Quantum polygons and the Fourier transform

David Wakeham*
Department of Physics and Astronomy, University of British Columbia, Vancouver, BC V6T 1Z1, Canada
(Dated: September 18, 2020)

The Hilbert space $\mathcal{H}_d \simeq \mathbb{C}^d$ of a qudit is impossible to directly visualize for $d \geq 2$. Here, we explore a shortcut which turns vectors into polygons in the complex plane. This suggests a basis of regular polygons, for which the active change of basis is the quantum Fourier transform (QFT). We give an intuitive, visual approach to the QFT, in particular when $d = a^\lambda$ is a perfect power.

INTRODUCTION

The Hilbert space $\mathcal{H}_d \simeq \mathbb{C}^d$ of a qudit is impossible to directly visualize for $d \geq 2$. There are many indirect methods for visualizing higher dimensions [4]. Here, we will represent vectors as polygons in the complex plane, following [3], and give visual representations for various basic operation in the natural basis of regular polygons.

Consider a vector $\vec{v} = (v_k) \in \mathbb{C}^d$, for $k \in [d]$, where $[d] := \{0, \ldots, d-1\}$. Viewing the vector as a map $\vec{v} : [d] \to \mathbb{C}$ suggests we place vertices at points $v_k \in \mathbb{C}$ and label with k. A less visually cluttered approach is to draw directed edges $v_k \to v_{k+1}$ and indicate v_0, e.g. with a white vertex. Then a vector $\vec{v} \in \mathbb{C}^d$ can be pictured as a polygon of d sides with a distinguished vertex (Fig. 1). We call this the *polygonal representation*.

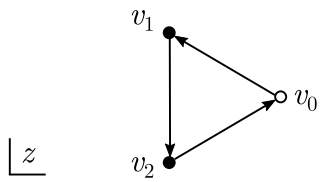

FIG. 1: A vector $\vec{v} \in \mathbb{C}^3$ represented as a triangle in $\mathbb{C}$.

For a state $|\psi\rangle \in \mathcal{H}_d$, we choose some basis $|k\rangle, k \in [d]$, to canonically identify $\mathcal{H}_d$ with $\mathbb{C}^d$. For coefficients $|\psi\rangle =: \sum_k \alpha_k |k\rangle$, $|\psi\rangle = (\alpha_k)$ is a (normalized) map we can represent as a d-gon. Note that the phase ambiguity for a single qudit means that polygons for states form an equivalence class under rotations around the origin.

REGULAR POLYGONS

The standard basis vectors $\vec{\delta}_k$ are useful for linear algebra, but ill-suited to the polygonal representation. This suggests we seek a new basis. The most natural choice is the set of *regular polygons* centred around the origin. These will be maps of the form $\chi_k = \omega^k \chi_0$ for some phase $|\omega| = 1$ and constant $\chi_0 \neq 0$.

The phase ω describes the angular difference between successive points. Since a regular polygon has the same angular difference (from the cetnre) for all points, this relation also holds for the first and last component, and we immediately have $\omega^d = 1$. We can always rotate and rescale a polygon, i.e. multiply by some complex $z \in \mathbb{C}$, so that $\chi_0 = 1$. Define the primitive dth root of unity $\omega_d := e^{2\pi i/d}$. Our two conditions, $\omega^d = 1$ and $\chi_0 = 1$, suggest we consider the regular polygons

$$\vec{\chi}^s_d := (\omega_d^{sk}) \, . \tag{1}$$

Convex regular polygons will only occur for values of $s = 1$ and $s = d-1$, as in Fig. 2. Stellated regular polygons occur for $1 < s < d-1$ coprime with d, i.e. $\gcd(d, s) = 1$.

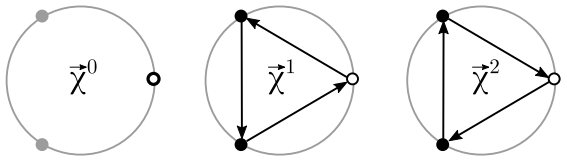

FIG. 2: The polygonal basis for $\mathbb{C}^3$.

