Quantum computing I

PHYS 246 class 13

https://lkwagner.github.io/IntroductionToComputationalPhysics/intro.html

```
from google.colab import drive
drive.mount('/content/drive')
!cp /content/drive/MyDrive/Colab\ Notebooks/Dynamics.ipynb ./
!jupyter nbconvert --to HTML "Dynamics.ipynb"
```

Announcements/notes

Apr 28 (next Tuesday)
CS+Physics lunch & discussion

- 'Fluids' is due tonight
- For QC1 there are a couple small clarifying updates but no big changes from previous version.
- May 13th NOON -- slides and ipynb file due. I may have to remake the assignment so the groups are correct so don't submit anything yet. :)
- Note that the final exam is REQUIRED to obtain a grade in this course

Quick quantum intro

Classical mechanics:

State $x, v, \mathscr{R}^{6N}$ vectors

Dynamical equation: $F = ma = m\dfrac{d^2x}{dx^2}$

Measurement:
x, v are definite!

Quantum mechanics:

State $\Psi(x)$, function $\mathscr{R}^{3N} \to \mathscr{C}$

Dynamical equation:
$$i\hbar\frac{\partial\Psi}{\partial t} = \hat{H}\Psi$$

Measurement:
$\rho(x) = |\Psi(x)|^2$

$$\rho(p) = |\Psi(p)|^2 = \left| \int e^{ipx/\hbar}\Psi(x)dx \right|^2$$

Two-state systems

Electron spin, other q-bit systems

Can be only up or down.

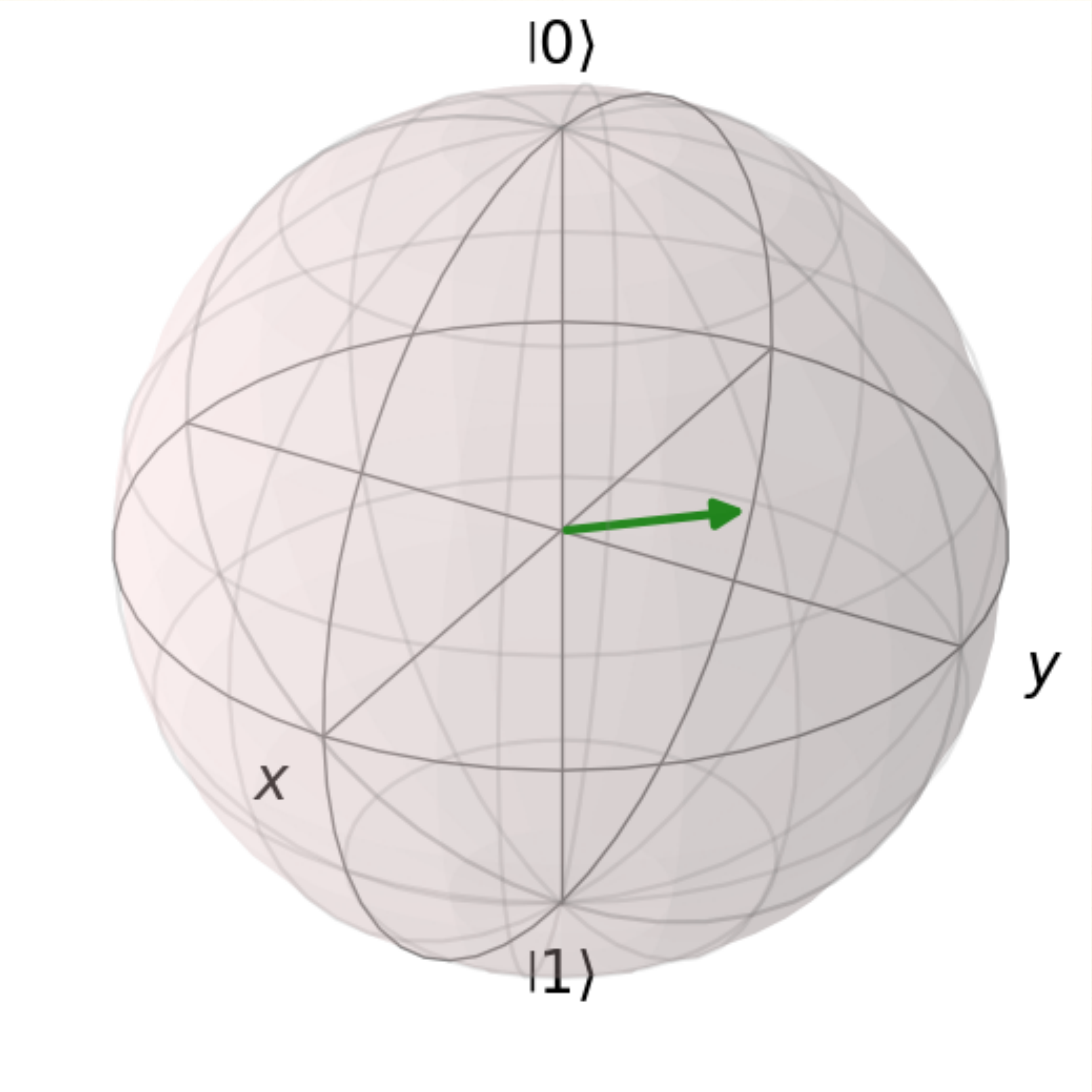
Classically, $x = [1,0] = \uparrow$ or $[0,1] = \downarrow$ (Ising model)

