INTRO TO QUANTUM COMPUTING

Week 10 Lab

INTRODUCTION TO NUMPY

<insert name>

<insert date>

PROGRAM FOR TODAY

- TA introduction
- Logistics and ground rules
- Canvas attendance quiz
- Pre-lab zoom feedback
- Lab content
- Post-lab zoom feedback

Canvas attendance quiz passcode:

TA INTRODUCTION

TA picture

<TA details>

Canvas attendance quiz passcode:

LOGISTICS

- **Homework:** The week 10 homework is completing the Technical Assessment. It is due, like all homework assignments, on Sunday, at 11:59 p.m. ET on Canvas.

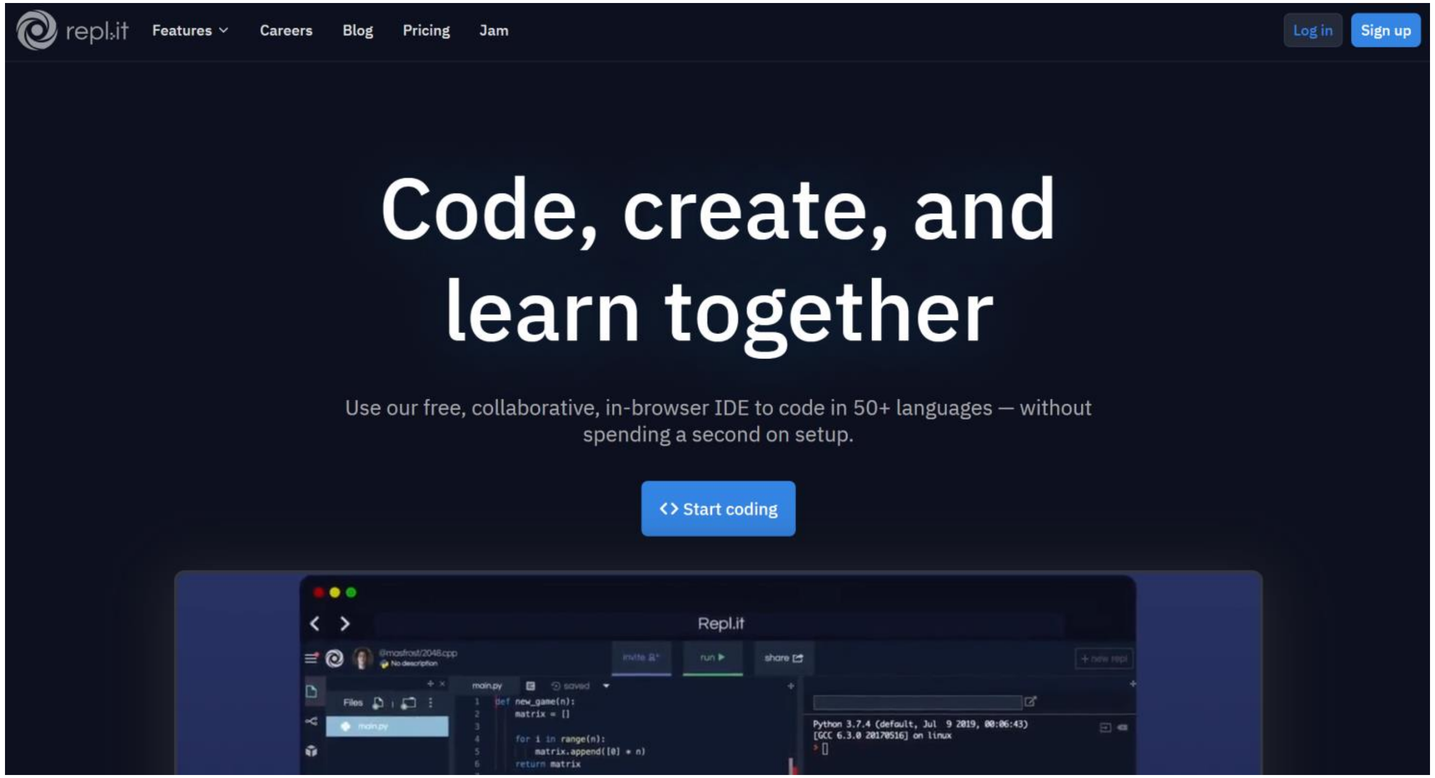
- **Canvas:** Contains all required course information and materials
  - TAs will not be able to respond to messages on Canvas
  - If you have a question on logistics, email **student@qubitbyqubit.org**

