INTRO TO QUANTUM COMPUTING

LECTURE #10

THE QUANTUM STACK & MATH REVIEW

FRANCISCA VASCONCELOS

1/10/2021

Welcome to Semester 2!

Connect with us on **social media**, and let us know what you're most looking forward to!

- **Facebook:** @thecodingschool
- **Instagram:** @the_coding_school
- **LinkedIn:** @the-coding-school
- **Twitter:** @qubitbyqubit
- #QxQ #FutureQuantumLeader

ANNOUNCEMENTS

1. All course announcements will be communicated to students through **Canvas**
   - **Check Canvas** AT LEAST 1-2 times weekly
   - Change your Canvas **notification settings** to make sure you receive emails when we post an announcement
2. **Labs** begin this week! (starting Tuesday, going through Saturday)
   - Remember to submit your **attendance quiz** during lab! Attendance is mandatory
   - 3 excused absences, won't affect your grade

ANNOUNCEMENTS

3. **Grades**: homework and attendance are both **required**
    - **Homework:** Your homework this week: Technical Assessment (only a *completion grade!)*
4. Course resources
    - **Piazza** for content-related questions answered by TAs
    - **Student Assistant Office Hours** (Fridays from 8:00 am-2:00 pm EST. Info posted later this week on Canvas).
    - **Friday Homework Review** (Fridays from 4:00-5:00 pm EST. Zoom link to be posted on Canvas).
5. Use **Discord** to participate in the course community [optional]

ABOUT YOUR INSTRUCTOR, FRAN

MIT Engineering Quantum Systems (EQuS)

Teaching Quantum Computing at MIT!

MSc in Statistics at Oxford
& teaching for Qubit x Qubit!

Intern at
Rigetti Computing
and
Microsoft Research
Quantum!

TODAY'S LECTURE

1. The Quantum Stack
   a) Classical Software
   b) Quantum Software
   c) Quantum Program Compiler
   d) Classical Control Hardware
   e) Quantum Hardware

2. Math Review
   a) Vectors
   b) Complex Numbers & Exponentials
   c) Matrices
   d) Probability
   e) Quantum States & Dirac Notation
   f) Quantum Operations & Gates
   g) Vector & Hilbert Spaces
   h) Bases
   i) Eigenvectors & Eigenvalues

THE QUANTUM STACK

BEGINNING OF LAST SEMESTER…

Lecture #1 (Classical Computing):

We learned abstractly and theoretically how computation is performed on classical computers…

- Binary Representation
- Boolean Logic

Lecture #2 (Quantum Computing in the Abstract):

We learned abstractly, *from a physics perspective*, how quantum computers work and compare to classical computers…

- Qubits
- Quantum Superposition
- Quantum Gates
- Quantum Entanglement

WATCH THIS LECTURE IF YOU HAVEN'T ALREADY!!!

TODAY'S LECTURE

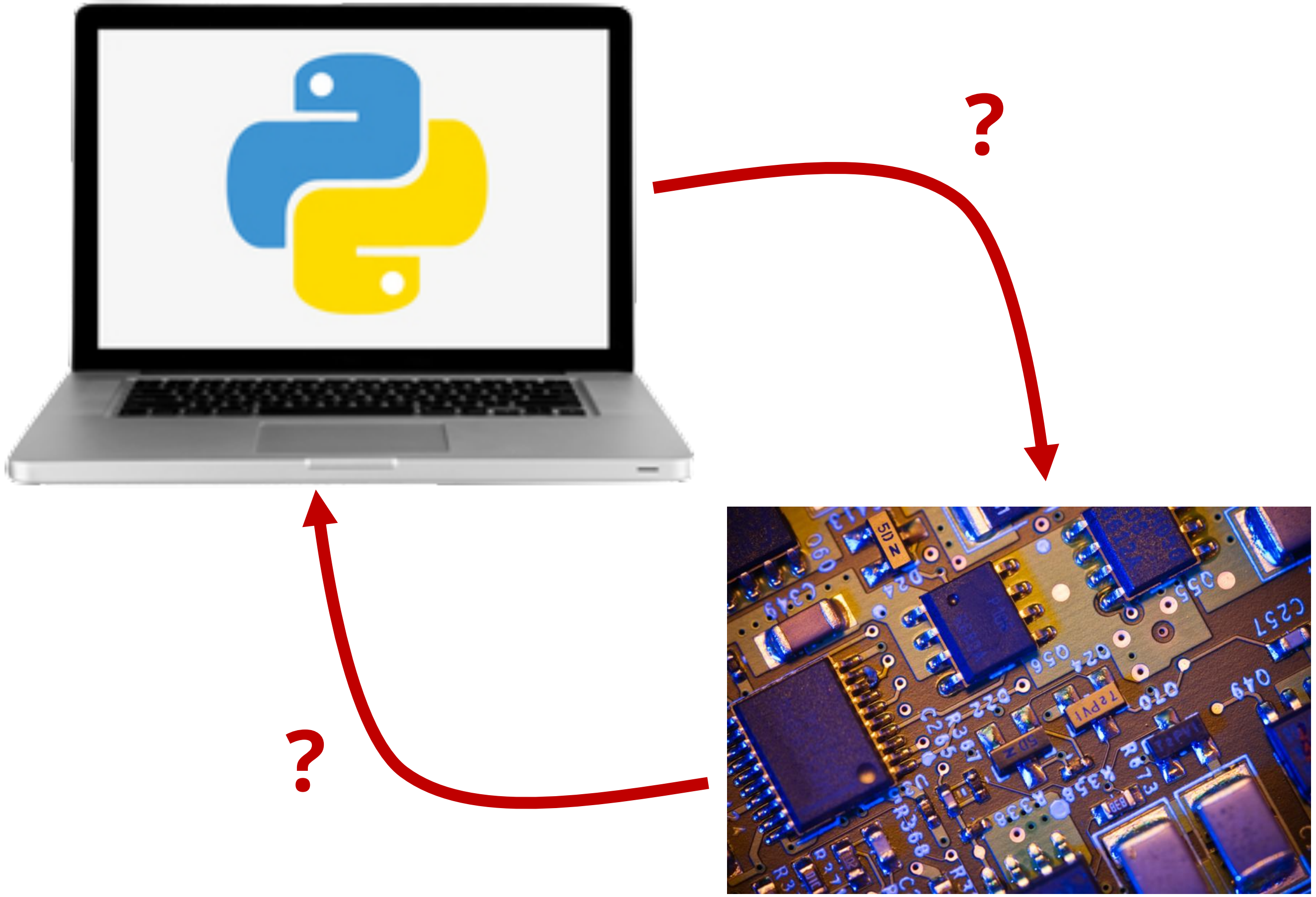
We are going to learn abstractly, *from an EECS perspective*, how quantum computers work and compare to classical computers…

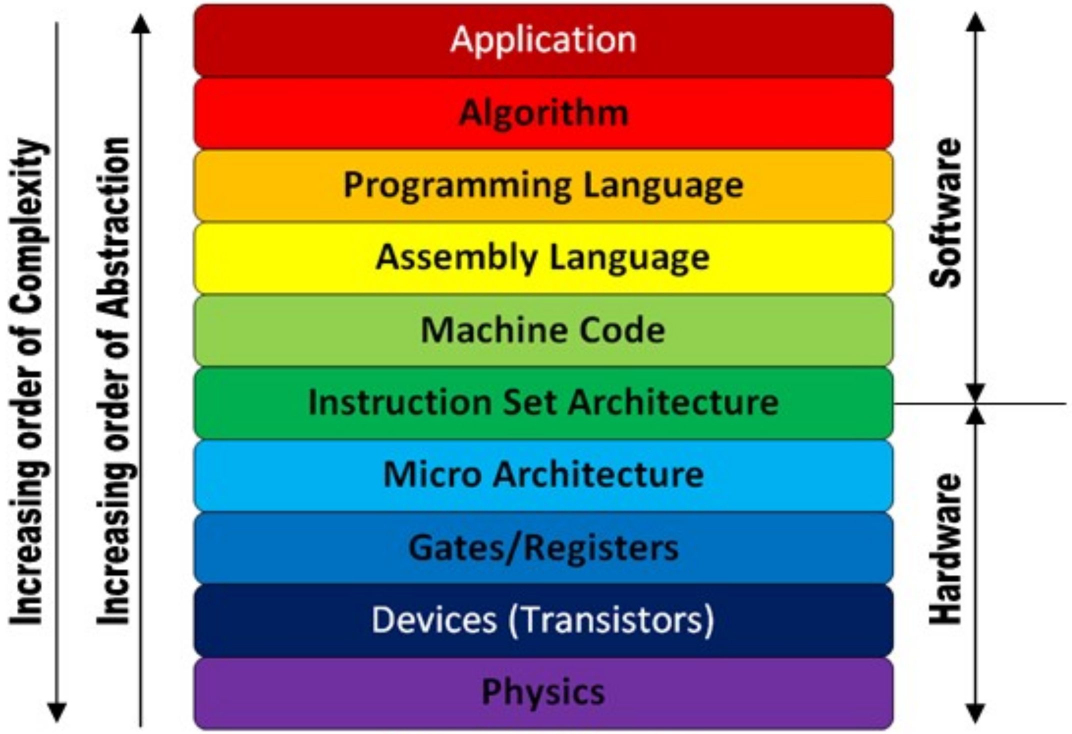
FULL-STACK?

Definition: the entirety of a computer ***system*** or application, comprising both the ***front end*** and the ***back end***.

STACK?

How do computers work, from the highest ***level of abstraction*** (programming) to the lowest level of abstraction (transistors)?





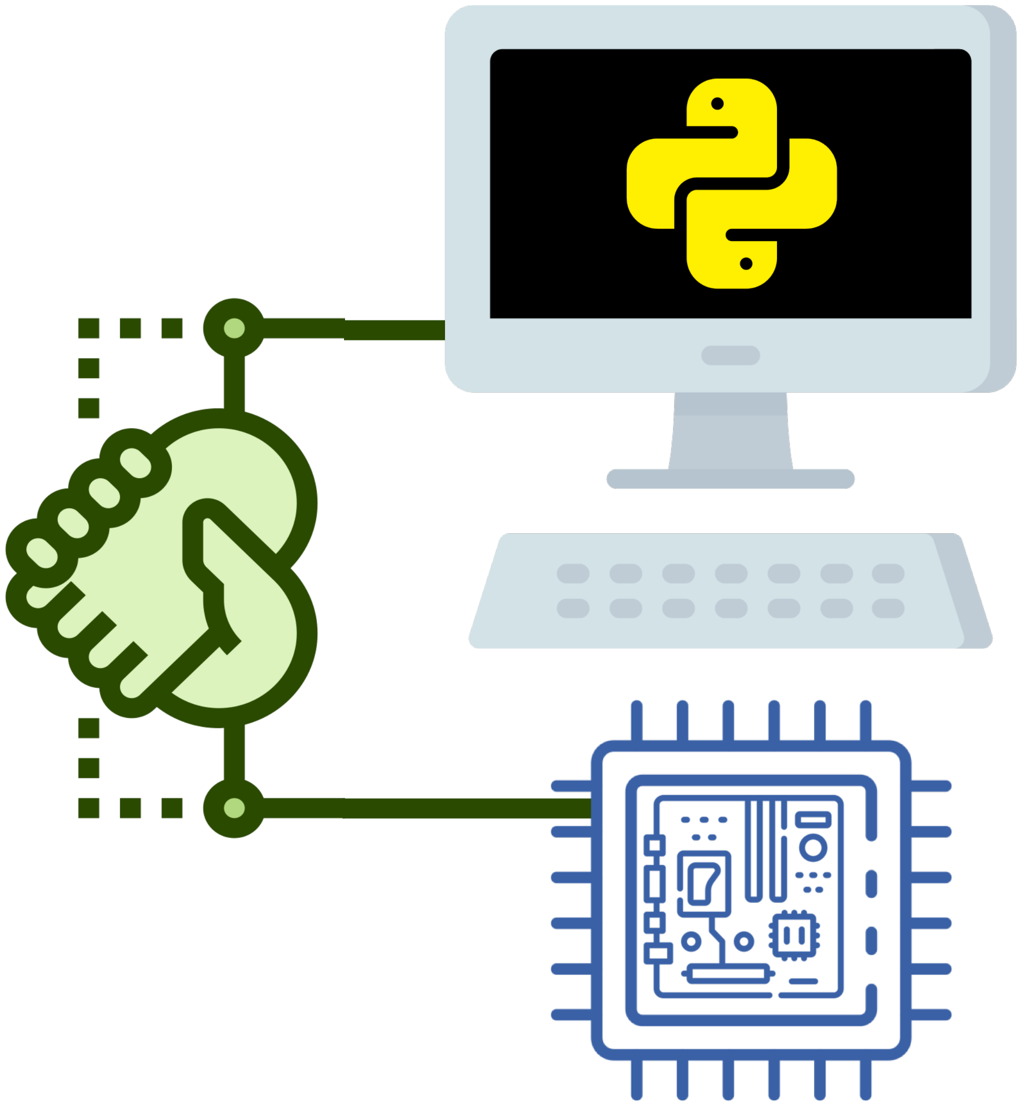
Layers of Abstraction

THE CLASSICAL COMPUTING STACK

CLASSICAL SOFTWARE

CLASSICAL COMPILER

CLASSICAL HARDWARE



HARDWARE AND SOFTWARE

https://youtu.be/xnyFYiK2rSY

COMPILERS

A ***compiler*** is a program that translates computer code written in a source programming language to a target programming language. Oftentimes, the target language is machine code.

PYTHON PROGRAM COMPILATION TO MACHINE CODE

QUBIT X QUBIT

The Coding School

THE CLASSICAL VS QUANTUM STACK

Classical	Quantum
	QUANTUM SOFTWARE
CLASSICAL SOFTWARE	CLASSICAL SOFTWARE
CLASSICAL COMPILER	QUANTUM PROGRAM COMPILER
CLASSICAL HARDWARE	CLASSICAL CONTROL HARDWARE
	QUANTUM HARDWARE

THE QUANTUM COMPUTING STACK

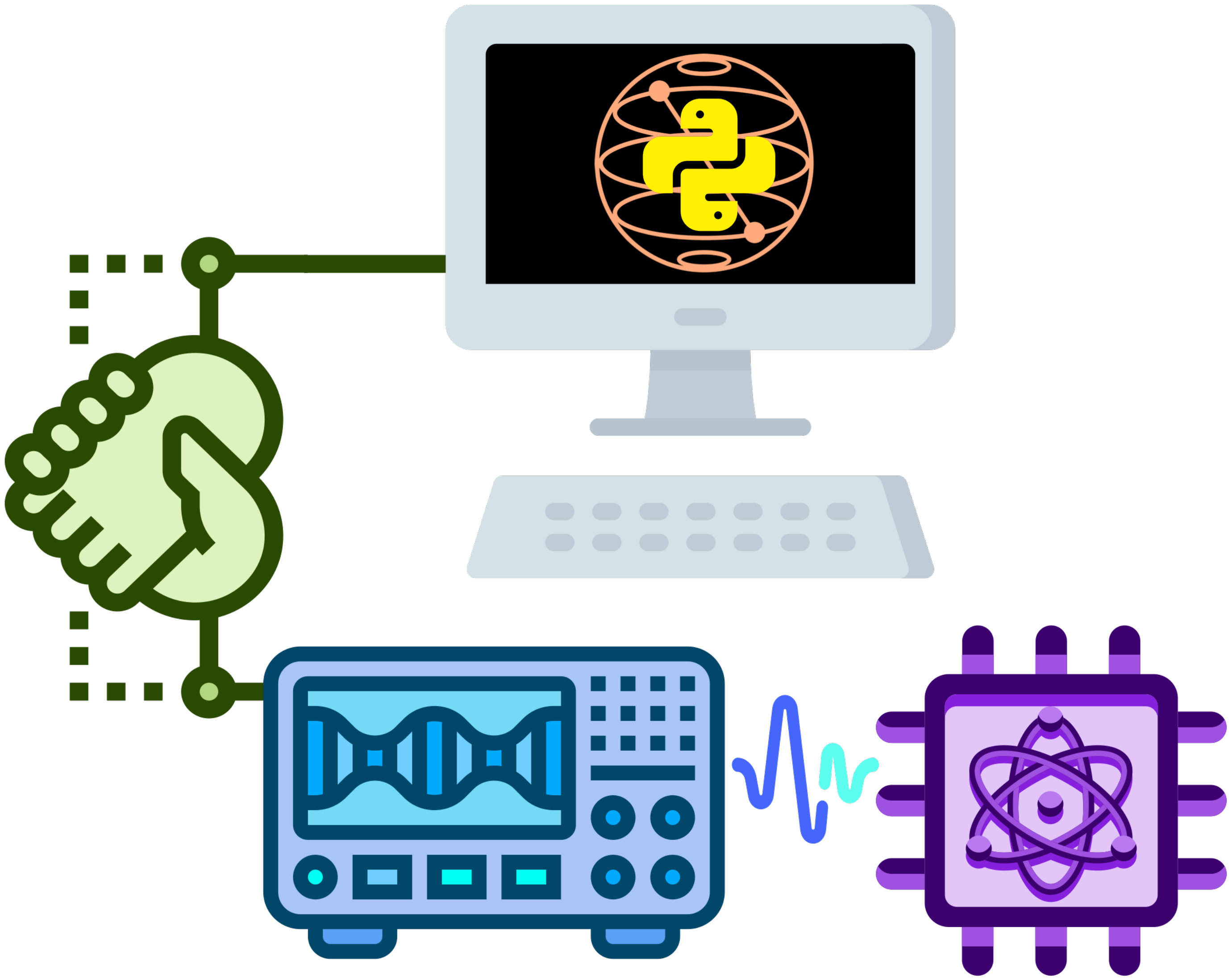
QUANTUM SOFTWARE

CLASSICAL SOFTWARE

QUANTUM PROGRAM COMPILER

CLASSICAL CONTROL HARDWARE

QUANTUM HARDWARE



THIS COURSE

QUANTUM SOFTWARE

CLASSICAL SOFTWARE

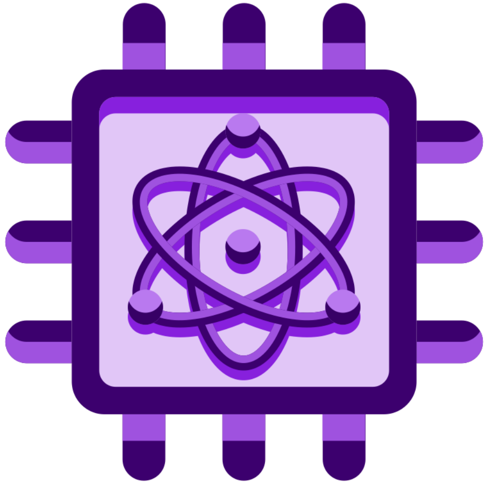
QUANTUM PROGRAM COMPILER

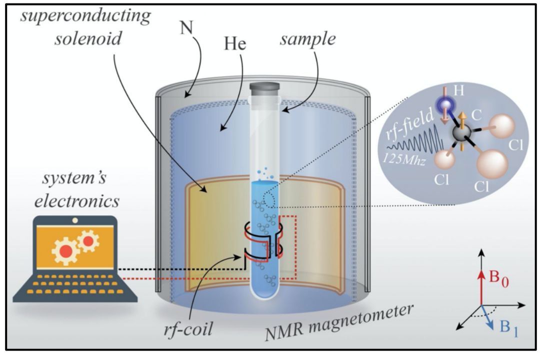
CLASSICAL CONTROL HARDWARE

QUANTUM HARDWARE

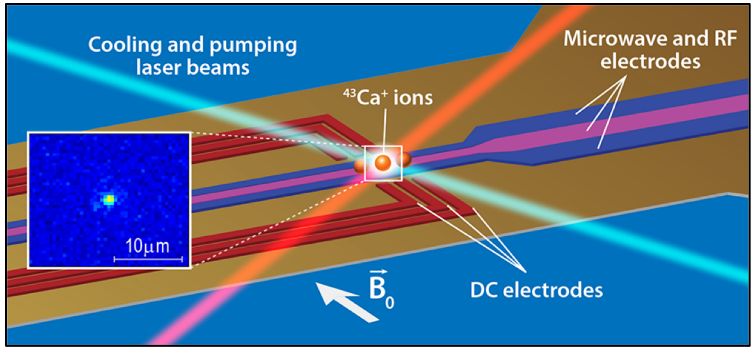
QUANTUM HARDWARE

What will be the '*quantum transistor*' ? Are we still in the '*quantum vacuum tube*' era ?

