Lecture 5, 17 Sep 2020

180 degree rotations, as demo'd

$$u(\hat{x}, \pi) = \cos\frac{\pi}{2} + i\sigma_x \sin\frac{\pi}{2} = i\sigma_x$$

$$i\sigma_y \, i\sigma_x = i\sigma_z$$

90° rotations, as demo'd

$$u(\hat{x}, \pi/2) = \frac{1}{\sqrt{2}}(1 + i\sigma_x)$$

$$u(\hat{x}, \pi/2)^2 = \frac{1}{2}(1 + 2i\sigma_x - 1) = i\sigma_x$$

$$\frac{1}{\sqrt{2}}(1 + i\sigma_y)\frac{1}{\sqrt{2}}(1 + i\sigma_x) = \frac{1}{2}(1 + i\sigma_x + i\sigma_y + i\sigma_z)$$

$$\cos\frac{\theta}{2} = 1/2 \qquad \frac{\theta}{2} = \pi/3 \qquad \theta = 2\pi/3$$

$$\sin\frac{\theta}{2} = \sqrt{3}/2,$$ 120° rotation about $\hat{n} = \frac{1}{\sqrt{3}}(1, 1, 1)$

Now rotate 90° around x-axis, bring z-axis to $\hat{y}$ direction.
Rotate around that new y-axis.
Then rotate back 90° around $\hat{x}$, restoring z-axis to original position

$$\frac{1}{\sqrt{2}}(1-i\sigma_x)\frac{1}{\sqrt{2}}(1+i\sigma_y)\frac{1}{\sqrt{2}}(1+i\sigma_x)$$

$$\frac{1}{2\sqrt{2}}(1-i\sigma_x)(1+i\sigma_x+i\sigma_y+i\sigma_z)$$

$$=\frac{1}{2\sqrt{2}}(1+\cancel{i\sigma_x}+\cancel{i\sigma_y}+i\sigma_z-\cancel{i\sigma_x}+1+i\sigma_z-\cancel{i\sigma_y})$$

$$=\frac{1}{\sqrt{2}}(1+i\sigma_z)$$

Result is rotation around original z-axis

Recall parametrization of single qubit wave function, here's how related to canonical form of transformation

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

$$= e^{i\hat{n}\cdot\sigma\frac{\theta}{2}}|0\rangle$$

where $\hat{n} = (\sin\varphi, -\cos\varphi, 0)$

$$\left[\cos\frac{\theta}{2}\mathbb{1} + \begin{pmatrix} & in_1+n_2 \\ in_1-n_2 & \end{pmatrix}\sin\frac{\theta}{2}\right]|0\rangle$$

$in_1 - n_2 = \cos\varphi + i\sin\varphi = e^{i\varphi}$

$$= \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}e^{i\varphi}|1\rangle$$

in python $\quad u(\hat{n},\theta) = \cos\frac{\theta}{2}\mathbb{1} + i\sin\frac{\theta}{2}\,\hat{n}\cdot\vec{\sigma}$

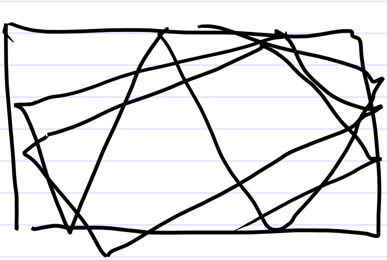
```
s0 = np.eye(2)
s1 = np.array([[0,1],[1,0]])
s2 = np.array([[0,-1j],[1j,0]])
s3 = np.array([[1,0],[0,-1]])
s = np.transpose([s1,s2,s3], [1,2,0])   # vector of sigma matrices

def u(n,th):  return np.cos(th/2)*s0 + 1j*np.sin(th/2)*s.dot(n)
```

e.g.
u(x, -np.pi/2).dot(u(y, np.pi/2)).dot(u(x, np.pi/2))

x = (1,0,0)
y = (0,1,0)
z = (0,0,1)

$\longrightarrow$ u(z, np.pi/2)



generically difficult to close on a discrete subgroup, so ...

