale quantum computers are ever realised.

The diagrams discussed in this section are not universally useful or applicable. The 'dancing rules' of Wen do not appear to be relevant to all instances where a TNS could be used. And cluster state diagrams and stabiliser formalism are not applicable to all quantum information protocols. Yet when they are useful they seem to be the most concise and elegant way to perform calculations. We see that their appears to be some trade off between how widely applicable a language is vs. how powerful it is for a particular situation. Another example is Hardy's duotensor notation [18], this is highly specialised for the particular reformulation of Hardy and is very powerful in that context, however in terms of general calculations in quantum mechanics it is much more cumbersome than the notation of Coecke and Abramsky.

3 Conclusion

3.1 Unification of the different notations

We have now seen various different examples of diagrammatic notation being used in physics. Many of these could be interpreted as diagrams in a monoidal category. All of the string diagrams, Penrose graphical notation and tensor networks [22] have a categorical interpretation. Feynman diagrams can also be interpreted as a string diagram [7]. So it seems like category theory may provide a rigorous underpinning for all of the diagrammatic notation used in physics, Baez and many others are working on turning anything that remotely looks like a network into diagrams in some category, from electric circuits [23] to Petri nets [24] to Bayesian networks[23].

One set of diagrams that we have talked about so far that hasn't been tied to category theory is the string nets of Xiao-Gang Wen , however it turns out that the possible states described by dancing rules is classified by using tensor category theory [20] so it seems that category theory is just under the surface here as well. The other diagram currently not linked to category theory are Hardy's operator tensor diagrams, as far as I know this has not yet been done but there is no obvious reason why they could not be interpreted as diagrams in a monoidal category.

It is not yet clear whether category theory will underly all diagrammatic languages but it is clear that it is a powerful unifying tool in physics as it was in mathematics. It provides a broad framework to work upon which can describe a wide range of physics, mathematics and computer science whilst once specialised for a particular task, e.g. the string diagrams of Coecke and Abramsky, also becomes a powerful tool for calculation. The hope is that the simplicity of the notation and graphical manipulations will lead to not just easier pen and paper calculations but also the possibility of automated reasoning [25].

3.2 Hardy's compositional principle

The line between diagrams used to describe a situation and a mathematical diagrammatic notation are often blurred. For example in the cluster state notation one could imagine that we had four qubits that were actually in a line, that then were passed through some entangling gates and then measured, the diagram could then correspond to just a picture of the experiment, where boxes represent the real physical processes and the lines real physical systems. Equally well the diagrams can be viewed purely as mathematical notation, used to calculate the final state. Hardy [26] takes this idea as a primitive and suggests that there should be a 'compositional principle' that theories should aspire to, that is that the mathematical calculation performed should be composed in the same way as the physical objects which the calculation is about.

From a category theory point of view this amounts to saying that if we can represent the composition of some object by a diagram in some 'real world' category, then their should exist some functor from that category to a concrete category in which the properties of that object can be calculated.

3.3 Beyond 2D

Throughout this review we have highlighted the increased descriptive power and ease of calculation that arises from moving from 1D to 2D mathematics, it is natural to consider what happens if we move beyond 2D. Within the context of monoidal categories it can be said that if we introduce a 'braiding' operator, i.e. one that interchanges lines, then this gives a 3D diagram as now we draw objects crossing which must take place off the plane. Symmetric monoidal categories are those in which if we apply this braiding twice we end up with the same, i.e. we can unknot any lines, this can be viewed as drawing a 4D diagram as in 4D all knots can be trivially untied. So in some sense we can already depict higher dimensions within the same framework.

This is not the whole story thought, we are still limiting ourselves to 1D strings living in potentially higher dimensional spaces. We can also consider what occurs if we allow for (hyper) surfaces, these sorts of objects arise naturally when one considers n-categories, for example in a string diagram for a 2-Category the diagrammatic elements are given by, points, lines and surfaces, there is recent work in using bicategories for quantum protocols [27].

3.4 Summary

To briefly summarise, in this overview we began by discussing some of the history of diagrammatic notations in physics, the original example being Penrose's graphical notation for tensors. We then briefly discussed how category theory plays a role in giving a formal mathematical basis for many of the diagrams that we use today in physics despite its routes lying in highly abstract mathematics. We then gave a couple of examples in which diagrams can be used to make calculations and descriptions of states far simpler, cluster state quantum computing and the toric code. Finally looking at what it is that gives these diagrammatic notations their power and considering what the natural extensions to these could be in terms of higher category theory.

I have mainly highlighted the use of these diagrammatic notations for their ability to make calculations easier and more natural, my main hope however is that by viewing quantum theory through this lens we will be able to gain a deeper understanding of it.

References

[1] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 221244, 1971.

[2] Predrag Cvitanovic. *Group Theory: Birdtracks, Lie's, and Exceptional Groups*. Princeton University Press, 2008.

[3] C Rovelli and L Smolin. Spin networks and quantum gravity. *Physical review D: Particles and fields*, 52(10):5743–5759, November 1995.

[4] Guifre Vidal. Entanglement Renormalization: an introduction. *arXiv preprint arXiv:0912.1651*, page 24, December 2009.

[5] Samuel Eilenberg and Saunders MacLane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945.

[6] Joseph A. Goguen. A categorical manifesto. In *Mathematical Structures in Computer Science*, pages 49–67, 1991.

[7] John C Baez and Aaron Lauda. A Prehistory of n -Categorical Physics. pages 1–129, 2009.

[8] Samson Abramsky and Nikos Tzevelekos. Introduction to categories and categorical logic. In *New structures for physics*, pages 3–94. Springer, 2011.

[9] André Joyal and Ross Street. The geometry of tensor calculus ii. *Draft available at http://www. math. mq. edu. au/˜ street/GTCII. pdf*, 585, 1991.

[10] Bob Coecke and Éric Oliver Paquette. Categories for the practising physicist. In *New structures for physics*, pages 173–286. Springer, 2011.

[11] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols.

[12] Peter Selinger. A survey of graphical languages for monoidal categories. In *New Structures for Physics*, pages 289–355. Springer Berlin/Heidelberg, 2011.

[13] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to Quantum Field Theory*. Perseus Books, Cambridge, Massachusetts, 1995.

[14] Nima Arkani-Hamed and Jaroslav Trnka. The Amplituhedron. page 36, December 2013.

[15] Michael A Nielsen. Cluster-state quantum computation. XX(X):15, April 2005.

[16] AY Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, page 27, July 2003.

[17] Bob Coecke. Quantum Picturalism. *Contemporary physics*, page 31, August 2009.

[18] Lucien Hardy. The Operator Tensor Formulation of Quantum Theory. page 37, January 2012.

[19] Roman Orus. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. page 51, June 2013.

[20] Xiao-gang Wen. Topological order: from long-range entangled quantum matter to a unified origin of light and electrons. pages 0–42.

[21] H.B. Curry. *Outlines of a Formalist Philosophy of Mathematics*. Studies in logic and the foundations of mathematics. Elsevier Science, 1951.

[22] JD Biamonte, SR Clark, and Dieter Jaksch. Categorical tensor network states. *AIP Advances*, page 39, December 2011.

[23] Brendan Fong. Causal Theories: A Categorical Perspective on Bayesian Networks. page 72, January 2013.

[24] John C Baez and Jacob D Biamonte. A Course on Quantum Techniques for Stochastic Mechanics.

[25] Lucas Dixon and Ross Duncan. Extending graphical representations for compact closed categories with applications to symbolic quantum computation. In Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics, 9th International Conference, AISC 2008, 15th Symposium, Calculemus 2008, 7th International Conference, MKM 2008, Birmingham, UK, July 28 - August 1, 2008. Proceedings*, volume 5144 of *Lecture Notes in Computer Science*, pages 77–92. Springer, 2008.

[26] Lucien Hardy. On the theory of composition in physics. *arXiv preprint arXiv:1303.1537*, March 2013.

[27] Mike Stay and Jamie Vicary. Bicategorical Semantics for Nondeterministic Computation. page 21, January 2013.