This gives at most $d-1$ polygons, not enough to form a basis for $\mathbb{C}^d$. To get d vectors, we simply take $s \in [d]$. This will include a "trivial" polygon for $s = 0$, represented by the thick white circle in Fig. 2. But in general, for composite d, it will include *wrapped* polygons when s and d share a common factor. We call this the set of *regular polygons*. We show some examples in Fig. 3.

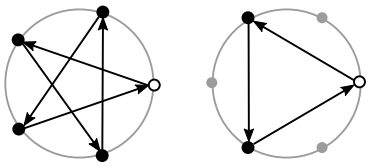

FIG. 3: A stellation $\vec{\chi}^2_5$ and a wrapped polygon $\vec{\chi}^2_6$.

Let (s, d) be the greatest common divisor of s and d, and define $\{s, d\} := d/(s, d)$. Then $\vec{\chi}^s_d$ is a polygon of $\{s, d\}$ sides, which wraps the unit circle g times, called a *polygram* in the mathematics literature, and denoted by Coxeter-Schläfi symbol $(s, d)\{d/s\}$ [2].This has an immediate number-theoretic consequence. The count of $s \in [d]$ relatively prime to d is given by *Euler's totient function* $\phi(d)$. Similarly, $\phi(d/f)$ counts s for which $(d, s) = f$,

since $(d,s) = f$ just in case $(d/f, s) = 1$. Thus, $\phi(d/f)$ counts the polygons wrapping f times. Since $d/f = f'$ is just another way of enumerating factors of d, and every polygon must wrap f times for some factor f, we find that

$$\sum_{f|d} \phi(f) = d. \tag{2}$$

This identity was first proved by Gauss.

LINKAGES

We have yet to prove that the regular polygons form a basis. A direct if clumsy method is to form a $d \times d$ matrix from the vectors $\vec{\chi}_d^j$ and compute the determinant.[6] A nicer method is to compute the inner product and show the regular polygons are orthogonal, and since we have d of them, form a basis.

Using the geometric series $\sum_k r^k = (1-r)^{-1}(1-r^d)$ for $r \neq 1$, we have

$$\langle \vec{\chi}_d^t, \vec{\chi}_d^s \rangle = \sum_{k=0}^{d-1} \omega_d^{(s-t)k} = \frac{1-\omega^{(s-t)d}}{1-\omega^{(s-t)}} = 0 \tag{3}$$

for $s \neq t$, using $\omega_d^{(s-t)d} = 1$. For $s = t$, each term is 1, so the inner product is d. Hence,

$$\langle \vec{\chi}_d^t, \vec{\chi}_d^s \rangle = d\delta_{s,t} \ . \tag{4}$$

We can also see this by geometrizing the inner product using *linkages*. The linkage associated with a vector $\vec{v}$ is the sequence of points $L(\vec{v}) := (V^k)$, where $V^k := \sum_{j=0}^k v_j$. Geometrically, this simply arranges the complex numbers v_k top to tail. We show example in Figs. 4 and 5, colouring the numbers v_k (considered as vectors from the origin) to distinguish them from the edges in the polygonal representation.

Note that the linkage for a regular polygon closes (Fig. 4), starting and ending at the origin, unless we have the trivial linkage for which $V_k = k$. The argument is simple: the linkage is geometrically similar to the original polygon, since the vectors v_k comprising the linkage, and $v_{k+1} - v_k$ comprising the polygon, have a fixed complex ratio (i.e. fixed scaling and rotation) $\omega - 1$.

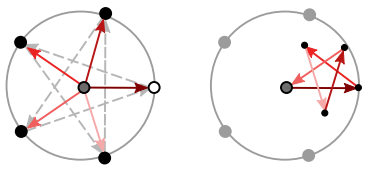

FIG. 4: The stellation $\vec{\chi}_5^2$ and associated linkage $L(\vec{\chi}_5^2)$.

Let us return to the question of orthogonality. The inner product $\langle \vec{\chi}_d^t, \vec{\chi}_d^s \rangle$ is the last component in the linkage $L(\vec{\chi}_d^{s-t})$. Since $\vec{\chi}_d^{s-t}$ is a regular polygon, it closes unless $s = t$. If $s = t$, then the linkage is simply a chain of length d, and hence we rediscover $\langle \vec{\chi}_d^t, \vec{\chi}_d^s \rangle = d\delta_{s,t}$ geometrically.

We can view inner products with a polygon as "mechanically" uncurling a linkage. Left-multiplying a vector $\vec{v}$ by $(\vec{\chi}_d^t)^\dagger$ leaves the first chain in the linkage, V^0, alone. But it rotates $V^1, V^2, \ldots, V^{d-1}$ by ω_d^{-t}. Then it rotates $V^2, V^3, \ldots, V^{d-1}$ by ω_d^{-t}, and so on, and ultimately, $V^k \to V^k \omega_d^{-tk}$. We give an example in Fig. 5, unzipping a linkage $L(\vec{v})$ with the triangle $\vec{\chi}_3^1$. The most general vector $\vec{w}$ for which $\langle \vec{w}, \vec{v} \rangle$ can be interpreted as a hinged motion of $L(\vec{v})$ is $\vec{w} = (\omega_k)$, for phases $|\omega_k| = 1$.

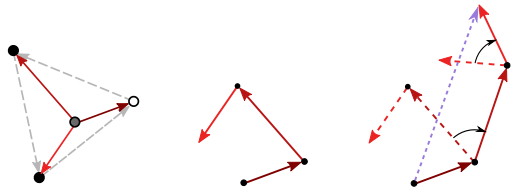

FIG. 5: A vector $\vec{v}$, with linkage $L(\vec{v})$, unzipped by $\vec{\chi}_3^1$. The blue line is the coefficient $d^{-1}\langle \vec{\chi}_3^1, \vec{v} \rangle$.

Since the $\vec{\chi}_d^s$ form an orthogonal basis, the results of uncurling a linkage $L(\vec{v})$ give coefficients for an expansion of $\vec{v}$ in regular polygons. Similarly, we can consider states $|\psi\rangle = \sum_k \alpha_k |k\rangle \in \mathcal{H}_d$. Our basis of polygonal vectors can be made into an orthonormal basis of polygonal *states*, $|\chi_d^s\rangle := \vec{\chi}_d^s / \sqrt{d}$. Then, in this new basis,

$$|\psi\rangle = \sum_s A_s |\chi_d^s\rangle, \quad A_s := \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} \omega_d^{-ks} \alpha_k \ . \tag{5}$$

This is the *discrete Fourier transform (DFT)*. It is a passive transformation in the sense that it leaves $|\psi\rangle$ alone but implements a basis change.

In a quantum computer, we cannot change the measurement basis, but actively change the state. Define the *quantum Fourier transform (QFT)* as the map

$$|s\rangle \mapsto \mathrm{QFT}_d |s\rangle = |\chi_d^s\rangle = \frac{1}{\sqrt{d}} \sum_k \omega_d^{ks} |k\rangle \ , \tag{6}$$

and extending by linearity to any $|\psi\rangle = \sum_s \alpha_s |s\rangle$. The coefficients in the computational basis after this active transformation will be precisely the Fourier-transformed coefficients A_s. We will discuss this operation in more detail below.

We can define a "generalized" polygon $\vec{\chi}_d^x := (\omega_d^{kx})$, for $x \in [0, d)$. As argued above, the associated linkage closes if and only if x is an integer. But how close is it to closing for arbitrary x? We can check explicitly. Using the geometric series from (3), for $x \neq 0$ the tip of the linkage lies at

$$\sum_{k=0}^{d} \omega_d^{kx} = \frac{1-\omega_d^{dx}}{1-\omega_d^x} \ . \tag{7}$$

Using Euler's formula $e^{i\theta} = \cos(\theta) + i\sin(\theta)$ and the double angle formula $1 - \cos(\theta) = 2\sin^2(\theta/2)$ distance from the origin is

$$\left|\frac{1-\omega_d^{dx}}{1-\omega_d^x}\right| = \left|\frac{1-\cos(2\pi x))^2+\sin^2(2\pi x)}{(1-\cos(2\pi x/d))^2+\sin^2(2\pi x/d)}\right|^{1/2} = \left|\frac{\sin(\pi x)}{\sin(\pi x/d)}\right| . \tag{8}$$

This can be used to bound the coefficients in the QFT, and a related procedure called *phase estimation* [1], but we will not discuss this further here.

TENSOR PRODUCTS

Things become more interesting when the Hilbert space factorizes. Assume that $d = ab$, so that $\mathcal{H}_d \simeq \mathcal{H}_a \otimes \mathcal{H}_b$. To canonically identify these spaces, we define $|k\rangle \simeq |m\rangle \otimes |n\rangle$, for $m \in [a]$, $n \in [b]$, and

$$|k(m,n)\rangle = |na+m\rangle \simeq |m\rangle \otimes |n\rangle . \tag{9}$$

This corresponds to arranging the $d = ab$ points in a rectangular array of height b and width a, with points labeled by (n, m).

Let us consider a regular polygon $\vec{\chi}_d^s$, and see how it factorizes across $\mathcal{H}_a \otimes \mathcal{H}_b$. The kth component is

$$\omega_d^{ks} = e^{2\pi i s(na+m)/ab} = \omega_a^{sm/b}\omega_b^{sn} . \tag{10}$$

This implies $\vec{\chi}_d^s = \vec{\chi}_a^{s/b} \otimes \vec{\chi}_b^s$. The second factor is a regular b-gon we call the *principal subgon*, while the first factor is a *copygon*, consisting of a points in a regular polygon of $\{d, s\}$ sides. It serves to make a copies offset by ccw angles $\Delta\phi = 2\pi s/d$, starting with the first copy at $\phi = 0$. The labelling (9) instructs us to iterate across the a copies of the b-gon, or arrange numbers into a $b \times a$ matrix and reading every sth entry.

We illustrate in Fig. 6, displaying two factorizations of a hexagon, $\vec{\chi}_6^1 \simeq \vec{\chi}_2^{1/3} \otimes \vec{\chi}_3^1$ (two triangles) and $\vec{\chi}_6^1 \simeq \vec{\chi}_3^{1/2} \otimes \vec{\chi}_2^1$ (three digons). The white vertices represent the copygon, and form the first vertex of each subgon.

For multiple factors, the procedure iterates, with $\vec{\chi}_{abc}^s \simeq \vec{\chi}_a^{s/bc} \otimes \vec{\chi}_b^{s/c} \otimes \vec{\chi}_c^s$ and so on. We first factor out the principal subgon, then iteratively factor the copygons. We indicate the possibilities for $\vec{\chi}_{12}^1$ in Fig. 7. Dashed lines are used to indicate a second level of nesting, and associated arrays are shown. Thus, tensor products induce a pleasing correspondence between integer factorizations, multi-dimensional arrays, and polygons.

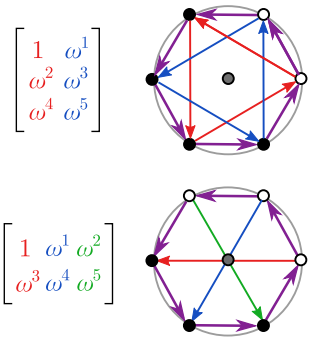

FIG. 6: Two tensor decompositions of $\vec{\chi}_6^1$.

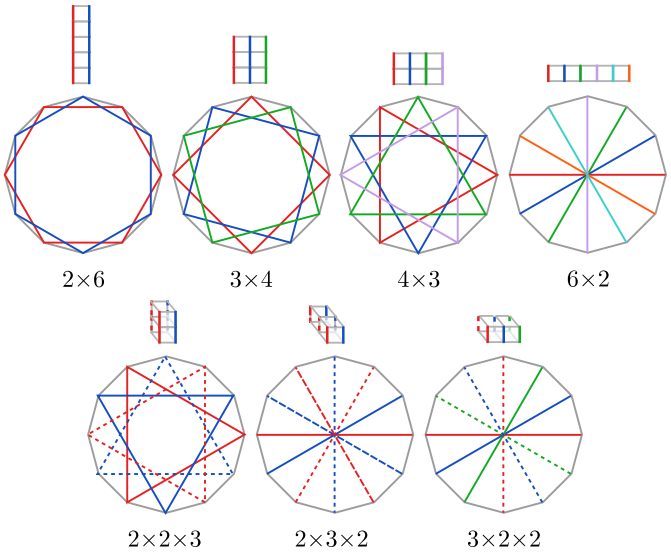

FIG. 7: The non-trivial factorizations of $\vec{\chi}_{12}^1$.

PERFECT POWERS

For a power $d = a^\lambda$, the labelling (9) extends to an expansion in base a. For $\mathbf{n} = (n_\ell) \in [a]^\lambda$,

$$k(\mathbf{n}) := n_0 + an_1 + a^2 n_2 + \cdots = \sum_{\ell=0}^{\lambda-1} n_\ell a^\ell , \tag{11}$$

The array is a hypercube of λ dimensions, latticized so that it has a points per side, and as above, entries in $\vec{\chi}_d^s$ process this array in steps of size s. The points are labelled with the base d expansion of the angle $\phi \in [0, 2\pi)$, since $\phi(\mathbf{n}) = 2\pi k(\mathbf{n})/d$.

We illustrate for binary powers in Fig. 8. The array is a hypercube of 2^λ points labelled by binary strings. In turn, these strings give the expansion in binary for anglular coordinates of points on the corresponding 2^λ-gon. In Fig. 8, we show to build the hypercube using a decomposition $\vec{\chi}_2^{s/2^{\lambda-1}} \simeq \vec{\chi}_{2^\lambda}^s \otimes \vec{\chi}_{2^{\lambda-1}}^s$, so that we have a simply copying digon, and a principal sub-$2^{\lambda-1}$-gon. We

can orthogonally copy an array, or copy the polygon with a twist. Similarly, we can build up polygons for $d = a^\lambda$ by starting with an a-gon, and iteratively making a copies which are fan uniformly around the circle.

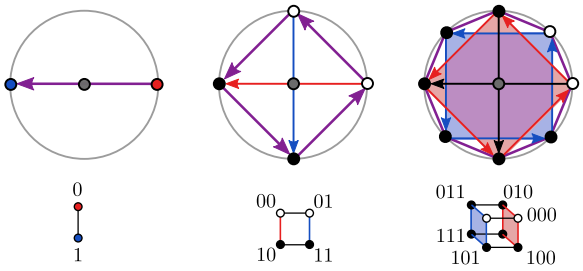

FIG. 8: Iteratively building a hypercube and 2^λ-gon from digons. Binary sequences label angles.

We can also completely factorize a polygon:

$$\vec{\chi}_{a^\lambda}^{s} = \vec{\chi}_a^{sa^{-(\lambda-1)}} \otimes \cdots \otimes \vec{\chi}_a^{sa^{-1}} \otimes \vec{\chi}_a^{s} \,. \tag{12}$$

Each terms corresponds to a place value in the base a expansion, with terms decreasing in significance as we go left to right. The binary case is illustrated in Fig. 9.

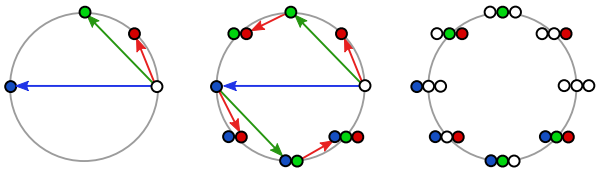

FIG. 9: Completely factorizing an octagon into digons. Digons correspond to bits.

A VISUAL QFT

Equation (12) captures the required sequence of operations needed to form $\vec{\chi}_d^s$, and hence implement the QFT on a quantum computer. We start by building the *initial copygon* $\vec{\chi}_a^{sa^{-(\lambda-1)}}$, and then expanding the points on the unit circle by a factor of a, a total of $\lambda - 1$ times. We call $\vec{\chi}_a^x \mapsto \vec{\chi}_a^{ax}$ *expansion*. If we can record the outcome of each expansion, we can take the tensor product and obtain $\vec{\chi}_d^s$ according to (12).

We picture this for $s = 20$, $d = 3^3$, in Fig. 10. In base 3 (used implicitly in this example), we have $s = 202$, so our first step is to construct $\vec{\chi}_3^{2.02}$. Expanding once gives $\vec{\chi}_3^{0.2}$, and again gives $\vec{\chi}_3^{2}$. The tensor product of these three vectors (not pictured) gives $\chi_{3^3}^{20}$.

If we want to implement the QFT on a quantum computer, we are only permitted to use unitary operations.

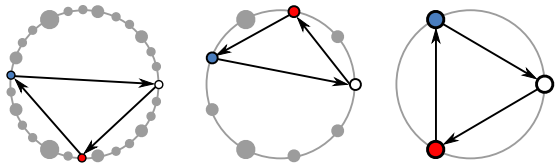

FIG. 10: Performing the QFT for $s = 20, d = 3^3$. Points have their arguments tripled mod 2π at each step.

Expansion is not unitary, in fact not linear, since

$$\vec{\chi}_a^x = \sum_k \omega_a^{kx} |k\rangle \mapsto \sum_k \omega_a^{(a-1)kx} \cdot \omega_a^{kx} |k\rangle$$

acts on basis vectors in a way that depends on x. But once x is fixed, it acts via the matrix

$$E_a^{(x)} = \mathrm{diag}(\omega_a^{kx(a-1)}) \,. \tag{13}$$

This is unitary since it has pure phases along the diagonal. Similarly, the map which constructs the initial copygon, defined by

$$C_a^{(x)}|0\rangle = |\chi_a^x\rangle = \mathrm{diag}(\omega_a^{kx}) \circ \mathrm{QFT}_a |0\rangle \,, \tag{14}$$

is unitary, since it composes a unitary matrix with the QFT, which is unitary.

The *no cloning theorem* [8] forbids us from recording the intermediate copygons as we go along. But we need these intermediate results to perform the tensor product (12). Thus, to implement the QFT on a quantum computer, we need to compute the factors separately, running λ parallel calculations. This is easily achieved using the operators in (13) and (14):

$$E_a^{(a^{j-1}x)} \cdots E_a^{(ax)} E_a^{(x)} C_a^{(x)} |0\rangle = |\chi_a^{xa^j}\rangle \,. \tag{15}$$

Setting $x = sa^{-(\lambda-1)}$ yields the factors in (13) for $j = 0, \ldots, \lambda - 1$. Since we have $j + 1$ operations in (15), the total number of operations required is

$$1 + 2 + \cdots + \lambda = \frac{1}{2}\lambda(\lambda + 1) \,. \tag{16}$$

Thus, the total number of operations required in $O(\lambda^2)$.

A final issue is that (15) is using the digits s *classically*, but we need a genuine linear map on $\mathcal{H}_d$ in order to take the QFT of an arbitrary state in $O(\lambda^2)$ steps This can be done using *controlled* versions of $E_a^{(x)}, C_a^{(x)}$. The details are beyond our scope, but see [5] for more.[7] We can generalize this picture of the QFT to an arbitrary composite $d = a_0 a_1 \cdots a_{\lambda-1}$, though it is less intuitive because the number of points in the factors will change.

ACKNOWLEDGMENTS

I am supported by an IDF scholarship from UBC. I would like to thank Olivia Di Matteo, Rafael Haenel, and Pedro Lopes for discussions.

* daw@phas.ubc.ca

[1] Cleve, R., Ekert, A., Macchiavello, C., and Mosca, M. Quantum algorithms revisited. *Proc. Roy. Soc. Lond. A 454* (1998), 339.

[2] Coxeter, H. *Regular Polytopes*. Dover Publications, 1973.

[3] Ion, P. Geometry and the Discrete Fourier Transform, 2010.

[4] MathOverflow. *Intuitive crutches for higher dimensional thinking*, 2010 (accessed September 10, 2020).

[5] Nielsen, M. A., and Chuang, I. L. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[6] It is a Vandermonde matrix, so the computation is not hard, merely uninsightful.

[7] In fact, the algorithm builds the intermediate copygons from scratch, rather than expanding the initial copygon. The perspective here is visually clearer.

[8] Wootters, W., and Zurek, W. A single quantum cannot be cloned. *Nature 299* (1982), 802–803.