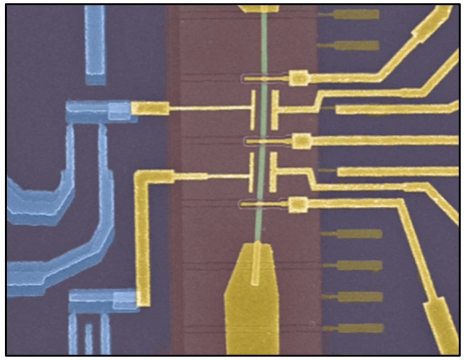




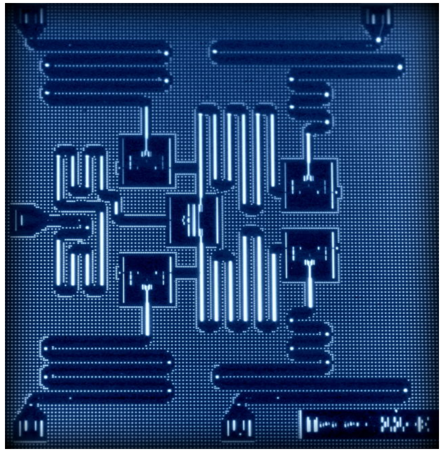
Nuclear Magnetic Resonance



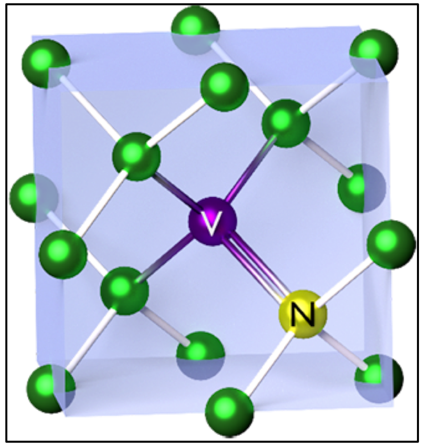
Trapped Ion



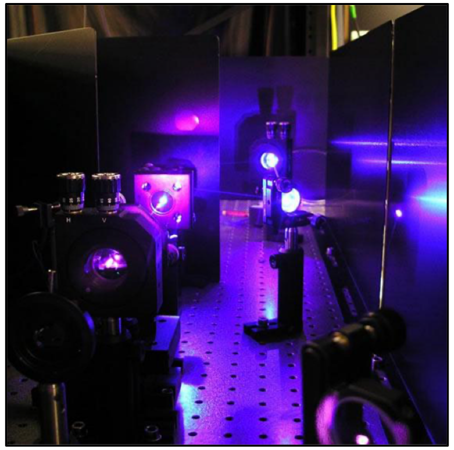
Majorana/Topological



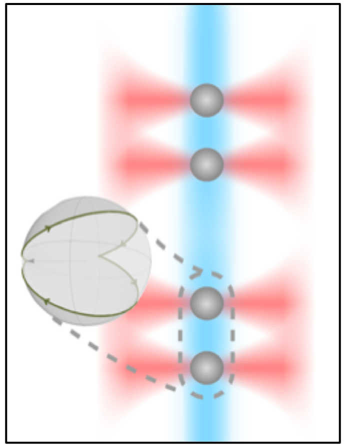
Superconducting



Diamond NV Centers

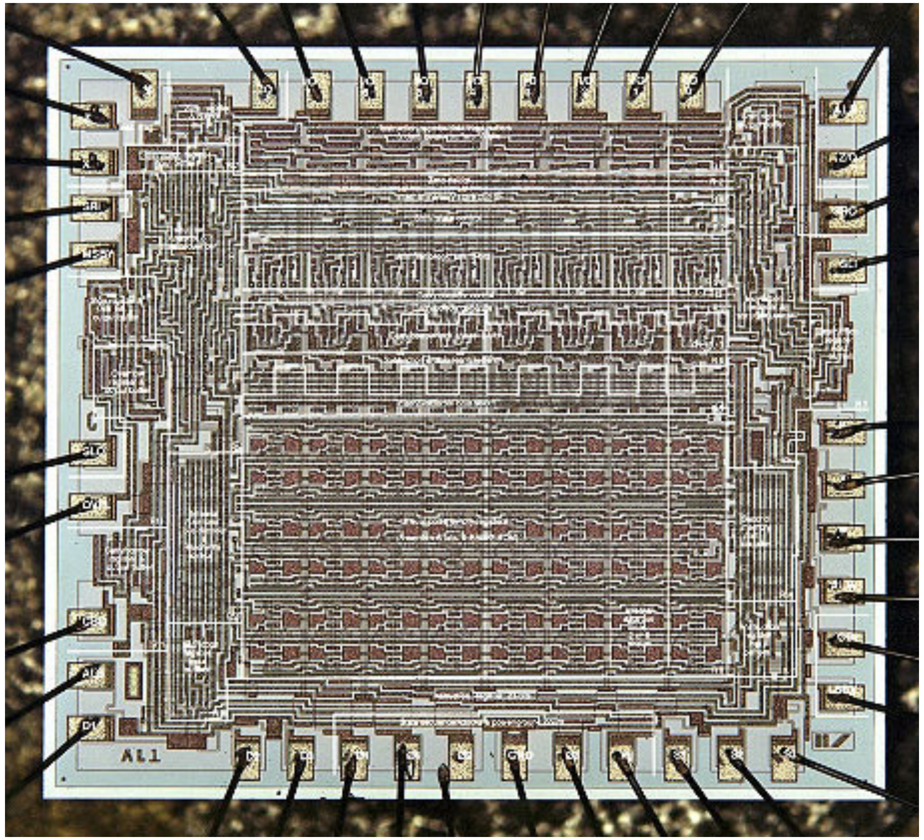


Photonics



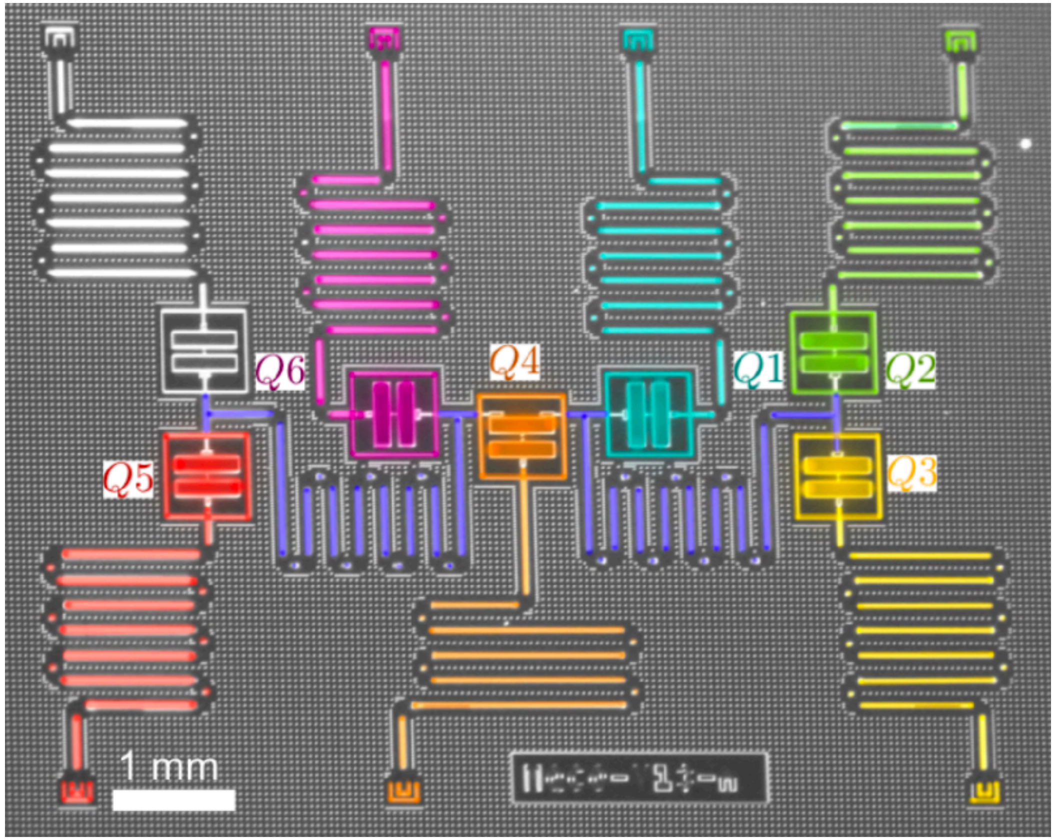
Neutral Atoms

QUANTUM HARDWARE



1969 Four-Phase Systems AL1 Processor.

(Classical)



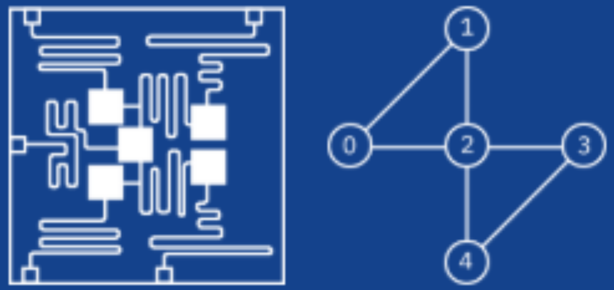
2017 IBM 7-Qubit Device

(Quantum)

QUANTUM HARDWARE

IBM Q 5 Tenerife [ibmqx4]

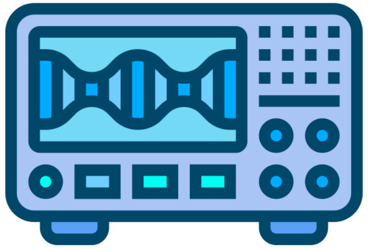
ACTIVE: USERS



Last Calibration: 2019-01-28 06:40:05

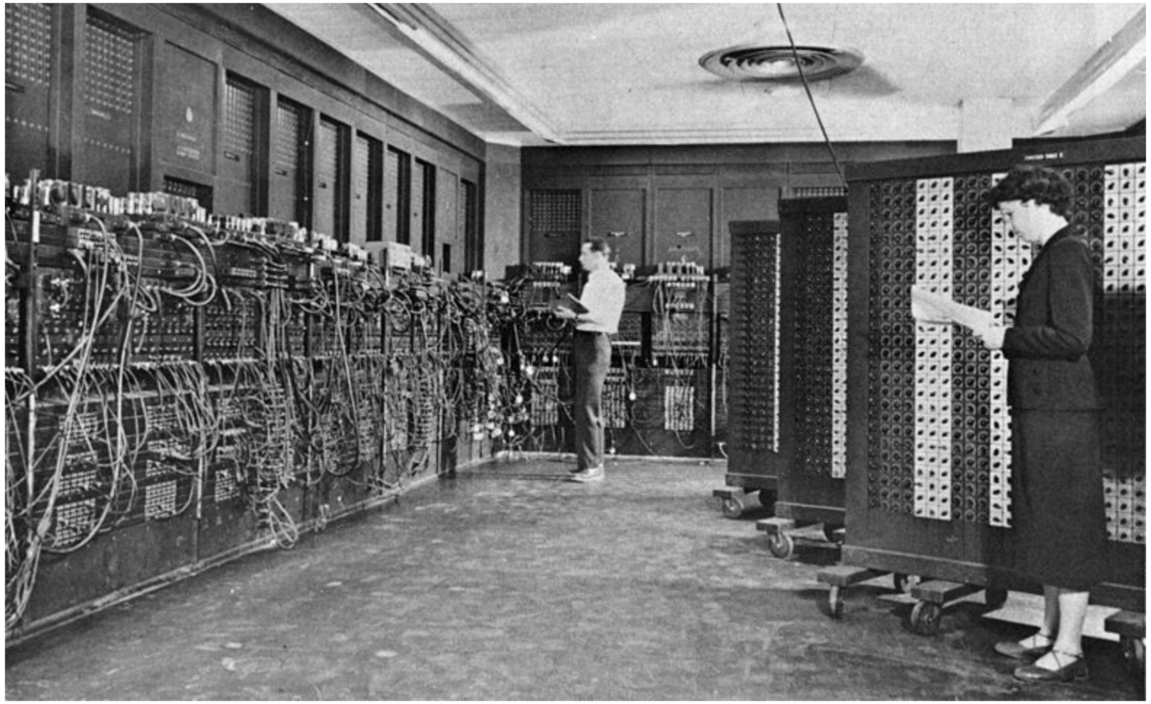
	Q0	Q1	Q2	Q3	Q4
Frequency (GHz)	5.25	5.30	5.35	5.43	5.18
T1 (µs)	53.90	43.70	44.20	48.70	54.90
T2 (µs)	39.60	26.20	31.00	19.60	13.70
Gate error (10^{-3})	0.60	0.94	1.37	1.97	1.46
Readout error (10^{-2})	6.30	9.30	3.20	2.80	5.80
		CX1_0	**CX2_0**	**CX3_2**	**CX4_2**
MultiQubit gate error (10^{-2})		3.32	2.82	7.64	6.43
			CX2_1	**CX3_4**	
			3.86	4.82	

CLASSICAL CONTROL HARDWARE



Superconducting qubits are controlled via microwave pulse sequences. Different pulse shapes and lengths correspond to different quantum gates.

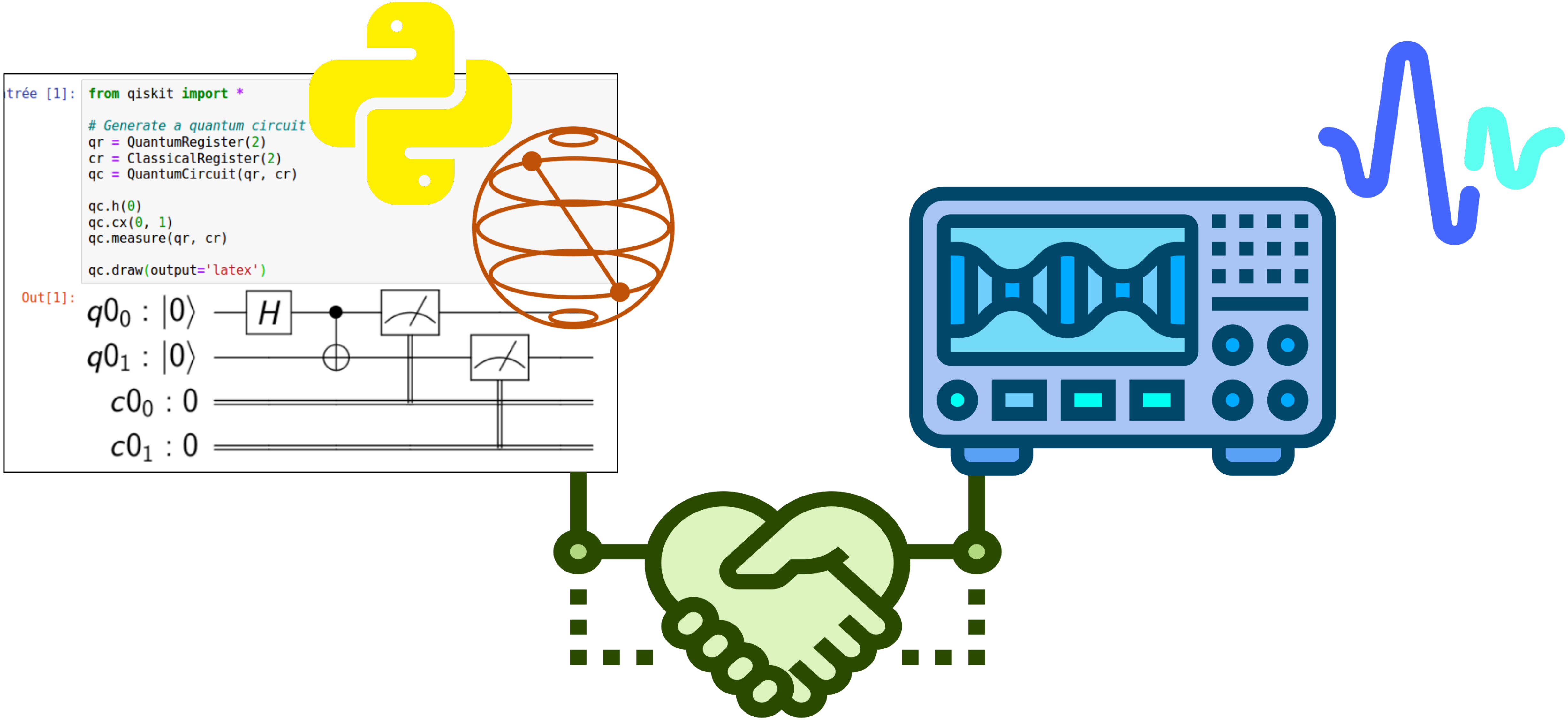
Quantum Engineer's Guide to Superconducting Qubits [arXiv: 1904.06560]



ENIAC (first classical computer)
1950

MIT EQuS Control Room (for quantum computers)
2020

QUANTUM PROGRAM COMPILER



CLASSICAL SOFTWARE

PYTHON

VARIABLE INSTANTIATION

```
string = “QxQ”
integer = 5
float = 6.7
boolean = True
```

DATA STRUCTURES

```
list = [4, ’a’, False, -20.9]
```

FUNCTIONS

```
def add(a, b, c):
    return a + b + c
```

FOR LOOPS

```
for a in [‘hello’, ‘world’, ‘!’]:
    print(a)
```

CLASSICAL SOFTWARE

Python

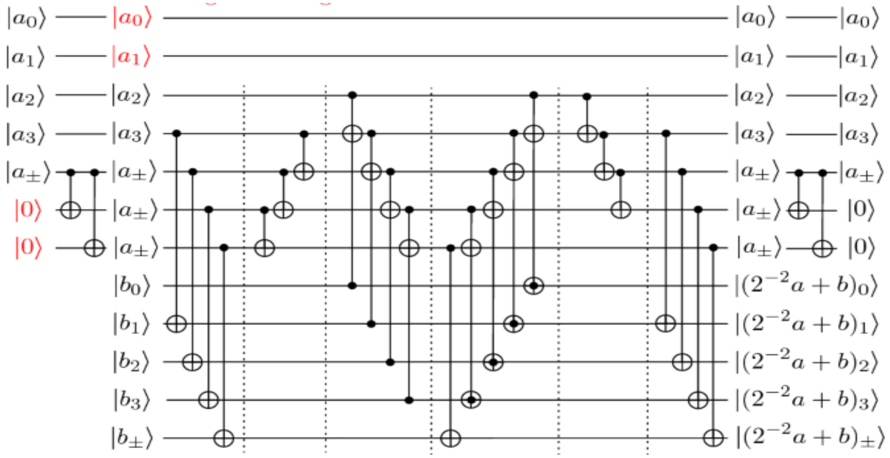
SciPy and Numpy are Python **libraries** for scientific and numerical computations.

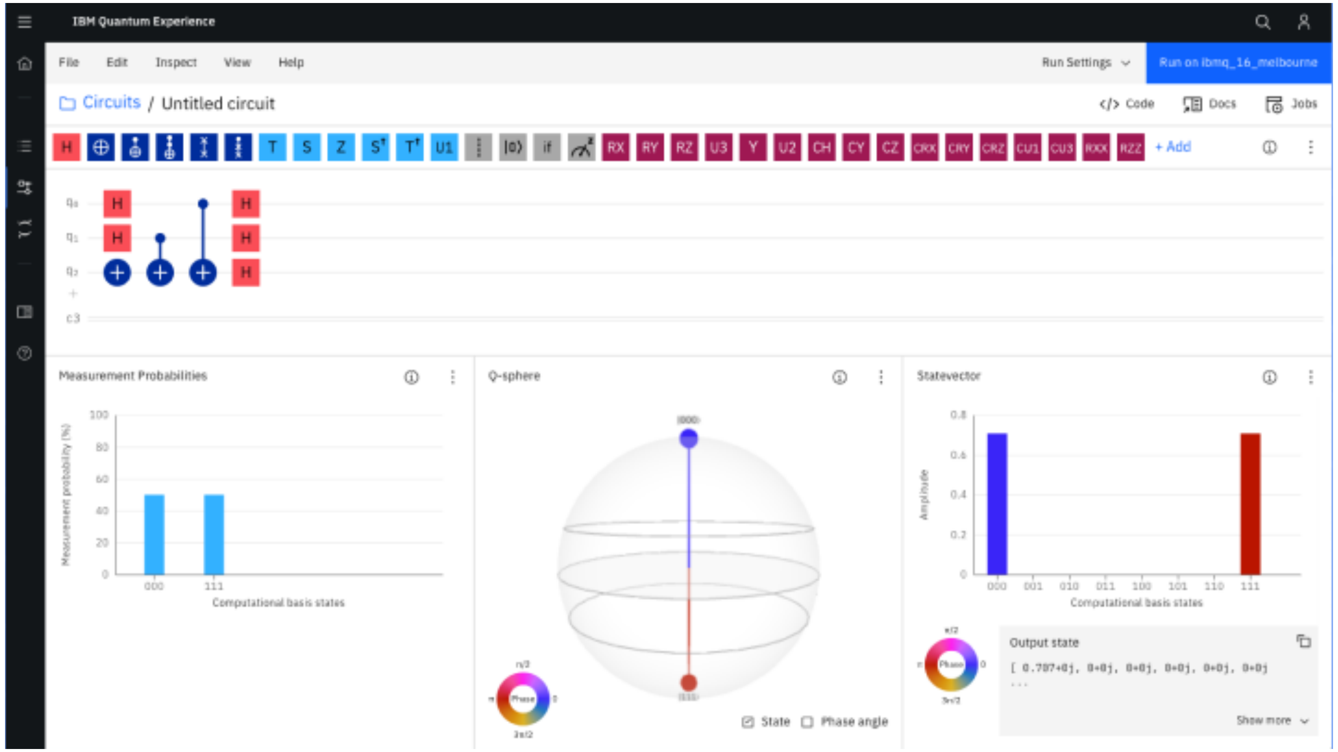
SciPy

NumPy

QUANTUM SOFTWARE

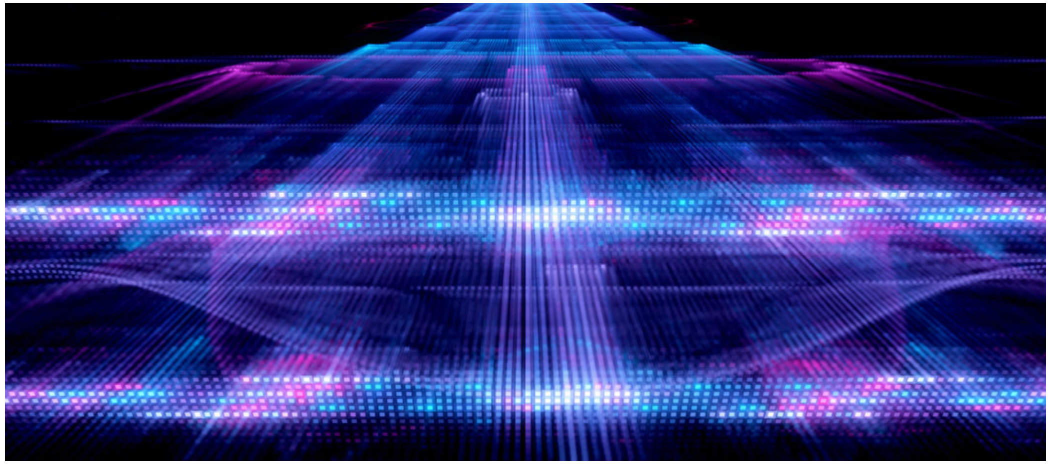
How can we implement large, complex circuits?



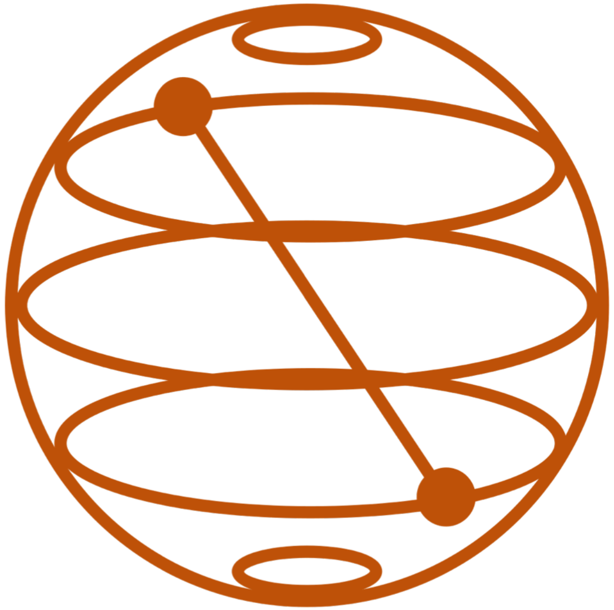


IBM QUANTUM EXPERIENCE

What about when we have large-scale quantum computers?



QUANTUM SOFTWARE

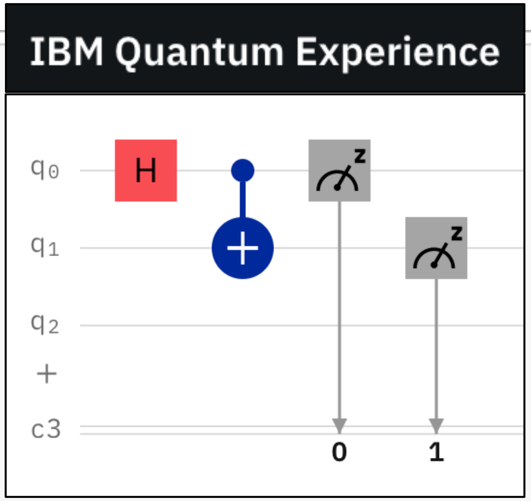


IBM QISKIT

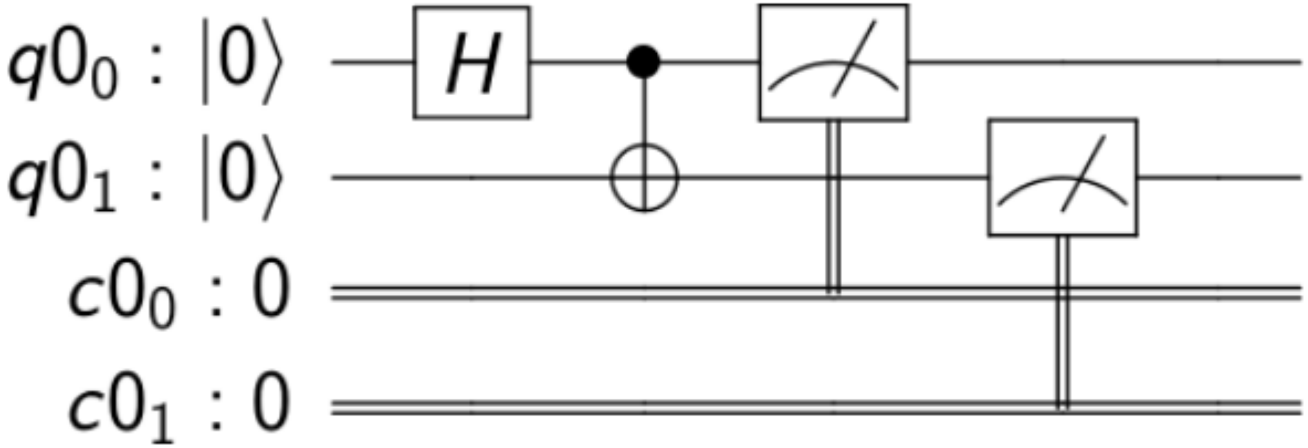
a Python library for quantum computing!

trée [1]:

```
from qiskit import *

# Generate a quantum circuit
qr = QuantumRegister(2)
cr = ClassicalRegister(2)
qc = QuantumCircuit(qr, cr)

qc.h(0)
qc.cx(0, 1)
qc.measure(qr, cr)

qc.draw(output='latex')
```



Out[1]:



THE QUANTUM COMPUTING STACK

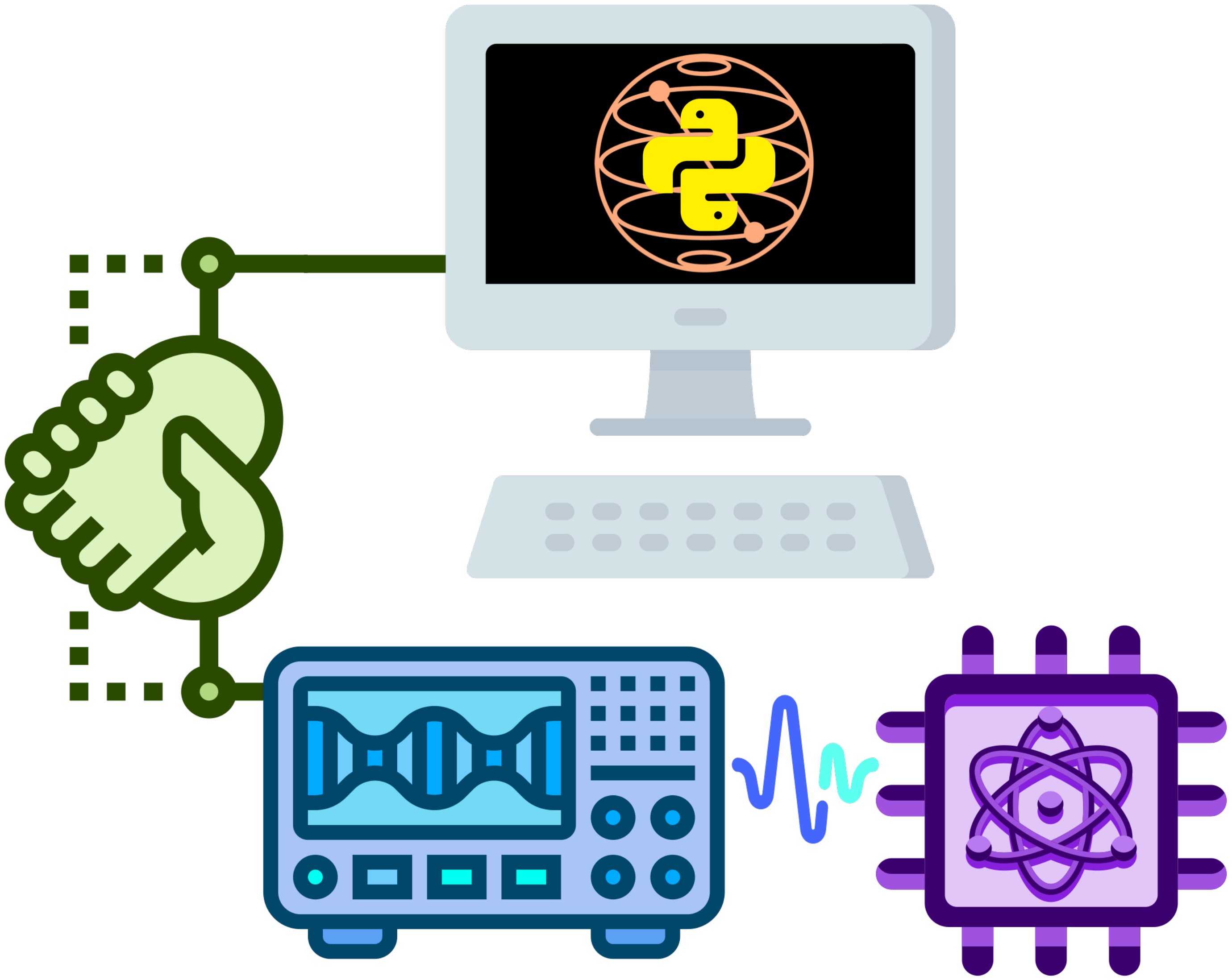
QUANTUM SOFTWARE

CLASSICAL SOFTWARE

QUANTUM PROGRAM COMPILER

CLASSICAL CONTROL HARDWARE

QUANTUM HARDWARE



BREAK TIME!

SEMESTER 1

MATH REVIEW

WHY ALL THE MATH?

Quantum mechanics is a beautiful generalization of the laws of probability: a generalization based on the 2-norm rather than the 1-norm, and on complex numbers rather than nonnegative real numbers. It can be studied completely separately from its applications to physics (and indeed, doing so provides a good starting point for learning the physical applications later). This generalized probability theory leads naturally to a new model of computation – the quantum computing model – that challenges ideas about computation once considered a priori, and that theoretical computer scientists might have been driven to invent for their own purposes, even if there were no relation to physics. In short, while quantum mechanics was invented a century ago to solve technical problems in physics, today it can be fruitfully explained from an extremely different perspective: as part of the history of ideas, in math, logic, computation, and philosophy, about the limits of the knowable.

