Quantum Imaging Magnetic Fields Sensors

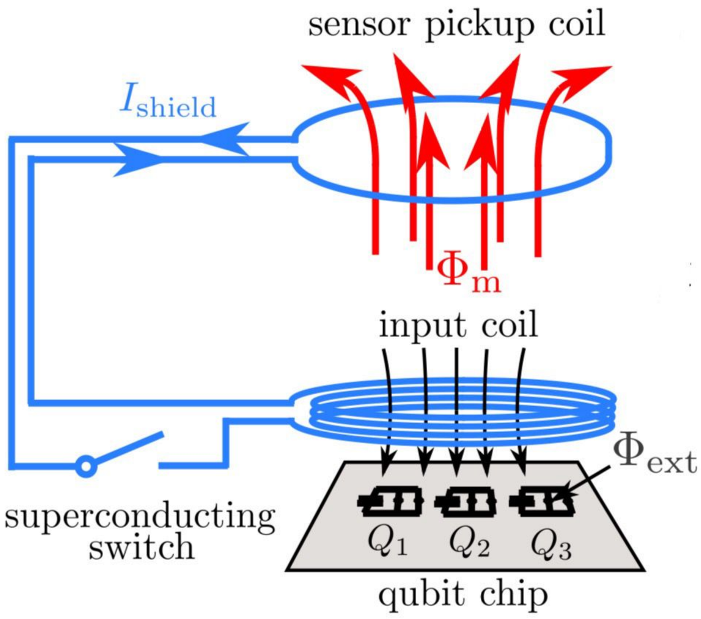
Yasir, Adrian, Munzir

Overview

- Qubits undergo phase shifts when exposed to magnetic fields
- Using **Ramsey Fringes Interferometry** to express the phase shifts
- **Kitaev's Algorithm** for phase estimation (Optimization)
  - Qiskit Simulation (comparing 1, 2, and 3 qubits)
- Conclusion

Quantum Sensors

- Φm: general flux to be measured
- Ishield: Current representing Φm
- Φext: magnetic flux representing Φm
  - They are exposed to the Φext
    - This results in a phase shift depending on exposure time
      - Exposure time is determined by the superconducting switch
    - Phase shift is dependent on the strength of the flux and the time of exposure



Introduction to Kitaev’s Algorithm

- Kitaev’s Phase Estimation Algorithm is a quantum procedure designed to estimate the phase ϕ of a unitary operator.
- It leverages quantum superposition and entanglement to achieve high precision in phase estimation.
- The algorithm achieves a precision that scales exponentially with the number of qubits.

$$U|\lambda\rangle = e^{i\theta}|\lambda\rangle$$

EIGENVALUE

$$\mathbb{P}(0) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

unit modulus.

$$\mathbb{P}(1) = \left|\frac{e^{i\theta}}{\sqrt{2}}\right|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

this relative phase has no observable effect when we try to measure only the first qubit

Kitaev's Algorithm for Phase Estimation

- A unitary gate of some phase shift is applied
  - This effect doesn't show during measurement
- The phase ϕ is estimated iteratively.
  - Uses each qubit to perform a controlled operation and measure the resulting phase shift
- The inverse Quantum Fourier Transform is applied to the set of qubits to extract the phase information encoded in the quantum state.
  - The QFT transforms the measured probabilities into a phase estimate

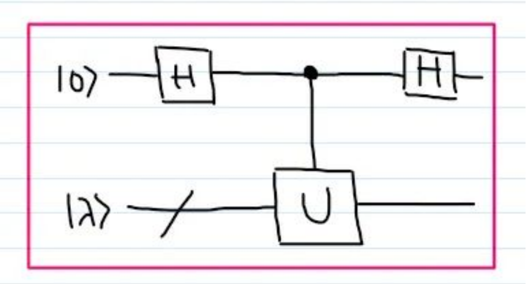
$$U|\lambda\rangle = e^{i\theta}|\lambda\rangle$$

$$\mathbb{P}(0) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

unit modulus.

$$\mathbb{P}(1) = \left|\frac{e^{i\theta}}{\sqrt{2}}\right|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$$

Kitaev’s Algorithm with 1 qubit



- Apply a second Hadamard gate
  - This encodes phase shift into the measurement

$$\left[\frac{|0\rangle + e^{i\theta}|1\rangle}{\sqrt{2}}\right] \otimes |\lambda\rangle \xrightarrow{H\otimes I} \left[\frac{H|0\rangle + e^{i\theta}H|1\rangle}{\sqrt{2}}\right] \otimes |\lambda\rangle$$

EXPAND | H gate

$$\left[\left[\frac{1+e^{i\theta}}{2}\right]|0\rangle + \left[\frac{1-e^{i\theta}}{2}\right]|1\rangle\right] \otimes |\lambda\rangle \xleftarrow{\text{GROUP TERMS}} \frac{1}{2}\left[|0\rangle + |1\rangle + e^{i\theta}(|0\rangle - |1\rangle)\right] \otimes |\lambda\rangle$$

$$\mathbb{P}(0) = \left|\frac{1+e^{i\theta}}{2}\right|^2$$

$$\mathbb{P}(1) = \left|\frac{1-e^{i\theta}}{2}\right|^2$$

- Using Euler’s Identity
  - e^(iΦ)+e^(-iΦ) = 2cos(Φ)
  - cos(Φ)= .5(e^(iΦ)+e^(-iΦ))

$$\mathbb{P}(0)_c = \left|\frac{1+e^{i\theta}}{2}\right|^2 = \left|\cos\frac{\theta}{2}\right|^2 = \cos^2\frac{\theta}{2} = \frac{1+\cos\theta}{2}$$

$$\mathbb{P}(1)_c = \left|\frac{1-e^{i\theta}}{2}\right|^2 = \left|\sin\frac{\theta}{2}\right|^2 = \sin^2\frac{\theta}{2} = \frac{1-\cos\theta}{2}$$

$\theta \in [0, 2\pi)$

Because

$\Rightarrow \mathbb{P}(0) + \mathbb{P}(1) = 1$

$\Rightarrow \mathbb{P}(1) = 1 - \mathbb{P}(0)$

Qiskit Implementation of Kitaev’s Algorithm

Simulation and Data

- We used multiple qubit states to represent degree of noise in measurement and highlight the algorithmic speedup
  - 1-qubit
  - 2-qubits
  - 3-qubit
- Expect that more qubits results in higher accuracy due to greater phase resolutions
- Sensing accuracy should scale inversely with time,

Qiskit Code Overview

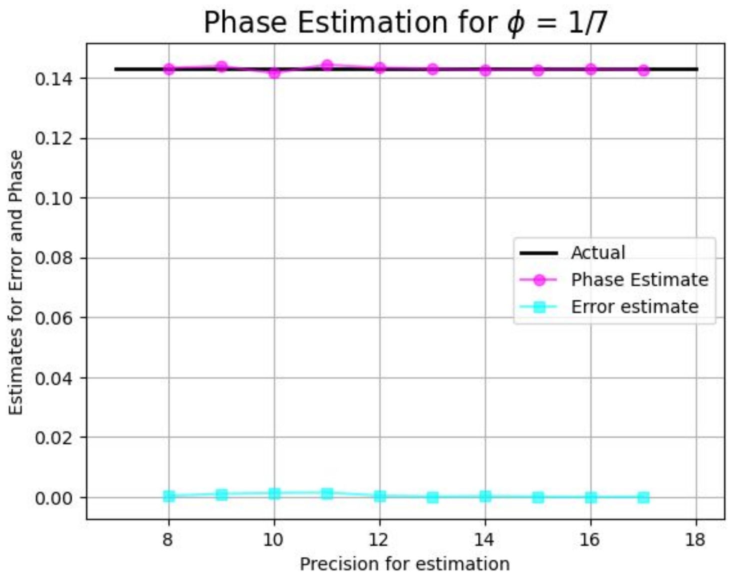
- The implementation involves constructing the quantum circuit, executing it on a quantum simulator, and extracting the estimated phase.
- Class KQPE:
  - Initialization: Configures the unitary matrix, precision level, and number of qubits.
  - get_phase Method: Executes the quantum circuit containing the Kitaev phase estimation sub-circuit and extracts the phase.
  - get_circuit Method: Constructs the Kitaev phase estimation circuit with the specified unitary and precision for integration with other quantum circuits.
- Kitaev QPE Jupyter Notebook
  - Run simulation experiments and generate visualizers

```
U = np.array([[1, 0], [0, np.exp(2*np.pi*1j*(1/3))]])
kqpe = KQPE(unitary=U, precision=16)
kq_circ = kqpe.get_circuit(show=True)
```