- **Piazza:** We will not be able to address all content-questions in lecture or lab
  - Look at answers to similar questions on Piazza or Discord
  - Post your question in the relevant folder in Piazza (see Piazza orientation video)
  - We will explore new content and will likely answer your question in future weeks ☺

Canvas attendance quiz passcode:

LOGISTICS: REPL.IT

- We'll be using **repl.it** for the lab today
- You can follow along on the TA's screen and try out the code later, or open a repl.it window and try it with the TA
- If you have never used repl.it before, you'll need to make a (free) account



Canvas attendance quiz passcode:

GROUND RULES

- We want to ensure that every student participating in this lab feels welcome and included

- We ask that you:
  - **Do not** spam the chat with repeated questions or messages
  - **Do not** put answers to problems in the chat, **unless your TA asks you to**
  - **Keep your questions relevant** to the topics being discussed. We have Piazza for other content-related questions
  - Only raise your hand if the TAs ask students to

- As instructors and TAs, we want to hear from diverse voices
  - **Step up, step back**

Canvas attendance quiz passcode:

CANVAS ATTENDANCE QUIZ

- Please log into Canvas and answer your lab section's quiz (using the password posted below and in the chat).
  - This is lab number:
  - Passcode:

- What topics are you MOST excited to learn this semester?

- **This quiz not graded, but counts for your lab attendance!**

Canvas attendance quiz passcode:

PRE-LAB ZOOM FEEDBACK

On a scale of 1 to 5, how would you rate your understanding of this week's content?

- 1 –Did not understand anything
- 2 – Understood some parts
- 3 – Understood most of the content
- 4 – Understood all of the content
- 5 – The content was easy for me/I already knew all of the content

Canvas attendance quiz passcode:

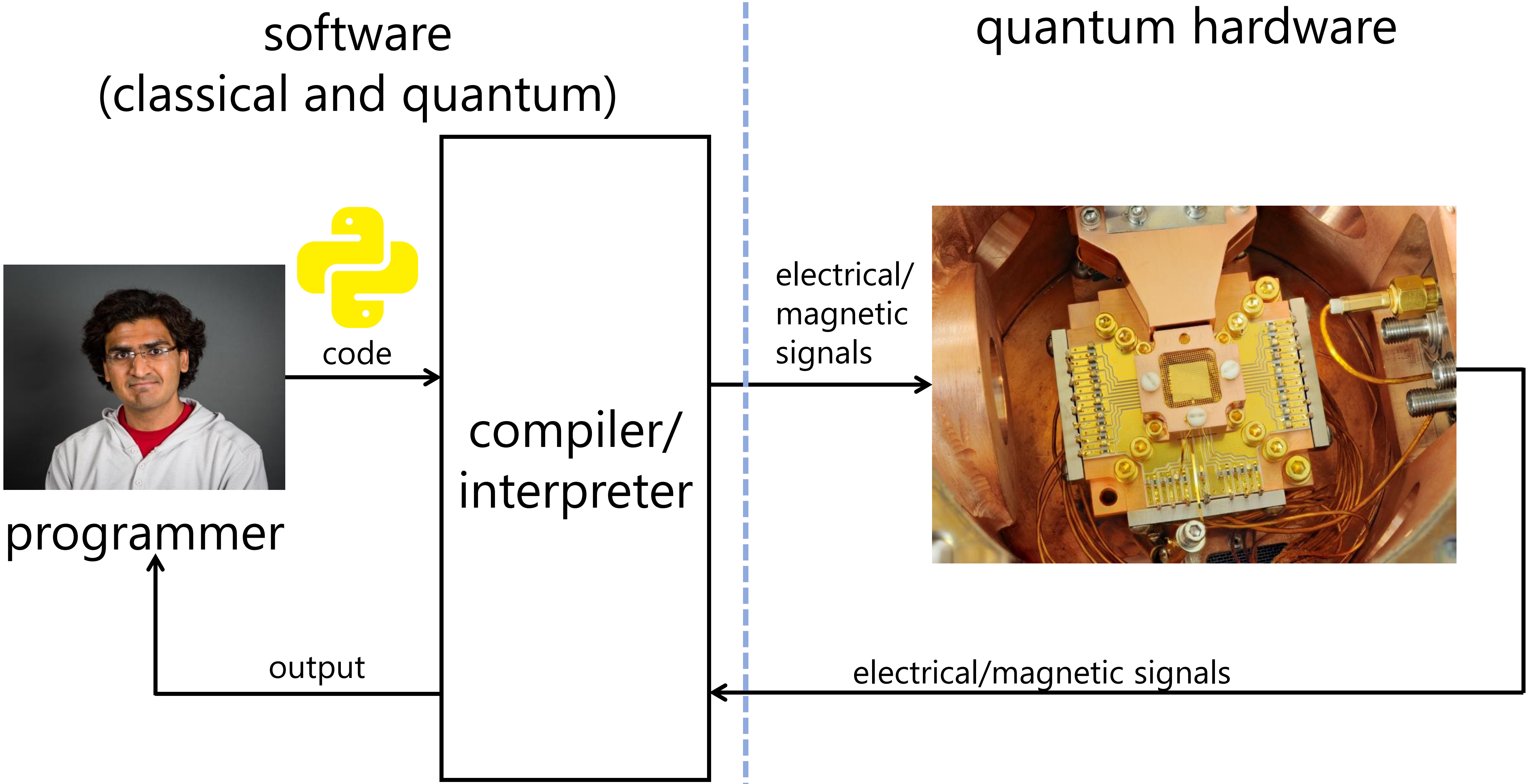
LEARNING OBJECTIVES FOR LAB 10

- Understanding the role of python in the quantum stack
    - The quantum stack
    - Python and its libraries
- Using numpy to create and manipulate arrays
    - Introduction to numpy
    - Creating and modifying vectors and matrices
    - Math with numpy arrays
    - Using numpy to apply quantum gates to qubits
- Regularly spaced arrays with numpy*

*Optional content

Canvas attendance quiz passcode:

(SIMPLIFIED) QUANTUM COMPUTING STACK



PYTHON AND LIBRARIES

- **Python:** general-purpose programming language
- **Libraries:** additional tools for programming specific types of problems
  - numpy, scipy, matplotlib
- **Why python?** Python is widely used to code, and IBM provides a library (**qiskit**) and an interpreter for python code to run quantum computers
- This lab: numpy ☺

Canvas attendance quiz passcode:

QUANTUM "APPS"

qiskit

cirq

Both qiskit and cirq are python libraries for designing quantum circuits

- Developed by IBM
- Can be used to run actual quantum hardware (coming up later this semester!)

- Developed by Google
- Cannot currently be used to run hardware

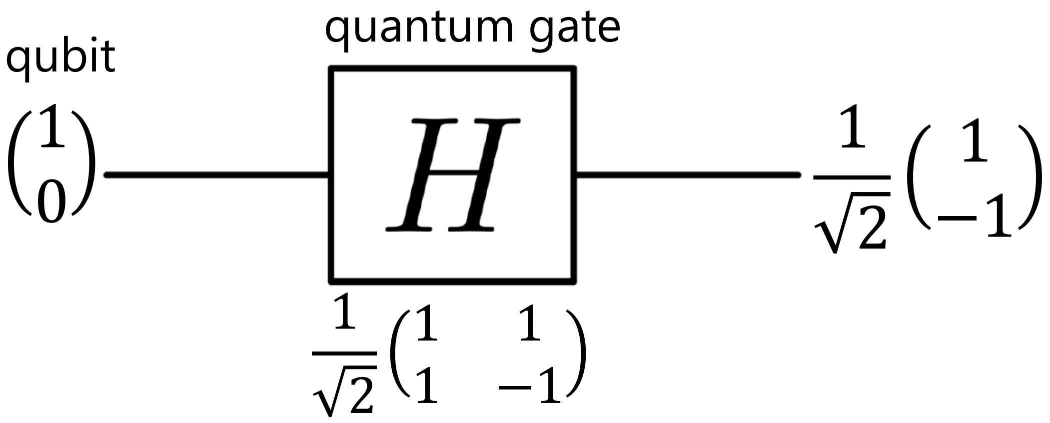
Canvas attendance quiz passcode:

WHAT IS NUMPY?

