Gates, States, and Circuits

Notes on the circuit model of quantum computation

Tech. Note 014 v0.7.0 BETA

http://threeplusone.com/gates

Gavin E. Crooks

2022-01-12

Contents

List of Figures

List of Tables

1 Introduction: Gates, states, and circuits

> *We shouldn't be asking 'where do quantum speedups come from?' we should say 'all computers are quantum, [...]' and ask 'where do classical slowdowns come from?'* — Charlie Bennett [0]

> *It appears that very rapid progress is now being made on the fundamentals of quantum computing. It is well to keep in mind, though, that many basic issues of the realization of quantum computers remain unsolved or very difficult.* — David P. Diviencero [13]

1.1 Additional reading

The canonical textbook for quantum computing and information remains Michael A. Nielsen's and Isaac L. Chuang's classic "Quantum Computation and Quantum Information" (affectionately know as Mike and Ike) [18]. If you have any serious interest in quantum computing, you should own this book[1]. These notes are going to take a different cut through the subject, with more detail in some places, some newer material, but neglecting other areas, since it is not necessary to repeat what Mike and Ike have already so ably covered. John Preskill's lecture notes [15] are another excellent (If perennially incomplete) treatment of the subject.

For a basic introduction to quantum mechanics, see "Quantum Mechanics: The Theoretical Minimum" by Leonard Susskind and Art Friedman [39]. The traditional quantum mechanics textbooks are not so useful, since they tend to rapidly skip over the fundamental and informational aspects, and concentrate on the detailed behavior of light, and atoms, and cavities, and what have you. Such physical details are important if you're building a quantum computer, obviously, but not so much for programming one, and I think the traditional approach tends to obscure the essentials of quantum information and how fundamentally different quantum is from classical physics. But among such physics texts, I'd recommend "Modern Quantum Mechanics" by J. J. Sakurai [31].

For gentler introductions to quantum computing see "Quantum Computing: A Gentle Introduction" by Eleanor G. Rieffel and Wolfgang H. Polak [37]. Another interesting take is "Quantum Country" by Andy Matuschak and Michael Nielsen. This is an online introductory course in quantum computing, with builtin spaced repetition [48]. Scott Aaronson's "Quantum Computing since Democritus" [33] is also a good place to start, particularly for computational complexity theory.

Mathematically, quantum mechanics is mostly applied linear algebra, and you can never go wrong learning more linear algebra. For a good introduction see "No Bullshit Guide to Linear Algebra" by Ivan Savov [55], and for a deeper dive "Linear Algebra Done Right" by Sheldon Axler [38].

For a deep dives into quantum information, both "The Theory of Quantum Information" by John Watrous [45] and "Quantum Information Theory" by Mark M. Wilde [43] are excellent, if weighty, tombs.

And if you have very young children, start them early with Chris Ferrie and whurely's "Quantum Computing for Babies" [44].

[1] And Mike and Ike.

2 Bloch sphere representation of a qubit

We'll begin by considering the action of a quantum gate on a single quantum bit. A single classical bit (cbit) is relatively boring; either it's in a zero state, or a one state. In contrast a quantum bit is a much richer object that can exist in a quantum superposition of zero and one. This state can be conveniently visualized as a point on the surface of a 3-dimensional ball, generally called the Bloch sphere [2, 0]. The action of a 1-qubit gate is to rotate this sphere around some axis.

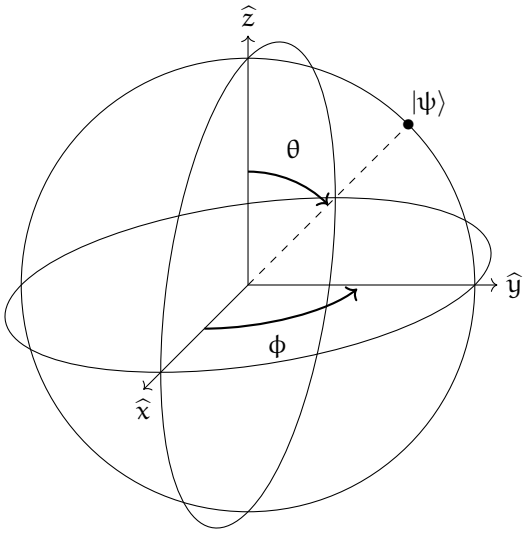


Figure 2.1: Bloch sphere representation of single qubit states.

Ultimately, a qubit is a physical system with two distinct states, which we conventionally label zero and one. The state of the qubit $|\psi\rangle$ can be written as a superposition of zero states $|0\rangle$, and one states $|1\rangle$.

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle\,, \qquad |a|^2 + |b|^2 = 1 \tag{1}$$

where the coefficients a and b are complex numbers. We can rewrite this as

$$|\psi\rangle = e^{i\alpha}\left(\cos(\tfrac{1}{2}\theta)\,|0\rangle + e^{i\phi}\sin(\tfrac{1}{2}\theta)\,|1\rangle\right) \tag{2}$$

where α, θ, and ϕ are real numbers. The phase factor $e^{i\alpha}$ has no observable physical effect and can be ignored. It is merely an artifact of the mathematical representation. (If we use a density matrix representation then the phase factor disappears altogether.)

$$|\psi\rangle \simeq \cos(\tfrac{1}{2}\theta)\,|0\rangle + e^{i\phi}\sin(\tfrac{1}{2}\theta)\,|1\rangle \tag{3}$$

We'll use $\simeq$ to indicate that two states (or gates) are equal up to a phase factor.

The parameters θ and ϕ, can be interpreted as spherical coordinates of a point on the surface of a unit sphere, where θ is the colatitude with respect to the $\hat{z}$-axis and ϕ the longitude with respect to the $\hat{x}$-axis, and $0 \leqslant \theta \leqslant \pi$ and $0 \leqslant \phi < 2\pi$. In cartesian coordinates the point on the 3-dimensional unit sphere is given by the

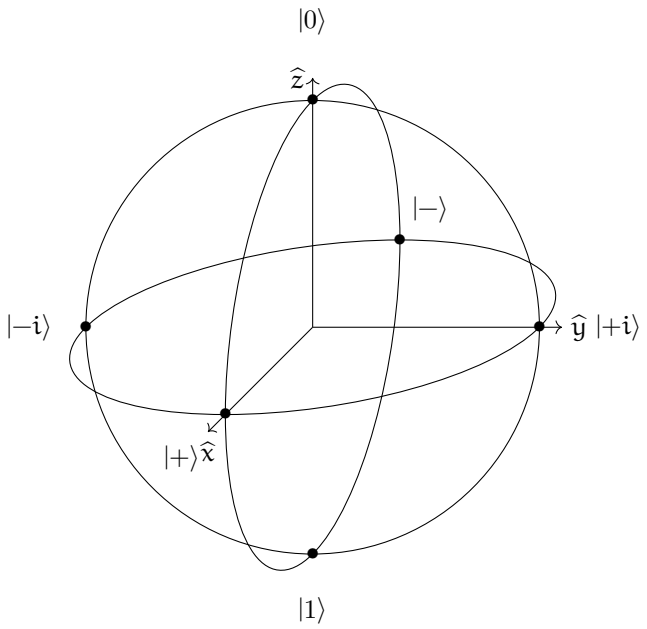


Figure 2.2: Location of standard basis states on the Bloch sphere.

Bloch vector $\hat{n} = (\sin\theta\cos\phi,\ \sin\theta\sin\phi,\ \cos\theta)$.

Note that the zero state, by convention, is located at the top of the Bloch sphere, and the one state at the bottom. States on opposite sides of the sphere are orthogonal, and any pair of such states provides a basis in which any state of a qubit can be represented. The other basis states located along the cartesian axes are common enough to have notation of their own.

X basis	$\lvert+\rangle = \frac{1}{\sqrt{2}}(\lvert0\rangle + \lvert1\rangle)$	$\hat{n} = (+1, 0, 0)$
	$\lvert-\rangle = \frac{1}{\sqrt{2}}(\lvert0\rangle - \lvert1\rangle)$	$\hat{n} = (-1, 0, 0)$
Y basis	$\lvert+i\rangle = \frac{1}{\sqrt{2}}(\lvert0\rangle + i\lvert1\rangle)$	$\hat{n} = (0, +1, 0)$
	$\lvert-i\rangle = \frac{1}{\sqrt{2}}(\lvert0\rangle - i\lvert1\rangle)$	$\hat{n} = (0, -1, 0)$
Z basis	$\lvert0\rangle$	$\hat{n} = (0, 0, +1)$
	$\lvert1\rangle$	$\hat{n} = (0, 0, -1)$

Generically we'll call these the X, Y, and Z bases. The Z-basis is also called the computational or standard basis, is the one we label with zero and ones, and is generally the only basis in which we can make measurements of the system. The X-basis is also called the Hadamard basis, since it can be generated from the computational basis with a Hadamard transform (§3.5).

Unfortunately, there aren't any real-space geometric representations of multi-qubit systems. The geometric representation of 1-qubit states by the Bloch sphere only works because of a mathematical accident that doesn't generalize.

3 Standard 1-qubit gates

Classically, there are only 2 1-bit reversible logic gates, identity and NOT (And 2 irreversible gates, reset to 0 and reset to 1). But in quantum mechanics the zero and one states can be placed into superposition, so there are many other interesting possibilities.

3.1 Pauli gates

The simplest 1-qubit gates are the 4 gates represented by the Pauli operators, I, X, Y, and Z. These operators are also sometimes notated as σ_x, σ_y, σ_z, or with an index σ_i, so that $\sigma_0 = I$, $\sigma_1 = X$, $\sigma_2 = Y$, $\sigma_3 = Z$.

We will explore the algebra of Pauli operators in more detail in chapter (§11). But for now, note that the Pauli gates are all Hermitian, $\sigma_i^\dagger = \sigma_i$, square to the identity $\sigma_i^2 = I$, and that the X, Y, and Z gates anti-commute with each other.

$$\begin{aligned} XY &= -YZ = iZ \\ YZ &= -ZY = iX \\ ZX &= -ZX = iY \\ XYZ &= iI \end{aligned}$$

Pauli-I gate (identity):

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4}$$

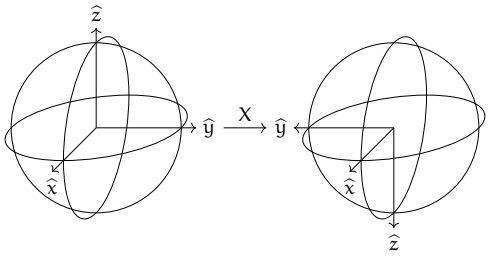
The trivial no-operation gate on 1-qubit, represented by the identity matrix. Acting on any arbitrary state, the gate leave the state unchanged.

$$\begin{aligned} I &= |0\rangle\langle 0| + |1\rangle\langle 1| \\ I\,|0\rangle &= |0\rangle \\ I\,|1\rangle &= |1\rangle \end{aligned}$$

Pauli-X gate (X gate, bit flip)

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{5}$$

The X-gate generates a half-turn in the Bloch sphere about the x axis.



With respect to the computational basis, the X gate is equivalent to a classical NOT operation, or logical negation. The computation basis states are interchanged, so that $|0\rangle$ becomes $|1\rangle$ and $|1\rangle$ becomes $|0\rangle$.

$$\begin{aligned} X &= |1\rangle\langle 0| + |0\rangle\langle 1| \\ X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned}$$

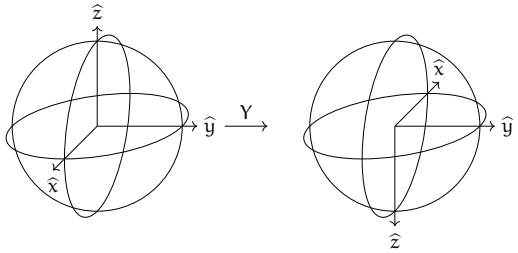
However, the X-gate is not a true quantum NOT gate, since it only logically negates the state in the computational basis. A true quantum logical negation would require mapping every point on the Bloch sphere to its antipodal point. But that would require an inversion of the sphere which cannot be generated by rotations alone. There is no general quantum NOT operation that would negate an arbitrary qubit state.

Pauli-Y gate (Y-gate):

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{6}$$

A useful mnemonic for remembering where to place the minus sign in the matrix of the Y gate is "Minus eye high" [0]. In some older literature the Y-gate is defined as $iY = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$ (e.g. [37]), which is the same gate up to a phase.

The Pauli-Y gate generates a half-turn in the Bloch sphere about the $\hat{y}$ axis.



The Y-gate can be thought of as a combination of X and Z gates, $Y = -iZX$. With respect to the computational basis, we interchange the zero and one states and apply a relative phase flip.

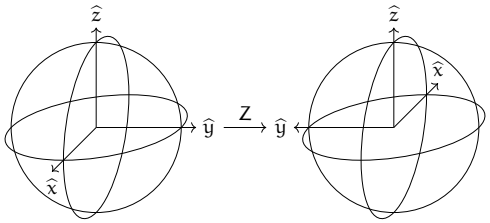
$$\begin{aligned} Y &= i|1\rangle\langle 0| - i|0\rangle\langle 1| \\ Y|0\rangle &= +i|1\rangle \\ Y|1\rangle &= -i|0\rangle \end{aligned}$$

Pauli-Z gate (Z-gate, phase flip)

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{7}$$

$$H_Z = -\pi\tfrac{1}{2}(1 - Z)$$

The Pauli-Z gate generates a half-turn in the Bloch sphere about the $\hat{z}$ axis.



With respect to the computational basis, the Z gate flips the phase of the $|1\rangle$ state relative to the $|0\rangle$ state.

$$\begin{aligned} Z &= |0\rangle\langle 0| - |1\rangle\langle 1| \\ Z\,|0\rangle &= +|0\rangle \\ Z\,|1\rangle &= -|1\rangle \end{aligned}$$

3.2 Rotation gates

The three Pauli-rotation gates[2] R_x, R_y, and R_z rotate the state vector by an arbitrary angle about the corresponding axis in the Bloch sphere, Fig. 3.1. They are generated by taking exponentials of the Pauli operators.

A useful identity to keep in mind is that given an operator A that squares to the identity $A^2 = I$ then

$$\exp(i\theta A) = \cos(\theta)\, I + i \sin(\theta)\, A \tag{8}$$

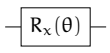
This is a generalization of the usual Euler's formula $e^{ix} = \cos x + i \sin x$. We expand the exponential as a power series, and gather the even powers into the cosine term, and the odd powers into the sin term.

$$\begin{aligned} \exp(i\theta A) &= I + i\theta A - \tfrac{\theta^2}{2!} I - i\tfrac{\theta^3}{3!} A - \tfrac{\theta^4}{4!} I - i\tfrac{\theta^5}{5!} A + + \cdots \\ &= \left(1 - \tfrac{\theta^2}{2!} + \tfrac{\theta^4}{4!} - \cdots\right) I + \left(\theta - \tfrac{\theta^3}{3!} + \tfrac{\theta^5}{5!} - \cdots\right) A \\ &= \cos(\theta)\, I + i \sin(\theta)\, A \end{aligned}$$

R_x gate [14] Rotate θ radians anti-clockwise about the $\hat{x}$ axis of the Bloch sphere.

$$\begin{aligned} R_x(\theta) &= e^{-i\frac{1}{2}\theta X} \\ &= \cos(\tfrac{1}{2}\theta) I - i \sin(\tfrac{1}{2}\theta) X \\ &= \begin{bmatrix} \cos(\frac{1}{2}\theta) & -i\sin(\frac{1}{2}\theta) \\ -i\sin(\frac{1}{2}\theta) & \cos(\frac{1}{2}\theta) \end{bmatrix} \\ H_{R_x} &= \tfrac{1}{2}\theta X \end{aligned} \tag{9}$$

The R_x gate is represented by the following circuit diagram.

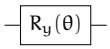


or, if we want to specify a generic R_x gate, and not a specific angle, we can drop the theta argument.

[2] The 1-qubit rotation gates are typically verbalized as *arr-ex*, *arr-why*, *arr-zee*, and *arr-en*.

R_y gate [14] Rotate θ radians anti-clockwise about the $\hat{y}$ axis of the Bloch sphere.

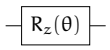
$$\begin{aligned} R_y(\theta) &= e^{-i\frac{1}{2}\theta Y} \\ &= \cos(\tfrac{1}{2}\theta)I - i\sin(\tfrac{1}{2}\theta)Y \\ &= \begin{bmatrix} \cos(\frac{1}{2}\theta) & -\sin(\frac{1}{2}\theta) \\ \sin(\frac{1}{2}\theta) & \cos(\frac{1}{2}\theta) \end{bmatrix} \end{aligned} \tag{10}$$



$$H_{R_y} = \tfrac{1}{2}\theta Y$$

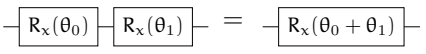
R_z gate [14] Rotate θ radians anti-clockwise about the $\hat{z}$ axis of the Bloch sphere.

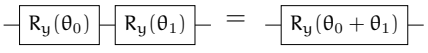
$$\begin{aligned} R_z(\theta) &= e^{-i\frac{1}{2}\theta Z} \\ &= \cos(\tfrac{1}{2}\theta)I - i\sin(\tfrac{1}{2}\theta)Z \\ &= \begin{bmatrix} e^{-i\frac{1}{2}\theta} & 0 \\ 0 & e^{+i\frac{1}{2}\theta} \end{bmatrix} \end{aligned} \tag{11}$$

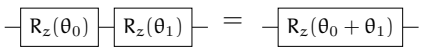


$$H_{R_z} = \tfrac{1}{2}\theta Z$$

Consecutive rotations about the same axis can be merged, with the total angle being the sum of angles.







Let us demonstrate that the R_z gate generates rotations about the $\hat{z}$ axis. Recall the definition of the Bloch vector of an arbitrary state $|\psi\rangle$, (§2).

$$\begin{aligned} R_z(\theta')|\psi\rangle &= \left(e^{-i\frac{1}{2}\theta'}|0\rangle\langle 0| + e^{+i\frac{1}{2}\theta'}|1\rangle\langle 1|\right)\left(\cos(\tfrac{1}{2}\theta)|0\rangle + e^{i\Phi}\sin(\tfrac{1}{2}\theta)|1\rangle\right) \\ &= e^{-i\frac{1}{2}\theta'}\left(\cos(\tfrac{1}{2}\theta)|0\rangle + e^{i(\theta'+\Phi)}|1\rangle\right) \\ &\simeq \cos(\tfrac{1}{2}\theta)|0\rangle + e^{i(\theta'+\Phi)}|1\rangle \end{aligned} \tag{12}$$

In the last line we drop an irrelevant phase. We can see that the R_z gate has left the elevation angle unchanged, but added θ' to the azimuth angle, which corresponds to a rotation about the $\hat{z}$-axis.

We can do the same exercise for the R_X and R_Y gates, although the trigonometry is slightly more involved.

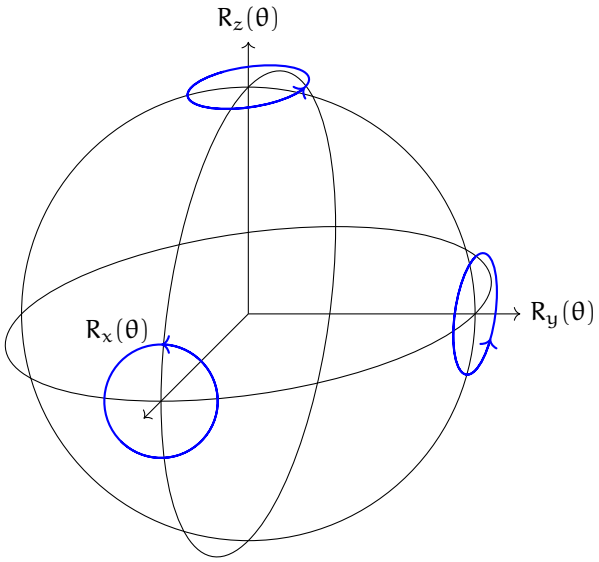


Figure 3.1: Pauli rotations of the Bloch Sphere

$R_{\vec{n}}$ gate A rotation of θ radians anti-clockwise about an arbitrary axis in the Bloch sphere.

$$\begin{aligned} R_{\vec{n}}(\theta) &= e^{-i\frac{1}{2}\theta(n_x X + n_y Y + n_z Z)} \\ &= \cos(\tfrac{1}{2}\theta) I - i \sin(\tfrac{1}{2}\theta)(n_x X + n_y Y + n_z Z) \\ &= \begin{bmatrix} \cos(\frac{1}{2}\theta) - i n_z \sin(\frac{1}{2}\theta) & -n_y \sin(\frac{1}{2}\theta) - i n_x \sin(\frac{1}{2}\theta) \\ n_y \sin(\frac{1}{2}\theta) - i n_x \sin(\frac{1}{2}\theta) & \cos(\frac{1}{2}\theta) + i n_z \sin(\frac{1}{2}\theta) \end{bmatrix} \end{aligned} \tag{13}$$

$$-\boxed{R_{\vec{n}}(\theta)}-$$

Every 1-qubit gate can be represented as a rotation gate (up to phase) with some coordinate $(\theta\, n_x, \theta\, n_y, \theta\, n_z)$, where $n_x^2 + n_y^2 + n_z^2 = 1$ and θ runs between π and $-\pi$. The Pauli gates are the rotations around the principal axes.

$$\begin{aligned} R_x(\theta) &= R_{\vec{n}}(\theta), \quad \vec{n} = (1,0,0) \\ R_y(\theta) &= R_{\vec{n}}(\theta), \quad \vec{n} = (0,1,0) \\ R_z(\theta) &= R_{\vec{n}}(\theta), \quad \vec{n} = (0,0,1) \end{aligned}$$

This representation provides a convenient visualization of 1-qubit gates: The 1-qubit gates form a spherical ball of radius θ. See figures 3.2 and 3.3. This sphere-of-gates is distinct from the Bloch sphere of states, although the underlying mathematical structures are related.

You might reasonably be wondering why there is a factor of half in the definitions of the rotation gates. A 1-qubit gate is represented by an element of the group $SU(2)$ (the group of 2×2 unitary matrices with unit determinate). Each element is a rotation in a 2-dimensional complex vector space. But we are visualizing the effect of these gates as rotations in 3-dimensional Euclidean space, which are elements of the special orthogonal group $SO(3)$. We can do this because there is an accidental correspondence between these two groups that allows us to visualize 1-qubit gates as rotations in 3-space. We can map two elements of $SU(2)$ (differing by only a -1

phase) to each element of $SO(3)$ while keeping the group structure. In the jargon, $SU(2)$ is a double cover of $SO(3)$. Because of this doubling up, a rotation of θ radians in the Bloch sphere corresponds to a rotation of only $\frac{1}{2}\theta$ in the complex vector space. We have to go twice around the Bloch sphere, $\theta = 4\pi$, to get back to the same gate with the same phase.

3.3 Pauli-power gates

It turns out to be useful to define powers of the Pauli-gates. This is slightly tricky because non-integer powers of matrixes aren't unique. Just as there are 2-square roots of any number, a diagonalizable matrix with n unique eigenvalue has 2^n unique square roots. We circumvent this ambiguity by defining the Pauli power gates via the Pauli rotation gates. We note that a π rotation is a Pauli gate up to phase, e.g.

$$R_X(\pi) = e^{-i\frac{\pi}{2}X} = -iX \tag{14}$$

and define powers of the Pauli matrices as

$$X^t = e^{-i\frac{\pi}{2}t(X-I)} \simeq R_x(\pi t)\,, \tag{15}$$

and similarly for Y and Z rotations. With this definition the Pauli-power gates spin states in the same direction around the Bloch sphere as the Pauli-rotation gates.

The Pauli rotation-representation is more natural from the point of view of pure mathematics. But the Pauli-power representation has computational advantages. In quantum circuits we most often encounter rotations of angles $\pm\pi/2^n$ for some integer n. Whereas it is easy to spot that $Z^{0.125}$ is a T gate, for example, it is less obvious that $R_z(0.78538\ldots)$ is the same gate up to phase. Moreover binary fractions have exact floating point representations, whereas fractions of π inevitably suffer from numerical round-off error.

X power gate

$$\begin{aligned} X^t &= e^{-i\frac{\pi}{2}t(X-I)} = e^{i\frac{\pi}{2}t}R_x(\pi t) \\ &= e^{i\frac{\pi}{2}t}\begin{bmatrix} \cos(\frac{\pi}{2}t) & -i\sin(\frac{\pi}{2}t) \\ -i\sin(\frac{\pi}{2}t) & \cos(\frac{\pi}{2}t) \end{bmatrix} \end{aligned} \tag{16}$$

Y power gate

$$\begin{aligned} Y^t &= e^{-i\frac{\pi}{2}t(Y-I)} = e^{i\frac{\pi}{2}t}R_y(\pi t) \\ &= e^{i\frac{\pi}{2}t}\begin{bmatrix} \cos(\frac{\pi}{2}t) & -\sin(\frac{\pi}{2}t) \\ \sin(\frac{\pi}{2}t) & \cos(\frac{\pi}{2}t) \end{bmatrix} \end{aligned} \tag{17}$$

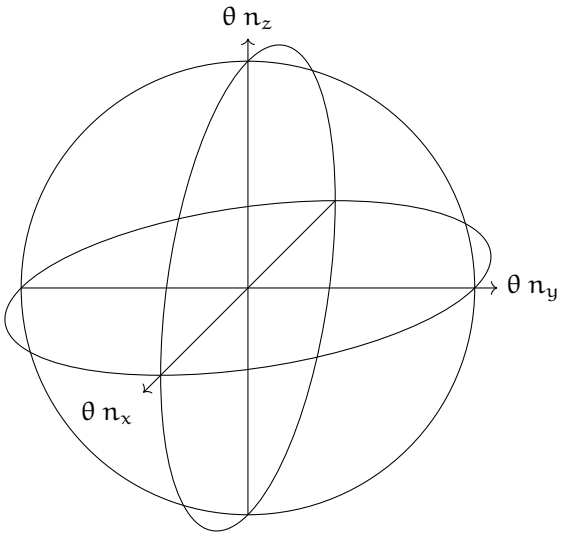


Figure 3.2: Spherical ball of 1-qubit gates (13). Each point within this sphere represents a unique 1-qubit gate (up to phase). Antipodal points on the surface represent the same gate. The Pauli rotation gates lie along the three principal axes.

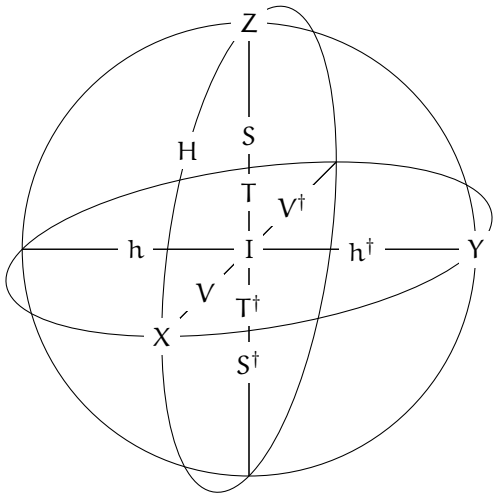


Figure 3.3: Coordinates of common 1-qubit gates (13).

Z power gate

$$\begin{aligned} Z^t &= e^{-i\frac{\pi}{2}t(Z-I)} = e^{i\frac{\pi}{2}t} R_z(\pi t) \\ &= e^{i\frac{\pi}{2}t} \begin{bmatrix} e^{-i\frac{\pi}{2}t} & 0 \\ 0 & e^{+i\frac{\pi}{2}t} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{+i\pi t} \end{bmatrix} \end{aligned} \tag{18}$$

Phase shift gate [0] The name arrises because this gate shifts the phase of the $|1\rangle$ state relative to the $|0\rangle$ state.

$$\begin{aligned} P(\theta) &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \\ &= e^{-i\frac{1}{2}\theta}\, R_z(\theta) \\ &= Z^{\frac{\theta}{\pi}} \end{aligned} \tag{19}$$

Sometimes favored over the R_z gate because special values are exactly equal to various other common gates. For instance, $R_\pi = Z$, but $R_z(\pi) = -iZ$. The CPhase gate (65) is a controlled phase shift. The phase shift gate also appears when considering the construction of controlled unitary gates (§7.5).

This gate is also commonly notated as R_θ, but I have adopted the notation $P(\theta)$ (which is also used in qiskit and QASM [0]), in an attempt to reduce confusion with all the other "R-subscript" gates. Note that historically 'P' was also used for the S gate, e.g. [0]

Fractional phase shift gate [0] Discrete fractional powers of the Z gate have their own notation. They most notably appear as controlled operations in the quantum Fourier transform (§**??**).

$$\begin{aligned} P_k &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{bmatrix} \\ &= P(2\pi/2^k) = Z^{2^{1-k}} \\ P_1 &= Z \\ P_2 &= S \\ P_3 &= T \end{aligned} \tag{20}$$

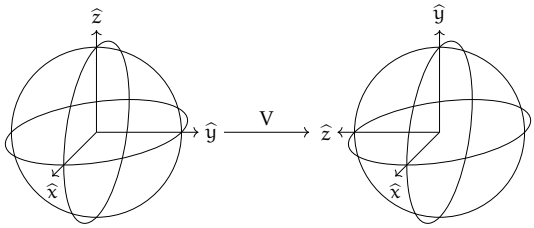
Thus P_1 is a half turn in the Bloch sphere, P_2 a quarter turn, P_3 an eighth turn, and so on. Most often notated as R_k, or sometimes as Z_k, here, as with the phase shift gate, I've adopted P_k is a vain hope of reducing ambiguity.

3.4 Quarter turns

V gate [0, 0] Square root of the X-gate, $VV = X$.

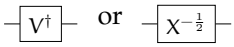
$$\begin{aligned} V &= X^{\frac{1}{2}} \\ &= \tfrac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \\ &= HSH \\ &\simeq R_x(+\tfrac{\pi}{2}) \end{aligned} \tag{21}$$

A quarter turn anti-clockwise about the $\hat{x}$ axis.

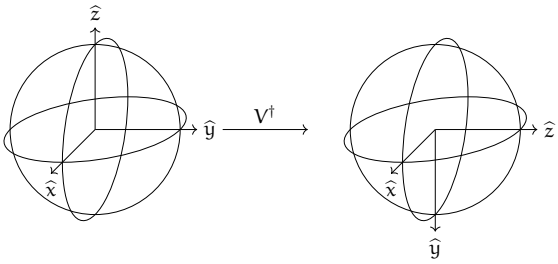


Inverse V gate Since the V-gate isn't Hermitian, the inverse gate, $V^\dagger$, is a distinct square root of X.

$$\begin{aligned} V^\dagger &= X^{-\frac{1}{2}} \\ &= \tfrac{1}{2}\begin{bmatrix} 1-i & 1+i \\ 1+i & 1-i \end{bmatrix} \\ &= HS^\dagger H \\ &\simeq R_x(-\tfrac{\pi}{2}) \end{aligned} \tag{22}$$



A quarter turn clockwise about the $\hat{x}$ axis.

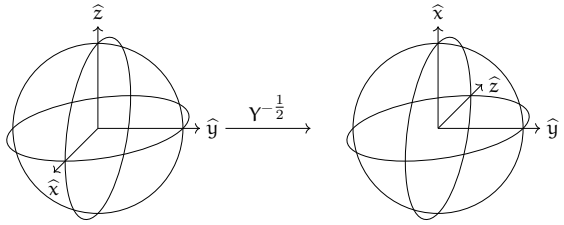


Pseudo-Hadamard gate [56, 20]: Inverse square root of the Y-gate.

$$\begin{aligned} h &= \tfrac{\sqrt{2}}{1+i} Y^{-\frac{1}{2}} \\ &= \tfrac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \end{aligned} \tag{23}$$

$$-\boxed{\mathrm{h}}- \quad \text{or} \quad -\boxed{Y^{-\frac{1}{2}}}-$$

A quarter turn clockwise about the $\hat{y}$ axis.



This square-root of the Y-gate is called the pseudo-Hadamard gate as it has the same effect on the computational basis as the Hadamard gate.

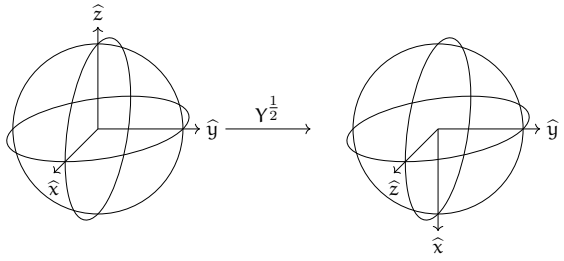
$$\mathrm{h}\,|0\rangle = |+\rangle$$
$$\mathrm{h}\,|1\rangle = |-\rangle$$

Inverse pseudo-Hadamard gate Principle square root of the Y-gate. Unlike the Hadamard gate, the pseudo-Hadamard gate is not Hermitian, and therefore not its own inverse.

$$\begin{aligned} \mathrm{h}^\dagger &= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \\ &= \tfrac{\sqrt{2}}{1+\mathrm{i}} Y^{\frac{1}{2}} \end{aligned} \tag{24}$$

$$-\boxed{\mathrm{h}^\dagger}- \quad \text{or} \quad -\boxed{Y^{\frac{1}{2}}}-$$

A quarter turn anti-clockwise about the $\hat{y}$ axis.

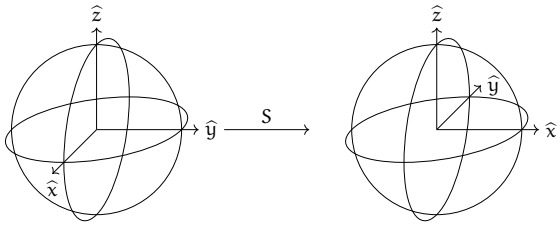


S gate (Phase, P, "ess") Square root of the Z-gate, $SS = Z$.

$$\begin{aligned} \mathrm{S} &= Z^{\frac{1}{2}} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{bmatrix} \\ &\simeq \mathrm{R}_z(+\tfrac{\pi}{2}) \end{aligned} \tag{25}$$

$$-\boxed{\mathrm{S}}-$$

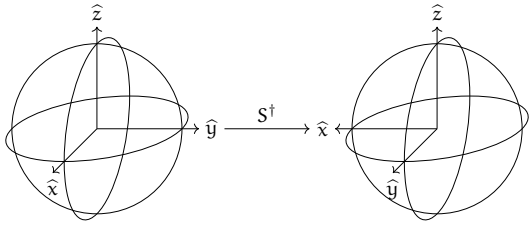
Historically called the phase gate (and denoted by P), since it shifts the phase of the one state relative to the zero state. This is a bit confusing because we have to make the distinction between the phase gate and applying a global phase. Often referred to as simple the S ("ess") gate in contemporary discourse.



Inverse S gate Hermitian conjugate of the S gate, and an alternative square-root of Z, $S^\dagger S^\dagger = Z$.

$$\begin{aligned} S^\dagger &= Z^{-\frac{1}{2}} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} \\ &\simeq R_z(-\tfrac{\pi}{2}) \end{aligned} \tag{26}$$

A quarter turn clockwise about the $\hat{z}$ axis.



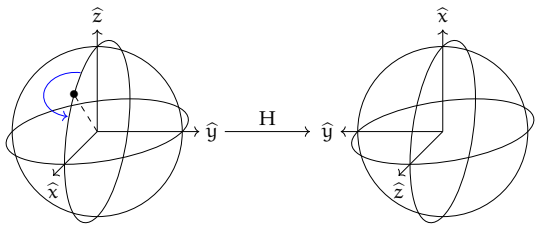
Can be generated from the S gate, $SSS = S^\dagger$.

3.5 Hadamard gates

Hadamard gate The Hadamard gate is one of the most interesting and useful of the common gates. Its effect is a π rotation (half turn) in the Bloch sphere about the axis $\frac{1}{\sqrt{2}}(\hat{x}+\hat{z})$. In a sense the Hadamard gate is half way between the Z and X gates (Fig. **??**).

$$\begin{aligned} H &= \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &\simeq R_{\vec{n}}(\pi), \quad \vec{n} = \tfrac{1}{\sqrt{2}}(1,0,1) \end{aligned} \tag{27}$$

In terms of the Bloch sphere, the Hadamard gate interchanges the $\hat{x}$ and $\hat{z}$ axes, and inverts the $\hat{y}$ axis.



A Hadamard similarity transform interchanges X and Z gates,

$$HXH = Z, \qquad HYH = -Y, \qquad HZH = X$$
$$HR_x(\theta)H = R_z(\theta), \qquad HR_y(\theta)H = R_y(-\theta), \qquad HR_z(\theta)H = R_x(\theta)$$

One reason that the Hadamard gate is so useful is that it acts on the computation basis states to create superpositions of zero and one states. These states are common enough that they have their own notation, $|+\rangle$ and $|-\rangle$.

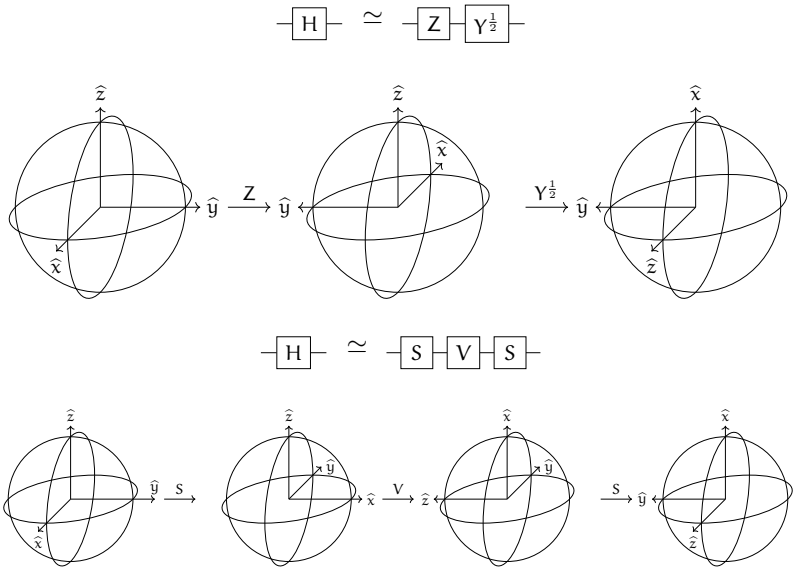
$$H|0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$$
$$H|1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$$

The square of the Hadamard gate is the identity $HH = I$. This is easy to show with some simple algebra, or by considering that the Hadamard is a 180 degree rotation in the Bloch sphere, or by noting that the Hadamard matrix is both Hermitian and unitary, so the Hadamard must be its own inverse. As a consequence, the Hadamard converts the $|+\rangle$, $|-\rangle$ Hadamard basis back to the $|0\rangle$, $|1\rangle$ computational basis.

$$H|+\rangle = |0\rangle$$
$$H|-\rangle = |1\rangle$$

The Hadamard gate is named for the *Hadamard transform* (Or *Walsh-Hadamard transform*), which in the context of quantum computing is the simultaneous application of Hadamard gates to multiple-qubits. We will return this transform presently (§**??**). The Hadamard gate is also the 1-qubit quantum Fourier transform (§**??**)

It is also worth noting a couple of useful decompositions (up to phase).



Here, V is the square-root of the X gate, and S is the square-root of Z, each of which is a quarter turn in the Bloch sphere.

Hadamard-like gates If we peruse the sphere of 1-qubit gates, Fig. 3.3, we can see that there are 6 different Hadamard-like gates that lie between the main $\hat{x}$, $\hat{y}$, and $\hat{z}$ axes. (Recall that gates on opposite sides of the sphere's surface are the same up to

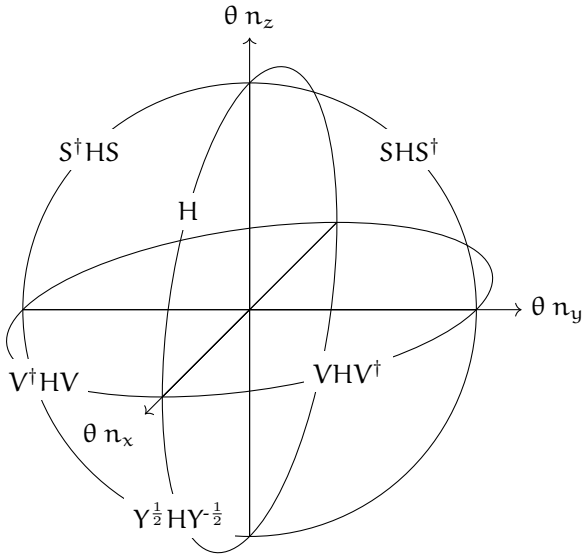
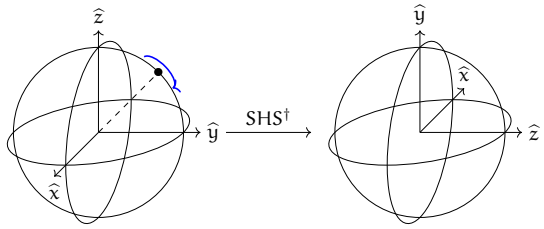


Figure 3.4: Coordinates of the 6-Hadamard like gates.

phase.) Each of these gates can be obtained from straightforward transform so the Hadamard gate. For instance, $H_{YZ} = SHS^\dagger$ is the Hadamard-like gate between the Z and Y gates, which interchanges the $\hat{y}$ and $\hat{z}$ axies, and flips the $\hat{x}$-axis.



This particular Hadamard-like gate takes the computational Z-basis to the Y-basis.

$$SHS^\dagger |0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) = |+i\rangle$$
$$SHS^\dagger |1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) = |-i\rangle$$

The coordinates of all 6 Hadamard-like gates are shown in Fig. 3.5, and listed in Table 12.1 in the same block as the Hadamard gate.

3.6 Axis cycling gates

Another interesting, but rarely discussed[3] class of gates are those that interchange three axes. These gates have periodicity 3 and represent 120 degree rotations of the Bloch sphere.

[3]Period 3 axis cycling gates are widely discussed abstractly in the context of Clifford gates (§**??**). I've borrowed the explicit realization and nomenclature from Craig Gidney's `stim` python package, a simulator for quantum stabilizer circuits. https://github.com/quantumlib/Stim [53]

C gate [53]

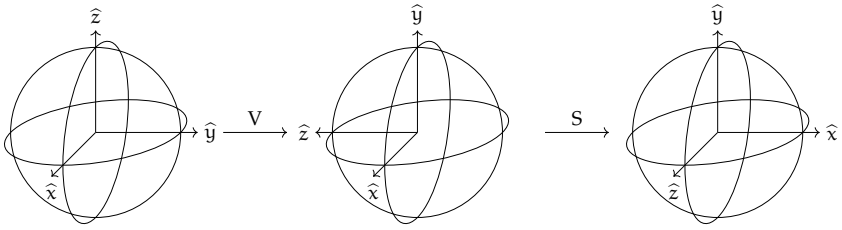
$$\begin{aligned} C &= \tfrac{1}{2}\begin{bmatrix} +1-i & -1-i \\ +1-i & +1+i \end{bmatrix} \\ &= R_n(\tfrac{2}{3}\pi), \quad n = (\tfrac{1}{\sqrt{3}}, \tfrac{1}{\sqrt{3}}, \tfrac{1}{\sqrt{3}}) \end{aligned} \tag{28}$$

A right handed period 3 axis cycling gate, cycling the axes in the permutation $\hat{x} \to \hat{y} \to \hat{z} \to \hat{x}$

$$\begin{aligned} C\, X^t\, C^\dagger &= Y^t \\ C\, Y^t\, C^\dagger &= Z^t \\ C\, Z^t\, C^\dagger &= X^t \end{aligned}$$

Note that this is a third root of the identity, $C^3 = I$, and that the square gives the inverse gate $C^2 = C^\dagger$ which cycles in the opposite direction.

There are 8 distinct axis cycling gates, which are all also Clifford gates, and listed in the last block of table 12.1. Each such gate can be broken down into a combination of two quarter turns, e.g. $C = SV$.