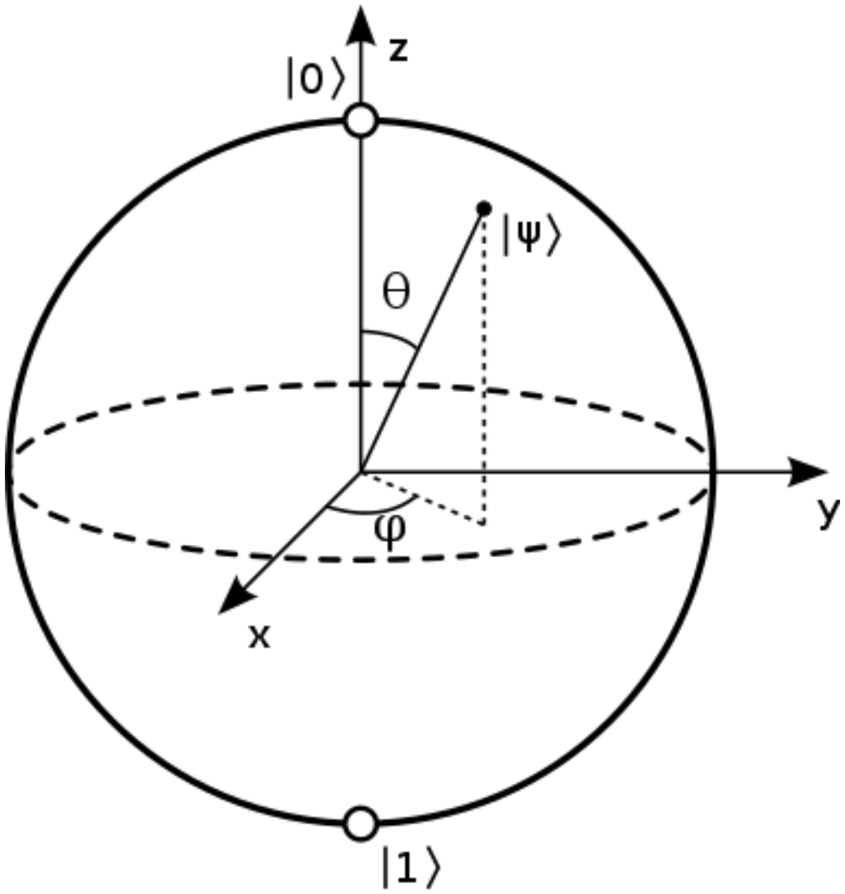
- Professor Scott Aaronson (UT Austin)

(excerpt from *Quantum Computing Since Democritus*)

Image Source: ETH Zurich

QUANTUM BIG IDEA

Quantum states are represented mathematically as ***complex vectors****!*



LECTURE #3 - VECTORS

Vector Representation:

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

Every vector has a direction and a magnitude!

2D Vector Direction:

$$\angle \vec{v} = \tan^{-1}\left(\frac{v_y}{v_x}\right)$$

Vector Magnitude:

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

Vector Addition:

$$\vec{a} + \vec{b} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{pmatrix}$$

Vector-Scalar Multiplication:

$$c * \vec{v} = \begin{pmatrix} c * v_1 \\ c * v_2 \\ \vdots \\ c * v_n \end{pmatrix}$$

LECTURE #3 – COMPLEX NUMBERS

The Imaginary Unit:

$$i = \sqrt{-1}$$

$$i^2 = -1$$

Complex Number:

$$a + i\,b$$

Complex Conjugation:

$$\overline{(a + i\,b)} = (a + ib)^* = (a - ib)$$

Complex Modulus:

$$|a + ib| = \sqrt{a^2 + b^2}$$

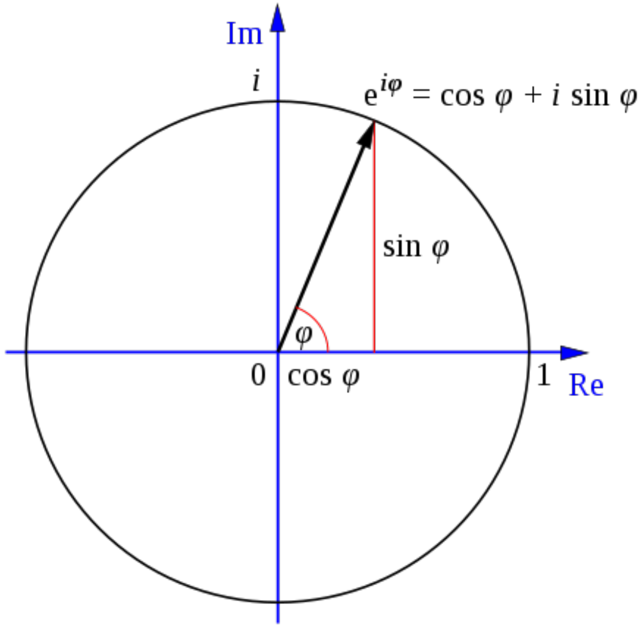
Complex Addition:

$$(a + i\,b) + (c + id) = (a + c) + i(b + d)$$

Complex Mult.:

$$(a + i\,b) * (c + id) = (ac - bd) + i(ad + bc)$$

LECTURE #3 – COMPLEX EXPONENTIALS



Euler's formula:

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

$$r = |z| = \sqrt{x^2 + y^2}$$

(vector radius)

Polar representation of complex numbers!

$$z = x + iy = |z|(\cos\varphi + i\sin\varphi) = re^{i\varphi}$$

$$\varphi = \tan^{-1}\left(\frac{y}{x}\right)$$

(vector angle)

Multiplication:

$$e^{i\varphi}\, e^{i\theta} = e^{i(\varphi+\theta)}$$

Trig Relations:

$$\cos\varphi = \frac{e^{i\varphi} + e^{-i\varphi}}{2}$$

$$\sin\varphi = \frac{e^{i\varphi} - e^{-i\varphi}}{2i}$$

Complex Conjugation:

$$\overline{e^{i\varphi}} = e^{-i\varphi}$$

Modulus

$$|z| = \left|re^{i\varphi}\right| = |r|$$

VECTOR SHAPE:

$$(\text{\# rows} \times \text{\# cols})$$

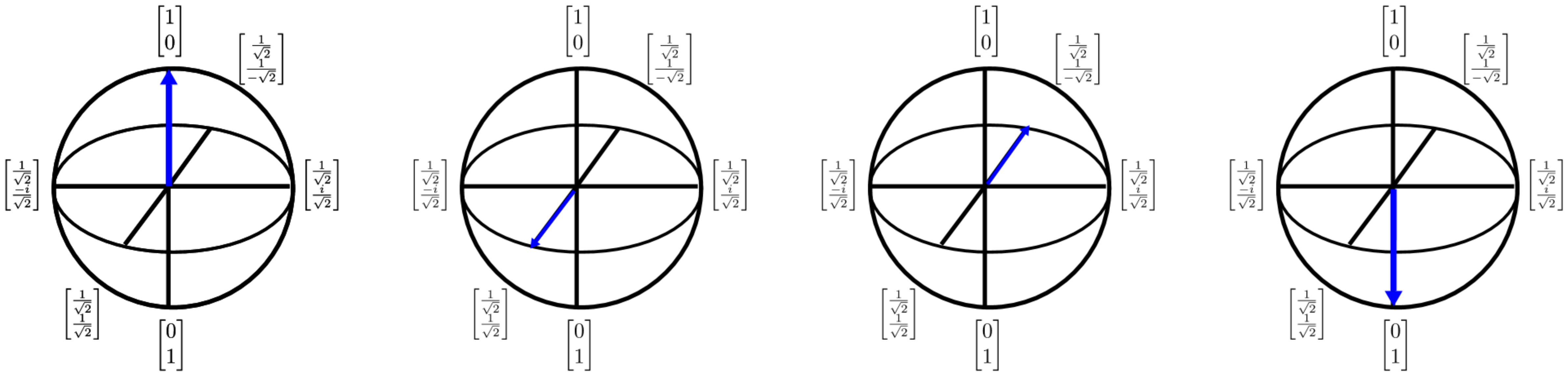
	COLUMN VECTOR	ROW VECTOR
	$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$	$\begin{pmatrix} r_1 & r_2 & \cdots & r_m \end{pmatrix}$
SHAPE:	$(n \times 1)$	$(1 \times m)$

If $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$, then its ***transpose*** is $\vec{v}^T = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix}$

If $\vec{w} = \begin{pmatrix} w_1 & w_2 & \cdots & w_n \end{pmatrix}$, then its transpose is $\vec{w}^T = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$

QUANTUM BIG IDEA

*The **inner product** enables us to compare quantum states and determine their **orthogonality**!*



$$\langle \psi | \phi \rangle$$

The ***inner product*** is:

$$\langle \vec{v}, \vec{w} \rangle = \vec{v}\,\vec{w}^T = \sum_{i=1}^{n} v_i w_i$$

1. **vector x vector** to **scalar** mapping
2. tool for calculating vector **magnitude**
3. tool for vector **normalization**
4. tool for **geometrically comparing** vectors
5. tool for determining vector **orthogonality**

MAGNITUDE:

$$\langle \vec{v}, \vec{v} \rangle = \|\vec{v}\|^2$$

NORMALIZATION:

$$\frac{\vec{v}}{\sqrt{\langle \vec{v}, \vec{v} \rangle}} = \frac{\vec{v}}{\|\vec{v}\|}$$

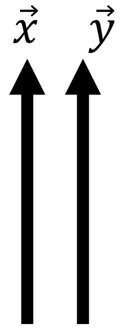
GEOMETRIC COMPARISON & VECTOR ORTHOGONALITY:

$$\langle \vec{x}, \vec{y} \rangle = \|\vec{x}\|\|\vec{y}\| \cos(\theta)$$

where $\theta = \angle(\vec{x}, \vec{y})$ is the angle between $\vec{x}$ and $\vec{y}$

$$\theta = \cos^{-1}\left(\frac{\langle \vec{x}, \vec{y} \rangle}{\|\vec{x}\|\|\vec{y}\|}\right)$$

PARALLEL

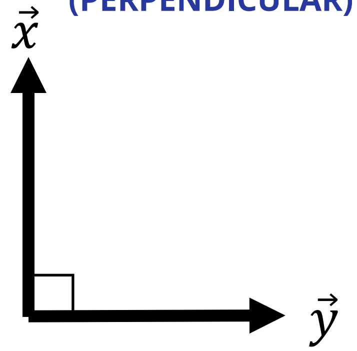


$\theta = 0°$

$\langle \vec{x}, \vec{y} \rangle =$

ORTHOGONAL

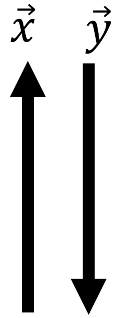
(PERPENDICULAR)



$\theta = 90°$

$\langle \vec{x}, \vec{y} \rangle =$

ANTI-PARALLEL

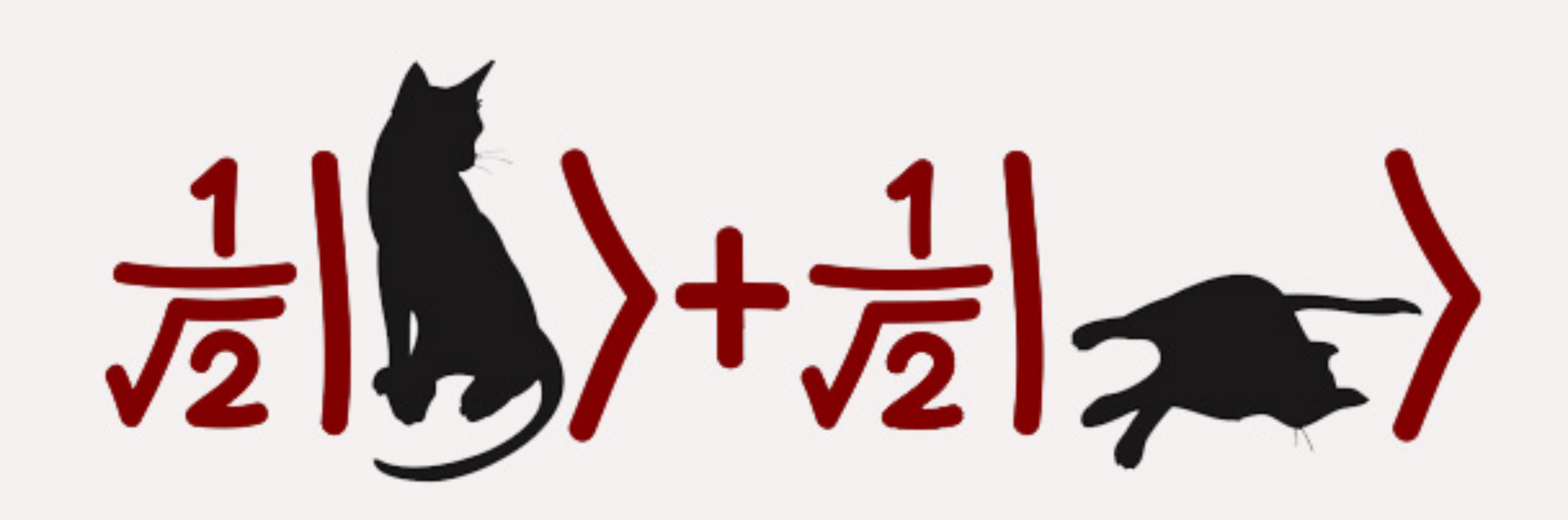


$\theta = 180°$

$\langle \vec{x}, \vec{y} \rangle =$

QUANTUM BIG IDEA

*A **linear combination** of quantum states is a **superposition**!*



LINEAR COMBINATION OF FRUITS :

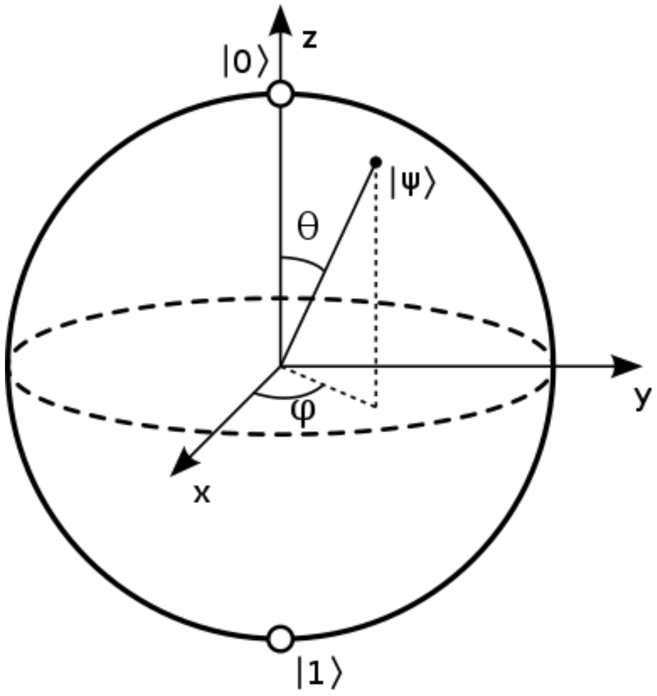
 $=\ 6\ *$  $+10\ *$  $+6\ *$  $+\cdots$

LINEAR COMBINATION OF VECTORS :

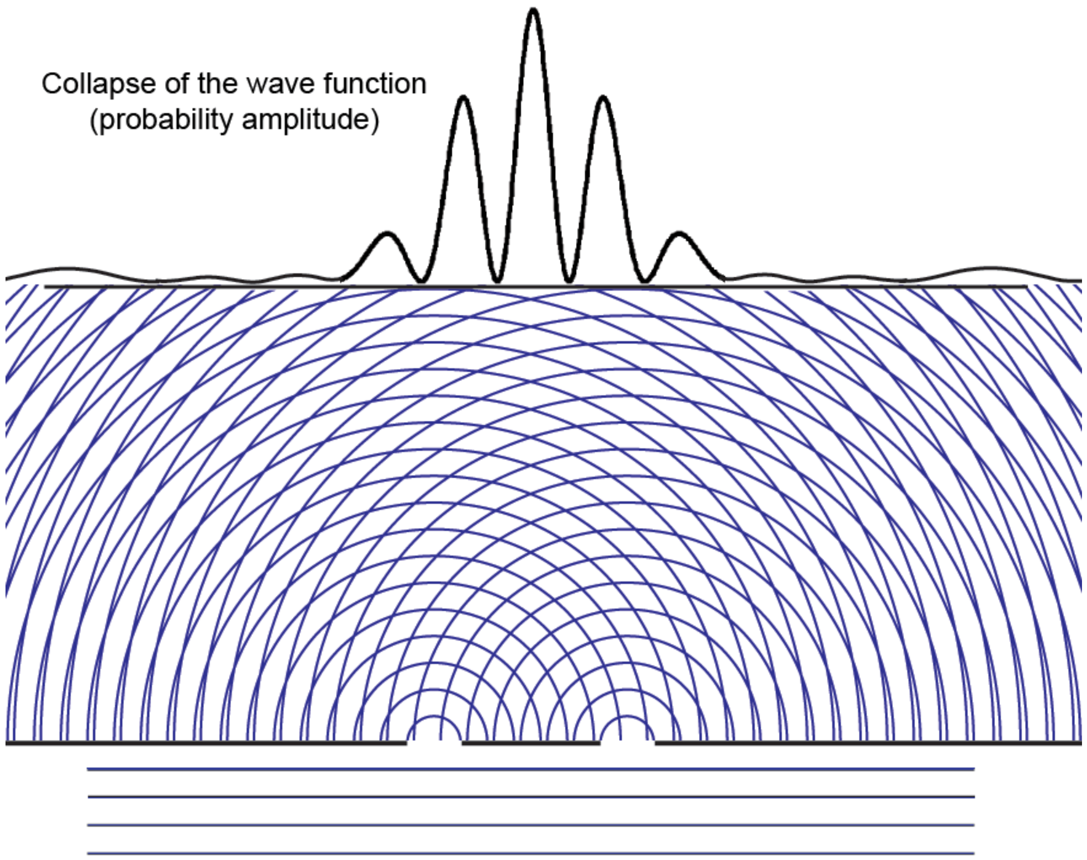
$$\vec{v} = a_1\vec{v}_1 + a_2\vec{v}_2 + \cdots + a_n\vec{v}_n = \sum_{i=1}^{n} a_i\vec{v}_i$$

QUANTUM BIG IDEA

The coefficients in a quantum superposition are ***probability amplitudes****!*



$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$



LECTURE #4 – COMPLEX VECTOR MANIPULATION

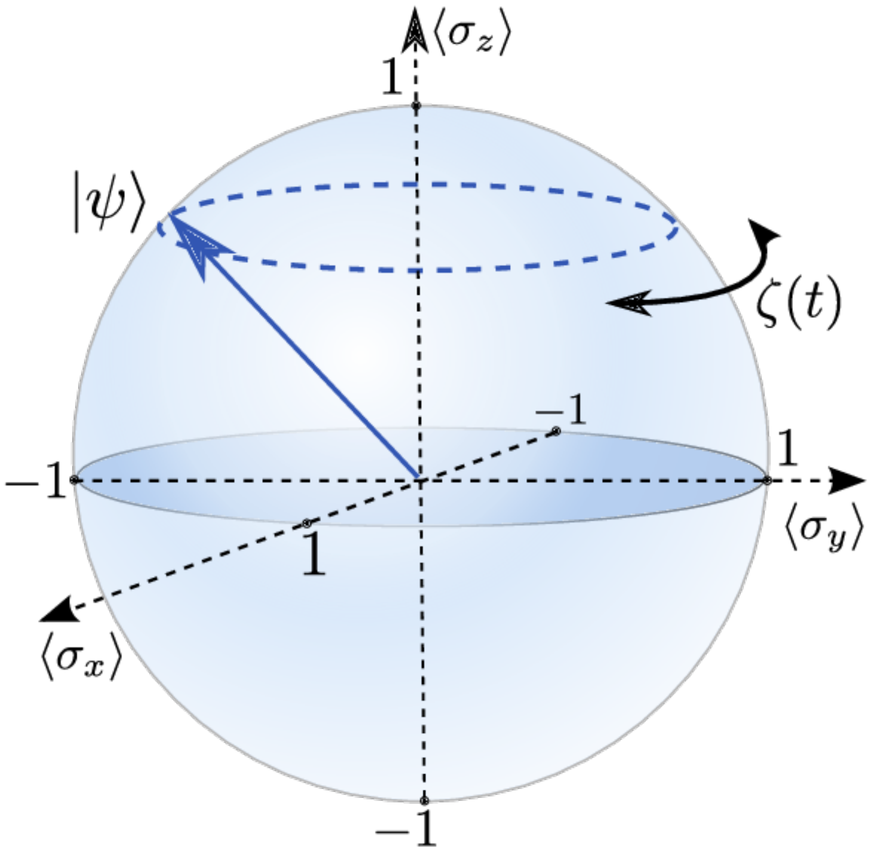
CONJUGATE TRANSPOSE: $\vec{v}^{\dagger} = (\vec{v}^{T})^{*} = (\vec{v}^{*})^{T}$

To avoid negative and complex magnitudes (not physically possible!), we use the ***complex inner product***:

$$\langle \vec{v}, \vec{w} \rangle = \vec{v}^{\dagger}\vec{w} = v_1^* w_1 + \cdots + v_n^* w_n = \sum_{i=1}^{n} v_i^* w_i$$

QUANTUM BIG IDEA

*Quantum operations perform the computation in a quantum computer and are represented mathematically as **matrices**!*



An $(\mathrm{n} \times \mathrm{m})$ matrix is written as,

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

MATRIX SHAPE:

$$(\ \#\ \mathrm{rows} \times \#\ \mathrm{cols}\ )$$

$$\boldsymbol{X} = \begin{pmatrix} \boxed{x_{11} \quad x_{12} \quad \cdots \quad x_{1m}} \\ \boxed{x_{21} \quad x_{22} \quad \cdots \quad x_{2m}} \\ \vdots \quad\quad \vdots \quad\quad \ddots \quad\quad \vdots \\ \boxed{x_{n1} \quad x_{n2} \quad \cdots \quad x_{nm}} \end{pmatrix} \qquad \boldsymbol{X} = \begin{pmatrix} \boxed{\begin{matrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{matrix}} & \boxed{\begin{matrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{matrix}} & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \boxed{\begin{matrix} x_{1m} \\ x_{2m} \\ \vdots \\ x_{nm} \end{matrix}} \end{pmatrix}$$

MATRIX TRANSPOSE:

$$\boldsymbol{X}^T = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1m} & x_{2m} & \cdots & x_{nm} \end{pmatrix}$$

MATRIX CONJUGATE TRANSPOSE:

$$\boldsymbol{X}^\dagger = \begin{pmatrix} x_{11}^* & x_{21}^* & \cdots & x_{n1}^* \\ x_{12}^* & x_{22}^* & \cdots & x_{n2}^* \\ \vdots & \vdots & \ddots & \vdots \\ x_{1m}^* & x_{2m}^* & \cdots & x_{nm}^* \end{pmatrix}$$

LECTURE #4 – MATRICES

MATRIX ADDITION:

$$A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2m} + b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \cdots & a_{nm} + b_{nm} \end{pmatrix}$$

MATRIX-VECTOR MULTIPLICATON:

$$A\vec{x} = \begin{pmatrix} \langle \vec{a}_1, \vec{x} \rangle \\ \langle \vec{a}_2, \vec{x} \rangle \\ \vdots \\ \langle \vec{a}_n, \vec{x} \rangle \end{pmatrix}$$

SCALAR-MATRIX MULTIPLICATION:

$$c * A = \begin{pmatrix} c * a_{11} & c * a_{12} & \cdots & c * a_{1m} \\ c * a_{21} & c * a_{22} & \cdots & c * a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c * a_{n1} & c * a_{n2} & \cdots & c * a_{nm} \end{pmatrix}$$

MATRIX-MATRIX MULTIPLICATION:

$$AB = \begin{pmatrix} \langle \vec{a}_1, \vec{b}_1 \rangle & \langle \vec{a}_1, \vec{b}_2 \rangle & \cdots & \langle \vec{a}_1, \vec{b}_k \rangle \\ \langle \vec{a}_2, \vec{b}_1 \rangle & \langle \vec{a}_2, \vec{b}_2 \rangle & \cdots & \langle \vec{a}_2, \vec{b}_k \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{a}_n, \vec{b}_1 \rangle & \langle \vec{a}_n, \vec{b}_2 \rangle & \cdots & \langle \vec{a}_n, \vec{b}_k \rangle \end{pmatrix}$$

MATRIX/VECTOR MULTIPLICATION SHAPE CHECK: $(\,n \times m\,)(\,m \times k\,)$

IDENTITY MATRIX:

$$\mathbb{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

$$\boldsymbol{X}\,\mathbb{I} = \boldsymbol{X}$$
$$\vec{x}\,\mathbb{I} = \vec{x}$$

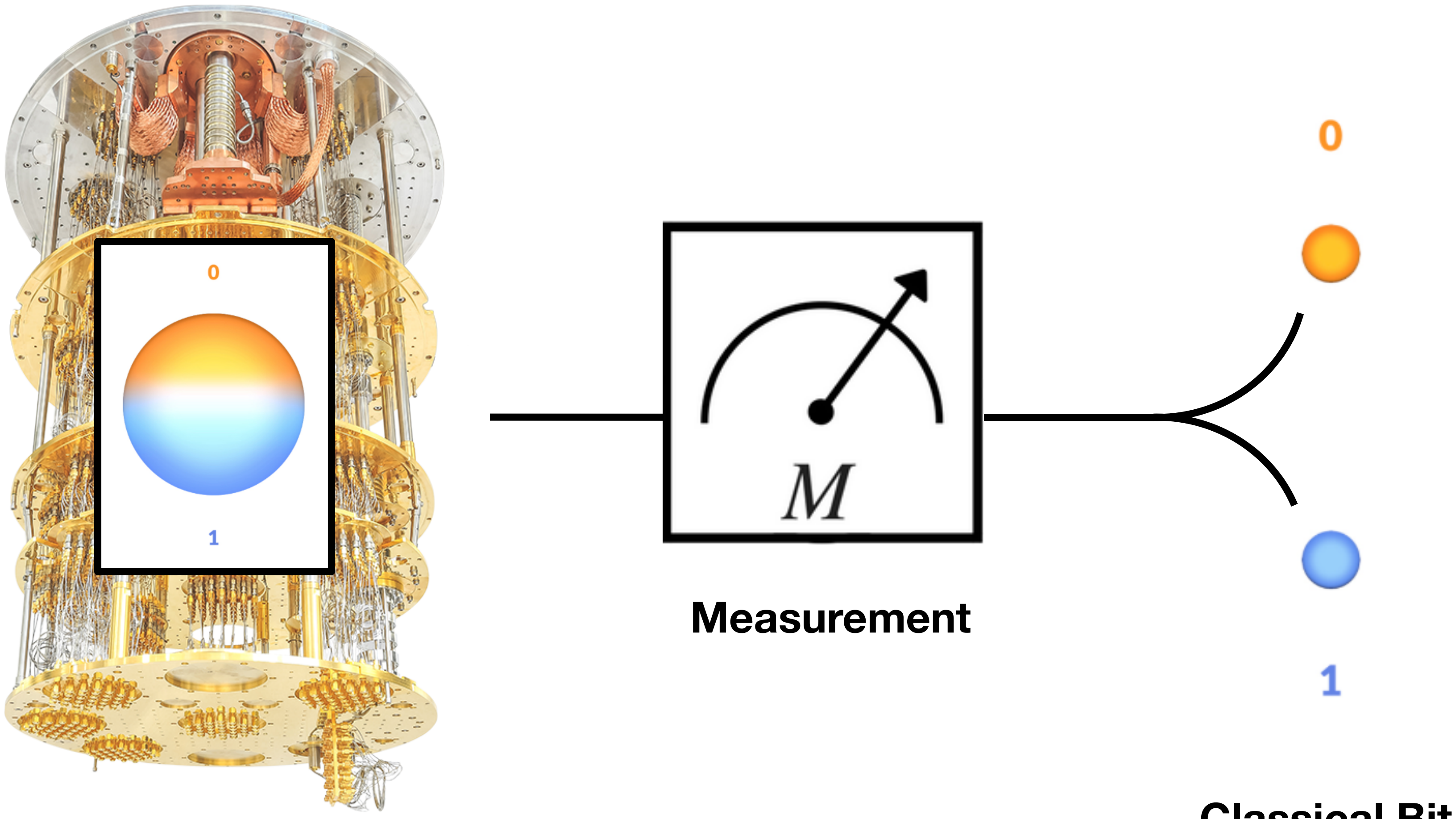
$$\mathbb{I}\,\boldsymbol{Y} = \boldsymbol{Y}$$
$$\mathbb{I}\,\vec{y} = \vec{y}$$

MATRIX INVERSION:

$$\boldsymbol{X}\boldsymbol{X}^{-1} = \boldsymbol{X}^{-1}\boldsymbol{X} = \mathbb{I}$$

QUANTUM BIG IDEA

*Quantum measurement is **probabilistic**!*



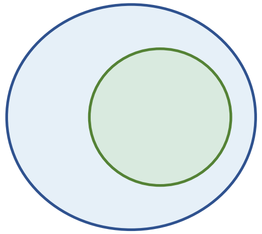
Quantum State

Classical Bit

LECTURE #5 – SETS

Comparing sets:

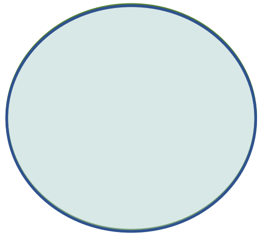
$A \subset B$



Set A is a ***subset*** of set B.

$A \subseteq B$

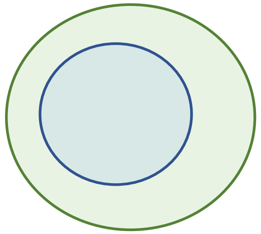
$A = B$



Set A is ***equal*** to set B.

$A \supseteq B$

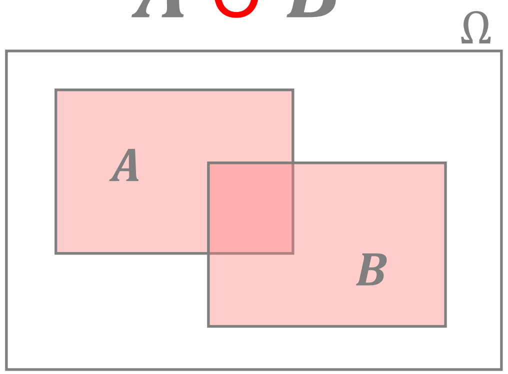
$A \supset B$



Set A is a ***superset*** of set B.

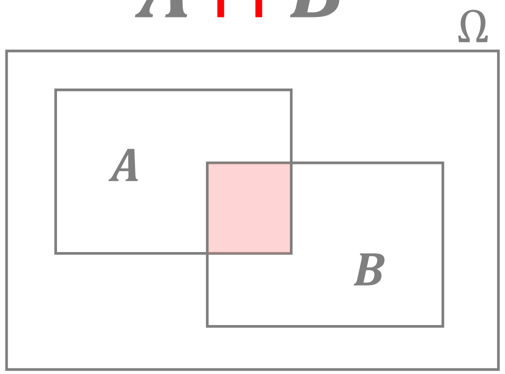
Operations on sets:

$A \cup B$



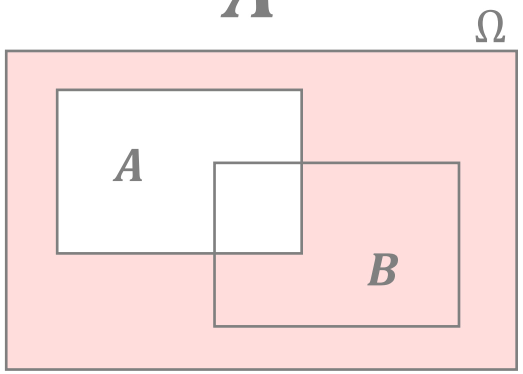
UNION (OR)

$A \cap B$



INTERSECT (AND)

A^c

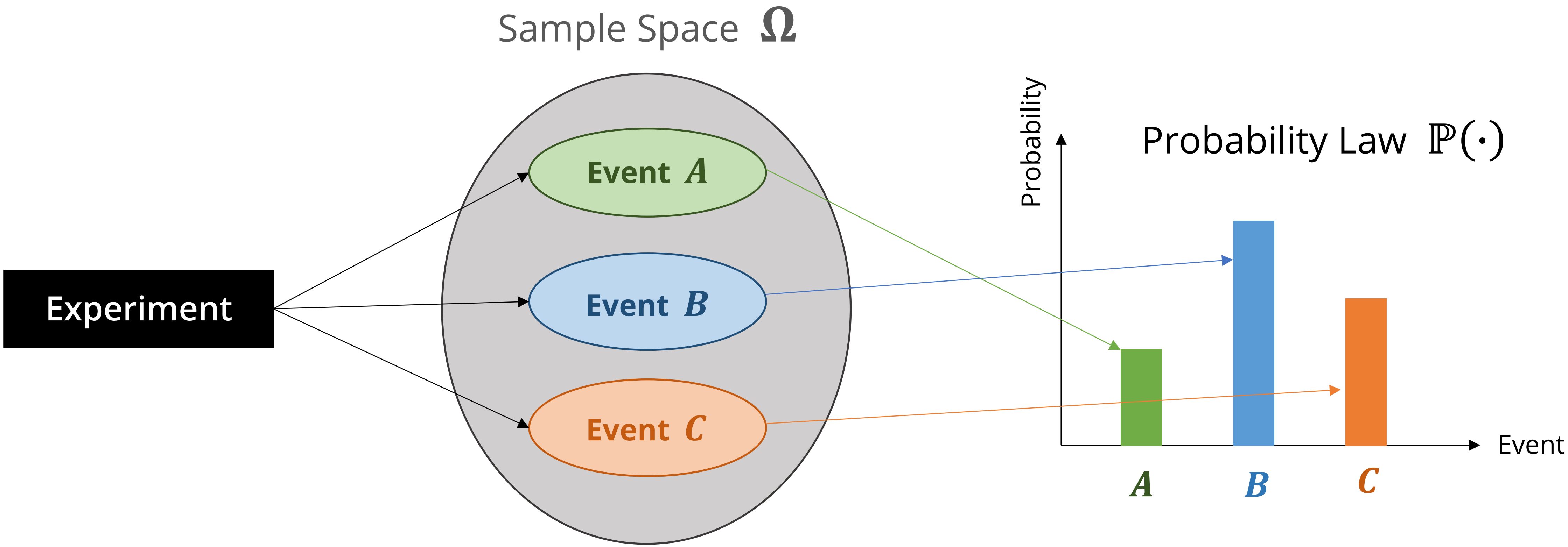


COMPLEMENT (SUBTRACT)

LECTURE #5 – PROBABILISTIC MODEL

A **probabilistic model** is a way to mathematically describe an unknown situation.

There are two key components: (1) **sample space** and (2) **probability law**.



LECTURE #5 – AXIOMS OF PROBABILITY

Every probability law must satisfy the following axioms:

1. **NONNEGATIVITY** : The probability of every event in the sample space must be greater than or equal to zero.

$$\mathbb{P}(\boldsymbol{A}) \geq 0, \text{ for every event } \boldsymbol{A}$$

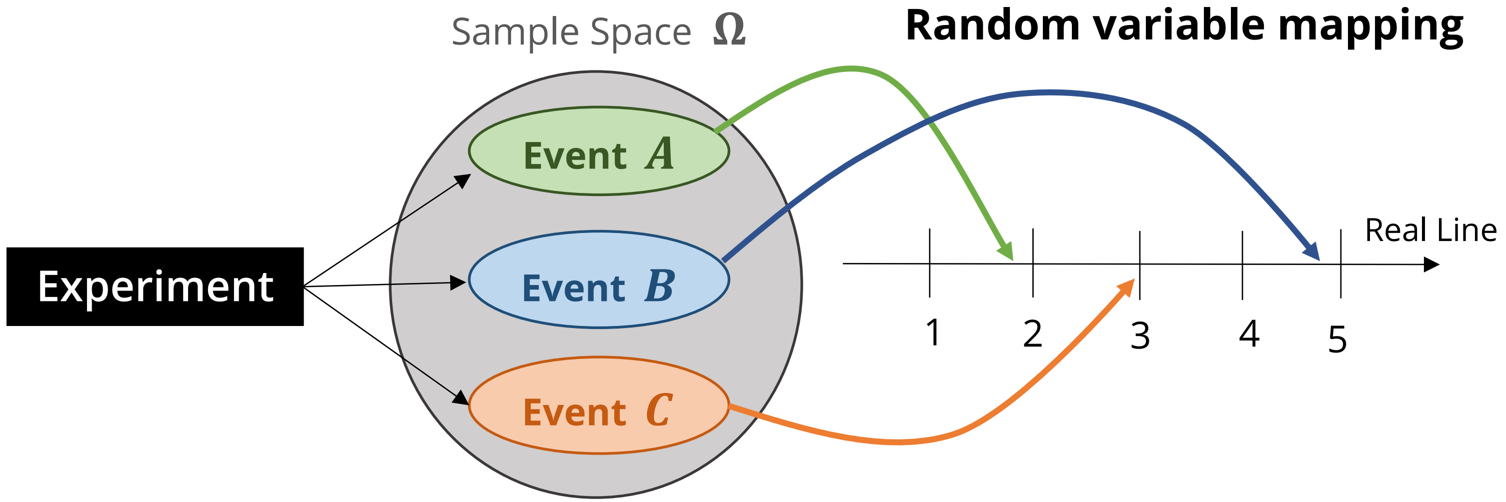
2. **ADDITIVITY** : If $\boldsymbol{A}$ and $\boldsymbol{B}$ are two ***disjoint*** events, then the probability of their union is the sum of their individual probabilities.

$$\mathbb{P}(\boldsymbol{A} \cup \boldsymbol{B}) = \mathbb{P}(\boldsymbol{A}) + \mathbb{P}(\boldsymbol{B})$$

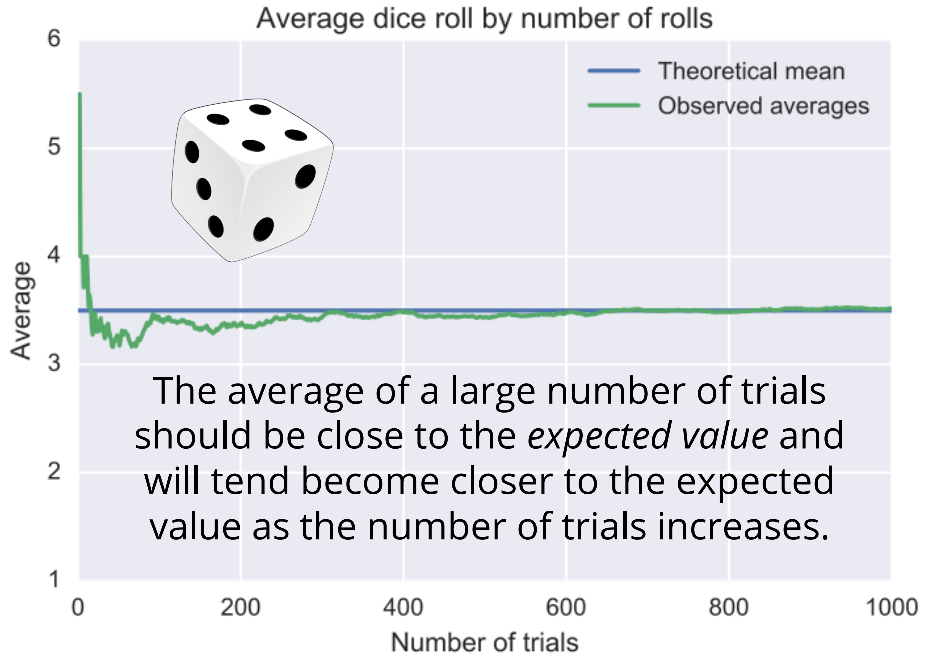
3. **NORMALIZATION** : The probability of the entire sample space (Ω) is equal to 1.

$$\mathbb{P}(\boldsymbol{\Omega}) = 1$$

LECTURE #5 – RANDOM VARIABLES



Law of Large Numbers



The average of a large number of trials should be close to the *expected value* and will tend become closer to the expected value as the number of trials increases.

EXPECTATION:

$$\langle \boldsymbol{X} \rangle = \mathbb{E}[\boldsymbol{X}] = \sum_{x} x \ \mathbb{P}(\boldsymbol{X} = x)$$

VARIANCE:

$$\mathrm{var}(\boldsymbol{X}) = \mathbb{E}[\,(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])^2\,] = \mathbb{E}[\boldsymbol{X}^2] - \mathbb{E}[\boldsymbol{X}]^2 = \sum_{x} (x - \mathbb{E}[\boldsymbol{X}])^2 \, \mathbb{P}(\boldsymbol{X} = x)$$

QUANTUM BIG IDEA

Dirac (bra-ket) notation *is how we mathematically represent quantum states!*

(An alternative linear algebra notation…)

Quantum states:

Inputs and outputs of a quantum computer

A ***ket*** is simply a column vector!

$$|v\rangle = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

A ***bra*** is the conjugate transpose of a ket (row vector)!

$$\langle v| = |v\rangle^\dagger = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix}$$

Superposition: $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$

Overlap between two quantum states:

braket (bra+ket): $\langle\psi|\phi\rangle$

Expectation of quantum operation $\boldsymbol{A}$ with respect to state $|\psi\rangle$:

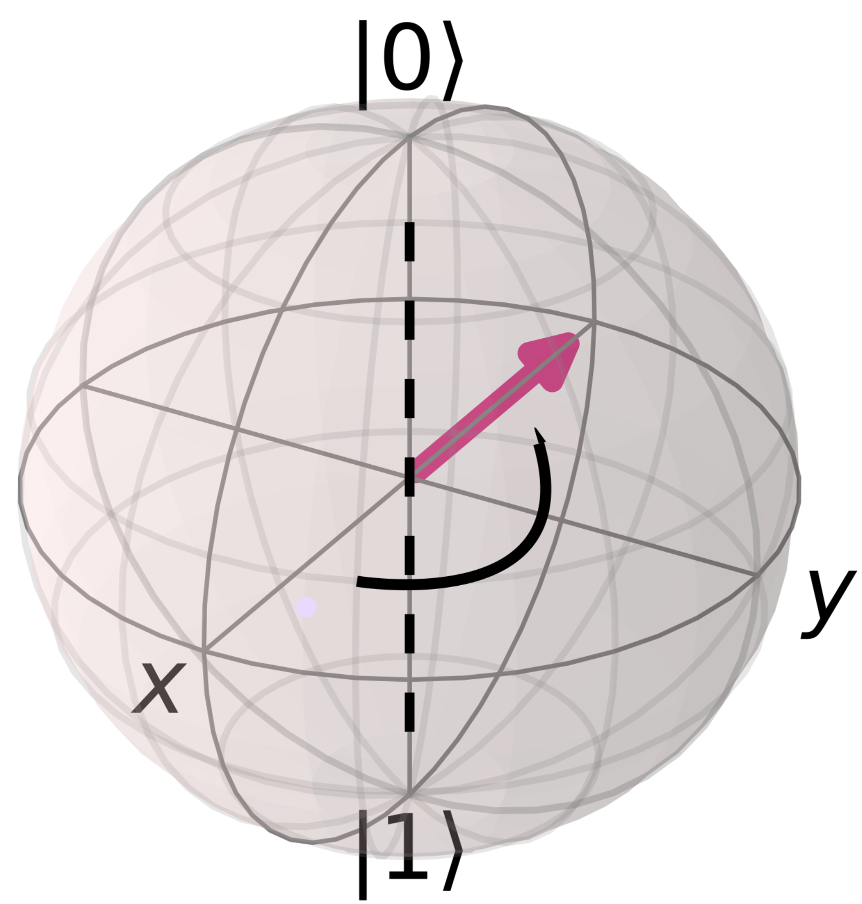
$$\langle\psi|\,\boldsymbol{A}\,|\psi\rangle$$

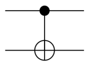
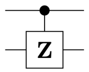
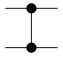
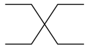
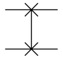
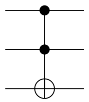
States $|\psi\rangle$ and $|\phi\rangle$ are **orthogonal** if: $\langle\psi|\phi\rangle =$

State $|\psi\rangle$ is **normal** if: $\langle\psi|\psi\rangle =$

QUANTUM BIG IDEA

*Many quantum gates are **Hermitian** and **unitary**!*



Operator	Gate(s)		Matrix
Pauli-X (X)	X	$\oplus$	$\begin{bmatrix}0&1\\1&0\end{bmatrix}$
Pauli-Y (Y)	Y		$\begin{bmatrix}0&-i\\i&0\end{bmatrix}$
Pauli-Z (Z)	Z		$\begin{bmatrix}1&0\\0&-1\end{bmatrix}$
Hadamard (H)	H		$\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix}$
Phase (S, P)	S		$\begin{bmatrix}1&0\\0&i\end{bmatrix}$
$\pi/8$ (T)	T		$\begin{bmatrix}1&0\\0&e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)			$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
Controlled Z (CZ)			$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$
SWAP			$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)			$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}$

LECTURE #8 –QUANTUM OPERATIONS & GATES

Quantum operations:

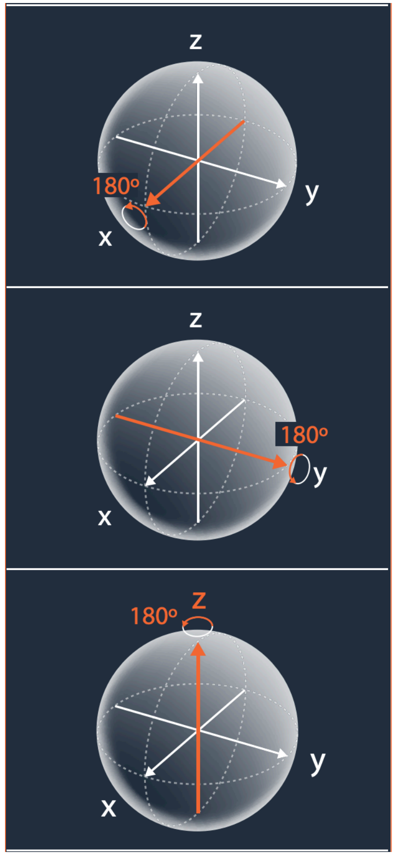
Perform the computation in a quantum computer

Pauli operators:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$ Pauli-X

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$ Pauli-Y

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$ Pauli-Z



Quantum operation properties:

- Linearity
- Can be composited
- Order matters

All observable operators are **Hermitian**:

$$\boxed{A = A^\dagger}$$

(operator is equal to its own conjugate transpose)

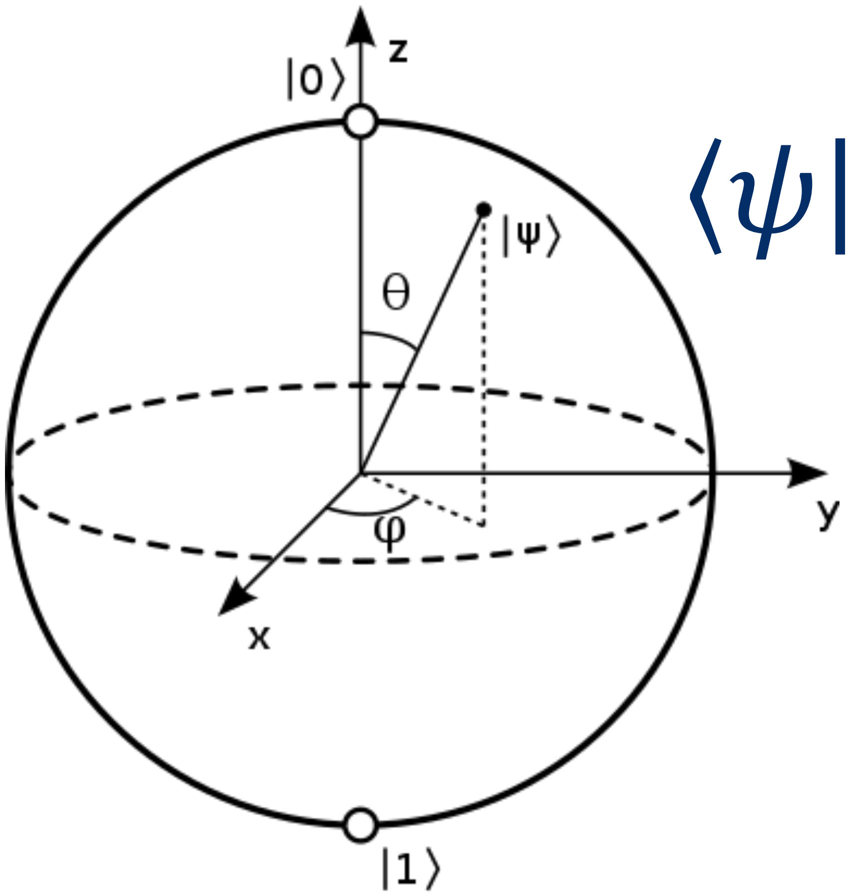
All reversible operations are **unitary:**

$$\boxed{AA^\dagger = A^\dagger A = I}$$

(conjugate transpose is matrix inverse)

QUANTUM BIG IDEA

Quantum states lie in ***Hilbert spaces****!*



$$\langle\psi|\phi\rangle$$

LECTURE #9 – VECTOR & HILBERT SPACES

A ***vector space*** is a collection of vectors which can be added and multiplied, adhering to the following axioms:

For all

$\vec{x}, \vec{y}, \vec{z} \in \boldsymbol{\mathcal{V}}$

(where $\boldsymbol{\mathcal{V}}$ is a vector space)

$a, b \in \mathbb{C}$

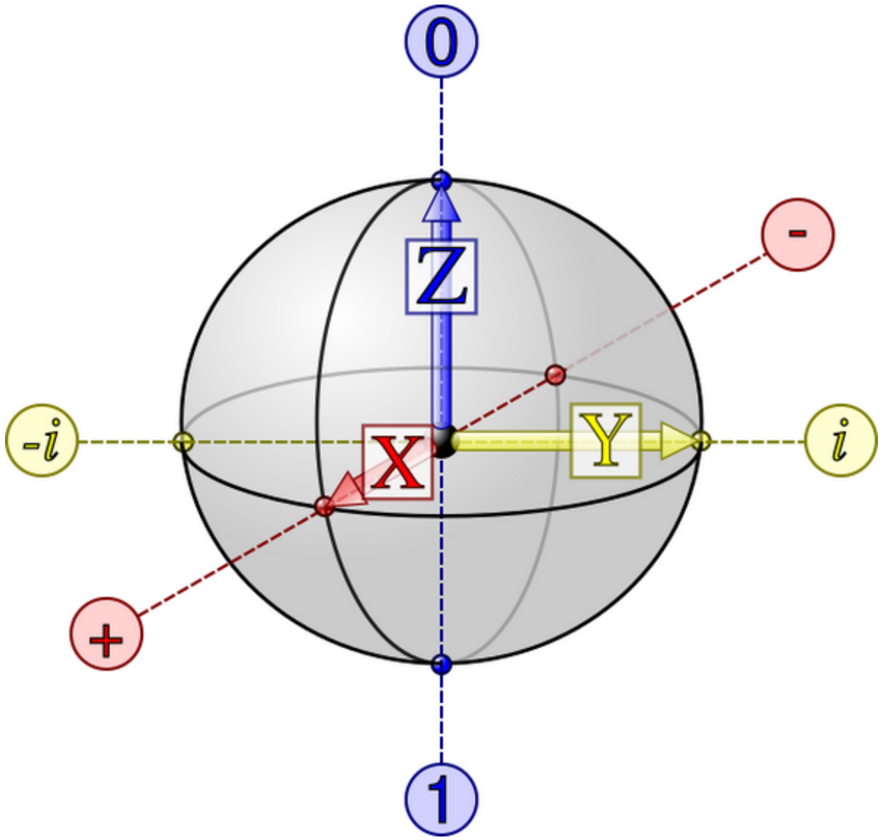
1. $\vec{x} + \vec{y} = \vec{y} + \vec{x}$
2. $\vec{x} + (\vec{y} + \vec{z}) = (\vec{x} + \vec{y}) + \vec{z}$
3. There is a unique zero vector ($\vec{0}$), such that $\vec{x} + \vec{0} = \vec{x}$
4. For each $\vec{x}$ there is a unique $-\vec{x}$, such that $\vec{x} + (-\vec{x}) = \vec{0}$
5. $1\vec{x} = \vec{x}$
6. $(ab)\,\vec{x} = a(b\vec{x})$
7. $a(\vec{x} + \vec{y}) = a\vec{x} + b\vec{x}$
8. $(a + b)\vec{x} = a\vec{x} + b\vec{x}$

EXAMPLES

Quantum states lie in ***Hilbert spaces,*** which are just ***vector space equipped with an inner product***!

QUANTUM BIG IDEA

*The **eigenvectors** of the Pauli matrices form **bases** for the quantum Hilbert space, corresponding to different measurement frames.*



LECTURE #9 – BASES

A **basis** for the vector space $\boldsymbol{\mathcal{V}}$ is the set of vectors $\{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_k\}$ having the following two properties at once:

1. The vectors are **linearly independent** (not too many vectors):

$$\boxed{c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_k\vec{v}_k = \vec{0} \text{ only when } c_1 = c_2 = \cdots = c_k = 0}$$

[Note: orthogonal vectors are linearly independent!!]

1. The vectors ***span*** $\boldsymbol{\mathcal{V}}$ (not too few vectors):

$$\boxed{\text{every vector } \vec{v} \in \boldsymbol{\mathcal{V}} \text{ can be expressed as } \vec{v} = c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_k\vec{v}_k}$$

Thus, every vector in the vector space $\boldsymbol{\mathcal{V}}$ is a **unique** linear combination of the basis vectors.

However, every vector space has **infinitely** many bases…

$$\boxed{A\vec{v} = \lambda\vec{v}}$$

If v is an eigenvector of A, our matrix-vector multiplication simplifies to a scalar-vector multiplication!

INTRO TO QUANTUM MECHANICS...