... Easy to find a universal gate set:

1 qubit: $\underbrace{1, H, T}_{\text{gives any } SU(2) \text{ element}}$

where $T = \begin{pmatrix} 1 & \\ & e^{i\pi/4} \end{pmatrix}$, $T^4 = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} = Z$

[though might take lots of ops to get arbitrarily close to desired unitary op

---

$1, H, \text{cNOTs}, T = \begin{pmatrix} 1 & \\ & e^{i\pi/4} \end{pmatrix}$

n-qubit

gives all of $SU(2^n)$ (!)

Reversible ops on cbits extend by linearity to unitary on qubits

$$U|x\rangle_n = |P(x)\rangle_n$$

$$U|\psi\rangle = \sum_x \alpha_x U|x\rangle = \sum_x \alpha_x |P(x)\rangle$$

$$|\psi\rangle = \sum_x \alpha_x |x\rangle$$

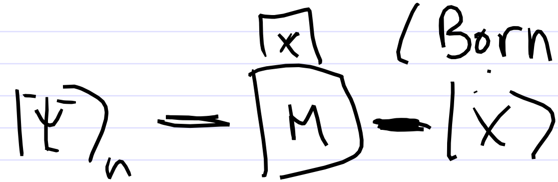
$$= \sum_x \alpha_{P^{-1}(x)} |x\rangle$$

$$\langle U\psi | U\psi\rangle = \sum_x \alpha^*_{P^{-1}(x)} \alpha_{P^{-1}(x)}$$

$$= \sum_x \alpha^*_x \alpha_x = \langle \psi | \psi \rangle$$

Also applies to Z, H

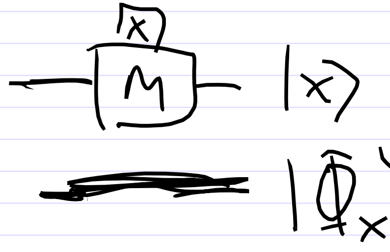
(Born rule)



$p(x) = |\alpha_x|^2$

Generalized Born:

$$|\Psi\rangle_{n+1} = \alpha_0|0\rangle|\Phi_0\rangle_n + \alpha_1|1\rangle|\Phi_1\rangle_n$$



$p = |\alpha_x|^2$

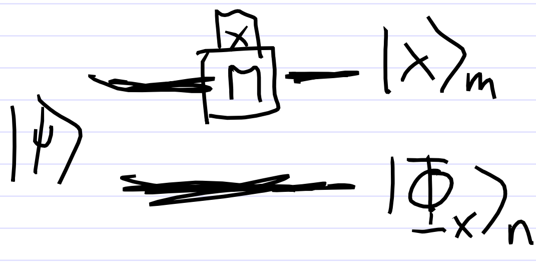
measuring one at a time

reproduces ✡.

Additional generalization: Measure 1st m of m+n qubits

$$|\Psi\rangle_{m+n} = \sum_{x=0}^{2^m-1} \alpha_x |x\rangle_m |\Phi_x\rangle_n$$

with probability $|\alpha_x|^2$ measure x and project to $|x\rangle_m |\Phi_x\rangle$:



State preparation

Think about 2 state system

$0, E$

$$p(\epsilon) = \frac{e^{-E/kT}}{1+e^{-E/kT}}$$

if in state ϵ, add dE (to make less likely)

$$\int_0^\infty dE\, p(E) = -kT \ln\left(1+e^{-E/kT}\right)\Big|_0^\infty$$

$$= kT \ln 2$$

Energy to erase a qubit

$dE = T\,dS \qquad dS = k \ln 2$

to erase each bit of info!

Probably easier: just "measure" and flip if projected to $|1\rangle$ (by applying X)

Next time:

General computational process

---

probset 2 out tomorrow (?)