3.7 T gates

All the of preceding discrete 1-qubit gates (Pauli gates, quarter turns, Hadamard and Hadamard-like gates, and axis cycling gates) are examples of a special class of gates called Clifford gates. Although important, the Clifford gates have the notable restricting that they aren't universal – you can't build an arbitrary qubit rotation from Clifford gates alone. The is because the Clifford gates always map the $\hat{x}$, $\hat{y}$ and $\hat{z}$ axes back onto themselves. In order to be computational universal, it is necessary to have at least one non-Clifford gate in our gate set, and the most common choice for that non-Clifford gate is the T gate, one eighth of a rotation anti-clockwise about the z axis. A gate set consisting of all Cliffords (including multi-qubit Cliffords) and the T gate is often written as "Clifford+T".

T gate ("tee", $\pi/8$) Forth root of the Z gate, $T^4 = Z$.

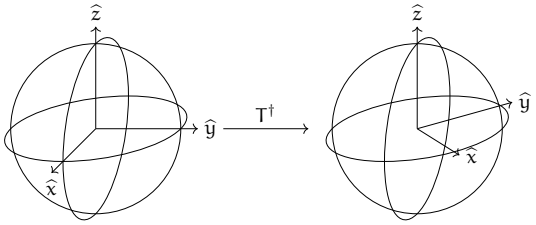
$$\begin{aligned} T &= Z^{\frac{1}{4}} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \end{aligned} \tag{29}$$

The T gate has sometimes been called the $\pi/8$ gate since we can extract a phase

and write the T gate as

$$T = e^{i\frac{\pi}{8}\pi}\begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{+i\frac{\pi}{8}} \end{bmatrix}$$

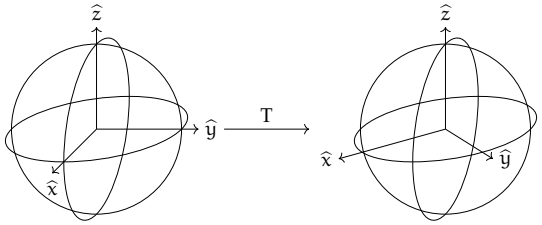
An eight turn anti-clockwise about the $\hat{z}$ axis.



Inverse T gate Hermitian conjugate of the T gate.

$$\begin{aligned} T^\dagger &= Z^{-\frac{1}{4}} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} \\ &\simeq R_z(\tfrac{pi}{4}) \end{aligned} \tag{30}$$

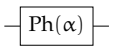
An eighth turn clockwise about the $\hat{z}$ axis.



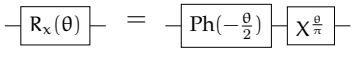
3.8 Global phase

Global phase gate (phase-shift) [14, 0, 0]

$$\begin{aligned} Ph(\alpha) &= e^{i\alpha} I \\ &= \begin{bmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \end{aligned} \tag{31}$$



To shift the global phase we multiply the quantum state by a scalar, so it is not necessary to assign a phase shift to any particular qubit. But on those occasions where we want to keep explicit track of the phase in a circuit, it is useful to assign a global phase shift to a particular qubit and temporal location, e.g.



This gate was originally called the phase-shift gate [14], but unfortunately the 1-qubit gate that shifts the phase of the 1 state relative the the zero state is also called the phase-shift gate (19), which is potentially confusing.

Omega gate [0, 0]

$$\begin{aligned} \omega^k &= \mathrm{Ph}(\tfrac{\pi}{4}k) \\ &= \begin{bmatrix} e^{i\frac{\pi}{4}k} & 0 \\ 0 & e^{i\frac{\pi}{4}k} \end{bmatrix} \end{aligned} \tag{32}$$

An alternative parameterization of a global phase shift. Note that this gate is an eight root of the identity, $\omega^8 = I$. This gate, with integer powers, crops up when constructing the 1-qubit Clifford gates from Hadamard and S gates, since $SHSHSH = \omega$ (see p. 66).

4 Decomposition of 1-qubit gates

A general 1-qubit gate corresponds to some 2 by 2 unitary matrix,

$$U = e^{i\alpha} \begin{bmatrix} a & -b^* \\ b & a^* \end{bmatrix} \tag{33}$$

where a and b are complex with $|a|^2 + |b|^2 = 1$, and α is real. Given such a generic unitary, we would like to represent this gate using standard parameterized gates.

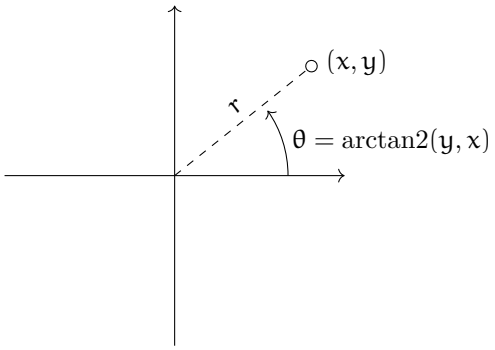
The first step to deke[4] a gate is to extract the phase factor α,

$$V = e^{i\alpha} U \tag{34}$$

so that V is a special unitary matrix with $\det V = 1$. In general, if we multiply a special unitary matrix by a complex phase c then $\det cU = c^k$ where k is the rank of the matrix, i.e. $k = 2^n$ for n qubits. This follows since the determinate is the product of the eigenvalues, and multiplying a matrix by a constant multiplies each eigenvalue by that constant. Thus the determinate of U is $\det U = e^{i2\alpha}$, and we can extract the phase factor α with some trigonometry.

$$\alpha = \tfrac{1}{2} \arctan2\big(\mathrm{Im}(\det U), \mathrm{Re}(\det U)\big) \tag{35}$$

The two-argument arctangent function $\arctan2(y, x)$ returns the angle θ between x-axis and the ray from the origin to (x, y). In contrast the single argument arctangent function $\arctan(y/x)$ only gives the correct answer for $x > 0$ since it can't distinguish between (x, y) and $(-x, -y)$.



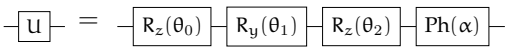
For a complex number $x + iy = re^{i\theta}$, the modulus (or magnitude) $r = \sqrt{x^2 + y^2}$ and the phase (or argument) is $\theta = \arctan2(y, x)$.

4.1 Z-Y-Z decomposition

Any 1-qubit gate can be decomposed as a sequence of Z, Y, and Z rotations, and a phase [14][5].

$$U = e^{i\alpha}\, R_z(\theta_2)\, R_y(\theta_1)\, R_z(\theta_0) \tag{36}$$

Or in circuit notation.



[4] *deke —dek—* verb — To decompile, deconstruct, or decompose.
1995 Neal Stephenson *The Diamond Age* "We gotta deke all this stuff now" Easy come, easy go.

[5] The Z-Y decomposition is of ancient origin, long know in the theory of light polarization [**?**]

Note that we have numbered the three angles in chronological order, and recall that time runs right-to-left in operator notation, but left-to-right in circuit notation.

If multiply out the circuit, then we get the following universal 1-qubit gate.

$$U = e^{i\alpha} \begin{bmatrix} +e^{+i(+\frac{1}{2}\theta_2+\frac{1}{2}\theta_0)}\cos(\frac{1}{2}\theta_1) & -e^{-i(-\frac{1}{2}\theta_2+\frac{1}{2}\theta_0)}\sin(\frac{1}{2}\theta_1) \\ +e^{+i(+\frac{1}{2}\theta_2-\frac{1}{2}\theta_0)}\sin(\frac{1}{2}\theta_1) & +e^{-i(+\frac{1}{2}\theta_2+\frac{1}{2}\theta_0)}\cos(\frac{1}{2}\theta_1) \end{bmatrix} \tag{37}$$

The first step in the decomposition is to extract the phase using Eq. (35), leaving a special unitary matrix $V = e^{-i\alpha}U$. The value of θ_1 can be calculated from the absolute value of either the diagonal or off-diagonal elements, provided those entries aren't close to zero. For instance, the Z-gate has zero off-diagonal entries, whereas the X-gate has zeros on the diagonal. But the diagonal and off-diagonal entries can't approach zero at the same time. So to calculate θ_1 with greatest numerical accuracy, we use whichever element has the largest absolute value.

$$\theta_1 = \begin{cases} 2\arccos(|V_{00}|), & |V_{00}| \geqslant |V_{01}| \\ 2\arcsin(|V_{01}|), & |V_{00}| < |V_{01}| \end{cases} \tag{38}$$

Having extracted θ_1, we can now calculate the sum $\theta_0 + \theta_1$ from V_{11} using the $\operatorname{arctan2}$ function.

$$\theta_0 + \theta_2 = 2\ \operatorname{arctan2}\left(\operatorname{Im}(\frac{V_{11}}{\cos(\frac{1}{2}\theta_1)}),\ \operatorname{Re}(\frac{V_{11}}{\cos(\frac{1}{2}\theta_1)})\right). \tag{39}$$

except if $\cos(\frac{1}{2}\theta_1) = 0$ then $\theta_0 + \theta_2 = 0$.

Similarly we can extract the difference $\theta_0 - \theta_2$ from V_{10}.

$$\theta_0 - \theta_2 = 2\ \operatorname{arctan2}\left(\operatorname{Im}(\frac{V_{10}}{\sin(\frac{1}{2}\theta_1)}),\ \operatorname{Re}(\frac{V_{10}}{\sin(\frac{1}{2}\theta_1)})\right) \tag{40}$$

again with an exception that if $\sin(\frac{1}{2}\theta_1) = 0$ then $\theta_0 - \theta_2 = 0$. Taking the sum and differences of (39) and (40) yields θ_0 and θ_2, which completes the decomposition.

Instead of rotation gates, we could express the same decomposition as Pauli-power gates with a reparameterization.

$$\begin{aligned} U =& e^{i\alpha'}\ Z^{t_2}\ Y^{t_1}\ Z^{t_0} \\ & \alpha' = \alpha - (\theta_0 + \theta_1 + \theta_2)/\pi \\ & t_0 = \theta_0/\pi \\ & t_1 = \theta_1/\pi \\ & t_2 = \theta_2/\pi \end{aligned} \tag{41}$$

4.2 V-Z decomposition

For some superconducting qubit architectures the natural 1-qubit gates are Z-rotations R_z and V (22)[6] the square root of X [0]. There isn't direct access to R_y rotations or general R_x rotations, but this is only a minor inconvenience since $R_z(\theta) = V^\dagger R_y(\theta) V$,

[6] Note that here V refers to a specific 1-qubit gate, the square-root of the X gate, whereas elsewhere V is used to denote a general unitary or special unitary matrix. Such notational ambiguities are inevitable since there's only so many squiggles to go around [0].

Table 4.1: Euler decompositions

Euler decomposition	Similarity transform to Z-Y-Z
X-Y-X	$H^\dagger$
X-Z-X	C
Y-X-Y	$C^\dagger$
Y-Z-Y	$VHV^\dagger$
Z-X-Z	$S^\dagger$
Z-Y-Z	I

(§**??**) and we can therefore decompose 1-qubit gates to a 5-gate sequence,

$$U = e^{i\alpha}\, R_z(\theta_2)\, V^\dagger\, R_z(\theta_1)\, V\, R_z(\theta_0)\,. \tag{42}$$

4.3 General Euler angle decompositions

Instead of a Z-Y-Z decomposition, we might instead desire a different decomposition, for example X-Y-X.

$$U = R_x(\theta_2)R_y(\theta_1)R_x(\theta_0) \tag{43}$$

The trick is to perform a similarity transform that takes us back to the Z-Y-Z decomposition that we already know how to perform.

$$\begin{aligned} V = CUC^\dagger &= CR_y(\theta_2)C^\dagger\, CR_z(\theta_1)C^\dagger\, CR_y(\theta_0)C^\dagger \\ &= R_z(\theta_2)R_y(\theta_1)R_z(\theta_0) \end{aligned} \tag{44}$$

Here we want the single qubit gate C that moves the $+\hat{y}$ axis to $+\hat{x}$, but leaves the $\hat{z}$ axis alone. Consulting page **??** we see that the required gate is $S^\dagger$. Therefore to find the parameters of a X-Y-X decomposition we carry out the similarity transform $V = S^\dagger\, U\, S$ and then perform a Z-Y-Z decomposition.

There are 6 distinct proper-Euler decompositions, and the appropriate similarity transforms to Z-Y-Z are listed in Table 4.1. These are all 1-qubit Clifford gates (Table 12.1).

4.4 Bloch rotation decomposition

Finally lets consider the decompositions of 1-qubit gates into single rotations about a particular axis (13).

$$R_{\vec{n}}(\theta) = \begin{bmatrix} \cos(\frac{1}{2}\theta) - in_z \sin(\frac{1}{2}\theta) & -n_y \sin(\frac{1}{2}\theta) - in_x \sin(\frac{1}{2}\theta) \\ n_y \sin(\frac{1}{2}\theta) - in_x \sin(\frac{1}{2}\theta) & \cos(\frac{1}{2}\theta) + in_z \sin(\frac{1}{2}\theta) \end{bmatrix} \tag{45}$$

Assuming that we have already extracted the phase and therefore V is a 1-qubit special unitary matrix, we can proceed as follows.

$$\begin{aligned} N &= \sqrt{(\operatorname{Im} V_{0,1})^2 + (\operatorname{Re} V_{0,1})^2 + (\operatorname{Im} V_{0,0})^2} \\ n_x &= -\operatorname{Im} V_{0,1}/N \\ n_y &= -\operatorname{Re} V_{0,1}/N \\ n_z &= -\operatorname{Im} V_{0,0}/N \\ s = \sin(\tfrac{1}{2}\theta) &= -\operatorname{Im} V_{0,0}/n_z \\ c = \cos(\tfrac{1}{2}\theta) &= \operatorname{Re} V_{0,0} \\ \theta &= 2\operatorname{arctan2}(s, c) \end{aligned} \tag{46}$$

The one ambiguous edge case that needs to be accounted for is that the identity can be represented as a zero-radians rotation about any axis.

4.5 Decomposition of Bloch rotation

A rotation about an arbitrary axis in the Bloch sphere can be analytically decomposed into a sequence of five R_z and R_y gates [30].

$$\begin{aligned} R_{\vec{n}}(\theta) &= R_z(+\alpha) R_y(+\beta) R_z(\theta) R_y(-\beta) R_z(-\alpha) \\ \alpha &= \operatorname{arctan2}(n_y, n_x) \\ \beta &= \arccos(n_z) \end{aligned} \tag{47}$$

5 The canonical gate

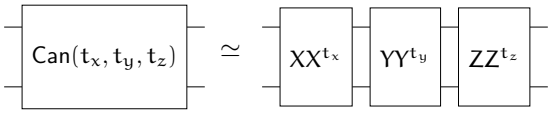
The canonical gate is a 3-parameter quantum logic gate that acts on two qubits [0, 0, 0].

$$\begin{aligned}&\mathsf{Can}(t_x, t_y, t_z)\\ &\quad = \exp\Big(-i\frac{\pi}{2}(t_x X\otimes X + t_y Y\otimes Y + t_z Z\otimes Z)\Big)\end{aligned} \tag{48}$$

Recall that $X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $Y = \left(\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right)$, and $Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ are the 1-qubit Pauli matrices, and that

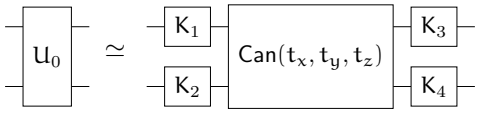
$$\begin{aligned} X\otimes X &= \begin{bmatrix} 0 & 0 & 0 & +1 \\ 0 & 0 & +1 & 0 \\ 0 & +1 & 0 & 0 \\ +1 & 0 & 0 & 0 \end{bmatrix},\\ Y\otimes Y &= \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & +1 & 0 \\ 0 & +1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \text{and}\\ Z\otimes Z &= \begin{bmatrix} +1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 \end{bmatrix}. \end{aligned}$$

Other parameterizations of the canonical gate are common in the literature. Often the sign is flipped, or the $\frac{\pi}{2}$ factor is absorbed into the parameters, or both. The parameterization used here the nice feature that it corresponds to powers of direct products of Pauli operators (up to phase) (see (62),(63),(64)).



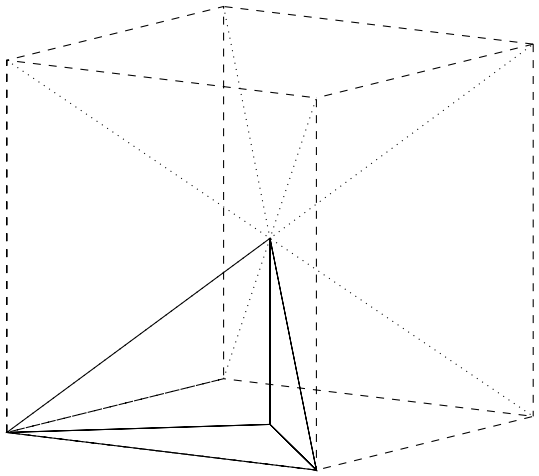
Here we use '$\simeq$' to indicate that two gates have the same unitary operator up to a global (and generally irrelevant) phase factor.

The canonical gate is, in a sense, the elementary 2-qubit gate, since any other 2-qubit gate can be decomposed into a canonical gate, and local 1-qubit interactions [0, 0, 23, 26, 28, 36].

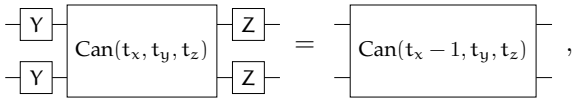


We will discuss the numerical decomposition of 2-qubit gates to canonical gates in section (§7.2). For know it is sufficient to now that the non-local properties of every 2-qubit gate can be characterized by the 3-parameters of the corresponding canonical gate. We'll use '$\sim$' to indicate that two gates are locally equivalent, in that they can be mapped to one another by local 1-qubit rotations.

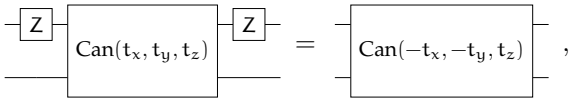
The parameters of the canonical gate are periodic with period 4, or period 2 if we neglect a -1 global phase factor. Thus we can constrain each parameter to the range $[-1, 1)$. Since $X\otimes X$, $Y\otimes Y$, and $Z\otimes Z$ all commute, the parameter space has the topology of a 3-torus.



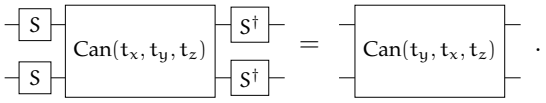
By applying local gates we can decrement any one of the canonical gate's parameters,



we can flip the signs on any pair of parameters,



or we can swap any pair of parameters,

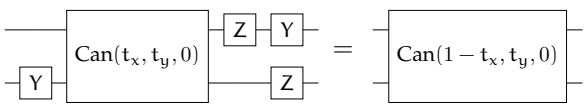


Because of these relations the canonical coordinates of any given 2-qubit gate are not unique since we have considerable freedom in the prepended and appended local gates. To remove these symmetries we can constraint the canonical parameters to a "Weyl chamber" [0, 0].

$$(\tfrac{1}{2} \geqslant t_x \geqslant t_y \geqslant t_z \geqslant 0) \cup (\tfrac{1}{2} \geqslant (1 - t_x) \geqslant t_y \geqslant t_z > 0) \tag{49}$$

This Weyl chamber forms a trirectangular tetrahedron. All gates in the Weyl chamber are locally inequivalent (They cannot be obtained from each other via local 1-qubit gates). The net of the Weyl chamber is illustrated in Fig. A.1, and the coordinates of many common 2-qubit gates are listed in table 5.1.

There is an additional symmetry across the bottom face of the chamber. Gates located at $\mathsf{Can}(t_x, t_y, 0)$ are locally equivalent to $\mathsf{Can}(1-t_x, t_y, 0)$, since we can now flip the sign of t_x without changing the other parameters.



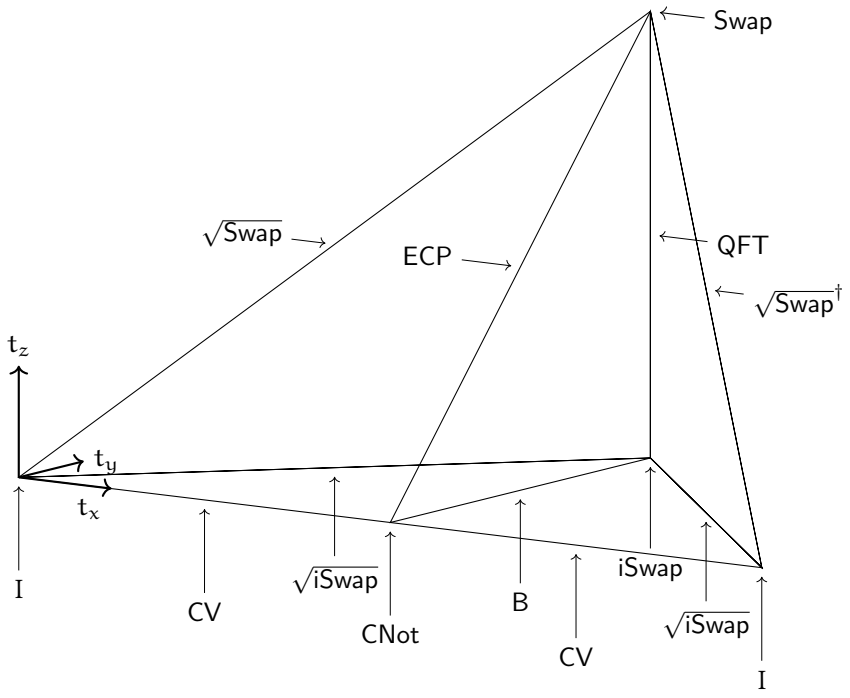
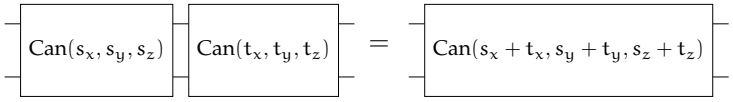


Figure 5.1: Location of the 11 principal 2-qubit gates in the Weyl chamber. All of these gates have coordinates of the form $\mathsf{Can}(\frac{1}{4}k_x, \frac{1}{4}k_y, \frac{1}{4}k_z)$, for integer k_x, k_y, and k_z. Note there is a symmetry on the bottom face such that $\mathsf{Can}(t_x, t_y, 0) \sim \mathsf{Can}(1 - t_x, t_y, 0)$.

Neighboring canonical gates can be merged by summing the parameters.



Taking the Hermitian conjugate of the canonical gate simple inverts the parameters $\mathsf{Can}(t_x, t_y, t_z)^\dagger = \mathsf{Can}(-t_x, -t_y, -t_z)$, and more generally powers of the canonical gate multiply the parameters, $\mathsf{Can}(t_x, t_y, t_z)^c = \mathsf{Can}(ct_x, ct_y, ct_z)$.

Table 5.1: Canonical coordinates of common 2-qubit gates

Gate	t_x	t_y	t_z	t'_x	t'_y	t'_z
	$\leqslant\frac{1}{2}$			¿$\frac{1}{2}$		
I_2	0	0	0	1	0	0
CNot / CZ / MS	$\frac{1}{2}$	0	0			
iSwap / DCNot	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{3}{4}$	$\frac{1}{2}$	0
Swap	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$			
CV	$\frac{1}{4}$	0	0	$\frac{3}{4}$	0	0
$\sqrt{\text{iSwap}}$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{3}{4}$	$\frac{1}{4}$	0
DB	$\frac{3}{8}$	$\frac{3}{8}$	0	$\frac{5}{8}$	$\frac{3}{8}$	0
$\sqrt{\text{Swap}}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$			
$\sqrt{\text{Swap}}^\dagger$				$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
B	$\frac{1}{2}$	$\frac{1}{4}$	0			
ECP	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$			
QFT_2	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$			
Sycamore	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{12}$			
Ising / CPhase	t	0	0			
XY	t	t	0	t	1-t	0
Exchange / Swap^α	t	t	t	t	1-t	1-t
PSwap	$\frac{1}{2}$	$\frac{1}{2}$	t			
Special orthogonal	t_x	t_y	0			
Improper orthogonal	$\frac{1}{2}$	t_y	t_z			
XXY	t	t	δ	t	1-t	δ
	δ	t	t	δ	t	t

6 Standard 2-qubit gates

There are four unique 2-qubits gates in the Clifford group (up to local 1-qubit Cliffords): the identity, CNot, iSwap, and Swap gates.

6.1 Identity

Identity gate The trivial no-operation gate on 2-qubits, represented by a 4x4 identity matrix. Acting on any arbitrary state, the gate leaves the state unchanged.

$$\begin{aligned} \mathrm{I}_2 &= \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix} = \mathrm{I}\otimes\mathrm{I} \\ &= \mathsf{Can}(0,0,0) \end{aligned} \tag{50}$$

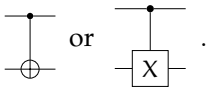
6.2 Controlled-Not gates

Controlled-Not gate (CNot, controlled-X, CX, Feynman) [0, 0]

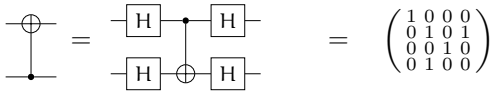
$$\begin{aligned} \mathsf{CNot} &= \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{pmatrix} \\ &\sim \mathsf{Can}(\tfrac{1}{2},0,0) \end{aligned} \tag{51}$$

$$\begin{aligned} \mathrm{H}_{\mathsf{CNot}} &= \tfrac{1}{2}(\mathrm{I}-\mathrm{Z})\otimes\mathrm{H}_{\mathrm{X}} \\ &= -\tfrac{\pi}{4}(\mathrm{I}-\mathrm{Z})\otimes(\mathrm{I}-\mathrm{X}) \end{aligned}$$

Typically represented by the circuit diagrams



The CNot gate is not symmetric between the two qubits. But we can switch control • and target ⊕ with local Hadamard gates.



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

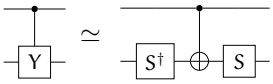
In classical logic a controlled-NOT has unambiguous control and target bits. The control bit influences the state of the target bit, and the target bit has no influence on the state of the control bit. But in quantum logic we can switch the apparent target and control with a local change of basis, which is essentially just a change in perspective as to which quantum states count as zero and one. In quantum logic there are no pure control operations *per se*. There is no unambiguous distinction between control and target. Joint operations on qubits create entanglement, and every action has a back reaction.

Controlled-Y gate

$$\begin{aligned} \mathsf{CY} &= \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&-\mathrm{i}\\ 0&0&+\mathrm{i}&0 \end{pmatrix} \\ &\sim \mathsf{Can}(\tfrac{1}{2},0,0) \end{aligned} \tag{52}$$

Commonly represented by the circuit diagram:

The CY gate is locally equivalent to CNot.

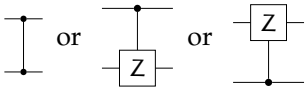


The CY gate is not encountered often, with the CNot (CX) and CZ gates being favored[7].

Controlled-Z gate (CZ, controlled-sign, or CSign)

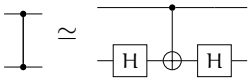
$$\begin{aligned} \mathsf{CZ} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\ &\sim \mathsf{Can}(\tfrac{1}{2}, 0, 0) \end{aligned} \tag{53}$$

Commonly represented by the circuit diagrams



Note that the controlled-Z gate is invariant to permutation of the qubits. So although we may conceive of this gate as a controlled operation, there is absolutely no distinction between control and target qubits.

The CZ gate is locally equivalent to the CNot gate.



The intuition is that the CNot gate applies an X gate to the $\oplus$ target qubit, and $\mathsf{HXH} = \mathsf{Z}$.

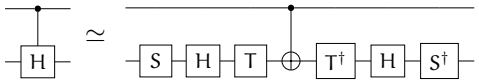
The CZ gate is frequently used as the elementary 2-qubit gate in circuit decompositions instead of the CNot gate. The CNot gate has the advantage that it directly corresponds to a classical reversible gate. On the other hand the CZ gate is intrinsically quantum (and therefore may be harder to reason about), but it has the advantages of being invariant to swapping qubits, and of being diagonal in the computational basis, which makes commutation relations easier to understand.

Controlled-Hadamard gate (CH) [**?** 35]

$$\mathsf{CH} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{54}$$

[7] Probably for no better reasons than that the CX and CZ gate operators don't feature imaginary numbers.

Occasionally turns up in applications, such as the decomposition of the W gate (**??**).



Mølmer-Sørensen gate (MS) [17, 57]

$$\begin{aligned} \mathsf{MS} &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 0 & \mathrm{i} \\ 0 & 1 & \mathrm{i} & 0 \\ 0 & \mathrm{i} & 1 & 0 \\ \mathrm{i} & 0 & 0 & 1 \end{pmatrix} \\ &= \mathsf{Can}(-\tfrac{1}{2}, 0, 0) \\ &\sim \mathsf{Can}(\tfrac{1}{2}, 0, 0) \\ &\sim \mathsf{CNot} \end{aligned} \tag{55}$$

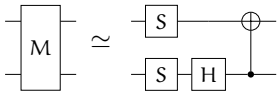
Proposed as a natural gate for laser driven trapped ions. Locally equivalent to CNot. The Mølmer-Sørensen gate, or more exactly its complex conjugate $\mathsf{MS}^\dagger = \mathsf{Can}(\frac{1}{2}, 0, 0)$ is the natural canonical representation of the CNot/CZ/MS gate family. (Note that Mølmer-Sørensen is also sometimes taken to be equivalent to the XX gate)

Magic gate (M) [0, 0, 0, 25]

$$\mathsf{M} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & \mathrm{i} & 0 & 0 \\ 0 & 0 & \mathrm{i} & 1 \\ 0 & 0 & \mathrm{i} & -1 \\ 1 & -\mathrm{i} & 0 & 0 \end{bmatrix} \tag{56}$$

$$\sim \mathsf{Can}(\tfrac{1}{2}, 0, 0) \tag{57}$$

The magic gate transforms to the *magic basis*, which has a number of useful properties. See (§7.2). Locally equivalent to CNot.



6.3 iSwap locally equivalent gates

iSwap **(imaginary swap) gate** [0, 0, 0] iSwap

$$\begin{aligned} \mathsf{iSwap} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & \mathrm{i} & 0 \\ 0 & \mathrm{i} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &\simeq \mathsf{Can}(-\tfrac{1}{2}, -\tfrac{1}{2}, 0) \end{aligned} \tag{58}$$

fSwap **(fermionic swap) gate** [0]fSwap

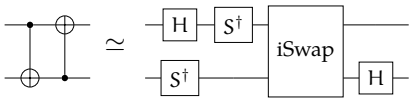
$$\begin{aligned} \mathsf{fSwap} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \\ &\sim \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{2}, 0) \end{aligned} \tag{59}$$

The fermionic swap gate swaps adjacent fermionic modes in the Jordan-Wigner representation. A qubit in a zero state represents a fermion (typically an electron) in an orbital, and a zero state represents a hole. Since the qubits are representing identical fermions, swapping two particles has to apply a -1 phase to the state.

Double Controlled NOT gate (DCNot)[21, 26]

$$\begin{aligned} \mathsf{DCNot} &= \begin{bmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&1&0&0\\ 0&0&1&0 \end{bmatrix} \\ &\sim \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{2}, 0) \end{aligned} \tag{60}$$

A CNot gate immediately followed by another CNot with control and target interchanged. The DCNot gate is in the iSwap locality class.



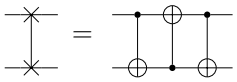
Note that unlike iSwap, action of DCNot is not invariant to the interchange of qubits.

6.4 Swap gate

A gate that swaps the state of two-qubits, located at the apex of the Weyl chamber [0, 0].

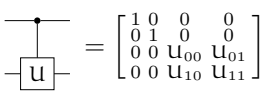
$$\begin{aligned} \mathsf{Swap} &= \begin{pmatrix} 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&0&1 \end{pmatrix} \\ &\simeq \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}) \\ \mathsf{H}_{\mathsf{Swap}} &= \tfrac{\pi}{4}(\mathsf{X}\otimes\mathsf{X} + \mathsf{Y}\otimes\mathsf{Y} + \mathsf{Z}\otimes\mathsf{Z} + \mathsf{I}\otimes\mathsf{I}) \end{aligned} \tag{61}$$

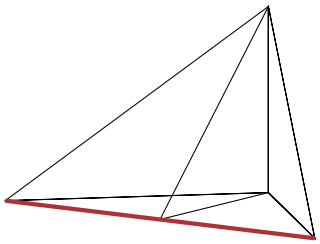
Swap gates are needed in physical realizations of quantum computers to move qubits into physical proximity so other gates can be performed between neighbors. In some cases this can be achieved by physically moving qubits. For example Honeywell's ion trap architecture can physically shift ions around the trap [0]. But in many cases physically moving qubits isn't possible. Swap gates can be synthesized from other quantum gates, most notable 1 Swap requires 3 CNot gates.



6.5 Ising gates

Gates in the Ising class have coordinates $\mathsf{Can}(t, 0, 0)$, which forms the front edge of the Weyl chamber. This includes the identity and CNot gates, and also all 2-qubit controlled unitary gates of the form





Ising gates

ZZ (Ising) gate

$$\begin{aligned} \mathsf{ZZ}(t) &= e^{-i\frac{\pi}{2}t\mathsf{Z}\otimes\mathsf{Z}} \\ &= \begin{pmatrix} e^{-i\frac{\pi}{2}t} & 0 & 0 & 0 \\ 0 & e^{+i\frac{\pi}{2}t} & 0 & 0 \\ 0 & 0 & e^{+i\frac{\pi}{2}t} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\pi}{2}t} \end{pmatrix} \\ &= \mathsf{Can}(0,0,t) \\ &\sim \mathsf{Can}(t,0,0) \end{aligned} \tag{62}$$

XX gate

$$\begin{aligned} \mathsf{XX}(t) &= e^{-i\frac{\pi}{2}t\mathsf{X}\otimes\mathsf{X}} \\ &= \begin{bmatrix} \cos(\frac{\pi}{2}t) & 0 & 0 & -i\sin(\frac{\pi}{2}t) \\ 0 & \cos(\frac{\pi}{2}t) & -i\sin(\frac{\pi}{2}t) & 0 \\ 0 & -i\sin(\frac{\pi}{2}t) & \cos(\frac{\pi}{2}t) & 0 \\ -i\sin(\frac{\pi}{2}t) & 0 & 0 & \cos(\frac{\pi}{2}t) \end{bmatrix} \\ &= \mathsf{Can}(t,0,0) \end{aligned} \tag{63}$$

YY gate

$$\begin{aligned} \mathsf{YY}(t) &= e^{-i\frac{\pi}{2}t\mathsf{Y}\otimes\mathsf{Y}} \\ &= \begin{pmatrix} \cos(\frac{\pi}{2}t) & 0 & 0 & +i\sin(\frac{\pi}{2}t) \\ 0 & \cos(\frac{\pi}{2}t) & -i\sin(\frac{\pi}{2}t) & 0 \\ 0 & -i\sin(\frac{\pi}{2}t) & \cos(\frac{\pi}{2}t) & 0 \\ +i\sin(\frac{\pi}{2}t) & 0 & 0 & \cos(\frac{\pi}{2}t) \end{pmatrix} \\ &= \mathsf{Can}(0,t,0) \\ &\sim \mathsf{Can}(t,0,0) \end{aligned} \tag{64}$$

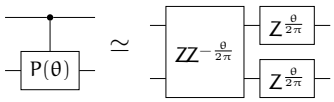
Notable the XX, YY and ZZ gates all commute with one another. This is because Pauli operators of different types anti-commute, but here we have pairs of Pauli's acting on separate qubits, so the gates commute.

CPhase (Controlled phase) gate [0, 41]

$$\begin{aligned}\mathrm{CPhase}(\theta) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\pi\theta} \end{bmatrix} \\ &\sim \mathrm{Can}(-\tfrac{\theta}{2\pi}, 0, t)\end{aligned} \tag{65}$$

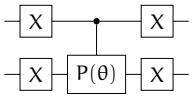
Controlled phase shift gate (**??**)

$$H_{\mathrm{CPhase}} = -\tfrac{\theta}{4}(I + Z_0 \otimes Z_1 - Z_0 - Z_1) \tag{66}$$

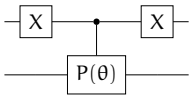


The QUIL quantum programming language [41, 0] defines several variants of the CPhase gate. Instead of the phase change occurring when both qubits are 1, instead the phase shift happens for qubits in the 00, 01, or 11 states. Each of these variants is closely related to the standard CPhase gate, and aren't explicitly used much in practice.

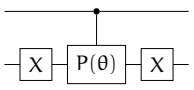
$$\mathrm{CPhase}_{00}(\theta) = \begin{bmatrix} e^{i\pi\theta} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

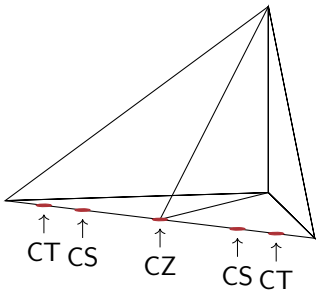


$$\mathrm{CPhase}_{01}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi\theta} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$



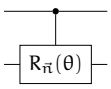
$$\mathrm{CPhase}_{10}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\pi\theta} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$





Controlled rotation gate [0] A controlled unitary represented as a rotation $R_{\vec{n}}(\theta)$ (13) around an arbitrary vector in the Bloch sphere.

$$CR_{\vec{n}}(\theta) = e^{-i\frac{1}{2}\theta(I-Z)\otimes(n_x X + n_y Y + n_z Z)} \tag{67}$$



Locally equivalent to $\mathsf{Can}(\frac{\theta}{2\pi}, 0, 0)$. The controlled-rotation is a convenient starting point for the decomposition of controlled unitaries. See (§7.5).

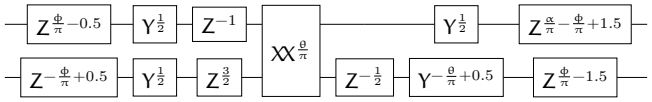
Barenco gate [12]: A 2-qubit gate of historical interest.

$$\mathsf{Barenco}(\phi, \alpha, \theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha}\cos(\theta) & -ie^{i(\alpha-\phi)}\sin(\theta) \\ 0 & 0 & -ie^{i(\alpha+\phi)}\sin(\theta) & e^{i\alpha}\cos(\theta) \end{pmatrix} \tag{68}$$

Barenco [12] showed that the 3-qubit Deutsch gate, which had previously been shown to be computationally universal, can be decomposed into 5 Barenco gates (See p. 55), demonstrating that 2-qubit gates can be computational universal for quantum logic. (In contrast, reversible classical logic requires 3-bit gates for computational universality (§**??**).)

The Barneco gate is locally equivalent to the XX gate, which can in turn be decomposed into two CNot gates.

$\mathsf{Barenco}(\phi, \alpha, \theta) \simeq$

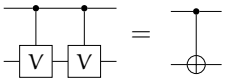


Controlled-V gate (square root of CNot gate):

$$\mathsf{CV} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1+i}{2} & \frac{1-i}{2} \\ 0 & 0 & \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix} \sim \mathsf{Can}(\tfrac{1}{4}, 0, 0) \tag{69}$$

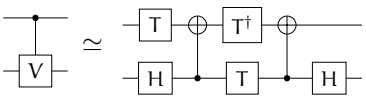
Commonly represented by the circuit diagram

The CV gate is a square-root of CNot, since the V-gate is the square root of the X-gate



Note that the inverse $\mathsf{CV}^\dagger$ is a distinct square-root of CNot. However CV and $\mathsf{CV}^\dagger$ are locally equivalent, which is a consequence of the symmetry about $t_x = \frac{1}{2}$ on the bottom face of the Weyl chamber.

The CV gate can be built from two-CNot gates.



6.6 XY gates

Gates in the XY class form two edges of the Weyl chamber with coordinates $\mathsf{Can}(t, t, 0)$ (for $t \leqslant \frac{1}{2}$) and $\mathsf{Can}(t, 1-t, 0)$ (for $t > \frac{1}{2}$). This includes the identity and iSwap gates.

XY-gate [0, 49] Also occasionally referred to as the piSwap (or parametric iSwap) gate [**?**].

$$\begin{aligned}\mathrm{XY}(t) &= e^{-i\frac{\pi}{2}t(X\otimes X+Y\otimes Y)} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\pi t) & -i\sin(\pi t) & 0 \\ 0 & -i\sin(\pi t) & \cos(\pi t) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \mathrm{Can}(t, t, 0) \\ &\sim \mathrm{Can}(t, 1-t, 0)\end{aligned} \tag{70}$$

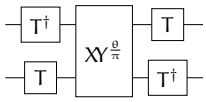
Here we have defined the XY gate here to match the parameterization of the canonical gate. An alternative parameterization is $\mathrm{XY}(\theta)$ where $\theta = -2\pi t$ [49**?**].

Givens gate [58]

$$\begin{aligned}\mathrm{Givens}(\theta) &= \exp(-i\theta(Y\otimes X - X\otimes Y)/2) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\sim \mathrm{Can}(\tfrac{\theta}{\pi}, \tfrac{\theta}{\pi}, 0)\end{aligned} \tag{71}$$

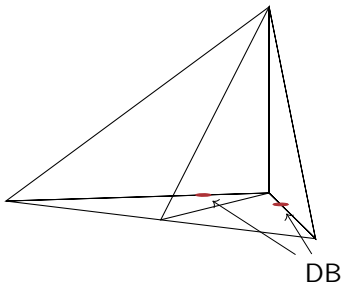
Occurs in quantum computational chemistry.

$\mathrm{Givens}(\theta) \simeq$



Dagwood Bumstead (DB) gate [51] Of all the gates in the XY class, the Dagwood Bumstead-gate makes the biggest sandwiches. [51, Fig. 4]

$$\begin{aligned}\mathrm{DB} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\frac{3\pi}{8}) & -i\sin(\frac{3\pi}{8}) & 0 \\ 0 & -\sin(\frac{3\pi}{8}) & \cos(\frac{3\pi}{8}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \mathrm{XY}(\tfrac{3}{8}) \\ &= \mathrm{Can}(\tfrac{3}{8}, \tfrac{3}{8}, 0)\end{aligned} \tag{72}$$

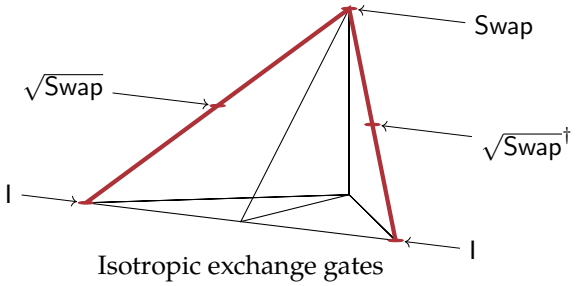


6.7 Isotropic exchange gates

Includes the identity and Swap gates.

Swap-alpha gate [? 28] Powers of the Swap gate Target

$$\mathrm{Swap}^\alpha = e^{+\frac{\pi}{2}\alpha}\begin{bmatrix} e^{-\mathrm{i}\frac{\pi}{2}\alpha} & 0 & 0 & 0 \\ 0 & \cos(\frac{\pi}{2}\alpha) & \mathrm{i}\sin(\frac{\pi}{2}\alpha) & 0 \\ 0 & \mathrm{i}\sin(\frac{\pi}{2}\alpha) & \cos(\frac{\pi}{2}\alpha) & 0 \\ 0 & 0 & 0 & e^{-\frac{\pi}{2}\alpha} \end{bmatrix} \simeq \mathrm{Can}(\tfrac{\alpha}{2}, \tfrac{\alpha}{2}, \tfrac{\alpha}{2}) \tag{73}$$



Isotropic exchange gates

$\sqrt{\mathrm{Swap}}$-gate [28]

$$\begin{aligned}\sqrt{\mathrm{Swap}} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1+\mathrm{i}) & \frac{1}{2}(1-\mathrm{i}) & 0 \\ 0 & \frac{1}{2}(1-\mathrm{i}) & \frac{1}{2}(1+\mathrm{i}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \mathrm{Can}(\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4})\end{aligned} \tag{74}$$

The square root of the Swap gate.

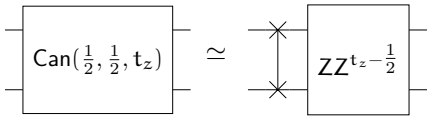
Inverse $\sqrt{\mathrm{Swap}}$-gate

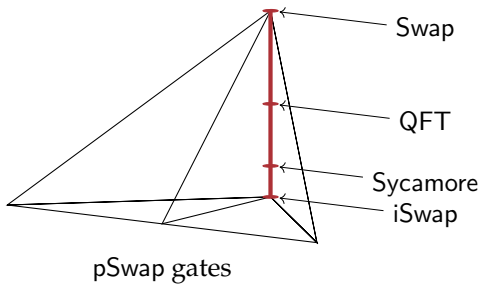
$$\begin{aligned}\sqrt{\mathrm{Swap}}^\dagger &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1-\mathrm{i}) & \frac{1}{2}(1+\mathrm{i}) & 0 \\ 0 & \frac{1}{2}(1+\mathrm{i}) & \frac{1}{2}(1-\mathrm{i}) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \mathrm{Can}(\tfrac{3}{4}, \tfrac{1}{4}, \tfrac{1}{4})\end{aligned} \tag{75}$$

Because of the symmetry around $t_x = \frac{1}{2}$ on the base of the Weyl chamber, the CNot and iSwap gates only have one square root. But the Swap has two locally distinct square roots, which are inverses of each other.

6.8 Parametric swap gates

The class of parametric Swap (PSwap) gates forms the back edge of the Weyl chamber, $\mathrm{Can}(\frac{1}{2}, \frac{1}{2}, t_z)$, connecting the Swap and iSwap gates. These gates can be decomposed into a Swap and ZZ gate, a combination that occurs naturally when considering Swap networks for routing QAOA style problems (§**??**).



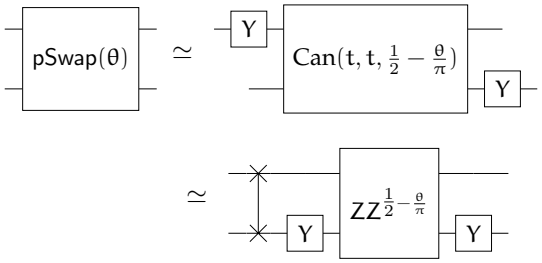


pSwap gates

The Sycamore gate is discussed under XXY gates (84).

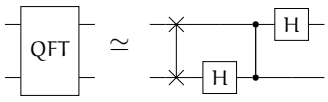
pSwap **gate** (parametric swap) [41] The parametric swap gate as originally defined in the QUIL quantum programming language.

$$\mathsf{pSwap}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \sim \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2} - \tfrac{\theta}{\pi}) \tag{76}$$



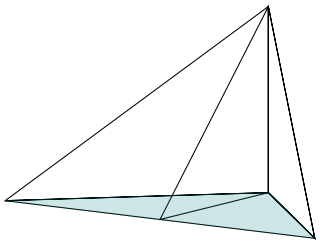
Quantum Fourier transform (QFT) [0] We will discuss the quantum Fourier transform (QFT) in detail latter (§**??**). The QFT can be applied to any number of qubits, and for 2-qubits, the QFT gate is in the PSwap class, half way between Swap and iSwap.

$$\mathsf{QFT}_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \sim \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{4}) \tag{77}$$



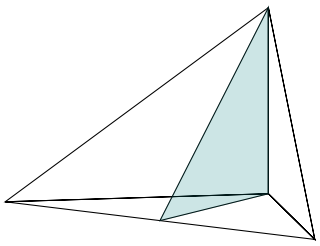
6.9 Orthogonal gates

An orthogonal gate, in this context, is a gate that can be represented by an orthogonal matrix (up to local 1-qubit rotations.) The special orthogonal gates have representations with determinant $+1$ and coordinates $\mathsf{Can}(t_x, t_y, 0)$, which covers the bottom surface of the canonical Weyl chamber.



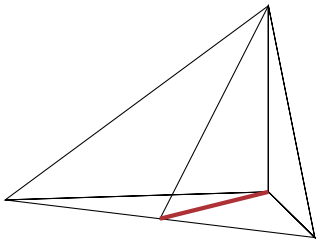
Special orthogonal gates

The improper orthogonal gates have representations with determinant -1 and coordinates $\mathsf{Can}(\frac{1}{2}, t_y, t_z)$, which is a plane connecting the CNot, iSwap, and Swap gates.



Improper orthogonal gates

The line of gates locally equivalent to $\mathsf{Can}(\frac{1}{2}, t_y, 0)$ are in both the special and improper orthogonal local equivalency classes. Theses are know as special perfect entangling (SPE) gates [27**? ?**], or super-controlled gates [**? ?**]).



Special perfect entangling gates

By way of illustration, these three orthogonal operators are all in the CNot local equivalency class, but have determinate of -1, -1, and 1, respectively.

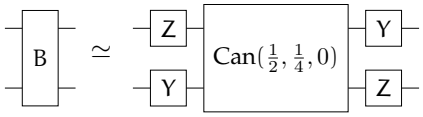
$$\begin{bmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{bmatrix} \qquad \begin{bmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&-1 \end{bmatrix} \qquad \begin{bmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&1\\ 0&0&-1&0 \end{bmatrix}$$

These are a CNot, a CZ, and a CNot followed by a CZ.

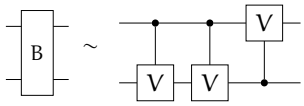
B (Berkeley) gate [59] Located in the middle of the bottom face of the Weyl chamber.

$$\begin{aligned} \mathsf{B} &= \begin{pmatrix} \cos(\frac{\pi}{8}) & 0 & 0 & i\sin(\frac{\pi}{8}) \\ 0 & \cos(\frac{3\pi}{8}) & i\sin(\frac{3\pi}{8}) & 0 \\ 0 & i\sin(\frac{3\pi}{8}) & \cos(\frac{3\pi}{8}) & 0 \\ i\sin(\frac{\pi}{8}) & 0 & 0 & \cos(\frac{\pi}{8}) \end{pmatrix} \\ &= \frac{\sqrt{2-\sqrt{2}}}{2} \begin{pmatrix} 1+\sqrt{2} & 0 & 0 & i \\ 0 & 1 & i(1+\sqrt{2}) & 0 \\ 0 & i(1+\sqrt{2}) & 1 & 0 \\ i & 0 & 0 & 1+\sqrt{2} \end{pmatrix} \\ &= \mathsf{Can}(-\tfrac{1}{2}, -\tfrac{1}{4}, 0) \end{aligned} \tag{78}$$

The B-gate, as originally defined, has canonical parameters outside our Weyl chamber due to differing conventions for parameterization of the canonical gate. But of course it can be moved into our Weyl chamber with local gates.



The B-gate is half way between the CNot and DCNot (~ iSwap) gates, and thus it can be constructed from 3 CV (square root of CNot) gates.



ECP-gate [51]

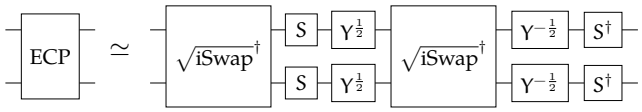
$$\mathsf{ECP} = \frac{1}{2}\begin{pmatrix} 2c & 0 & 0 & -i2s \\ 0 & (1+i)(c-s) & (1-i)(c+s) & 0 \\ 0 & (1-i)(c+s) & (1+i)(c-s) & 0 \\ -i2s & 0 & 0 & 2c \end{pmatrix} \tag{79}$$

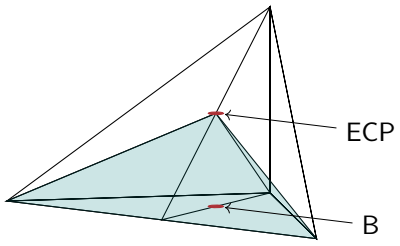
$$c = \cos(\tfrac{\pi}{8}) = \sqrt{\tfrac{2+\sqrt{2}}{2}}$$

$$s = \sin(\tfrac{\pi}{8}) = \sqrt{\tfrac{2-\sqrt{2}}{2}}$$

$$= \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{4})$$

The peak of the pyramid of gates in the Weyl chamber that can be created with a square-root of iSwap sandwich. Equivalent to $\mathsf{Can}(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$.



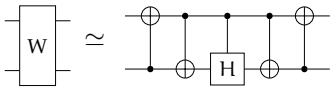


B and ECP gates, and the ECP pyramid

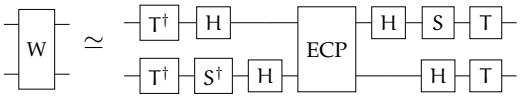
W-gate [0] A 2-qubit orthogonal and Hermitian gate (and therefore also symmetric) $\mathsf{W}^\dagger = \mathsf{W}$, that applies a Hadamard gate to a duel-rail encoded qubit.

$$\mathsf{W} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{80}$$

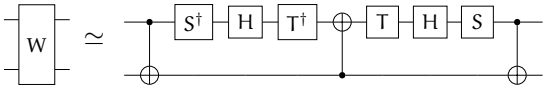
$$\sim \mathsf{ECP} = \mathsf{Can}(\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{4})$$



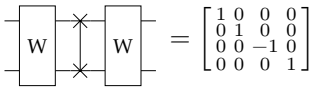
This W gate is locally equivalent to ECP,



and thus three CNot gates are necessary (and sufficient) to generate the gate.



The W gate has the useful property that it diagonalizes the swap gate [0].

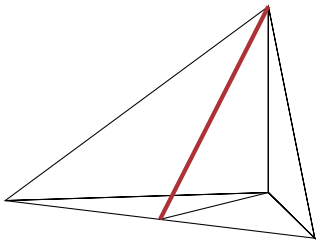


A-gate [47, 50][8] A 2-qubit 2-parameter gate in the improper-orthogonal local-equivalency class.

$$\mathsf{A}(\theta,\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & e^{+i\phi}\sin(\theta) & 0 \\ 0 & e^{-i\phi}\sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \sim \mathsf{Can}(\tfrac{1}{2}, \tfrac{\theta}{\pi}, \tfrac{\theta}{\pi}) \tag{81}$$

This gate is notable in that it conserves the number of 1s (versus 0s) in the computational basis [47, 50]. This has utility in VQE (§**??**) ansatzs as a particle-conserving mixer.

In the Weyl chamber, the A-gates span the line connecting the CNot and Swap gates [50].



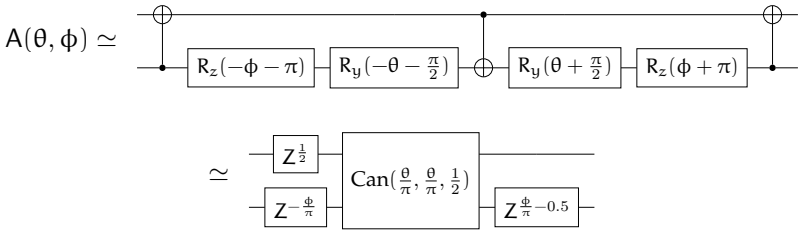
A gates

The W and Swap gates are special cases, and $\mathsf{A}(0,0)$ is locally equivalent to CNot.

$$\mathsf{A}(0,0) \sim \mathsf{CNot}$$
$$\mathsf{A}(\tfrac{\pi}{4},0) = \mathsf{W}$$
$$\mathsf{A}(\tfrac{\pi}{2},0) = \mathsf{Swap}$$

[8] Open problem: Find the analytic, Pauli basis decomposition of the A-gate Hamiltonian in terms of ϕ and θ.

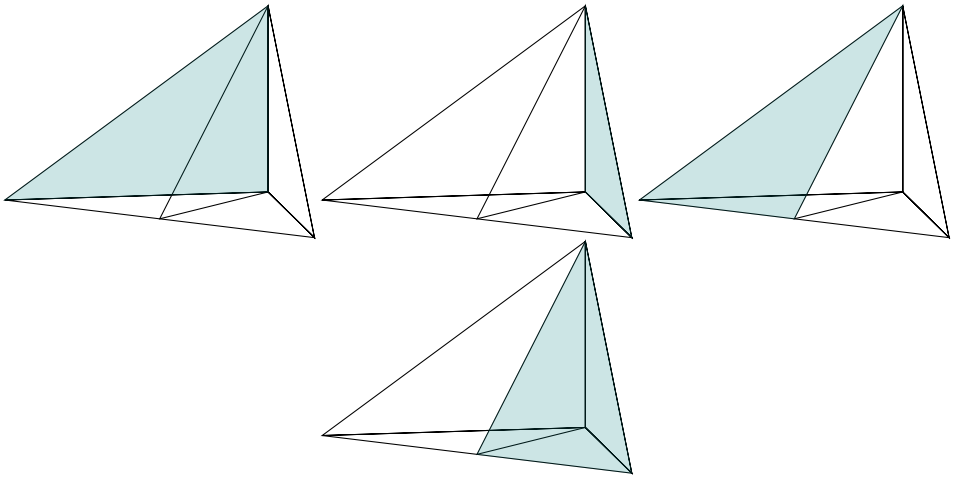
The A-gate requires a 3-CNot decomposition [50].



6.10 XXY gates

The remaining faces of the Weyl chamber are the XXY family. Thanks to the Weyl symmetries, this family covers all three faces that meet at the Swap gate.

$$\mathrm{XXY}(t, \delta) = \mathrm{Can}(t, t, \delta) \tag{82}$$



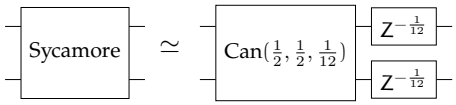
FSim **(Fermionic Simulator) gate** [0]

$$\mathrm{FSim}(\theta, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -i\sin(\theta) & 0 \\ 0 & -i\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & e^{-i\phi)} \end{bmatrix} \sim \mathrm{Can}(\tfrac{\theta}{\pi}, \tfrac{\theta}{\pi}, \tfrac{\phi}{2\pi}) \tag{83}$$

Sycamore (Syc) gate [**?** 54] The native 2-qubit gate on Google's Sycamore transmon quantum computer architecture. A carefully tuned instance of the fermionic simulator gate that for reasons that have to do with the details of the hardware can be performed particularly fast and with relatively low error [**?**].

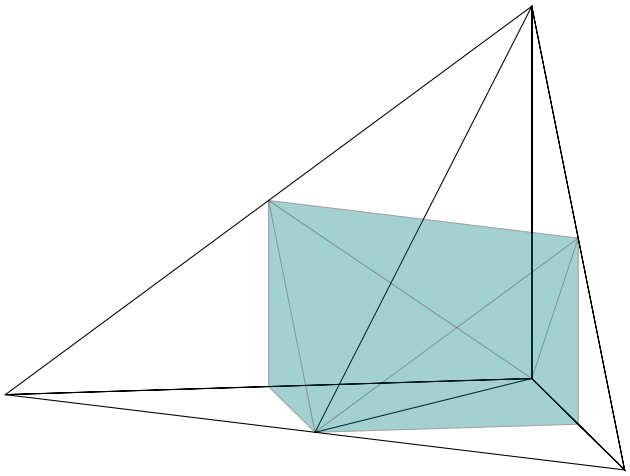
$$\mathrm{Syc} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & e^{-i\frac{\pi}{6}} \end{bmatrix} \simeq \mathrm{FSim}(\tfrac{\pi}{2}, \tfrac{\pi}{6}) \sim \mathrm{Can}(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{12}) \tag{84}$$

In the Weyl chamber the sycamore gate is located $\frac{1}{6}$ of the way up the back edge, between iSwap and the 2-qubit Quantum Fourier transform (See the Weyl chamber figure in (§6.8)).



Synthesizing other gates from sycamore gates is mathematically somewhat involved [**?** **?** 54]. Two sycamores are required to build CNot [**?**], B [0], or any gate in the Ising (CPHASE) class [54], and three for iSwap or Swap [54] or any gate not in the special-orthogonal locality class. One approach to general gate synthesis is to build B gates from pairs of Sycamores and then any 2-qubit gate from B gate sandwiches **??**, although this requires a total of 4 sycamore gates [**?**].

6.11 Perfect entanglers



Perfect entanglers

7 Decomposition of 2-qubit gates

A general 2-qubit gate corresponds to some 4 by 4 unitary matrix, with 16 free parameters. We can (as per usual) factor out an irrelevant phase (???), but that still leaves a rather unwieldy 15 parameters. Fortunately, the Kraus and Cirac demonstrated that any 2-qubit gate can be decomposed into a canonical gate, plus 4 local 1-qubit gates [0, 0, 0, 0]. The local gates account for $4 \times 3 = 12$ parameters, which leaves just the 3 parameters of the canonical gate. The canonical gate can be further decomposed into CNot gates, or other sets of 2-qubit gates as desired.

7.1 Kronecker decomposition

We'll first consider a simpler decomposition problem that we will use as a subalgorithm of the full decomposition. Suppose we have two 1-qubit gates, A and B, acting on separate qubits, but we are given only the full 2-qubit unitary operator C. Our task is to recover the two 1-qubit gates.

Mathematically, C is the Kronecker product of the two 1-qubit gates.

$$\begin{aligned} C &= A \otimes B \\ &= \begin{bmatrix} A_{11}B_{11} & A_{11}B_{12} & A_{12}B_{11} & A_{12}B_{12} \\ A_{11}B_{21} & A_{11}B_{22} & A_{12}B_{21} & A_{12}B_{22} \\ A_{21}B_{11} & A_{21}B_{12} & A_{22}B_{11} & A_{22}B_{12} \\ A_{21}B_{21} & A_{21}B_{22} & A_{22}B_{21} & A_{22}B_{22} \end{bmatrix} \end{aligned} \tag{85}$$

We will undo the Kronecker product using the Pitsianis-Van Loan algorithm[8, 19]. If the matrix C isn't constructed from a single Kronecker product, the algorithm still guarantees that $A \otimes B$ is the closest Kronecker product to C in the Frobenius norm.

The trick is to first view C as 4th order $2\times2\times2\times2$ tensor. The Kronecker product can then be written as the outer product of A and B, followed by a transpose of the last index of A and the first index of B.

$$C_{mpnq} = A_{mn} \otimes B_{pq} = [A_{mn}\ B_{pq}]^{T_{n\leftrightarrow p}} \tag{86}$$

If we flatten the matrices, we have a normal outer product of vectors, which can be undone with a singular-value decomposition.

In code, the algorithm can be expressed as follows.

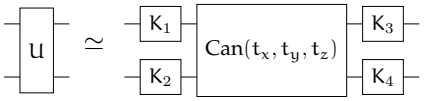
```
import numpy as np
def nearest_kronecker_product(C):
    C = C.reshape(2, 2, 2, 2)
    C = C.transpose(0, 2, 1, 3)
    C = C.reshape(4, 4)

    u, sv, vh = np.linalg.svd(C)

    A = np.sqrt(sv[0]) * u[:, 0].reshape(2, 2)
    B = np.sqrt(sv[0]) * vh[0, :].reshape(2, 2)

    return A, B
```

We first shape C to a 4th order tensor, so that in the next line we can undo the axes transposition, before reshaping to a matrix. The singular value decomposition takes this matrix apart, and we retain the rank-one approximation, retaining only

the largest singular value and corresponding left and right singular vectors. We reshape the singular vectors to matrices to obtain our desired result.

7.2 Canonical decomposition

Any 2-qubit gate can expressed as a canonical gate (**??**) plus 4 local 1-qubit gates [0].



Decomposition to the canonical gate is also known as the magic-, Kraus-Cirac- [0], or KAK-decomposition.

The trick to decomposing a 2-qubit gate unitary to the canonical representation is a similarity transform to the *magic basis*.

$$V = M\,U\,M^\dagger \tag{87}$$
$$M = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{bmatrix}$$

Here M is the magic gate (56). The magic basis has two remarkable properties. The first is that if U is a special orthogonal matrix (Real, $U^T = U$, and $\det U = 1$), then in the magic basis U is the Kronecker product of two 1-qubit gates.

$$V = MUM^\dagger = A \otimes B \quad \text{if } U \in SO(4) \tag{88}$$

For a sussinct proof see [].

The second useful property is that M diagonalizes the canonical gate.

$$\mathrm{Can}(t_x, t_y, t_z) = M\,D\,M^\dagger \tag{89}$$
$$D = \mathrm{diag}(e^{i\frac{1}{2}(+t_x-t_y+t_z)}, e^{i\frac{1}{2}(-t_x+t_y+t_z)}, e^{i\frac{1}{2}(+t_x+t_y-t_z)}, e^{i\frac{1}{2}(-t_x-t_y-t_z)})$$

With these two properties we can decompose any 2-qubit gate. We'll assume that the phase (**??**) has already been extracted and that U is therefore special unitary. We write U as a decomposition into the canonical gate, and Kronecker products of local gates before and after.

$$U = (K_3 \otimes K_4)\ \mathrm{Can}(t_x, t_y, t_z)\ (K_1 \otimes K_2) \tag{90}$$

We then make the transform to the magic basis, to give us a diagonal matrix D sandwiched between two special orthogonal matrices, Q_1 and Q_2.

$$\begin{aligned} V &= M\,U\,M^\dagger \\ &= M(K_3 \otimes K_4)M^\dagger\ M\ \mathrm{Can}(t_x, t_y, t_z)\ M^\dagger\ M(K_1 \otimes K_2)M^\dagger \\ &= Q_2\ D\ Q_1 \end{aligned}$$

The next trick is to take a transpose of V. This inverts the orthogonal matrices, but leaves the complex diagonal matrix unchanged. The product V^TV is therefore a similarity transform of the diagonal matrix D squared.