- numpy: python library for numerical computing

- Qubits → vectors
- Quantum gates → matrices

- numpy allows creation and manipulation of vectors and matrices → called **arrays** in numpy



WHY ALL THE MATH?

- All the math we learnt in semester 1 can be implemented in code using numpy
- So why did we learn to do it by hand?
- **Debugging!**

USING NUMPY

- We add libraries to python using **import**

- **import…as:** Give the library a pseudonym or abbreviation

```
import numpy as np
```


Canvas attendance quiz passcode:

DEFINING ARRAYS IN NUMPY

Use function **array** to define vectors and matrices

$$A = \begin{pmatrix} 1 & 4 & 6 & 7i \end{pmatrix}$$

```
A=np.array([1,4,6,7j])
```


Canvas attendance quiz passcode:

FINDING SPECIFIC ARRAY INDICES

- How do I find a particular element of an array?

Use **[]**

- **Array indices begin at 0!!**

```
A=np.array([1,4,6,7j])

print(A[2])
```


Canvas attendance quiz passcode:

FINDING THE SIZE OF AN ARRAY

- What are the dimensions of the array?

Use function **shape**

```
A=np.array([1,4,6,7j])

A.shape
```

- **Practice:** Define an array

$$B = (0.5 \quad 0.6 \quad -4)$$

  - Print the second element of B
  - Find the size of B

Canvas attendance quiz passcode:

MATH WITH NUMPY ARRAYS

- Addition and subtraction

- **Practice:** Create two arrays $A = (1 \quad 0)$ and $B = (0 \quad 1)$. Create two additional arrays C and D such that:

$$C = \frac{1}{\sqrt{2}}(\mathrm{A} + \mathrm{B})$$

$$D = \frac{1}{\sqrt{2}}(\mathrm{A} - \mathrm{B})$$

Square root: **np.sqrt()**

Canvas attendance quiz passcode:

ARRAY TRANSPOSE

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 $\vec{w}^T = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$

```
B = np.array([[1,3,4j]])
```

Use **<arrayname>.T**

```
D = B.T
```


Canvas attendance quiz passcode:

ARRAY CONJUGATE

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

Use **<arrayname>.conj()**

```
B = np.array([[1,3,4j]])

E = B.conj()
```

- **Practice:** Create an array $A = \frac{1}{\sqrt{2}}(1 \quad 1j)$ and print its conjugate transpose

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

Canvas attendance quiz passcode:

QUESTIONS?

Questions on content so far?

Canvas attendance quiz passcode:

DEFINING MATRICES

Use **array**

$$C = \begin{pmatrix} 1 & 4 & 6 & 7j \\ 1 & 5 & 6 & 4 \end{pmatrix}$$

```
C=np.array([[1,4,6,7j],[1,5,6,4]])
```

- **Practice:** Define a matrix

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

  - Find the size of Z
  - Print the (1,1) element of Z

Canvas attendance quiz passcode:

MULTIPLICATION WITH ARRAYS

Use @ symbol

```
A = np.array([2,3,4])

B = np.array([1,3,4j])

C = A @ B
```

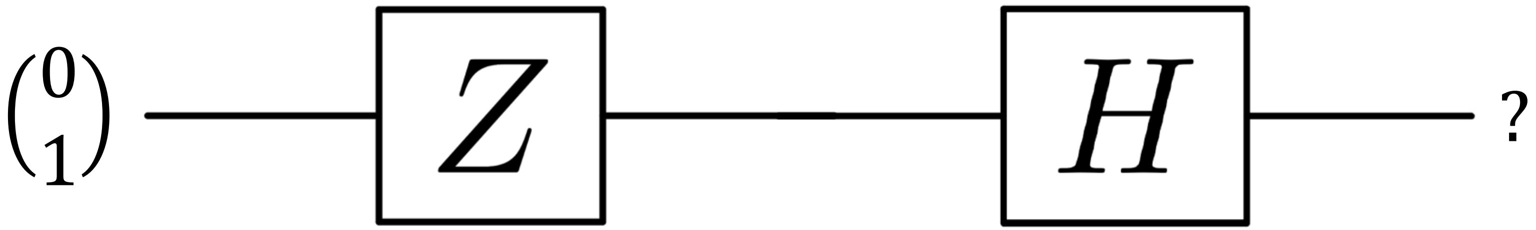

Canvas attendance quiz passcode:

PUTTING IT ALL TOGETHER - PROBLEM

- **Problem:** A qubit starts off in the state $\begin{pmatrix}0\\1\end{pmatrix}$. For a particular quantum computation, we apply a Z gate followed by an H gate to this qubit, and then measure the result. What is the state of the qubit at the end?

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

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$



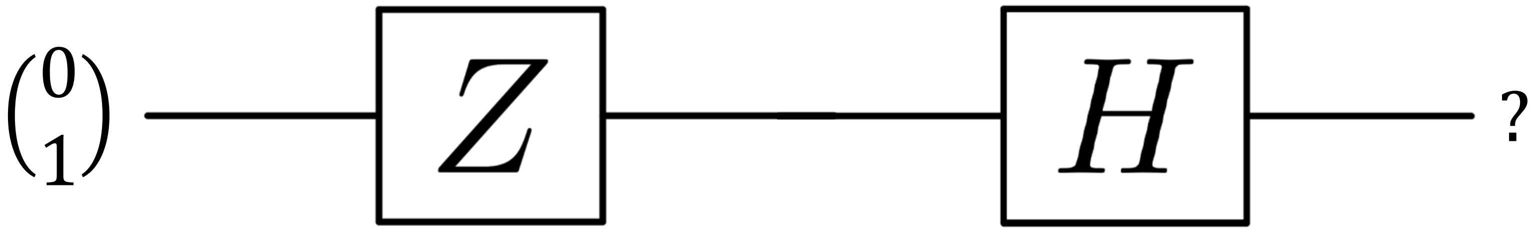
Canvas attendance quiz passcode:

PUTTING IT ALL TOGETHER - PROBLEM

- **Problem:** A qubit starts off in the state $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. For a particular quantum computation, we apply a Z gate followed by an H gate to this qubit, and then measure the result. What is the state of the qubit at the end?
  - Define the qubit state $psi = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
  - Apply the gate $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ to it
  - Apply the gate $\mathrm{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ to the result

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

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



KEY TAKEAWAYS

- Python is widely used to write programs that can run on quantum computers

- Libraries extend the functionality of python.
  - numpy and scipy – library for scientific computing
  - qiskit – library to create and run quantum circuits

- Qubit states can be represented as vectors, and quantum gates can be represented as matrices
  - Using numpy, we can define and manipulate both

QUESTIONS?

Questions on content so far?

Canvas attendance quiz passcode:

POST-LAB ZOOM FEEDBACK

After this lab, on a scale of 1 to 5, how would you rate your understanding of this week's content?

- 1 –Did not understand anything
- 2 – Understood some parts
- 3 – Understood most of the content
- 4 – Understood all of the content
- 5 – The content was easy for me/I already knew all of the content

Canvas attendance quiz passcode:

OPTIONAL CONTENT

DEFINING REGULARLY SPACED ARRAYS

- Manually defining arrays can get annoying for large arrays
- Examples of large arrays
  - Vectors defining multiple qubit states
  - Matrices for multi-qubit gates
  - Vectors defining time

- np.zeros
- np.ones
- np.full

$$a = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad C = \begin{pmatrix} \pi & \pi & \pi \\ \pi & \pi & \pi \\ \pi & \pi & \pi \end{pmatrix}$$

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

```
a = np.zeros((4,1))
```

```
C= np.full((3,3),np.pi)
```


Canvas attendance quiz passcode:

DEFINING REGULARLY SPACED ARRAYS

- np.arange

$$d = (2\ 4\ 6\ 8\ \ldots\ldots 98)$$

```
d = np.arange(2,100,2)
```

- np.linspace

$$e = (\overbrace{3\ \ldots\ldots\ldots\ 45}^{\text{50 numbers total}})$$

```
e=np.linspace(3,45,50)
```


Canvas attendance quiz passcode: