Meromorphic Quantum Computing

Simon Burton, Hussain Anwar
Quantinuum

We consider the kinematic axioms of quantum mechanics projectively. Instead of normalized (pure) states up to global phase, states become one-dimensional subspaces of vector spaces. This process of projectivization is functorial and lax monoidal. For qubits it identifies the Bloch sphere with the Riemann sphere. We interpret a fragment of the ZXW-calculus projectively and thereby provide an alternate derivation of the arithmetic GHZ/W-calculus of Coecke et al. We find meromorphic functions that characterize the coherent behaviour of circuits for logical state preparation of quantum codes and magic state distillation.

1 Introduction or: How I stopped worrying and learnt to love the Bloch sphere

To a mathematician a *qubit* is the complex projective line or *Riemann sphere*. To a physicist, this sounds like crazy-talk, or at best a gimmick. What is going on? It turns out there are good reasons for this language, although it may not be immediately obvious. We end up with a description of quantum states and operators that looks quite different to the textbook linear-algebra version, and we spend some time to dwell on the differences. This formulation of quantum mechanics using projective geometry is found in the literature [7, 17, 16] which we review below.

The usual axioms of quantum mechanics (see [15] §2.2) begin with defining a *state space* as a Hilbert space $\mathscr{H}$, and (pure) *states* as being unit vectors therein. Moreover, we are only interested in states up to *global phase*. Unit vectors $v \in \mathscr{H}$ modulo global phase $\phi \in U(1)$ are in one-to-one bijection with 1-dimensional subspaces of $\mathscr{H}$. This is called the *projective space* of $\mathscr{H}$, denoted as $\mathscr{P}(\mathscr{H})$:

$$\mathscr{P}(\mathscr{H}) := \{v \in \mathscr{H} - \{0\}\}/\sim \text{ with } u \sim \lambda u, \lambda \in \mathbb{C} - \{0\}.$$

These are compact manifolds. The zero Hilbert space 0 is sent to the empty manifold $\mathscr{P}(0) = \phi$.

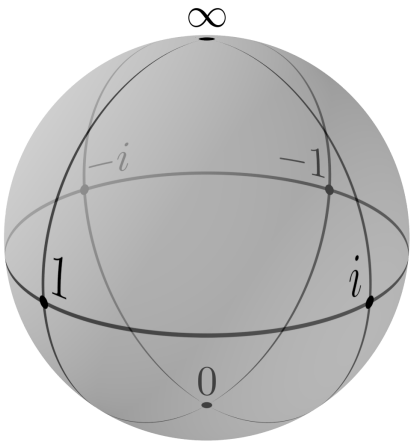
For our purposes we will restrict to finite dimensional complex vector spaces $\mathbb{C}$FVec. The projective space of the d-dimensional space $\mathbb{C}^d$ we write as $\mathbb{P}^{d-1} := \mathscr{P}(\mathbb{C}^d)$. It follows that any invertible linear map $T \in \mathrm{GL}(d,\mathbb{C})$ respects the equivalence relation $\sim$ and therefore acts on $\mathbb{P}^{d-1}$.

Proposition 1.1. *Projectivization $\mathscr{P}$ is a functor from the groupoid of finite dimensional complex vector spaces and invertible linear maps to the category of compact manifolds $\mathscr{P} : \mathscr{GL}(\mathbb{C}) \to$ CMan.*

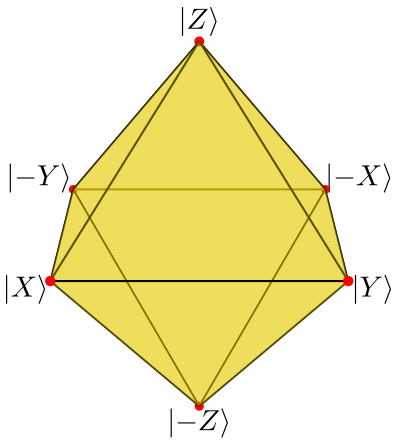
A common notation for elements of $\mathbb{P}^{d-1}$ is $[v] = [v_0 : \ldots : v_{d-1}]$ with $v_i \in \mathbb{C}$ and $[v_0 : \ldots : v_{d-1}] = [\lambda v_0 : \ldots : \lambda v_{d-1}]$ for $\lambda \in \mathbb{C} - \{0\}$. The zero based indexing is supposed to remind us that we lost a degree of freedom when projectivizing.

The textbook treatment of quantum mechanics picks out states using unit vectors as a canonical representative, which leaves some annoying fluff on the carpet in the form of a global phase. In other words, this fails as a (unique) canonical representative. When we projectivize instead, we pick out canonical vectors by just choosing $v_i = 1$ for some i. This gets rid of all the fluff, but now there is a hole in the carpet, corresponding to states with $v_i = 0$. These are states "at infinity". So we have traded in our old shaggy carpet, but at the cost of now needing multiple carpets, each with missing points at infinity.

Submitted to:
QPL 2026



(i) The Riemann sphere: states are labelled by a point z in the extended complex plane.



(ii) The Bloch octahedron: states are labelled by +1 eigenvectors of Pauli matrices.

Figure 1: Two ways to label qubit states.

A particularly nice thing happens with *qubits* which have state space $\mathbb{P}^1$. We can label the points $\begin{bmatrix} z \\ 1 \end{bmatrix}$ with a single variable $z \in \mathbb{C}$, which leaves out only one point, $\infty := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. The *extended complex plane* is defined as $\widehat{\mathbb{C}} := \mathbb{C} \cup \{\infty\} = \mathbb{P}^1$. This space is also called the *Riemann sphere*. See Fig. 1i.

Matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{GL}(2,\mathbb{C})$ then act on qubits as $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{bmatrix} z \\ 1 \end{bmatrix} = \begin{bmatrix} az+b \\ cz+d \end{bmatrix} = \begin{bmatrix} \frac{az+b}{cz+d} \\ 1 \end{bmatrix}$. This map $z \mapsto \frac{az+b}{cz+d}$ is known as a fractional linear transform, or *Möbius transform*. The point at infinity goes to a/c if $c \neq 0$ otherwise to ∞.

Going forwards we will be loose with evaluating expressions such as $1/z$ at $z = 0$, which is ∞, and so on.

For unitary operators $g \in U(2)$ the maps $\mathscr{P}(g)$ are rotations, $\mathscr{P}(g) \in SO(3)$. Below we use the notation $g(z) := \mathscr{P}(g)(z)$ for $z \in \widehat{\mathbb{C}}$.

Once we apply $\mathscr{P}$ to a linear automorphism, we will loose information about the complex eigenvalues, but not the eigenvectors.

Proposition 1.2. *Eigenvectors of linear operators $T \in \mathrm{GL}(n,\mathbb{C})$ correspond bijectively to fixed points of $\mathscr{P}(T) : \mathscr{P}(\mathbb{C}^n) \to \mathscr{P}(\mathbb{C}^n)$.*

This proposition is slightly confusing in the case of qubits: the projective fixed points are complex values $z \in \widehat{\mathbb{C}}$, but these correspond to eigen*vectors* of the linear operator. We have forgotten the eigen*value*.

In the following table we show *Pauli matrices* $g \in U(2)$ together with the action on $\widehat{\mathbb{C}}$ and the +1

eigenvectors $|g\rangle$ as fixed points $g(z) = z$:

$g \in U(2)$	$z \mapsto g(z)$	$\lvert\pm g\rangle$	$g(z) = z$
$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$Z(z) = -z$	$\lvert\pm Z\rangle$	$z = 0, \infty$
$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$X(z) = 1/z$	$\lvert\pm X\rangle$	$z = \pm 1$
$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$Y(z) = -1/z$	$\lvert\pm Y\rangle$	$z = \pm i$

This correspondence between points of the Riemann sphere and eigenvectors of operators is illustrated in Fig. 1. These six octahedral vertices label the *stabilizer states*. The matrices $\{X, Y, Z\}$ generate the *single qubit Pauli group* P_1 of order sixteen. The image $\mathscr{P}(\mathrm{P}_1)$ is a group of order four isomorphic to the Klein-four group $\mathbb{Z}_2 \times \mathbb{Z}_2$. As well as the identity rotation, these are the order two rotational symmetries of the Bloch octahedron that fix opposite vertices.

In total there are twenty-four rotational symmetries of the octahedron, and this group is isomorphic to S_4. These rotations all come from the image $\mathscr{P}(\mathrm{Cliff}_1)$ of the *single qubit Clifford group* generated by matrices $\{S, H\}$ where

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

These act on the Riemann sphere as

$$S(z) = -iz, \quad H(z) = \frac{z+1}{z-1}.$$

The complete list of non-trivial rotations that fix octahedral-edges follows, along with fixed points $z \in \widehat{\mathbb{C}}$ and minimal polynomial $p(z) \in \mathbb{Z}[z]$ for those fixed points:

$g(z) = z$	$\lvert g\rangle$	$p(z)$
$1 \pm \sqrt{2}$	$\lvert\pm H\rangle$	$z^2 - 2z - 1$
$-(1 \pm \sqrt{2})$	$\lvert\pm XZH\rangle$	$z^2 + 2z - 1$
$+i(1 \pm \sqrt{2})$	$\lvert\pm ZSHS\rangle$	$z^4 + 6z^2 + 1$
$-i(1 \pm \sqrt{2})$	$\lvert\pm YSHS\rangle$	$z^4 + 6z^2 + 1$
$\pm(1+i)/\sqrt{2}$	$\lvert\pm 1^{3/8} XS\rangle$	$z^4 + 1$
$\pm(1-i)/\sqrt{2}$	$\lvert\pm 1^{3/8} YS\rangle$	$z^4 + 1$

We call any of these fixed points *H-states*.

The non-trivial rotations that fix an octahedral-face are $F := SH$ and XF, ZF, YF. The fixed points, one at the center of each face, are $z = \pm(1 \pm \sqrt{3})(1 \pm i)/2$. These are the eight complex roots of the polynomial

$$z^8 + 14z^4 + 1 = (z^4 - 2z^3 + 2z^2 + 2z + 1)(z^4 + 2z^3 + 2z^2 - 2z + 1).$$

We call any of these eight fixed points *F-states*.

The stabilizer states, H-states and F-states are also shown in Fig. 2 below.

1.1 The impossible outcome

For us, measurement in quantum mechanics will be axiomatised via postselection maps. These are rank degenerate linear maps Π, which have non-trivial nullspace in general. The zero vector $\Pi(v) = 0$ is interpreted as an impossible outcome of the measurement. However none of the projective spaces have an element representing the zero vector, so we cannot just apply $\mathscr{P}$ to Π.

The solution we take is to adjoin (disjointly) an *impossible* element $\perp$ to all our projective spaces:

$$\mathscr{P}_\perp(\mathbb{C}^n) = \mathscr{P}(\mathbb{C}^n) \cup \{\perp\}.$$

Instead of topologizing this pointed space, we just take these *pointed sets*. This category is denoted $\mathrm{Set}_\perp$: objects are sets with a distinguished element and *pointed functions* that preserve the distinguished element. Given a linear map $T : \mathbb{C}^n \to \mathbb{C}^m$, we define the pointed function:

$$\mathscr{P}_\perp(T)(\perp) = \perp, \quad \mathscr{P}_\perp(T)([v]) = \begin{cases} [T(v)] & \text{if } T(v) \neq 0, \\ \perp & \text{otherwise.} \end{cases}$$

Proposition 1.3. *We have a functor $\mathscr{P}_\perp : \mathbb{C}FVec \to Set_\perp$ from the category of finite dimensional complex vector spaces to the category of pointed sets.*

We also think of pointed functions as partially defined functions. The *domain of definition* of a pointed function $f : A \cup \{\perp\} \to B \cup \{\perp\}$ is the set $\{x \in A | f(x) \neq \perp\}$, and $f(z)$ is *undefined* at z when $f(z) = \perp$.

1.2 And

The single qubit statespace $\mathbb{P}^1$ is a one-dimensional complex manifold: it has one (complex) degree of freedom. Therefore, preparing two qubits separably (classically), is the product space $\mathbb{P}^1 \times \mathbb{P}^1$ which has two degrees of freedom. The whole two qubit statespace is $\mathbb{P}^3 = \mathscr{P}(\mathbb{C}^4)$, and these states have three degrees of freedom, such as $[u : v : w : 1]$. Therefore, the two qubit separable statespace is a subspace of the two qubit statespace. The embedding $\mathbb{P}^1 \times \mathbb{P}^1 \rightarrowtail \mathbb{P}^3$ is given by the projectivization of the tensor product $([u : v], [w : z]) \mapsto [uw : uz : vw : vz]$. In general this is called the *Segre embedding* (see [7]):

$$\Sigma_{a,b} : \mathbb{P}^{a-1} \times \mathbb{P}^{b-1} \rightarrowtail \mathbb{P}^{ab-1}$$
$$\big([v], [w]\big) = \big([v_0 : \ldots : v_{a-1}], [w_0 : \ldots : w_{b-1}]\big) \mapsto [v_0 w_0 : v_0 w_1 : \ldots : v_{a-1} w_{b-1}] = [v \otimes w].$$

We will also need higher arity Segre embeddings, $\Sigma_{a,b,c,\ldots}$ which are defined straightforwardly, as well as the unit Segre embedding $\Sigma_a : \mathbb{P}^{a-1} \to \mathbb{P}^{a-1}$ which is just the identity.

The cartesian category of projective spaces we are taking is just sets with cartesian product: $\mathrm{Set}_\times$. When we include the impossible element, we must take monoidal product to be the *smash product*, written as $\wedge$. For pointed sets $A \cup \{\perp\}$ and $B \cup \{\perp\}$ their smash product is $(A \times B) \cup \{\perp\}$. The monoidal unit is the two element set $1_\wedge = \{\star\} \cup \{\perp\}$. The smash product of pointed functions f, g is

$$(f \wedge g)(\perp) = \perp, \quad (f \wedge g)(a, b) = \begin{cases} (f(a), g(b)) & \text{if } f(a) \neq \perp \text{ and } g(b) \neq \perp, \\ \perp & \text{otherwise.} \end{cases}$$

The monoidal category of pointed functions and smash product is written $\mathrm{Set}_{\perp,\wedge}$.

Extending the Segre embedding to pointed projective spaces is also straightforward. The pointed Segre embedding, also written as Σ, is the map:

$$\begin{aligned} \mathbb{P}_\perp^{a-1} \wedge \mathbb{P}_\perp^{b-1} &\to \mathbb{P}_\perp^{ab-1} \\ \perp,\ ([u],\perp),\ (\perp,[v]) &\mapsto \perp \\ ([u],[v]) &\mapsto \Sigma_{a,b}([u],[v]). \end{aligned}$$

Proposition 1.4. *Using the pointed Segre embedding as laxator, $\mathscr{P}_\perp$ becomes a lax monoidal functor*

$$\mathscr{P}_\perp : \mathbb{C}FVec_\otimes \to Set_{\perp,\wedge}.$$

Proof. See App. A. □

1.3 Summary

In summary, the kinematic axioms of projective quantum mechanics are:

(1) states are points in projective spaces, or otherwise impossible

(2) operators are maps of projective spaces that preserve the impossible state

(3) composition of states is lax monoidal

2 Meromorphic functions

A *meromorphic function* on the Riemann sphere is a fractional linear transform

$$f(z) = \frac{p(z)}{q(z)}$$

where $p, q \in \mathbb{C}[z]$ are polynomials in z with coefficients in $\mathbb{C}$, without any roots in common, and $q \neq 0$. [1] This set of meromorphic functions is the field of *rational functions in one variable*, denoted $\mathbb{C}(z)$. (See [8] Prop. 1.26.) The *degree* of f is the maximum of the degrees of p and q. (See [8] Ex. 1.58.) This counts, with multiplicity, the number of points in any fiber $f^{-1}(w)$ over $w \in \widehat{\mathbb{C}}$. A *branch point* of f is $z \in \widehat{\mathbb{C}}$ such that $f'(z) = 0$. These are the stationary points of f. The multiplicity of such a branch point is the "number of sheets that meet at z", written $m_z(f)$. In practice, this is one plus the order of the zero of $f'(z)$. To find all the stationary points sometimes we need to change variables, such as $z \mapsto 1/z$, in order to deal with the point at infinity.

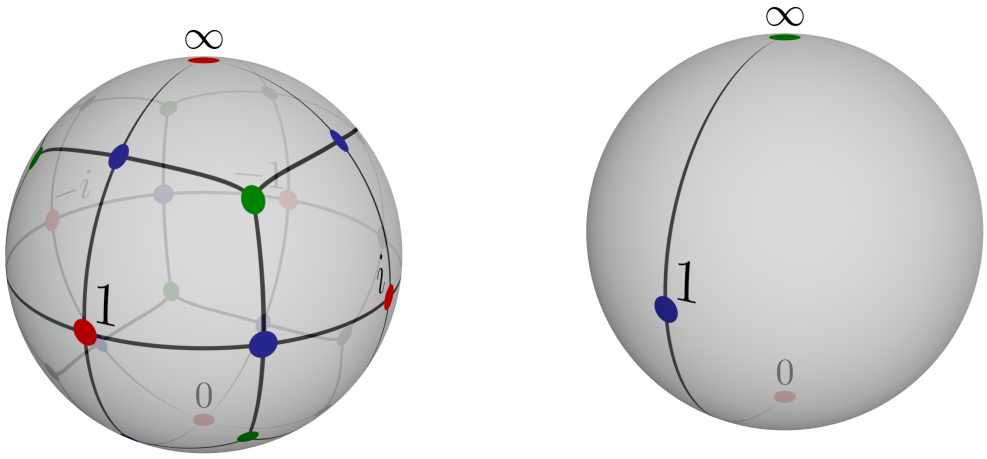
The following theorem is a baby version of the Riemann-Hurwitz theorem (see [8] Thm. 1.60.):

Theorem 2.1. *Given $f \in \mathbb{C}(z)$ with degree d and branch points $B \subset \widehat{\mathbb{C}}$, we have*

$$\sum_{z \in B} (m_z(f) - 1) = 2(d-1).$$

We also use *multivariate meromorphic functions* $f(z_1, ..., z_n)$ which are defined to be quotients p/q of polynomials $p, q \in \mathbb{C}[z_1, ..., z_n]$ without common zeros, and $q \neq 0$.

[1] At times we will forget the distinction between formal elements such as $p \in \mathbb{C}[z]$ and functions $p : \mathbb{C} \to \mathbb{C}$.



(i) The Riemann sphere $\widehat{\mathbb{C}} = \mathbb{P}^1$ with the six stabilizer states in red, twelve H-states in blue, and eight F-states in green.

(ii) The Riemann sphere with the point $z = 0$ in red, $z = 1$ in blue, and $z = \infty$ in green. This is the image of Klein's octahedral function E_7.

Figure 2: We show an octahedral "design" on the Riemann sphere, and its image under E_7 on the right. Each of the twenty-four tiles on the left is mapped to a single region on the right which covers the sphere.

Proposition 2.2. *Given a linear operator $T : (\mathbb{C}^2)^{\otimes n} \to \mathbb{C}^2$, the function $f : \bigtimes_n \mathbb{P}^1 \to \mathbb{P}^1$ given by $f = \mathscr{P}_\perp(T)\Sigma_{2,\ldots,2}$ is a meromorphic function of n variables on its domain of definition:*

$$f(z_1,\ldots,z_n) = \begin{cases} \frac{p(z_1,\ldots,z_n)}{q(z_1,\ldots,z_n)} \\ \\ \perp \end{cases}$$

Proof. Applying the Segre embedding to the n-qubit state $([z_1 : 1],\ldots,[z_n : 1])$ gives $[z_1 z_2 \ldots z_n : \ldots : 1]$, where each entry is a monomial in the z_i or 1. The action of T on this (projective) vector is $[p(z_1,\ldots,z_n) : q(z_1,\ldots,z_n)] = [\frac{p(z_1,\ldots,z_n)}{q(z_1,\ldots,z_n)} : 1]$ where $p,q \in \mathbb{C}[z_1,\ldots,z_n]$ with common zeros eliminated. □

2.1 Klein's Octahedral function

We define a meromorphic function, the *octahedral function* as

$$E_7(z) := \frac{108z^4(z^4-1)^4}{(z^8+14z^4+1)^3}$$

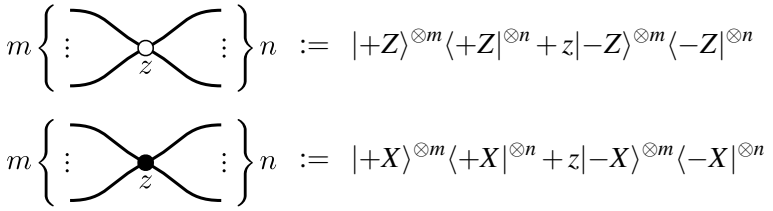
This is used to detect when points are related by a Clifford rotation. We could also think of this as a coequalizer of the image of Cliff_1 under $\mathscr{P}$.

Proposition 2.3. *Given $w,z \in \widehat{\mathbb{C}}$, then $w \in Orbit_{Cliff_1}(z)$ if and only if $E_7(w) = E_7(z)$.*

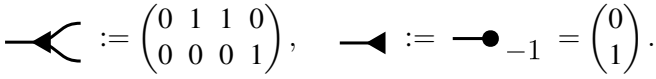
The proof of this proposition goes back to Klein's book from 1888 [13] on invariant theory and is connected to the modern theory of modular curves [21], and dessin d'enfants [8]. See Fig. 2.

3 Circuits and diagrams

We recall the tensor network notation known as ZX-calculus [19]. Our diagrams flow from right-to-left, in agreement with algebraic (Dirac) notation. We define the Z-type (white) or X-type (black) *spider* with m outputs and n inputs, labelled with a non-zero *multiplicative phase* $z \in \mathbb{C}/\{0\}$:



When the multiplicative phase z is omitted, it defaults to $z = 1$. We also define the following two W-*spiders*:



We recall the three structures $\left(\text{—○} , \text{—○<} \right)$, $\left(\text{—●} , \text{—●<} \right)$ and $\left(\text{—◀} , \text{—◀<} \right)$ each give complex unital associative algebras on $\mathbb{C}^2$ [11, 4].

Proposition 3.1. *Projectivization induces the three (associative, unital) monoid structures on the object* $\mathbb{P}^1_\perp$ *in* $Set_{\perp,\wedge}$*:*

$\mathbb{C}$*FVec matrix*	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}1&0&0&0\\0&0&0&1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1&0&0&1\\0&1&1&0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0&1&1&0\\0&0&0&1\end{pmatrix}$
diagram	—○	—○<	—●	—●<	—◀	—◀<
projectivization	$1 \leftarrow$	$\left.\begin{matrix} wz \\ \perp \end{matrix}\right\} \leftarrow (w,z)$	$\infty \leftarrow$	$\left.\begin{matrix} \frac{wz+1}{w+z} \\ \perp \end{matrix}\right\} \leftarrow (w,z)$	$0 \leftarrow$	$\left.\begin{matrix} w+z \\ \perp \end{matrix}\right\} \leftarrow (w,z)$
undefined at		$(\infty,0),(0,\infty)$		$(1,-1),(-1,1)$		(∞,∞)

Proof. The formulas come from applying the matrix to projective coordinates given by the Segre embedding. These give monoid structures either by direct inspection, or by noting that lax monoidal functors preserve monoids. □

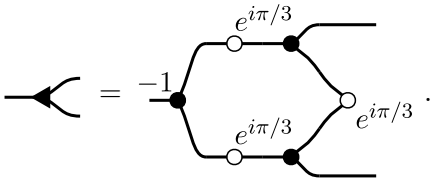
In Ref.[5] they build the field of rational numbers formally out of the Z and W spider, which forms the ZW-calculus [4]. We now see that this comes directly from reformulating linear algebra as projective algebraic geometry: the multiplication and addition of complex numbers is the projectivization of the Z- and W-spiders.

But what about the X-spider? Taking the function $(w,z) \mapsto \frac{wz+1}{w+z}$ and switching to the reciprocal chart $z \mapsto 1/z$ of the Riemann sphere, we find

$$(w,z) \mapsto \frac{w+z}{wz+1}$$

which is the familiar formula for Lorentz velocity addition.

From [3], page 28, we see the W-spider is expressed in ZX-calculus as



With these tools we can now express such things as Mandelbrot iteration $z \mapsto z^2 + c$ which belong in the field of "complex dynamics", see [21]§6.

4 Application to stabilizer codes

The n-qubit *Pauli group* denoted P_n is generated by n-fold tensor products $\otimes_{i=1}^n g_i$ with g_i an element of the (1-qubit) Pauli group. We also write indexed Pauli group elements such as X_i to mean a tensor product $I \otimes ...X... \otimes I$ with X in the i'th component, etc. An *X-type* Pauli operator is a tensor product of $I, X \in \mathrm{P}_1$. Similarly for *Z-type* Pauli operators. The n-qubit *Clifford group* denoted Cliff_n is the normalizer of P_n in the unitary group $U(2^n)$. It acts on P_n by conjugation. An element $g \in \mathrm{P}_n$ has *weight* $w(g)$ which counts how many of the n components in the n-fold tensor product g are not proportional to the identity.

A (quantum stabilizer) *code* [9] on n-qubits, denoted by C, is specified by an abelian subgroup $S \subset \mathrm{P}_n$ such that $-I \notin S$. The *logical subspace* of C is the linear subspace $\{v \in (\mathbb{C}^2)^{\otimes n} | gv = v \text{ for all } g \in S\}$. Given independent stabilizer generators $S = \langle g_i \rangle_{i=1}^m$, we find k pairs of anti-commuting independent *logical operators* $\{L_{X_i}, L_{Z_i}\}_{i=1}^k$, that together with S generate the centralizer $C_{\mathrm{P}_n}(S)$. An *encoder* is a unitary $E \in \mathrm{Cliff}_n$ such that $E^\dagger X_i E = g_i$ for $i = 1, .., m$ and $E^\dagger X_{i+m} E = L_{X_i}, E^\dagger Z_{i+m} E = L_{Z_i}$ for $i = 1, ..., k$. The *logical encoder* is the linear map $L = E(|Z\rangle^{\otimes m} \otimes I^{\otimes k})$. The *logical decoder* is $L^\dagger$. The projector onto the logical subspace is then $\Pi = LL^\dagger$. The *distance* of a code d is the minimum weight of a logical operator up to multiplication by a stabilizer. We will take the label C to denote any or all of this data as needed; the notation $[[n,k,d]]$ is also used to refer to these parameters.

Proposition 4.1. *Given a code C with parameters $[[n,1,d]]$ and logical decoder $L^\dagger$, the function $f : \bigtimes_n \mathbb{P}^1 \to \mathbb{P}^1$ given by $f = \mathscr{P}_\perp(L^\dagger)\Sigma_{2,...,2}$ is a meromorphic function of n variables on its domain of definition.*

Proof. Apply proposition 2.2 to the linear operator $L^\dagger$. □

For simplicity of exposition, we frame the following results in terms of a univariate meromorphic function $f(z) = f(z, ..., z)$. We call such a function $f(z)$ the *meromorphic decoder* of C. We say that a meromorphic decoder $f(z)$ *weakly distills* a point $z \in \widehat{\mathbb{C}}$ if z is a stationary point of $f(z)$. The degree of such a stationary point is $m_z(f)$ and we call this *coherent error suppression* to order $O(\varepsilon^{m_z(f)})$.

Theorem 4.2. *Given a meromorphic decoder of degree n that weakly distill points $D \subset \widehat{\mathbb{C}}$ then*

$$2(n-1) = \sum_{z \in D} (m_z(f) - 1).$$

Proof. This is a restatement of Thm. 2.1. □

We say a meromorphic decoder $f(z)$ *distills* a point $z \in \widehat{\mathbb{C}}$ when $f(z) = z$, and *distills up to a Clifford* when $f(z) = g(z)$ with $g \in \text{Cliff}_1$. Furthermore, $f(z)$ *coherently distills* a point $z \in \widehat{\mathbb{C}}$ when $f'(z) = 0$ and also $f(z) = z$, and *coherently distills up to a Clifford* if $f'(z) = 0$ and $f(z) = g(z)$ with $g \in \text{Cliff}_1$.

Given a code C with encoder E, the *dual code* $C^\top$ has encoder $H^{\otimes n}E$.

Proposition 4.3. *Given $k = 1$ code C with meromorphic decoder f, and dual code $C^\top$ with meromorphic decoder $f^\top$, we have*

$$f(z) = \frac{f^\top(\frac{z+1}{z-1}) + 1}{f^\top(\frac{z+1}{z-1}) - 1}.$$

Proof. Conjugate the logical decoder $L^\dagger$ with H and use functoriality of $\mathscr{P}$. □

A CSS code C has stabilizer group $S = S_X S_Z$ where S_Z has only Z-type Pauli operators, and S_X has only X-type. Given a set of operators $A \subset \text{P}_n$, we define the *weight enumerator* of A to be the following homogeneous polynomial in $\mathbb{C}[x,y]$:

$$W_A(x,y) = \sum_{g \in A} x^{n-w(g)} y^{w(g)}.$$

Theorem 4.4. *Given CSS code C with stabilizers S_X, S_Z and logical operators $X^{\otimes n}, Z^{\otimes n}$ the meromorphic decoder of C is given by*

$$f(z) = \frac{W_{\langle S_X \rangle}(z,1)}{W_{\langle S_X \rangle}(1,z)}$$

Proof. See App. B. □

Note that for CSS codes we can always recover the Z-type stabilizers from the X-type stabilizers and X-type logical operators, and vice-versa.

Corollary 4.5. *Given CSS code C with stabilizers S_X, S_Z and logical operators $X^{\otimes n}, Z^{\otimes n}$ we have*

$$f(z) = \frac{W_{\langle S_Z \rangle}(\frac{z+1}{z-1}, 1) + W_{\langle S_Z \rangle}(1, \frac{z+1}{z-1})}{W_{\langle S_Z \rangle}(1, \frac{z+1}{z-1}) - W_{\langle S_Z \rangle}(1, \frac{z+1}{z-1})}.$$

Proof. Combine the previous proposition and theorem. □

Prop. 4.3 and/or Cor. 4.5 can be seen as a kind of MacWilliams theorem for meromorphic decoders [14].

Given a code C with $k = 1$, and another code D on n qubits we can use the logical operators of n copies of C to express the stabilizers of D. The resulting code is the *concatenated* code $C \otimes D$.

Proposition 4.6. *Given two codes C and D both with $k = 1$, and meromorphic decoders f and g respectively, the meromorphic decoder of the concatenated code $C \otimes D$ is $f \circ g$.*

Proof. The logical decoders of C and D compose to give the logical decoder of $C \otimes D$, so the proposition follows from functoriality of $\mathscr{P}$. □

We will also define the *r-polynomial* for a meromorphic decoder $f(z) = \frac{p(z)}{q(z)}$ as $r(z) = p(z) - zq(z)$. This gives an algebraic characterization of the fixed points of a distillation routine.

Proposition 4.7. *The roots of $r(z) = 0$ correspond bijectively to distillation points $z \in \mathbb{C}$.*

Conjecture 4.8. *Any CSS code* $[[n,1,d]]$ *with logical operators* $X^{\otimes n}, Z^{\otimes n}$ *exhibits coherent error suppression* $O(\varepsilon^d)$ *at the four stabilizer states* $z = 0, \infty, \pm 1$.

This conjecture says that the meromorphic decoder $f(z)$ for these codes will have order $d-1$ zeros of $f'(z)$ at $z = 0, \infty, \pm 1$. We will see this in examples below as the factor $z^{d-1}(z-1)^{d-1}(z+1)^{d-1}$ appearing in the numerator of the expression for $f'(z)$.

5 Examples

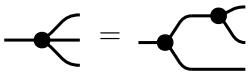
Many of the following calculations were performed using sage python [18], which has powerful algorithms for polynomial factorization and root-finding, see App. C.

In the following examples we suppress the tensor product symbol $\otimes$ when notating elements of P_n, as well as writing a dot . for $I \in \mathrm{P}_1$.

5.1 Logical state preparation

Given a code C, by applying the code projector $\Pi = LL^\dagger$ to n physical qubits we find the encoded state to be given by the meromorphic decoder.

Example 5.1. Here we use logical decoder circuits from the *phase-free* ZX-calculus, see [12]. The stabilizer code $\langle XXI, IXX\rangle$ has logical decoder circuit $L^\dagger =$  from which we derive the meromorphic decoder via Prop. 3.1: $g(z_1,z_2,z_3) = z_1z_2z_3$. For the other stabilizer code $\langle ZZI, IZZ\rangle$, the logical decoder circuit is



Once again using Prop. 3.1 we find the meromorphic decoder,

$$\begin{aligned} h(z_1,z_2,z_3) &= \frac{1+z_1z_2}{z_1+z_2} \text{ with } z_2 \mapsto \frac{1+z_2z_3}{z_2+z_3} \\ &= \frac{1+z_1\frac{1+z_2z_3}{z_2+z_3}}{z_1+\frac{1+z_2z_3}{z_2+z_3}} \\ &= \frac{z_1+z_2+z_3+z_1z_2z_3}{1+z_1z_2+z_1z_3+z_2z_3} \end{aligned}$$

Concatenating these two codes together produces the $[[9,1,3]]$ Shor code:

S	XX.
	.XX
L_X	XXX
L_Z	ZZZ

$\otimes$

S	ZZ.
	.ZZ
L_X	XXX
L_Z	ZZZ

$=$

S	ZZ.....
	...ZZ....
	.....ZZ.
	.ZZ.....
	....ZZ...
	.....ZZ
	XXXXXX...
	...XXXXXX
L_X	XXXXXXXXX
L_Z	ZZZZZZZZZ

For simplicity we write the univariate meromorphic decoders of the concatenating codes as $g(z) = \frac{z^3+3z}{3z^2+1}$ and $h(z) = z^3$, respectively. Using Prop. 4.6, the Shor code has meromorphic decoder, $f(z)$, r-polynomial, and derivative $f'(z)$:

$$\begin{aligned} f(z) &= g(h(z)) = \frac{z^9+3z^3}{3z^6+1} \\ r(z) &= (z-1)z(z+1)(z^2-z-1)(z^2+1)(z^2+z-1) \\ f'(z) &= \frac{9(z-1)^2z^2(z+1)^2(z^2-z+1)^2(z^2+z+1)^2}{(3z^6+1)^2} \end{aligned}$$

The r-polynomial has roots at $z = 0, \pm 1, \pm i$ and so these stabilizer states are distilled by Prop. 4.7. Of these, $z = 0, \pm 1$ are stationary points of degree three, giving coherent distillation at $z = 0, \pm 1$ that suppresses coherent error to $O(\varepsilon^3)$. □

5.2 Magic state distillation

We repeat the same analysis of the meromorphic decoder, but now we are interested in *magic states*, which are non-stabilizer states prepared by the decoder $L^\dagger$. See [2, 20].

Example 5.2. Here we consider the original magic state distillation procedure from [2]. The $[[5,1,3]]$ code has stabilizer group $S = \langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle$. The meromorphic decoder is given by

$$f(z) = \frac{z^5-5z}{5z^4-1}$$

with derivative

$$f'(z) = \frac{5(z^8+14z^4+1)}{(5z^4-1)^2}$$

The degree of $f(z)$ is five, and there are eight weakly distilled points, all of degree two. These are the eight F-states. We see $2 \times (5-1) = 8$ which verifies Thm. 4.2. The r-polynomial for this code is

$$r(z) = -4z(z^4+1)$$

and so the meromorphic decoder distills $z = 0$ as well as four of the H-states. Using the octahedral function, we find that $E_7(f(z)) - E_7(z) = 0$ has solutions at all stabilizer states, H-states and F-states. So by Prop. 2.3, up to a Clifford correction, this meromorphic decoder can be used to coherently distill F-states suppressing coherent error up to $O(\varepsilon^2)$. □

Example 5.3. The $[[7,1,3]]$ *Steane code* has stabilizer group

$$\begin{aligned} S = \langle &IIIXXXX, IXXIIXX, XIXIXIX, \\ &IIIZZZZ, IZZIIZZ, ZIZIZIZ \rangle \end{aligned}$$

which is a CSS code with logical operators $X^{\otimes 7}, Z^{\otimes 7}$. This has weight enumerator $W_{\langle S_X \rangle}(x,y) = x^7 + 7x^3y^4$ and so by Thm. 4.4:

$$\begin{aligned} f(z) &= \frac{W_{\langle S_X \rangle}(z,1)}{W_{\langle S_X \rangle}(1,z)} = \frac{z^7+7z^3}{7z^4+1} \\ r(z) &= (z-1)z(z+1)(z^2-2z-1)(z^2+2z-1) \\ f'(z) &= \frac{21(z-1)^2z^2(z+1)^2(z^2+1)^2}{(7z^4+1)^2} \end{aligned}$$

The distillation map has degree seven, and we see it has stationary points only at the six stabilizer states, each of degree three. From the r-polynomial, it distills the four H-states at $z^2 \pm 2z - 1 = 0$, but these are not stationary points, and so f does not suppress coherent errors at these points. □

Example 5.4. For the $[[15,1,3]]$ triorthogonal code ([2]§VI) we find:

$$\begin{aligned} f(z) &= \frac{z^{15} + 15z^7}{15z^8 + 1} \\ r(z) &= (z-1)z(z+1)(z^{12} + z^{10} + z^8 - 14z^6 + z^4 + z^2 + 1) \\ f'(z) &= \frac{105z^6(z-1)^2(z+1)^2(z^2+1)^2(z^4+1)^2}{(15z^8+1)^2} \end{aligned}$$

This suppresses coherent error of the four H-states at $z^4 + 1 = 0$ to order $O(\varepsilon^3)$, which are distilled up to Clifford rotation. □

6 Departing thoughts and questions

We are making use of algebraic geometry ideas to reformulate quantum computing primitives. By projectivizing in a functorial way, we begin to translate the axioms of quantum mechanics into a different, non-linear language. These are the algebraic bones of quantum computing. We end up with something that is "the same" as the usual formulation of quantum physics, although we would like to see a more precise statement of this. Apparently Heisenberg learnt about linear algebra from Max Born [10]. Perhaps if he had wandered a bit further down the hallway to find an algebraic geometer instead we would now be rewriting quantum mechanics in terms of linear algebra.

Meromorphic functions are found in many places in the literature, such as in signal processing and analytic combinatorics. Stationary points of our meromorphic distillation functions hold the key to coherent error suppression in (magic) state preparation, and these appear quite generically by Thm.4.2.

A more refined analysis would differentiate the multivariate meromorphic decoders, in order to see the effect of independent coherent errors on each input qubit. We have simplified to the univariate case for simplicity of exposition. We also gloss over consideration of the point $z = \infty$ which involves a change of coordinates such as $z \mapsto 1/z$.

Physicists, especially those doing quantum physics, blithely write the symbol i in equations, but in terms of algebraic number theory this is already problematic: which i do you mean? There are two solutions of $z^2 + 1 = 0$. In this work, we are seeing how states are elements of number fields, and have names which are polynomials. And these names are ambiguous up to a group called the Galois group. What is the physical or computational significance of this group?

By restricting to the surface of the Bloch-Riemann sphere we are considering pure states and coherent operations. Extending to incoherent states requires considering anti-holomorphic structure: functions of $\bar{z}$.

Finally, we mention a closely related idea known as "geometric quantization" [6, 1]. In this work we are treating the projective space $\mathscr{P}(\mathscr{H})$ as a kind of configuration space for a system. Going the other way, we can ask, given a complex manifold $\mathscr{M}$, what extra information is needed to construct a quantum Hilbert space for which $\mathscr{M}$ is the classical configuration space?

Acknowledgements. To be added.

A Proof of Prop. 1.4

We prove that $\mathscr{P}_\perp : \mathbb{C}\text{FVec}_\otimes \to \text{Set}_{\perp,\wedge}$ is lax monoidal with unitor given by the identity map (the unary Σ), and laxator given by Σ.

Given $\text{Set}_\times$ as a strict monoidal (cartesian) category, we claim that $\text{Set}_{\perp,\wedge}$ is also a strict monoidal category. For example,

$$\begin{aligned}((A\cup\{\perp\})\wedge(B\cup\{\perp\}))\wedge(C\cup\{\perp\}) &= ((A\times B)\times C)\cup\{\perp\}\\ &= (A\times(B\times C))\cup\{\perp\}\\ &= (A\cup\{\perp\})\wedge((B\cup\{\perp\})\wedge(C\cup\{\perp\}))\end{aligned}$$

and similarly for the units. We will also take $FVec_\otimes$ as a strict monoidal category.

We have,

$$\mathscr{P}_\perp(1_\otimes) = \mathscr{P}_\perp(\mathbb{C}^1) = \{\star\}\cup\{\perp\},$$

and so we can take the unitor as the identity map on $\{\star\}\cup\{\perp\}$.

Given vector spaces $V,W \in \mathbb{C}\text{FVec}_\otimes$ we define the laxator as the natural transform

$$\begin{aligned}\Sigma_{V,W} : \mathscr{P}_\perp(V)\wedge\mathscr{P}_\perp(W) &\to \mathscr{P}_\perp(V\otimes W)\\ \perp &\mapsto \perp\\ ([v],\perp) &\mapsto \perp\\ (\perp,[w]) &\mapsto \perp\\ ([v],[w]) &\mapsto [v\otimes w]\end{aligned}$$

For associativity we need to show, given vector spaces $U,V,W \in \mathbb{C}\text{FVec}$ the following two maps are equal:

$$\mathscr{P}_\perp(U)\wedge\mathscr{P}_\perp(V)\wedge\mathscr{P}_\perp(W) \xrightarrow{\Sigma_{U,V}\wedge 1_W} \mathscr{P}_\perp(U\otimes V)\wedge\mathscr{P}_\perp(W) \xrightarrow{\Sigma_{U\otimes V,W}} \mathscr{P}_\perp(U\otimes V\otimes W)$$

and

$$\mathscr{P}_\perp(U)\wedge\mathscr{P}_\perp(V)\wedge\mathscr{P}_\perp(W) \xrightarrow{1_U\wedge\Sigma_{V,W}} \mathscr{P}_\perp(U)\wedge\mathscr{P}_\perp(V\otimes W) \xrightarrow{\Sigma_{U,V\otimes W}} \mathscr{P}_\perp(U\otimes V\otimes W)$$

Given $u\in U, v\in V, w\in W$ these maps both take $([u],[v],[w])$ to $[u\otimes v\otimes w]$ and any appearance of $\perp$ results in $\perp$.

For unitality, we check that both $\Sigma_{\mathbb{C}^1,U}$ and $\Sigma_{U,\mathbb{C}^1}$ are identity maps on $\mathscr{P}_\perp(\mathbb{C}^1\otimes U) = \mathscr{P}_\perp(U)$.

B Proof of Thm. 4.4

Given a CSS code with parameters $[[n,1,d]]$, stabilizer generators S_X, S_Z, logical operators L_X, L_Z, and logical decoder $L^\dagger$, we claim that

$$L^\dagger\begin{pmatrix}x\\y\end{pmatrix}^{\otimes n} = \begin{pmatrix}W_{\langle S_X\rangle}(x,y)\\ W_{L_X\langle S_X\rangle}(x,y)\end{pmatrix}. \tag{1}$$

When $L_X = X^{\otimes n}$ we have that $W_{L_X\langle S_X\rangle}(x,y) = W_{\langle S_X\rangle}(y,x)$ and the result follows by projectivizing.

To prove the claim (1), we write the logical decoder $L^\dagger$ as a phase-free ZX diagram called the *Z-X normal form* [12]. By bending wires this becomes a $(n+k)$-qubit state, and the result follows from [12] Eq.(5).

This proof straightforwardly generalizes in two ways. Firstly, to CSS codes $[[n,k,d]]$ with $k > 1$. Secondly, we obtain multivariate meromorphic decoder functions from multivariate versions of the weight enumerators called "full weight enumerators".

C Sage python example code

The following interactive session shows an example of using sage via the python interpreter.

```
Python 3.10.12
Type "help", "copyright", "credits" or "license" for more information.
>>> from sage.all_cmdline import *
>>> R = PolynomialRing(ZZ, "z") # polynomial ring in "z" over the integers
>>> z, = R.gens()
>>> F = z**8 + 14*z**4 + 1 # make a polynomial in the z variable
>>> F
z^8 + 14*z^4 + 1
>>> factor(F) # factorize over the integer polynomial ring
(z^4 - 2*z^3 + 2*z^2 + 2*z + 1) * (z^4 + 2*z^3 + 2*z^2 - 2*z + 1)
>>> F(z=1) # evaluate the polynomial at z=1
16
>>> K = PolynomialRing(CyclotomicField(24), "z") # a bigger polynomial ring
>>> KF = K(F) # promote F to this new ring
>>> factor(KF) # KF now completely splits into linear factors
(z - zeta24^4 - zeta24^2) * (z - zeta24^4 + zeta24^2) *
(z + zeta24^4 - zeta24^2) * (z + zeta24^4 + zeta24^2) *
(z - zeta24^6 - zeta24^4 + zeta24^2 + 1) * (z - zeta24^6 + zeta24^4 + zeta24^2 - 1) *
(z + zeta24^6 - zeta24^4 - zeta24^2 + 1) * (z + zeta24^6 + zeta24^4 - zeta24^2 - 1)
>>> f = (z**5 - 5*z) / (5*z**4 - 1) # a meromorphic function
>>> f.parent() # sage knows how to promote fractions
Fraction Field of Univariate Polynomial Ring in z over Integer Ring
>>> diff(f) # differential with respect to the variable z
(5*z^8 + 70*z^4 + 5)/(25*z^8 - 10*z^4 + 1)
>>>
```

References

[1] M. Blau. Symplectic geometry and geometric quantization. *Lecture notes*, 1992.

[2] S. Bravyi and A. Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A—Atomic, Molecular, and Optical Physics*, 71(2):022316, 2005.

[3] B. Coecke and B. Edwards. Three qubit entanglement within graphical Z/X-calculus. *arXiv preprint arXiv:1103.2811*, 2011.

[4] B. Coecke and A. Kissinger. The compositional structure of multipartite quantum entanglement. In *International Colloquium on Automata, Languages, and Programming*, pages 297–308. Springer, 2010.

[5] B. Coecke, A. Kissinger, A. Merry, and S. Roy. The GHZ/W-calculus contains rational arithmetic. *EPTCS*, 52:34–48, 2011.

[6] A. Echeverria-Enriquez, M. C. Munoz-Lecanda, N. Román-Roy, and C. Victoria-Monge. Mathematical foundations of geometric quantization. *arXiv preprint math-ph/9904008*, 1999.

[7] M. Gharahi, S. Mancini, and G. Ottaviani. Fine-Structure Classification of Multiqubit Entanglement by Algebraic Geometry. *Physical Review Research*, 2(4):043003, 2020.

[8] E. Girondo and G. González-Diez. *Introduction to compact Riemann surfaces and dessins d'enfants*, volume 79. Cambridge University Press, 2012.

[9] D. Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, 1997.

[10] J. L. Heilbron and C. Rovelli. Matrix mechanics mis-prized: Max born's belated nobelization. *The European Physical Journal H*, 48(1):11, 2023.

[11] C. Heunen and J. Vicary. *Categories for Quantum Theory: an introduction*. Oxford University Press, 2019.

[12] A. Kissinger. Phase-free zx diagrams are css codes (... or how to graphically grok the surface code). *arXiv preprint arXiv:2204.14038*, 2022.

[13] F. Klein. *Lectures on the Icosahedron and the Solution of Equations of the Fifth Degree*. Courier Corporation, 2003.

[14] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[15] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[16] L. L. Sanchez-Soto, A. B. Klimov, A. Z. Goldberg, and G. Leuchs. The quantum sky of majorana stars, 2026.

[17] A. H. A. Sumadi, N. M. Shah, U. A. Halim, and H. Zainuddin. Qubit geometry through holomorphic quantization. *arXiv preprint arXiv:2504.16426*, 2025.

[18] The Sage Developers. *SageMath, the Sage Mathematics Software System*. DOI 10.5281/zenodo.6259615.

[19] J. Van De Wetering. Zx-calculus for the working quantum computer scientist. *arXiv preprint arXiv:2012.13966*, 2020.

[20] Y. Zheng and D. E. Liu. From magic state distillation to dynamical systems. *arXiv preprint arXiv:2412.04402*, 2024.

[21] A. K. Zvonkin. Belyi functions: Examples, properties and applications. In *Applications of Group Theory to Combinatorics*, pages 171–190. CRC Press, 2008.