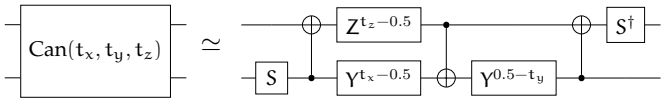
$$V^TV = Q_1^T D Q_2^T\ Q_2 D Q_1 = Q_1^T D^2 Q_1$$

An eigen-decomposition of V^TV yields the square eigenvalues of D, and Q_1 as the

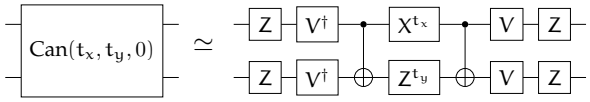
matrix of eigenvectors. We can then extract the canonical gate coordinates from the eigenvalues, and undo the magic basis transform to recover the local gates. These Kronecker products of local gates can be decomposed into separate 1-qubit gates using the Kronecker decomposition (§7.1) and then further into elementary gates using a 1-qubit decomposition (§4.1).

7.3 CNot decomposition

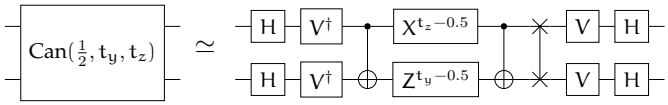
The elementary 2-qubit gate is most often taken to be the CNot gate. In general we can build any canonical gate from a circuit of 3 CNots [**?** **?** 25].



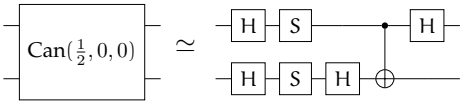
Gates on the bottom surface of the Weyl chamber (special orthogonal local equivalency class (§**??**)) require only 2 CNot gates [**?** 25**?**].



Gates in the improper orthogonal equivalency class (§**??**) require 3 CNot gates, or 2-CNots and 1 Swap [25].



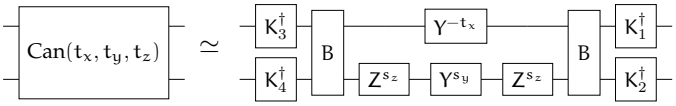
Clearly gates locally equivalent to CNot $\sim$ Can$(\frac{1}{2}, 0, 0)$ require only one CNot gate,



and those locally equivalent to the identity $I_2 = \text{Can}(0, 0, 0)$ require none.

7.4 B-gate decomposition

The canonical gate can be decomposed in to a B-gate sandwich [59].

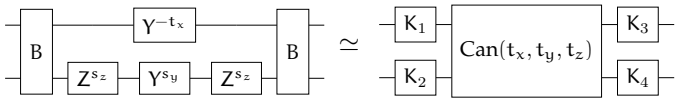


where

$$
\begin{aligned}
s_y &= +\tfrac{1}{\pi} \arccos\left(1 - 4\sin^2 \tfrac{1}{2}\pi t_y \cos^2 \tfrac{1}{2}\pi t_z\right) \\
s_z &= -\tfrac{1}{\pi} \arcsin \sqrt{\frac{\cos \pi t_y \cos \pi t_z}{1 - 2\sin^2 \tfrac{1}{2}\pi t_y \cos^2 \tfrac{1}{2}\pi t_z}}
\end{aligned}
\tag{91}
$$

Notably two B-gates are sufficient to create any other 2-qubit gate (whereas we need 3 CNOTs in general).

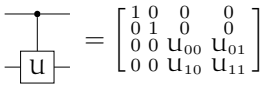
To recover the local gates K_n we perform another canonical decomposition on the B gate sandwich sans the terminal local gates [**?**][9].



The B-gate is not a native gate on any extant quantum computer, and thus the B-gate decomposition isn't used for gate synthesis directly. But the B-gate sandwich has been used as a compilation strategy for Google's Sycamore architecture [**?**]. The native sycamore gate (84) is locally equivalent to $\mathrm{Can}(\frac{1}{2}, \frac{1}{2}, \frac{1}{12})$. A sycamore-gate sandwich can generate a subset of gates in the special-orthogonal local equivalency class, including CNOT, the entire Ising class, and B (but notable not iSwap) [**?** 54]. A B-gate sandwich can then be used to synthesis any other gate using 4-sycamores.

7.5 ABC decomposition

A 2-qubit controlled-unitary gate has an arbitrary 1-qubit unitary U that acts on the target qubit if the control qubit is in the one state.

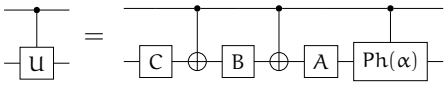


Controlled-unitaries are all in the Ising gate class (as we'll show), and can be implemented with at most 2 CNot gates.

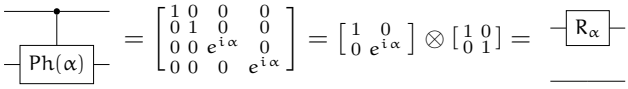
The trick is to express the 1-qubit unitary U as an ABC decomposition [14],

$$U = e^{i\alpha}\, A\, X\, B\, X\, C \tag{92}$$

where the gates A, B, and C are chosen such that $ABC = I$. We can then express the controlled unitary as



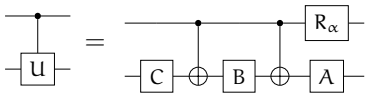
Note that this one situation that the phase of the gate actually matters. A controlled Z gate is not the same as a controlled-$R_z(\pi)$ because the 1-qubit unitary had different phases. Happily, a "controlled-global-phase" reduces to a 1 qubit phase shift gate (19).



The result is a decomposition of a 2-qubit controlled unitary into 5 1-qubit uni-

[9] Open problem: Zang et al.[59] derived the analytic decomposition of the canonical gate to a B-gate sandwich only up to local gates. Derive an analytic formula for the necessary local gates to complete the canonical to B-gate sandwich decomposition.

taries and 2 CNOT gates.



If the control bit is set we apply the desired gate, and if not nothing happens since $ABC = I$.

We can construct an ABC decomposition by a rearrangement of a Z-Y-Z decomposition (§4.1).

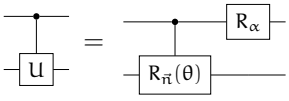
$$\begin{aligned} U &= e^{i\alpha}\, R_z(\theta_2)\, R_y(\theta_1)\, R_z(\theta_0) \\ &= e^{i\alpha} R_z(\theta_2)\, R_y(\tfrac{1}{2}\theta_1) \quad R_y(+\tfrac{1}{2}\theta_1) \quad R_z(+\tfrac{1}{2}\theta_0 + \tfrac{1}{2}\theta_2) \quad R_z(\tfrac{1}{2}\theta_0 - \tfrac{1}{2}\theta_2) \\ &= e^{i\alpha} R_z(\theta_2)\, R_y(\tfrac{1}{2}\theta_1)\, X\, R_y(-\tfrac{1}{2}\theta_1)\, X\, X\, R_z(-\tfrac{1}{2}\theta_0 - \tfrac{1}{2}\theta_2)\, X\, R_z(\tfrac{1}{2}\theta_0 - \tfrac{1}{2}\theta_2) \\ &= e^{i\alpha}\, A\, X\, B\, X\, C \end{aligned} \tag{93}$$

where

$$\begin{aligned} A &= R_z(\theta_2)\, R_y(\tfrac{1}{2}\theta_1)\,, \\ B &= R_y(-\tfrac{1}{2}\theta_1)\, R_z(-\tfrac{1}{2}\theta_0 - \tfrac{1}{2}\theta_2)\,, \\ C &= R_z(\tfrac{1}{2}\theta_0 - \tfrac{1}{2}\theta_2)\,. \end{aligned}$$

Note that $X\, R_z(\theta)\, X = R_z(-\theta)$ and $X\, R_y(\theta)\, X = R_y(-\theta)$. We can understand these relations by looking at the Bloch sphere. The X gate is a half turn rotation about the $\hat{x}$ axis, so the $\hat{z}$ and $\hat{y}$ axes are inverted, and the respective rotation gates induce an anti-clockwise rather than clockwise rotations relative to the original axes.

Another approach is to deke U into a general 1-qubit rotation gate.



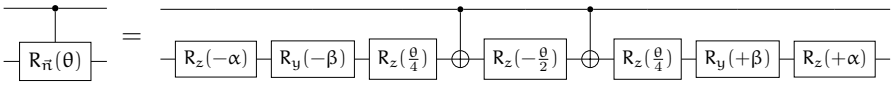
The rotation gate $R_{\vec{n}}(\theta)$ can be analytically decomposed into a 5 gate sequence (**??**), which can be rearranged into an ABC decomposition.

$$\begin{aligned} R_{\vec{n}}(\theta) &= R_z(+\alpha) R_y(+\beta) R_z(\theta) R_y(-\beta) R_z(-\alpha) \\ &= AXBXC \end{aligned} \tag{94}$$

where

$$\begin{aligned} A &= R_z(+\alpha) R_y(+\beta) R_z(\tfrac{\theta}{4})\,, \\ B &= R_z(-\tfrac{\theta}{2})\,, \\ C &= R_z(\tfrac{\theta}{4}) R_y(-\beta)\,. \end{aligned}$$

Thus a controlled-rotation gate can be expressed as



Note that the parameters α and β do not depend on rotation angle θ. If we compare to the CNot decompositions of the canonical gate (§**??**), we can see that a controlled-

rotation gate $\mathsf{R}_{\vec{n}}(\theta)$ is locally equivalent to $\mathsf{Can}(\frac{\theta}{2\pi}, 0, 0)$. Not only does this demonstrate that controlled-unitaries are in the Ising gate class, but we also see that the position of the gate along the front edge of the Weyl chamber is directly proportional to the controlled-unitary's angle of rotation in the Bloch sphere.

8 Standard 3-qubit gates

Regrettable there doesn't appear to be an easy way to characterize and visualizes the space of 3-qubit gates in the same way there is for 1-qubit (Bloch ball) and 2-qubit gates (Weyl chamber). Which is perhaps not surprising since a general 3-qubit gate has $(2^3)^2 = 64$ parameters.

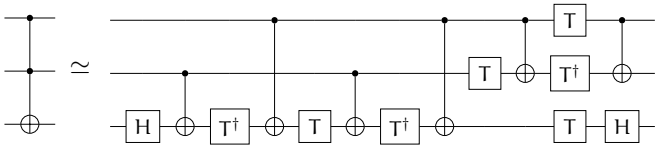
Fortunately there are only a few specific 3-qubit gates that show up in practice, most of which are directly related to the Toffoli (or controlled-controlled-not) gate.

Toffoli gate (controlled-controlled-not, CCNot) [3, 5, 14] A 3-qubit gate with two control and one target qubits. Originally studied in the context of reversible classical logic, where 3-bit gates are necessary for universal computation [3**?**]. The target bit flips only if both control bits are one. We often encounter this gate when converting classical logic circuits to quantum circuits.

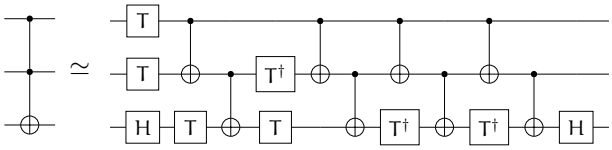
$$\text{CCNot} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{bmatrix} =$$

$$H_{\text{CCNot}} = -\tfrac{\pi}{8}(I_0 - Z_0)(I_1 - Z_1)(I_2 - X_2) \tag{95}$$
$$= \tfrac{\pi}{2}\begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&-1&1\\ 0&0&0&0&0&0&1&-1 \end{bmatrix}$$

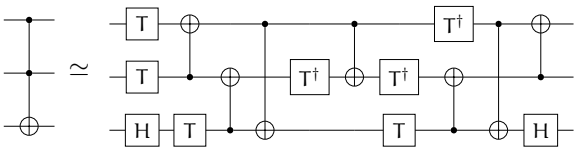
The CCNot can be decomposed into a circuit of at least 6 CNots [18].



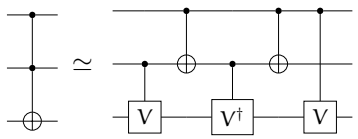
The above circuit assumes that we can apply CNot gates between any of the 3 qubits. If we are instead restricted to CNots between adjacent qubits, then we can decompose into 8 CNot gates, which is fewer than if we added explicit Swap operations.



This depth 9 decomposition requires 7 CNots [34]. Since more gates can be applied at the same time, the gate depth is less despite more 2-qubit gates.



Another decomposition requires 3 CV and 2 CNot gates [0].



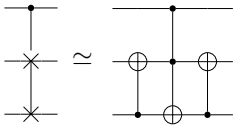
Fredkin gate (controlled-swap, CSwap) [4, 0]

$$\text{CSwap} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} =$$

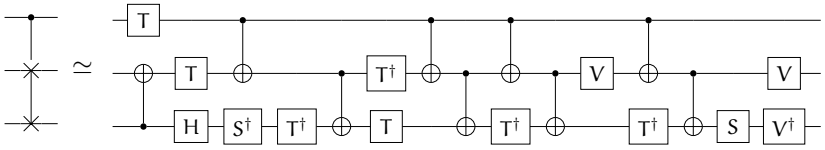
A controlled swap gate. Another logic gate for reversible classical computing.

$$\begin{aligned} H_{\text{CSwap}} &= -\tfrac{\pi}{8}(I_0 - Z_0)(X_1X_2 + Y_1Y_2 + Z_1Z_2 - I_1I_2) \\ &= \tfrac{\pi}{2}\begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&-1&1&0\\ 0&0&0&0&0&1&-1&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix} \end{aligned} \tag{96}$$

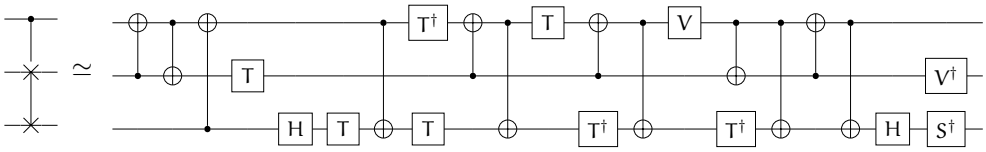
A CSwap can be built from 2 CNot gates and 1 CCNot (or 8 CNots in total).



An adjacency respecting decomposition of the CSwap can be formed with 10 CNots if the target is the first qubit [0],

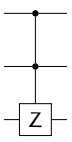


or 12 CNots if the target is between the two swapped qubits [0].



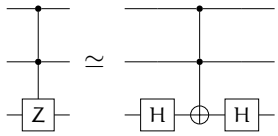
CCZ gate (controlled-controlled-Z) [0, 0]

$$\text{CCZ} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&-1 \end{bmatrix} =$$



$$H_{CCZ} = -\frac{\pi}{8}(I_0 - Z_0)(I_1 - Z_1)(I_2 - Z_2) \tag{97}$$
$$= \frac{\pi}{2}\begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&-1 \end{bmatrix}$$

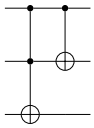
The CCNot gate can be converted to the CCZ gate by conjugating the target qubit with Hadamard gates (in the same way that we can convert a CNot to CZ)



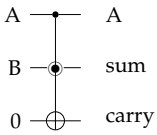
Peres gate [6, 24] Another gate that is universal for classical reversible computing.

$$\text{Peres} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&0&0&0 \end{bmatrix} \tag{98}$$

The Peres gate is equivalent to a Toffoli followed by a CNot gate, and decomposes into 5 CNots (compared to 6 for Toffoli gates).



The Peres gate is a reversible half-adder. (Recall that a half-adder sums two bits, whereas a full-adder sums three bits) If we feed a zero bit into the third position, then the output of the second bit is the sum (mod 2) of the first two bits, and the third bit is the carry.



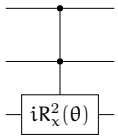
The above diagram is seen occasionally, where the middle bit of the Peres gate is denoted by a fisheye.

Deutsch gate [7, 12, 46] Mostly of historical interest, since this was the first quantum gate to be shown to be computationally universal [7].

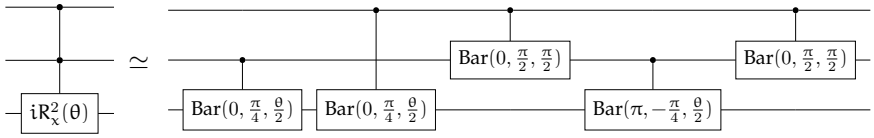
$$\text{Deutsch}(\theta) = \begin{bmatrix} 1&0&0&0&0&0 & 0 & 0\\ 0&1&0&0&0&0 & 0 & 0\\ 0&0&1&0&0&0 & 0 & 0\\ 0&0&0&1&0&0 & 0 & 0\\ 0&0&0&0&1&0 & 0 & 0\\ 0&0&0&0&0&1 & 0 & 0\\ 0&0&0&0&0&0 & i\cos(\theta) & \sin(\theta)\\ 0&0&0&0&0&0 & \sin(\theta) & i\cos(\theta) \end{bmatrix} \tag{99}$$

Examining the controlled unitary sub-matrix, the Deutsch gate can be thought of as a controlled-controlled-i$R_x{}^2(\theta)$ gate.

$$\text{Deutsch}(\theta) =$$



Barenco [12] demonstrated a construction of the Deutsch gate from 2-qubit "Barenco" gates, demonstrating that a single type of 2-qubit gate is sufficient for universality.

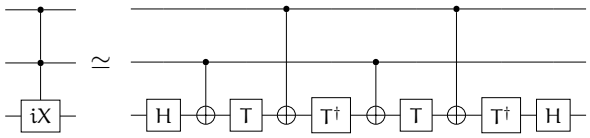


CCiX gate [60**?** , 61] A doubly controlled iX gate.

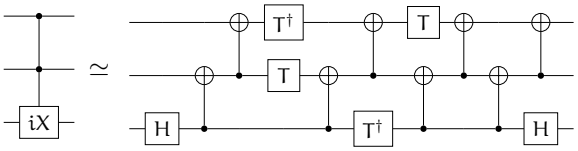
$$\text{CCiX} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&i\\ 0&0&0&0&0&0&i&0 \end{bmatrix} \tag{100}$$

$$H_{\text{CCiX}} = -\frac{\pi}{8} X_2(1 - Z_1)(1 - Z_0)$$

Can be decomposed into 4 CNot gates,



or 8 CNots respecting adjacency [60, 35].

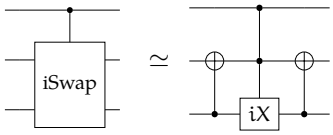


CiSwap gate A controlled iSwap gate.

$$\text{CiSwap} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&i&0\\ 0&0&0&0&0&i&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \tag{101}$$

$$H_{\text{CiSwap}} = \frac{\pi}{8}(Z_0X_1X_2 + Z_0Y_1Y_2 - X_1X_2 - Y_1Y_2)$$

Can be decomposed into a 2 CNots and a doubly controlled-iX gate (100) [**?**].



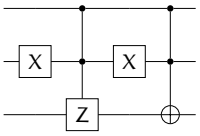
Rasmussen and Zinner (2020) [52] discuss possible implementations using superconducting circuits.

Margolus gate [11, 14, 16, 22, 61, 42] A"simplified" Toffoli gate, that differs from the Toffoli only by a relative phase, in that the $|101\rangle$ state picks up a -1 phase. In certain circuits Toffoli gates can be replaced with such relative phase Toffoli gates, leading to lower overall gate counts [61].

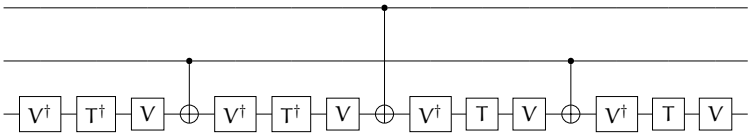
$$\text{Margolus} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{bmatrix} \tag{102}$$

$$\mathrm{H}_{\text{Margolus}} = \tfrac{\pi}{8}(1-\mathrm{Z}_0)(-2-\mathrm{Z}_1\mathrm{X}_2+\mathrm{Z}_1\mathrm{Z}_2+\mathrm{X}_2+\mathrm{Z}_2)$$

The Margolus gate is equivalent to a Toffoli gate plus a CCZ gate,



and can be implemented with only 3 CNot gates.

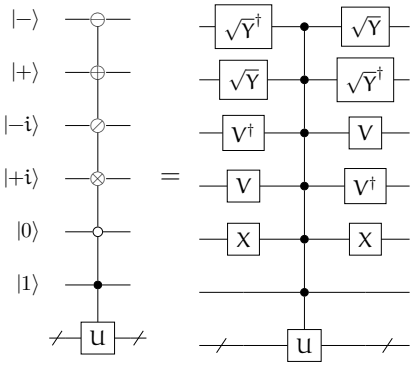


Note that this decomposition is often expressed in terms of $\mathrm{R}_y(\frac{\pi}{4})$, which is the same as $VTV^\dagger$ up to phase, e.g. Nielsen and Chuang [18, Ex 4.26]. This is a T-like gate: a counter-clockwise eighth turn of the Bloch sphere about the $\hat{y}$-axis.

9 Controlled unitary gates

9.1 Anti-control gates

9.2 Alternative axis control



9.3 Conditional gates

9.4 Multiplexed gates

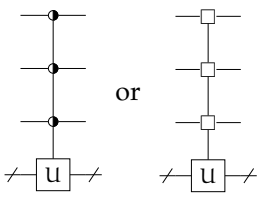
A multiplexed [0, 0, 0] gate (also called is a uniformly controlled gate [0, 0]) is a generalization of the conditional unitary gate with an arbitrary number of control qubits. For each different bitwise configuration of the control qubits, a different unitary operator is applied to the target qubits.

If we place the control qubits first, then the matrix of a multiplexed gate has a block diagonal structure, here illustrated for 3 control qubits.

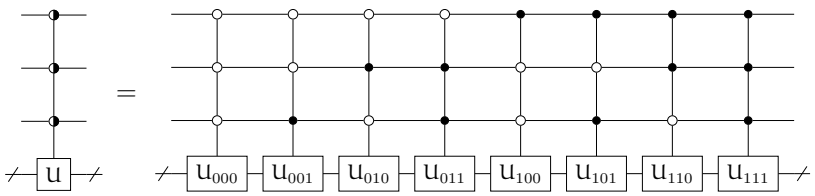
$$\mathsf{Mux}(\{U\}) = \begin{bmatrix} U_{000} & & & & & & & \\ & U_{001} & & & & & & \\ & & U_{010} & & & & & \\ & & & U_{011} & & & & \\ & & & & U_{100} & & & \\ & & & & & U_{101} & & \\ & & & & & & U_{110} & \\ & & & & & & & U_{111} \end{bmatrix} \tag{103}$$

Each block U is a unitary operator acting on the same number of qubits. Here each block is labelled with the corresponding control state (where we are using our standard tensor conventions (§**??**)), e.g. The operator U_{011} acts on the target qubits when the controls are in the $|011\rangle$ state.

In circuit diagrams the control qubits are typically represents as half-filled circles, or as boxes.



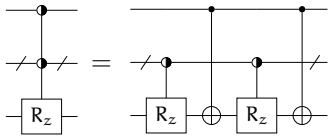
With C controls, a multiplex gate corresponds to 2^C controlled-unitary gates, one for each possible control-bit configuration.



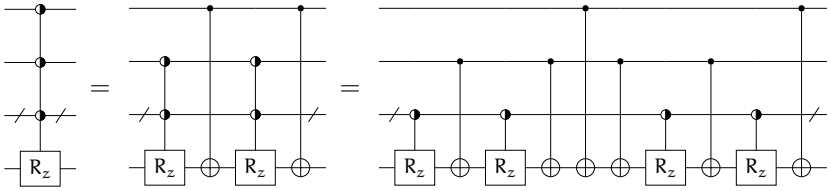
This decomposition provides an intuitive picture of the function of a multiplexed gate, but we'll describe a more efficient decomposition below.

9.5 Demultiplexing a multiplexed-R_z gate

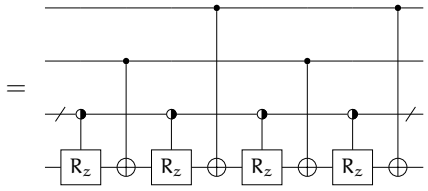
We can systematically demultiplex a multiplexed-R_z gate, splitting an N qubit gate into two $N-1$ qubit multiplexed-R_z gates, and two CNots.



[TODO: WHY THIS WORKS] [TODO: Explicit parameters]



The trick here is that we have reversed the order of the last decomposition, so that we can place two CNot gates in juxtaposition, separated only by another CNot with a different control qubit. Since CNot gates with different controls commute (§**??**) we can cancel these juxtaposed CNots.



This trick extend all the way to the final decomposition of 2-qubit conditional-R_z gates, which would normally require 2 CNots apiece, but instead require only 2 per pair. The net upshot is that the decomposition of an N qubit multiplexed-R_z gate required 2^{N-1} CNot gates and 2^{N-1} R_z gates.

10 Decomposition of multi-qubit gates

Let us now consider the problem of decomposing an arbitrary multi-qubit gate, without necessarily any special structure, into a circuit of 1 and 2 qubit gates. Such a generic N-qubit quantum gate is represented by a 2^N by 2^N unitary matrix, with $2^{2N} = 4^N$ free parameters (one of which is the phase). The (classical) computational resources needed to specify such a gate rapidly becomes prohibitive with qubit count (e.g. a 10 qubit gate has over a million parameters), but decomposition of gates with a dozen qubits or so is quite feasible.

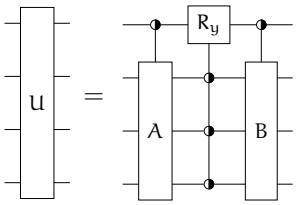
10.1 Quantum Shannon decomposition

The key to quantum Shannon decomposition is the sine-cosine decomposition, a standard relation from linear algebra [0]. Any unitary matrix can be written in a 2x2 block structure, and then decomposed into the following form,

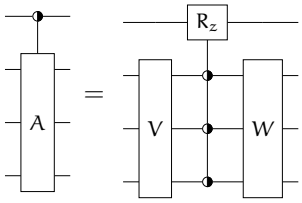
$$\left(\begin{array}{c|c} U_{00} & U_{01} \\ \hline U_{10} & U_{11} \end{array}\right) = \left(\begin{array}{c|c} B_0 & 0 \\ \hline 0 & B_1 \end{array}\right) \left(\begin{array}{c|c} C & -S \\ \hline +S & C \end{array}\right) \left(\begin{array}{c|c} A_0 & 0 \\ \hline 0 & A_1 \end{array}\right) \tag{104}$$

where the A's and B's are unitary, and C and S are diagonal matrices with $C^2 + S^2 = I$. It follows that we can write $C = \mathrm{diag}(\cos(\theta_0), \cos(\theta_1), \ldots)$, and $S = \mathrm{diag}(\sin(\theta_0), \sin(\theta_1), \ldots)$, wherefore the name of this decomposition[10].

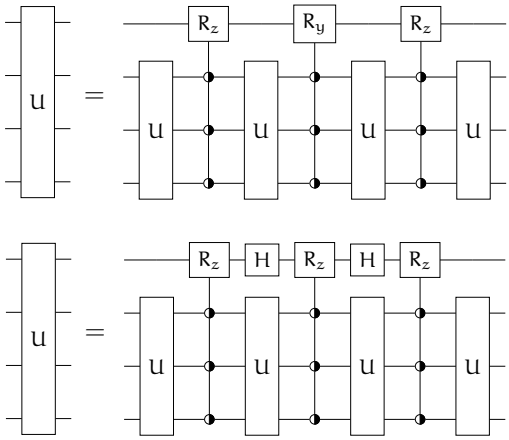
The initial and final blocks are equivalent to conditional gates, with a single control qubit, and potentially many target qubits. The central block is a multiplexed-Ry gate with the first qubit as target. We can spot this pattern because the overall $\left(\begin{smallmatrix} C & -S \\ +S & C \end{smallmatrix}\right)$ block structure has the same relationship between parameters as the R_Y gate. (Although the θ parameters of the R_Y gates and those of the cosine-sine decomposition differ by a factor of 2, for the usual reasons [0])



$$\left(\begin{array}{c|c} A_0 & 0 \\ \hline 0 & A_1 \end{array}\right) = \left(\begin{array}{c|c} W & 0 \\ \hline 0 & W \end{array}\right) \left(\begin{array}{c|c} D & 0 \\ \hline 0 & D^\dagger \end{array}\right) \left(\begin{array}{c|c} V & 0 \\ \hline 0 & V \end{array}\right) \tag{105}$$

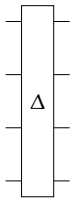


[10] This is a special case of the fully general cosine-sine decomposition, but sufficient for our purposes.

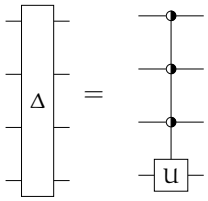


10.2 Decomposition of diagonal gates

A diagonal gate is any gate whose matrix representation is diagonal in the computation basis. Examples we have already encountered include the identity, Z, CZ, and CCZ gates. We'll notate a generic diagonal gate with a delta 'Δ'.



A diagonal gate can be thought of as a multiplexed gate. In particular, if we take the last qubit as the target, then a diagonal gate on N qubits is a multiplex gate with N-1 control qubits, and 2^{N-1} conditional unitaries, each of which is an arbitrary diagonal 1-qubit gate.

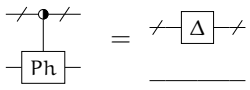


We can deke a diagonal 1-qubit gate into a R_z gate and a global phase. (this is one of those situations where we can't ignore the phase.)

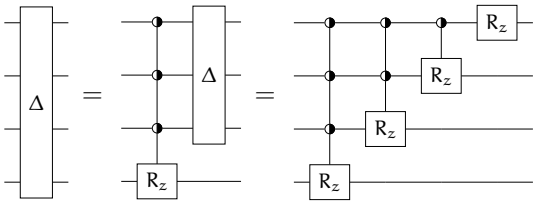
$$U = \begin{bmatrix} u_{00} & 0 \\ 0 & u_{11} \end{bmatrix} = \begin{bmatrix} e^{-ih_{00}} & 0 \\ 0 & e^{-ih_{11}} \end{bmatrix} = R_z(\theta)\mathrm{Ph}(\alpha) \tag{106}$$

$$h = i \ln u$$
$$\theta = \tfrac{1}{2}(h_{11} + h_{00})$$
$$\alpha = -(h_{11} - h_{00})$$

A diagonal gate is therefore equivalent to a multiplexed-R_z gate, and a "multiplexed-phase". Each sub-block of the "multiplexed-phase" has the same two values, so the "multiplexed-phase" breaks apart into a diagonal gate on the N-1 control qubits, and an identity on the target qubit. (This is the same effect as when a 2-qubit "controlled-global-phase" gate reduces to a 1-qubit phase shift gate. (§**??**)) [0]

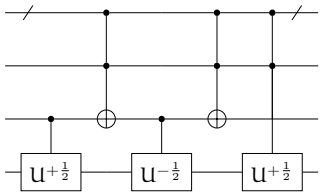


The net upshot is that a diagonal gate reduces to a multiplexed-R_z gate, and another diagonal gate on one fewer qubits. We can then recurse the diagonal gate decomposition, and deke a diagonal gate into a series of multiplexed-R_z gates[62].



Since each N qubit multiplexed gate requires 2^{N-1} CNot gates, a general N qubit diagonal gate requires $2^{N-1} + 2^{N-2} + \cdots + 2 = 2^N - 2$ CNot gates.

10.3 Decomposition of controlled-unitary gates



10.4 Two-level decomposition

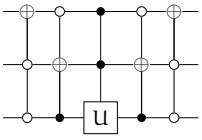
A 2-level unitary is a unitary operation that acts non-trivially on only 2-states. Any controlled 1-qubit unitary gate is 2-level, e.g. for a single qubit gate $U = \left[\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right]$ and 2 control qubits

$$CCU = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a & c \\ 0 & 0 & 0 & 0 & 0 & 0 & b & d \end{bmatrix} \tag{107}$$

But the active states need not be the last two. Any permutation of a two-level unitary gate is also a two-level unitary, such as

$$\begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix} . \tag{108}$$

Similarly any multi-controlled 2x2 unitary, or permutation of the same, is a 2-level unitary.



A d-dimensional unitary operator can be decomposed into a product of, at most, $\frac{1}{2}d(d-1)$ 2-level unitaries [10, 0, 0].

We'll use a 2-qubit gate A as illustration, with dimension $d = 2^2 = 4$.

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{109}$$

The trick is that we can set any off-diagonal entry to zero by multiplying by a carefully constructed 2-level unitary. Lets start with the $(1,0)$ entry.

$$B = U_{10}A = \begin{bmatrix} 1 & b_{01} & b_{02} & b_{03} \\ 0 & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix} \quad U_{10} = \begin{bmatrix} \frac{a_{00}^*}{w} & \frac{a_{10}^*}{w} & 0 & 0 \\ \frac{a_{10}}{w} & \frac{-a_{00}}{w} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad w = \sqrt{|a_{00}| + |a_{10}|}$$

Following through the matrix multiplication, we see that $b_{10} = (a_{10}a_{00} - a_{00}a_{10})/w = 0$, and $b_{00} = (a_{00}^* a_{00} - a_{10}^* a_{10})/w = w/w = 1$

We can now set $(2,0)$ to zero using the same procedure,

$$C = U_{20}B = \begin{bmatrix} 1 & c_{01} & c_{02} & c_{03} \\ 0 & c_{11} & c_{12} & c_{13} \\ 0 & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \quad U_{20} = \begin{bmatrix} \frac{b_{00}^*}{w} & 0 & \frac{b_{20}^*}{w} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{b_{20}}{w} & 0 & \frac{-b_{00}}{w} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad w = \sqrt{|b_{00}| + |b_{20}|}$$

and then set $(3,0)$ to zero.

$$D = U_{30}C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & d_{11} & d_{12} & d_{13} \\ 0 & d_{21} & d_{22} & d_{23} \\ 0 & d_{31} & d_{32} & d_{33} \end{bmatrix} \quad U_{30} = \begin{bmatrix} \frac{c_{00}^*}{w} & 0 & 0 & \frac{c_{20}^*}{w} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{c_{20}}{w} & 0 & 0 & \frac{-c_{00}}{w} \end{bmatrix} \quad w = \sqrt{|c_{00}| + |c_{20}|}$$

Once we have set all the off diagonal elements of the left column to zero, then the off-diagonal elements of the top row must also be zero.

Once we repeat this procedure $\frac{1}{2}d(d-1)$ times, setting all the lower off-diagonal entries to zero, we are left with the identity matrix.

$$I = U_{32}U_{31}U_{21}U_{30}U_{20}U_{10}A \tag{110}$$

Inverting this circuit, we obtain the original unitary as a product of 2-level unitaries.

$$A = U_{10}^\dagger U_{20}^\dagger U_{30}^\dagger U_{21}^\dagger U_{31}^\dagger U_{32}^\dagger \tag{111}$$

11 Pauli Group and Pauli Algebra

Recall the 4 1-qubit Pauli operators: $I = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $X = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $Y = \left[\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right]$, $Z = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$.

$$\begin{aligned} X^2 = Y^2 = Z^2 = I \\ XY = -YX = iZ \\ ZX = -XZ = iY \\ YZ = -ZY = iX \end{aligned} \tag{112}$$

Every pair of Pauli matrices either commutes or anti-commutes.

The *Pauli group* of 1 qubit operators consists of the 4 Pauli operators multiplied by factors of ± 1 or $\pm i$. This extra phase ensures that these 16 elements form a group under matrix multiplication. The Pauli group P_n of n qubit operators contains 4^{n+1} elements is formed from the 4 phase factors and tensor products of 1-qubit Pauli matrices,

$$P_n = \{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n} \tag{113}$$

12 Clifford Gates

The Clifford gates are an important subgroup of quantum gates. Familiar examples include the Pauli gates (I, X, Y, Z), phase (S), Hadamard (H), controlled-Z (CZ), controlled-not (CNOT), and swap. Common non-Clifford gates include T, B, and Toffolli (CCNOT).

The *Clifford group* C_n of gates acting on n qubits consists of those gates that *normalize* the corresponding Pauli group P_n. In the context of groups, *normalize* means that if p is an element of the Pauli group, and V is a Clifford gate, then $p' = VpV^\dagger$ is also an element of the Pauli group.

$$\mathbf{C}_n = \{V \in U_{2^n} \mid V\mathbf{P}_n V^\dagger = \mathbf{P}_n\} \tag{114}$$

The Clifford gates are defined this way because of important applications in quantum error correcting, which we will come to presently. An alternative approach is to define the Clifford group as all gates that can be constructed from S, H, and CNOT. This is the same group, up to phase.

12.1 Single qubit Clifford gates

Consider the X gate as a Clifford acting on the X, Y, and Z single-qubit Pauli elements.

$$\begin{aligned}
XXX^\dagger &= \left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right] = \left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right] = +X \\
XYX^\dagger &= \left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]\left[\begin{smallmatrix}0&-i\\i&0\end{smallmatrix}\right]\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right] = \left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right] = -Z \\
XZX^\dagger &= \left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]\left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right]\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right] = \left[\begin{smallmatrix}0&-i\\i&0\end{smallmatrix}\right] = -Y
\end{aligned}$$

For every Pauli element we recover another Pauli element when conjugated with the X gate, and therefore we can confirm that X gate is a Clifford gate.

Similarly we can consider the action of the Hadamard gate on the Pauli basses.

$$\begin{aligned}
HXH^\dagger &= \tfrac{1}{2}\left[\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right]\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]\left[\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right] = \left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right] = +Z \\
HYH^\dagger &= \tfrac{1}{2}\left[\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right]\left[\begin{smallmatrix}0&-i\\i&0\end{smallmatrix}\right]\left[\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right] = \left[\begin{smallmatrix}0&i\\-i&0\end{smallmatrix}\right] = -Y \\
HZH^\dagger &= \tfrac{1}{2}\left[\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right]\left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right]\left[\begin{smallmatrix}1&-1\\1&1\end{smallmatrix}\right] = \left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right] = +X
\end{aligned}$$

Note that we only ever pick up a ± 1 phase, and never an imaginary phase. This is because any element of the Pauli group with ± 1 phase is Hermitian, and the transformed gate $UpU^\dagger$ must also be Hermitian.

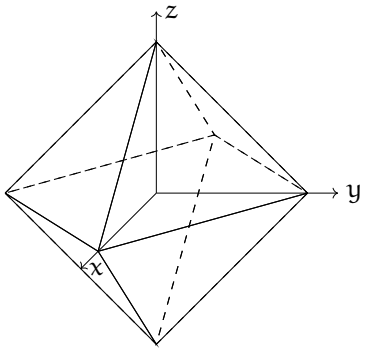
On the other hand, if we look at these transformations for a non-Clifford gate such as the T gate, we do not recover Pauli elements.

$$\begin{aligned}
TXT^\dagger &= \left[\begin{smallmatrix}1&0\\0&e^{+i\frac{\pi}{4}}\end{smallmatrix}\right]\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]\left[\begin{smallmatrix}1&0\\0&e^{-i\frac{\pi}{4}}\end{smallmatrix}\right] = \left[\begin{smallmatrix}0&e^{+i\frac{\pi}{4}}\\e^{-i\frac{\pi}{4}}&0\end{smallmatrix}\right] \\
TYT^\dagger &= \left[\begin{smallmatrix}1&0\\0&e^{+i\frac{\pi}{4}}\end{smallmatrix}\right]\left[\begin{smallmatrix}0&-i\\i&0\end{smallmatrix}\right]\left[\begin{smallmatrix}1&0\\0&e^{-i\frac{\pi}{4}}\end{smallmatrix}\right] = -\left[\begin{smallmatrix}0&e^{+i\frac{\pi}{4}}\\e^{-i\frac{\pi}{4}}&0\end{smallmatrix}\right] \\
TZT^\dagger &= \left[\begin{smallmatrix}1&0\\0&e^{+i\frac{\pi}{4}}\end{smallmatrix}\right]\left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right]\left[\begin{smallmatrix}1&0\\0&e^{-i\frac{\pi}{4}}\end{smallmatrix}\right] = \left[\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right] = Z
\end{aligned}$$

We need only consider the action of the Clifford element on each of the $4n$ single qubit Pauli gates, because the Pauli group elements are tensor products acting independently on separate qubits. Up to phase, a Clifford gate is completely determined by the transformation of these Pauli elements [TODO: Why?]. Moreover, the action of the identity is trivial, and the action on Y can be determined by that

on X and Y, since $VYV^\dagger = -iVXZV^\dagger = -i\ VXV^\dagger\ VZV^\dagger$. For single qubit Cliffords, X can map to 6 possibilities, $\{\pm X, \pm Y, \pm Z\}$, leaving 4 possibilities for the action on Z. This gives a total of $6 \times 4 = 24$ distinct 1-qubit Clifford groups.

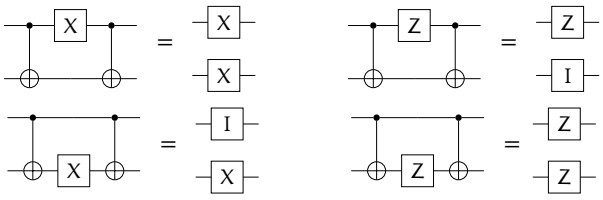
The 24 1-qubit Clifford gates are isomorphic to the group of rotations of an octahedron. The coordinates $R_n(\theta)$ of these are listed in table 12.1, along with the Pauli mappings. If we think of the Pauli gates X, Y, Z as the 3 cartesian axes x, y, z, then the elements of the Clifford group correspond to rotations that map axes to axes. We have 3 elements that rotate 180° about vertices (X, Y, Z); 6 elements (the square roots of X, Y, and Z) that rotate 90° or 270° degrees around vertices; 6 Hadamard like gates that rotate 180° about edges; 8 gates elements that rotate 120° or 240° degrees around faces; and the identity. This is a subgroup of the full octahedral group (which includes inversions), and also equal(??) to S_4, the group of permutations of 4 objects.



All 24 single qubit Clifford gates can be generated from just 2 elements, traditionally chosen to be S and H. For instance $X = HSSH$. Since $(SH)^3 = e^{2\pi i/8} I = \omega$ (32) we can generate each Clifford gate with 8 different phases. This is why you'll sometimes see the number of 1-qubit Clifford gates reported as $8 \times 24 = 192$, which includes in the possible Clifford gates integers powers of a phase $\omega^k = e^{2\pi i k/8}$, $k = 0, 1, \ldots, 7$.

12.2 Two qubit Clifford gates

Lets now consider the action of the 2-qubit CNOT gate on the X and Z single-qubit Pauli elements. Recall that CNOT is its own inverse. We can commute an X gate past the CNOT target, and a Z past the CNOT control, which leads to 2 trivial cases. But the other 2 cases are more interesting. An X gate acting on the control qubit becomes a pair of X gates, and a Z on the target qubit becomes a pair of Z gates.



For a CZ gate, the action on Z gates is trivial, but the action on X generates an extra Z gate.

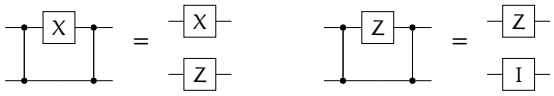
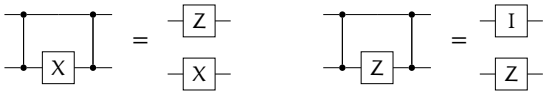


Table 12.1: Coordinates of the 24 1-qubit Clifford gates.

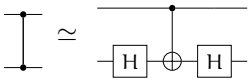
Gate	θ	n_x	n_y	n_z	X	Y	Z
I	0				$+X$	$+Y$	$+Z$
X	π	1	0	0	$+X$	$-Y$	$-Z$
Y	π	0	1	0	$-X$	$+Y$	$-Z$
Z	π	0	0	1	$-X$	$-Y$	$+Z$
V	$\frac{1}{2}\pi$	1	0	0	$+X$	$+Z$	$-Y$
$V^\dagger$	$-\frac{1}{2}\pi$	1	0	0	$+X$	$-Z$	$+Y$
$h^\dagger$	$\frac{1}{2}\pi$	0	1	0	$-Z$	$+Y$	$+X$
h	$-\frac{1}{2}\pi$	0	1	0	$+Z$	$+Y$	$-X$
S	$\frac{1}{2}\pi$	0	0	1	$+Y$	$-X$	$+Z$
$S^\dagger$	$-\frac{1}{2}\pi$	0	0	1	$-Y$	$+X$	$+Z$
H	π	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	$+Z$	$-Y$	$+X$
	π	$\frac{1}{\sqrt{2}}$	0	$\frac{1}{\sqrt{2}}$			
	π	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$			
	π	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0			
	π	$\frac{1}{\sqrt{2}}$	0	$-\frac{1}{\sqrt{2}}$			
	π	0	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$			
C	$\frac{2}{3}\pi$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$+Y$	$+Z$	$+X$
$C^\dagger$	$-\frac{2}{3}\pi$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$+Z$	$+X$	$+Y$
	$\frac{2}{3}\pi$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$			
	$-\frac{2}{3}\pi$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$			
	$\frac{2}{3}\pi$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$			
	$-\frac{2}{3}\pi$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$			
	$\frac{2}{3}\pi$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$			
	$-\frac{2}{3}\pi$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$			

Table 12.2: Clifford tableaus for select 2-qubit gates

Gate	qubit	X	Z	Gate	qubit	X	Z
I	0	$+X \otimes I$	$+Z \otimes I$				
	1	$+I \otimes X$	$+I \otimes Z$				
CNOT	0	$+X \otimes X$	$+Z \otimes I$	CZ	0	$+X \otimes Z$	$+Z \otimes I$
	1	$+I \otimes X$	$+Z \otimes Z$		1	$+Z \otimes X$	$+I \otimes Z$
iSWAP	0	$-Z \otimes Y$	$+I \otimes Z$	DCNOT	0	$+X \otimes X$	$+I \otimes Z$
	1	$-Y \otimes Z$	$+Z \otimes I$		1	$+X \otimes I$	$+Z \otimes Z$
SWAP	0	$+I \otimes X$	$+I \otimes Z$				
	1	$+X \otimes I$	$+Z \otimes I$				



The CNOT and CZ gate are locally equivalent, and are interrelated by 1-qubit Clifford gates.



Up to local equivalence there are only 4 classes of 2-qubit Clifford gates: the 2-qubit identity; the CNOT/CZ class, iSwap/DCNOT class, and SWAP. In canonical coordinates these are $\mathrm{CAN}(0,0,0)$, $\mathrm{CAN}(\frac{1}{2},0,0)$, $\mathrm{CAN}(\frac{1}{2},\frac{1}{2},0)$, and $\mathrm{CAN}(\frac{1}{2},\frac{1}{2},\frac{1}{2})$. Any canonical gate with integer or half integer arguments is a Clifford, and can be converted to the archetype of one of the classes with 1-qubit Cliffords.

12.3 Clifford tableau

A Clifford gate can be uniquely specified by the gate's actions on the Pauli matrices (this follows from the definition of the Clifford gates as the group that normalizes the Pauli group). And for an n qubit gate we only need to consider the action on each of the The X and Z Paulis on each of the n qubits. This is because we can deduce the action on Y from that on X and Z, and the Pauli group factorizes as a direct product of single qubit Pauli. Some examples of such *Clifford tableaus* for two-qubit gates are shown fig. 12.2.

The Clifford tableau representation is redundant, because there are additional restraints: The resultant Pauli product can't be the identity, and the X and Z actions must anti-commute (???). But the redundancy isn't large. For an n qubit gate we need to specify the action on $2n$ Paulis, each of which requires 2 bits for the 4 possibilities (I, X, Y, Z) on each qubit, plus a sign bit. So the number of bits needed to

Table 12.3: Number of Clifford gates $|C_n|$ for n qubits [0]

$$|C_n| = 2^{n^2+2n} \prod_{j=1,n} 4^j - 1$$

n	$\lvert C_n \rvert$	$\log_2 \lvert C_n \rvert$	2n(2n+1)
1	24	4.58	6
2	11520	13.49	20
3	92897280	26.47	42
4	12128668876800	43.46	72
5	25410822678459187200	64.46	110
6	852437556169034724016128000	89.46	156
7	4576209955296803515123703815864320000	118.46	210

specify a Clifford is at most $2n(2n+1) \approx 4n^2$. The exact number of Clifford gates for given n is

$$|C_n| = 2^{n^2+2n} \prod_{j=1,n} 4^j - 1 \tag{115}$$

The minimum number of bits required to uniquely specific a Clifford is asymptotically $2n^2$, so the Clifford tableau redundancy is no more than a factor of 2. See table 12.3 for the first few numerical values.

A Weyl Chamber

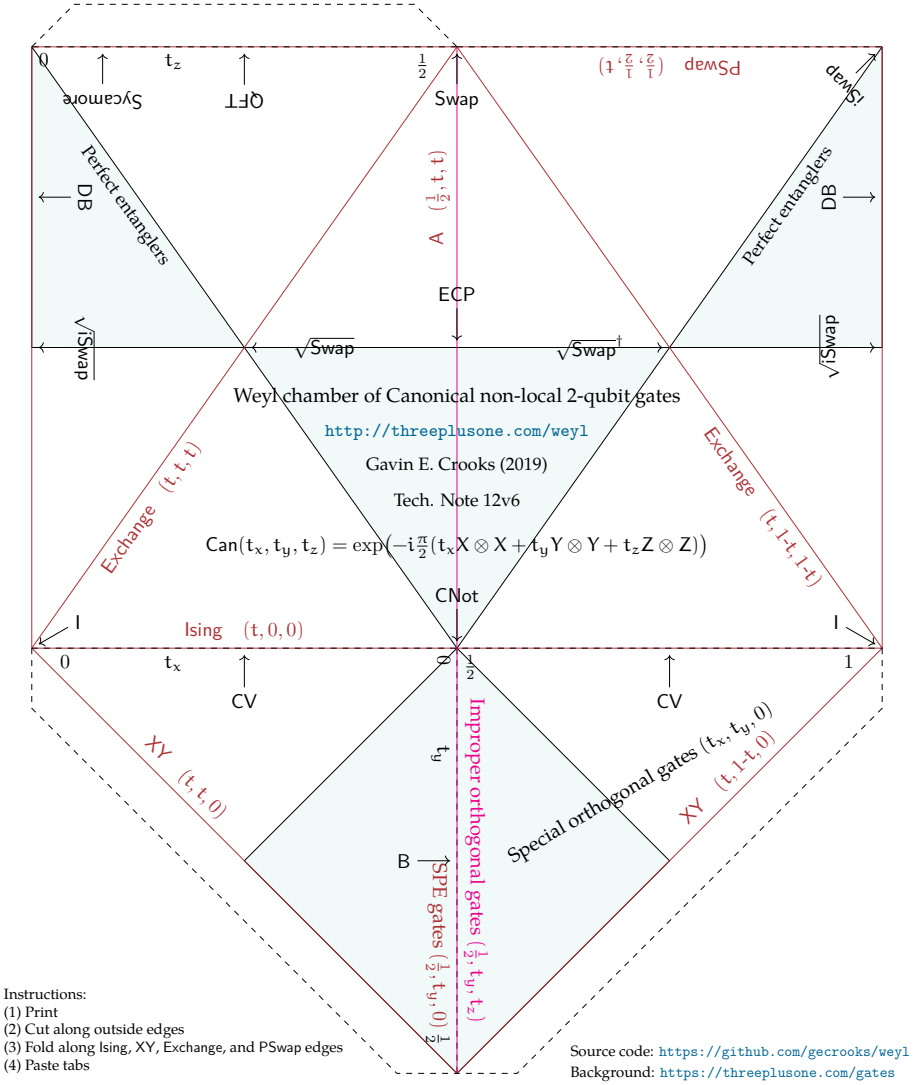


Figure A.1: The Weyl chamber of canonical non-local 2 qubit gates. (Print, cut, fold, and paste)

References

[0] [citation needed]. (pages 4, 5, 6, 9, 15, 15, 15, 15, 16, 16, 22, 22, 23, 23, 25, 25, 28, 28, 28, 28, 28, 29, 29, 32, 32, 34, 34, 34, 34, 34, 34, 34, 35, 35, 35, 37, 37, 37, 39, 41, 43, 44, 45, 46, 47, 47, 47, 47, 48, 48, 54, 54, 54, 54, 54, 54, 58, 58, 58, 58, 58, 60, 60, 61, 63, 63, and 69).

[1] P. A. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(2):416–418 (1939). DOI: 10.1017/S0305004100021162. Origin of bra-ket notation. No citations.

[2] F. Bloch. Nuclear induction. *Phys. Rev.*, 70:460–474 (1946). DOI: 10.1103/PhysRev.70.460. (page 6).

[3] Tommaso Toffoli. Reversible computing. In J. W. de Bakker and J. van Leeuwen, editors, *Automata, Languages and Programming, Seventh Colloquium. Noordwijkerhout, Netherlands*, pages 632–644. Springer Verlag (1980). Technical Report MIT/LCS/TM-151 (1980). Exposition of reversible classical computing, and demostration that computation can be dissipationless in principle. Origin of Toffoli (CCNOT) gate. (page 53 and 53).

[4] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3–4):219–253 (1982). DOI: 10.1007/BF01857727. (page 54).

[5] R. P. Feynman. Quantum mechanical computers. *Optics News*, 11:11 (1985). (page 53).

[6] A. Peres. Reversible logic and quantum computers. *Phys. Rev. A*, 32(6):3266–3276 (1985). DOI: 10.1103/physreva.32.3266. Origin of Peres 3-qubit gate. (page 55).

[7] David Elieser Deutsch. Quantum computational networks. *Proc. R. Soc. Lond. A*, 425(1868):73–90 (1989). (page 55 and 55).

[8] C. F. Van Loan and N. Pitsianis. Approximation with kronecker products. In Marc S. Moonen, Gene H. Golub, and Bart L. R. De Moor, editors, *Linear Algebra for Large Scale and Real-Time Applications*, pages 293–314. Springer Netherlands, Dordrecht (1993). ISBN 978-94-015-8196-7. DOI: 10.1007/978-94-015-8196-7_17. Van Loan–Pitsianis algorithm used to decompose Kronecker products of matrices. (page 47).

[9] Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of the 1993 IEEE 34th Annual Foundations of Computer Science*, SFCS '93, pages 352–361. IEEE Computer Society, USA (1993). ISBN 0818643706. DOI: 10.1109/SFCS.1993.366852. Development of quantum circuit model of quantum computation. Demonstration that the quantum circuit model is equivalent to the quantum Turing machine model. No citations.

[10] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61 (1994). DOI: 10.1103/PhysRevLett.73.58. (page 63).

[11] Norman Margolus. Simple quantum gates. Unpublished manuscript. Cited by [22] as the origin of the Margolus gate. (page 57).

[12] Adriano Barenco. A universal two-bit gate for quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 449(1937):679–683 (1995). DOI: 10.1098/rspa.1995.0066. (pages 38, 38, 55, and 56).

[13] David P. DiVincenzo. Two-bit gates are universal for quantum computation. *Phys. Rev. A*, 51:v (1995). DOI: 10.1103/PhysRevA.51.1015. ArXiv:cond-mat/9407022. (page 5).

[14] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467 (1995). DOI: 10.1103/PhysRevA.52.3457. (pages 10, 11, 11, 22, 22, 24, 50, 53, and 57).

[15] John Preskill. Quantum computation lecture notes (1997–2018). http://www.theory.caltech.edu/people/preskill/ph229/. (page 5).

[16] David P. DiVincenzo. Quantum gates and circuits. *Proc. R. Soc. Lond. A*, 454:261–276 (1998). DOI: 10.1098/rspa.1998.0159. (page 57).

[17] Klaus Mølmer and Anders Sørensen. Multiparticle entanglement of hot trapped ions. *Phys. Rev. Lett.*, 82:1835–1838 (1999). (page 34).

[18] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press (2000). (pages 5, 53, and 57).

[19] Charles F. Van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(1):85–100 (2000). DOI: 10.1016/S0377-0427(00)00393-9. (page 47).

[20] Kavita Dorai, Arvind, and Anil Kumar. Implementing quantum-logic operations, pseudopure states, and the deutsch-jozsa algorithm using noncommuting selective pulses in nmr. *Phys. Rev. A*, 61:042306 (2000). DOI: 10.1103/PhysRevA.61.042306. (page 16).

[21] Daniel Collins, Noah Linden, and Sandu Popescu. Nonlocal content of quantum operations. *Phys. Rev. A*, 64:032302 (2001). DOI: 10.1103/PhysRevA.64.032302. (page 35).

[22] Guang Song and Andreas Klappenecker. The simplified toffoli gate implementation by margolus is optimal. Quant-ph/0312225. Demonstration that the Margolus gate requires 3 CNot gates. (pages 57 and 72).

[23] Jun Zhang, Jiri Vala, Shankar Sastry, and K. Birgitta Whaley. Geometric theory of nonlocal two-qubit operations. *Phys. Rev. A*, 67:042313 (2003). arXiv:quant-ph/0209120. (page 28).

[24] D. Maslov and G.W. Dueck. Improved quantum cost forn-bit toffoli gates. *Electronics Letters*, 39:1790–1791 (2003). DOI: 10.1049/el:20031202. (page 55).

[25] Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Phys. Rev. A*, 69:032315 (2004). ArXiv:quant-ph/0308006. (pages 34, 49, 49, and 49).

[26] Jun Zhang, Jiri Vala, Shankar Sastry, and K. Birgitta Whaley. Optimal quantum circuit synthesis from controlled-unitary gates. *Phys. Rev. A*, 69:042309 (2004). ArXiv:quant-ph/0308167. (pages 28 and 35).

[27] A. T. Rezakhani. Characterization of two-qubit perfect entanglers. *Phys. Rev. A*, 70:052313 (2004). DOI: 10.1103/PhysRevA.70.052313. (page 42).

[28] M. Blaauboer and R. L. de Visser. An analytical decomposition protocol for optimal implementation of two-qubit entangling gates. *J. Phys. A : Math. Theor*, 41:395307 (2008). DOI: 10.1088/1751-8113/41/39/395307. arXiv:cond-mat/0609750. (pages 28, 40, and 40).

[29] Byron Drury and Peter Love. Constructive quantum Shannon decomposition from Cartan involutions. *J. Phys. A : Math. Theor*, 41(39):395305 (2008). arXiv:0806.4015. No citations.

[30] Ian Glendinning. Rotations on the bloch sphere. (page 27).

[31] J. J. Sakurai and Jim J. Napolitano. *Modern Quantum Mechanics*. Pearson, 2nd edition (2010). (page 5).

[32] Craig Gidney. Algorithmic assertions (2012–present). This blog by Craig Gidney is full of useful gems. Posts of particular interest include the trilogy "Constructing Large Controlled Not", "Constructing Large Increment Gates", and "Using Quantum Gates instead of Ancilla Bits" (2015), ... No citations.

[33] Scott Aaronson. *Quantum Computing since Democritus*. Cambridge University Press (2013). (page 5).

[34] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6):818–830 (2013). DOI: 10.1109/TCAD.2013.2244643. (page 53).

[35] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. Quipper: A scalable quantum programming language. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '13, pages 333–342. Association for Computing Machinery, New York, NY, USA (2013). ISBN 9781450320146. DOI: 10.1145/2491956.2462177. (pages 33 and 56).

[36] Paul Watts, Maurice O'Connor, and Jiří Vala. Metric structure of the space of two-qubit gates, perfect entanglers and quantum control. *Entropy*, 15:1963–1984 (2013). DOI: 10.3390/e15061963. (page 28).

[37] Eleanor G. Rieffel. *Quantum Computing: A Gentle Introduction*. The MIT Press (2014). (pages 5 and 9).

[38] Sheldon Axler. *Linear algebra done right*. Springer, 3rd edition (2015). (page 5).

[39] Leonard Susskind and Art Friedman. *Quantum Mechanics: The Theoretical Minimum*. Basic Books (2015). (page 5).

[40] Rigetti Computing. pyquil. Https://github.com/rigetticomputiong/pyquil. No citations.

[41] Robert S. Smith, Michael J. Curtis, and William J. Zeng. A practical quantum instruction set architecture. ArXiv:1608.03355. (pages 37, 37, and 41).

[42] Norbert M. Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A. Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. *Proc. Natl. Acad. Sci. U.S.A.*, 114(13):3305–3310 (2017). DOI: 10.1073/pnas.1618020114. (page 57).

[43] Mark M. Wilde. *Quantum information theory*. Cambridge University Press, 2nd edition (2017). (page 5).

[44] Chris Ferrie and whurley. *Quantum Computing for Babies*. Sourcebooks Explore (2018). (page 5).

[45] John Watrous. *The theory of quantum information*. Cambridge University Press (2018). (page 5).

[46] Xiao-Feng Shi. Deutsch, toffoli, and cnot gates via rydberg blockade of neutral atoms. *Phys. Rev. Applied*, 9:051001 (2018). (page 55).

[47] Panagiotis Kl. Barkoutsos, Jerome F. Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J. Egger, Matthias Troyer, Antonio Mezzacapo, Stefan Filipp, and Ivano Tavernelli. Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions. *Phys. Rev. A*, 98:022322 (2018). DOI: 10.1103/PhysRevA.98.022322. (page 44 and 44).

[48] Andy Matuschak and Michael Nielsen. Quantum country (2019). https://quantum.country/. (page 5).

[49] Deanna M. Abrams, Nicolas Didier, Blake R. Johnson, Marcus P. da Silva, and Colm A. Ryan. Implementation of xy entangling gates with a single calibrated pulse. *Nature Electronics*, 3(12):744–750 (2020). DOI: 10.1038/s41928-020-00498-1. ArXiv:1912.04424. (page 39 and 39).

[50] Bryan T. Gard, Linghua Zhu, George S. Barron, Nicholas J. Mayhall, Sophia E. Economou, and Edwin Barnes. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Information*, 6(1):10 (2020). DOI: 10.1038/s41534-019-0240-1. (pages 44, 44, 44, and 45).

[51] Eric C. Peterson, Gavin E. Crooks, and Robert S. Smith. Two-qubit circuit depth and the monodromy polytope. *Quantum*, 4:247 (2020). DOI: 10.22331/q-2020-03-26-247. arXiv:1904.10541. (pages 39, 39, and 43).

[52] S. E. Rasmussen and N. T. Zinner. Simple implementation of high fidelity controlled-iswap gates and quantum circuit exponentiation of non-hermitian gates. *Phys. Rev. Research*, 2:033097 (2020). DOI: 10.1103/PhysRevResearch.2.033097. (page 57).

[53] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497 (2021). DOI: 10.22331/q-2021-07-06-497. (pages 20 and 21).

[54] Matthew P. Harrigan, Kevin J. Sung, Matthew Neeley, Kevin J. Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Sean Demura, Andrew Dunsworth, Daniel Eppens, Austin Fowler, Brooks Foxen, Craig Gidney, Marissa Giustina, Rob Graff, Steve Habegger, Alan Ho, Sabrina Hong, Trent Huang, L. B. Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Cody Jones, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Seon Kim, Paul V. Klimov, Alexander N. Korotkov, Fedor Kostritsa, David Landhuis, Pavel Laptev, Mike Lindmark, Martin Leib, Orion Martin, John M. Martinis, Jarrod R. McClean, Matt McEwen, Anthony Megrant, Xiao Mi, Masoud Mohseni, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Charles Neill, Florian Neukart, Murphy Yuezhen Niu, Thomas E. O'Brien, Bryan O'Gorman, Eric Ostby, Andre Petukhov, Harald Putterman, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Andrea Skolik, Vadim Smelyanskiy, Doug Strain, Michael Streif, Marco Szalay, Amit Vainsencher, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Leo Zhou, Hartmut Neven, Dave Bacon, Erik Lucero, Edward Farhi, and Ryan Babbush. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics*, 17(3):332–336 (2021). DOI: 10.1038/s41567-020-01105-y. (pages 45, 46, 46, 46, and 50).

[55] Ivan Savov. *No bullshit guide to linear algebra*. Minireference Co. (2017). (page 5).

[56] J. A. Jones, R. H. Hansen, and M. Mosca. Quantum logic gates and nuclear magnetic resonance pulse sequences. *Journal of Magnetic Resonance*, 135(2):353–360 (1998). arXiv:quant-ph/9805070. (page 16).

[57] H. Häffner, C.F. Roos, and R. Blatt. Quantum computing with trapped ions. *Physics Reports*, 469(4):155–203 (2008). (page 34).

[58] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Phys. Rev. Lett.*, 120:110501 (2018). ArXiv:1711.04789. (page 39).

[59] Jun Zhang, Jiri Vala, Shankar Sastry, and K. Birgitta Whaley. Minimum construction of two-qubit quantum operations. *Phys. Rev. Lett.*, 93:020502 (2004). Quant-ph/0312193. (pages 42, 49, and 50).

[60] Peter Selinger. Quantum circuits of t-depth one. *Phys. Rev. A*, 87:042302 (2013). DOI: 10.1103/PhysRevA.87.042302. 1210.0974. (page 56 and 56).

[61] Dmitri Maslov. Advantages of using relative-phase toffoli gates with an application to multiple control toffoli optimization. *Phys. Rev. A*, 93:022311 (2016). DOI: 10.1103/PhysRevA.93.022311. (pages 56, 57, and 57).

[62] V. V. Shende, S. S. Bullock, and I. L. Markov. Synthesis of quantum-logic circuits. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 25(6):1000–1010 (2006). DOI: 10.1109/TCAD.2005.855930. (page 62).

Index

http://threeplusone.com/gates
Berkeley Institute for Theoretical Sciences (BITS)
typeset on 2022-01-12 with XeTeX version 0.99999
fonts: Trump Mediaeval (text), Euler (math)
2 7 1 8 2 8 1 8

This monograph is inevitably incomplete, inaccurate, and otherwise imperfect — *caveat emptor*.