Then $\Psi(x)$ is specified by two complex numbers $[a, b] = a \uparrow + b \downarrow$.

Probability we measure the state $\uparrow$: $\dfrac{|a|^2}{|a|^2 + |b|^2}$

Normalization: set $|a|^2 + |b|^2 = 1$

Bloch sphere



$\uparrow$ is $[1{,}0] = |0\rangle$
$\downarrow$ is $[0{,}1] = |1\rangle$

$\rightarrow$(x) is $[1{,}1] = |0\rangle + |1\rangle$
and so on

Spin direction	State
$\hat{z}$	$\uparrow$
$-\hat{z}$	$\downarrow$
$\hat{x}$	$\frac{1}{\sqrt{2}}(\uparrow + \downarrow)$
$-\hat{x}$	$\frac{1}{\sqrt{2}}(\uparrow - \downarrow)$
$\hat{y}$	$\frac{1}{\sqrt{2}}(\uparrow + i \downarrow)$
$-\hat{y}$	$\frac{1}{\sqrt{2}}(\uparrow - i \downarrow)$

You might think this is 4d since there are two complex numbers, but an overall phase does not change anything, so the states can be represented in 3d.

Multiple two-state systems

Classically, the phase space is $\uparrow\uparrow, \uparrow\downarrow, \downarrow\uparrow, \downarrow\downarrow$

Each gets multiplied by a different complex number:
$\Psi = a \uparrow\uparrow + b \uparrow\downarrow + c \downarrow\uparrow + d \downarrow\downarrow$ (four complex numbers)

Entanglement

Consider the state $\frac{1}{\sqrt{2}} \uparrow \uparrow - \frac{1}{\sqrt{2}} \downarrow \downarrow .$

P(both up) = 1/2
P(both down) = 1/2
P(one up, one down) = 0
P(one down, one up) = 0

This is called an entangled state
(sometimes a 'cat' state)

Measuring one spin will also collapse the state of the other spin

Discussion:
Is $\frac{1}{2}(\uparrow \uparrow + \uparrow \downarrow + \downarrow \uparrow + \downarrow \downarrow)$ entangled?

Gates

In quantum computing, we idealize the time dependence of the wave function to 'gates'

Hadamard gate:

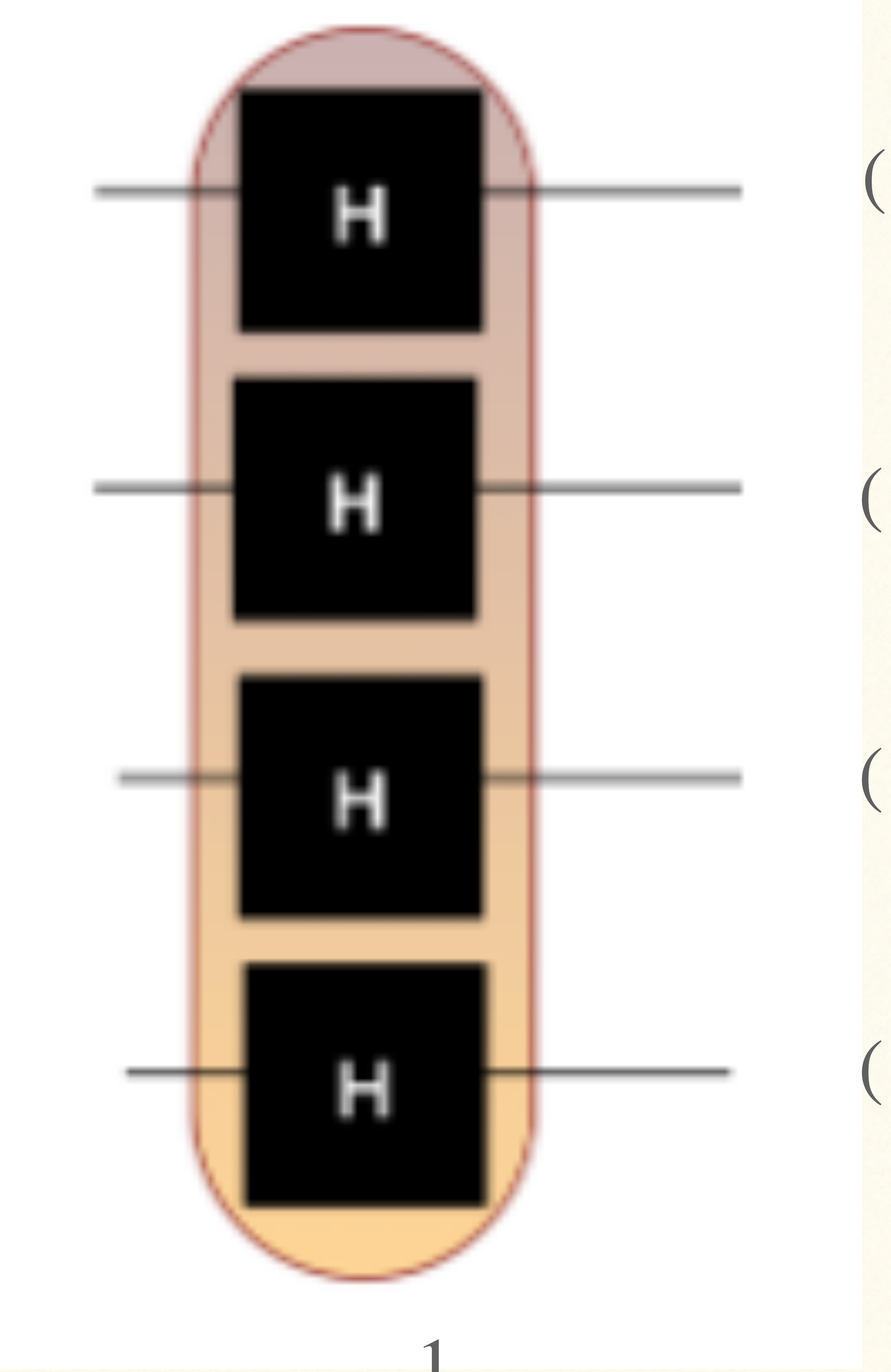
$$H\uparrow = \frac{1}{\sqrt{2}}(\uparrow + \downarrow)$$

$\uparrow$ $\quad$ $(\uparrow + \downarrow)$

$\uparrow$ $\quad$ $(\uparrow + \downarrow)$

$\uparrow$ $\quad$ $(\uparrow + \downarrow)$

$\uparrow$ $\quad$ $(\uparrow + \downarrow)$

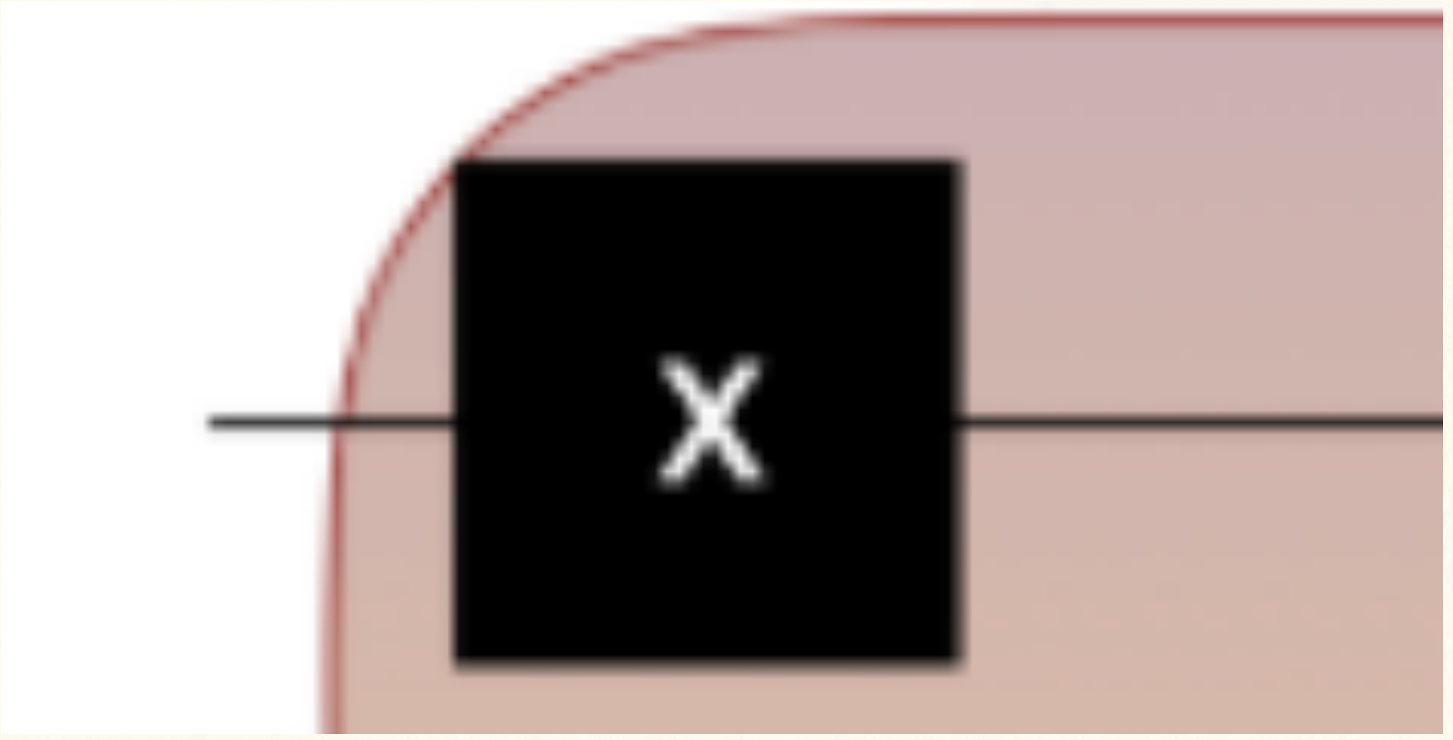


$$\uparrow\uparrow\uparrow\uparrow \qquad \frac{1}{4}(\uparrow + \downarrow)(\uparrow + \downarrow)(\uparrow + \downarrow)(\uparrow + \downarrow)$$

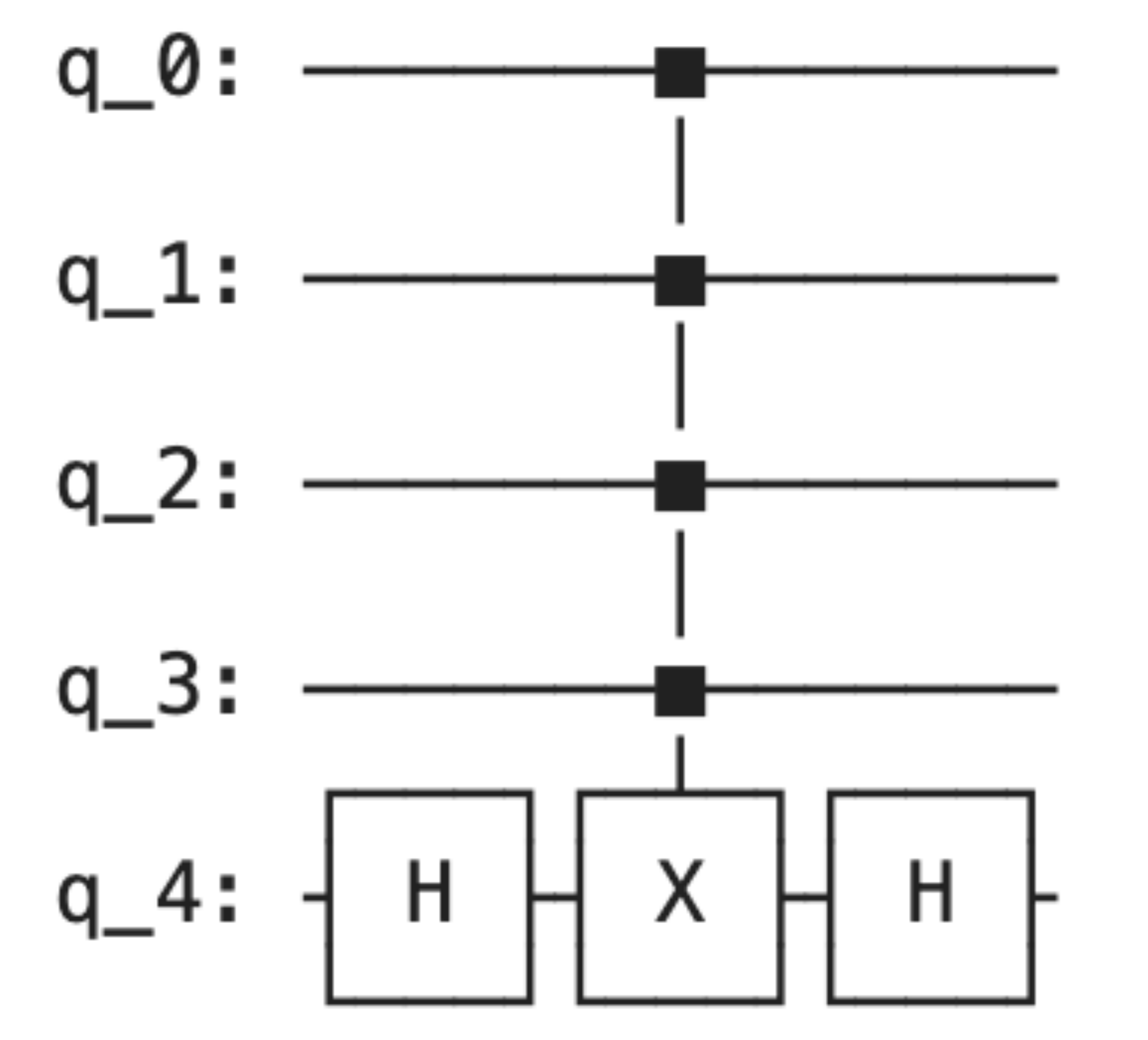
x-gate

Named after the Pauli matrix

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$\uparrow$  $\downarrow$

Control gates



Often the only way we introduce entanglement!

Apply an x on qbit 4 if 0, 1, 2, and 3 are all 1.

quiskit and circuits

```
def Mark(r,N):
  circuit=QuantumCircuit(N,N)
  circuit.barrier()


  myString=np.binary_repr(r,width=N)[::-1]
  for i in range(0,len(myString)):
    if myString[i]=='0':
      circuit.x(i)
  circuit.barrier()

  circuit.h(N-1)
  circuit.mcx(list(range(0,N-1)), N-1,mode='noancilla')
  circuit.h(N-1)
  circuit.barrier()

  for i in range(0,len(myString)):
    if myString[i]=='0':
      circuit.x(i)
  circuit.barrier()
  return circuit
```

Create an empty circuit with N bits

Adds a gate to the circuit

Classical search

N items.

We would like to find one element that matches $f(i) = 1$ ($f(i) = 0$ for all other elements)

Classically you go through items until you find the right one. On average $N/2$.
$\mathcal{O}(N)$

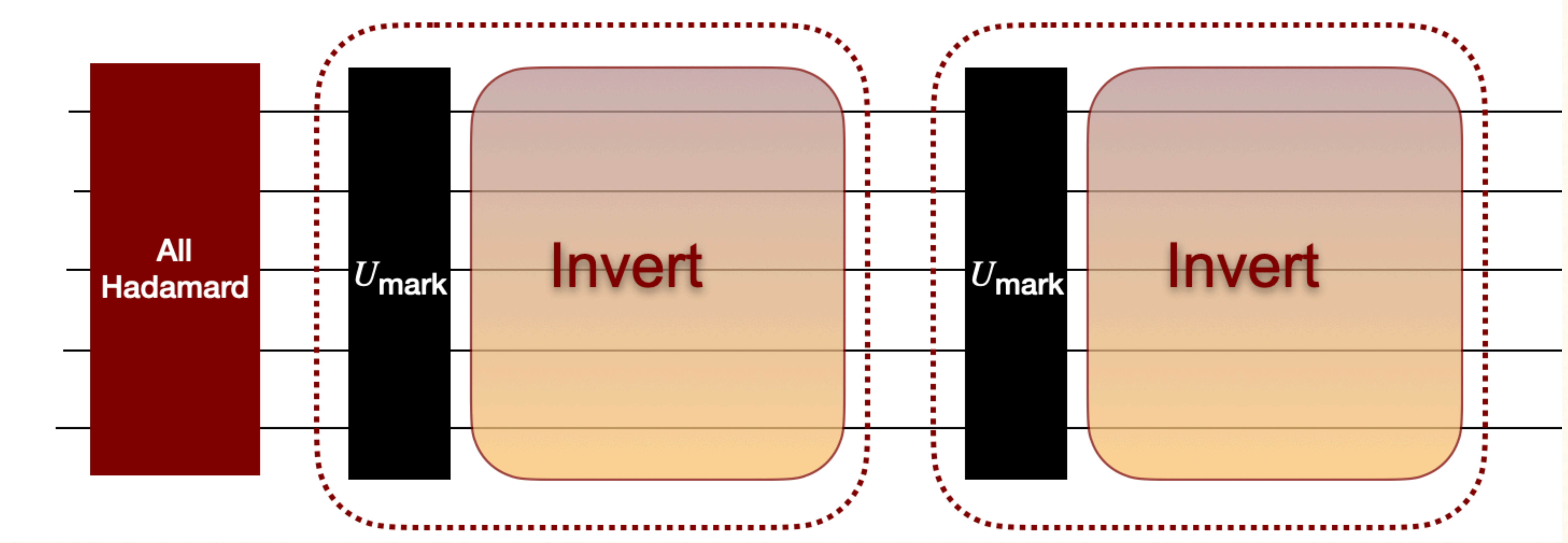
Databases, factorization, find the winning chess move, find the path for a robot to move in...

Grover's algorithm

Quantum search.

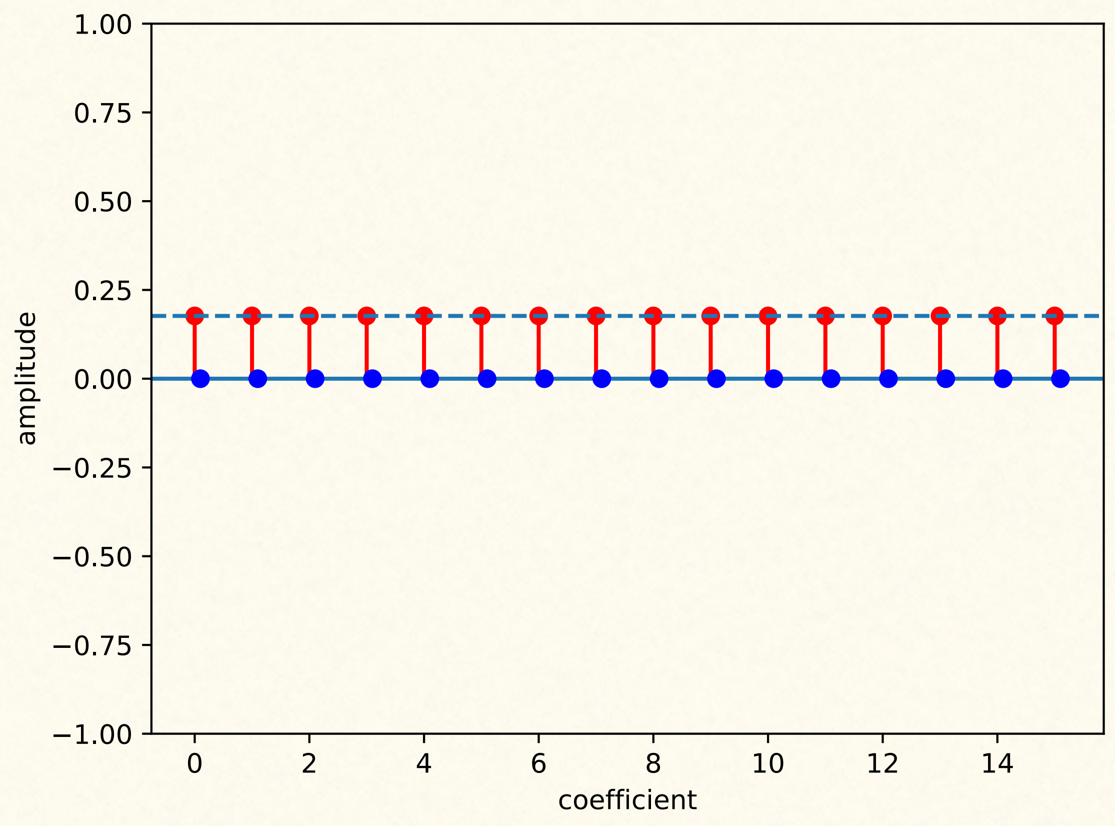
Scales as $\mathcal{O}(\sqrt{N})$. One of the few (only?) useful quantum algorithms with a **proven** speedup over classical algorithms.

The algorithm

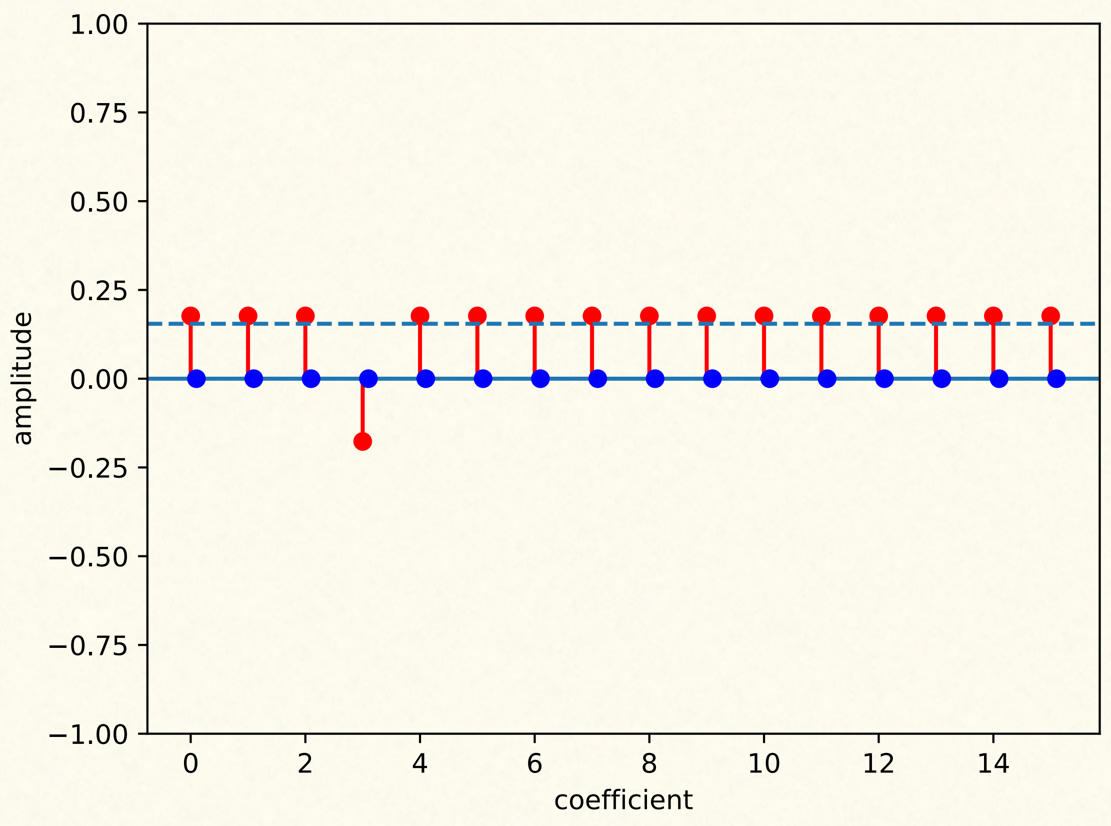


Measure

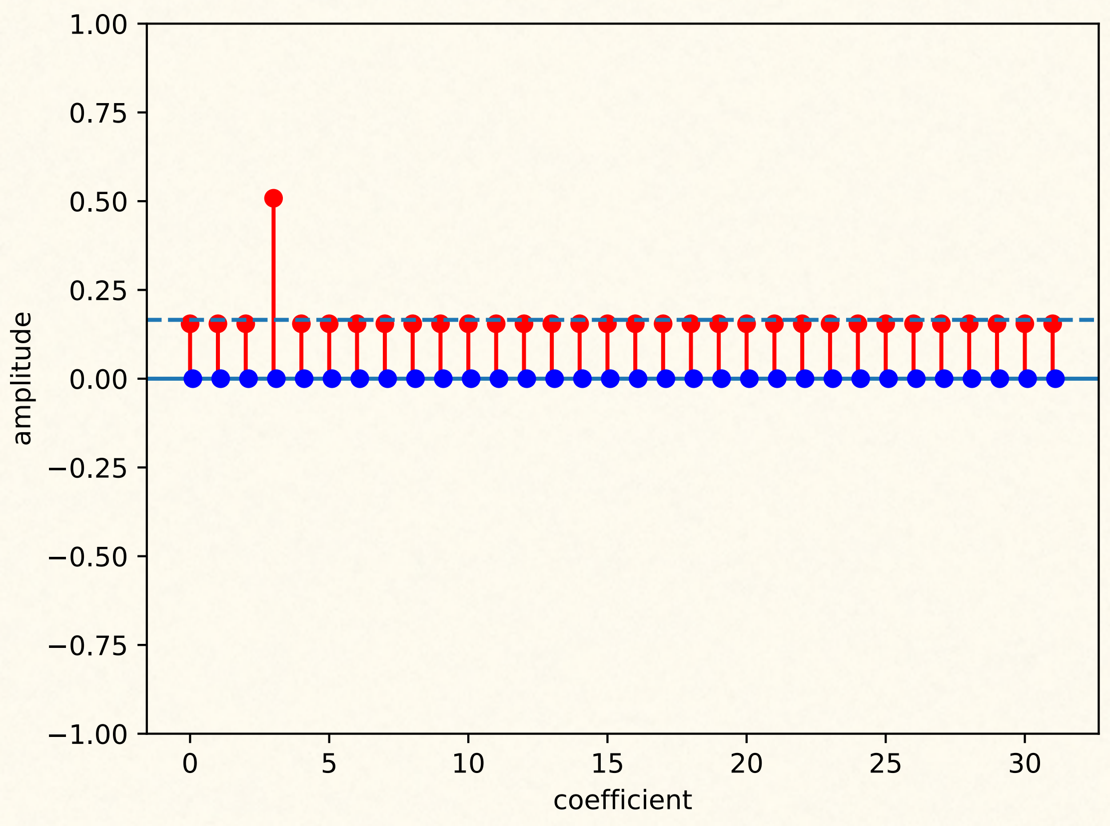
Start 00000, all hadamard



Mark



Invert



Repeat