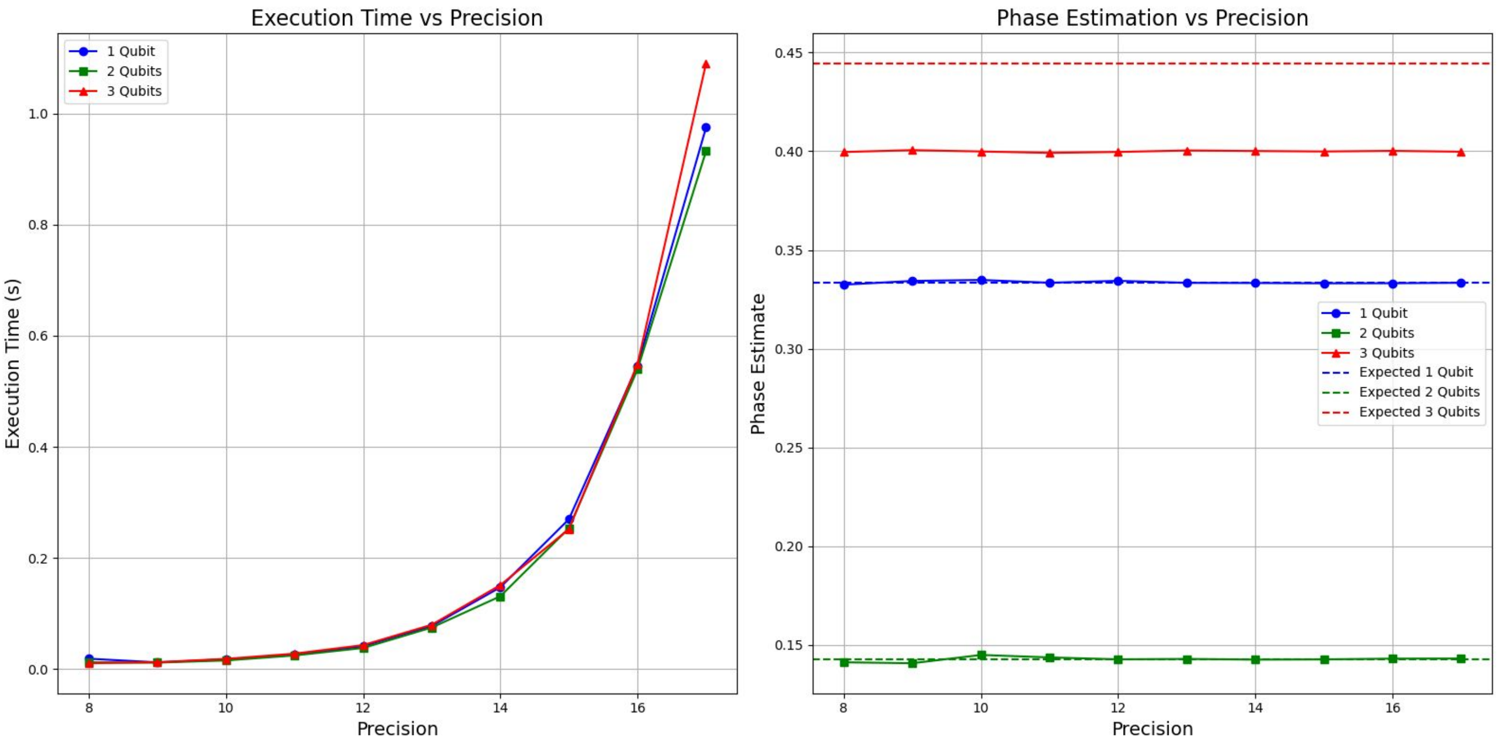
```
q = QuantumCircuit(5, 6)
q.x(3)
q.append(kq_circ, qargs=[1, 2, 3])
phase = kqpe.get_phase(backend=Aer.get_backend('qasm_simulator'),
                       QC=q, ancilla=[1, 2], clbits=[0, 1], show=True)
print("Estimated Phase:", phase)
```

Kitaev Optimization Results

- For one qubit, we can see that the phase estimation has some slight variance, meaning the residual error is present.
- However, it is very promising

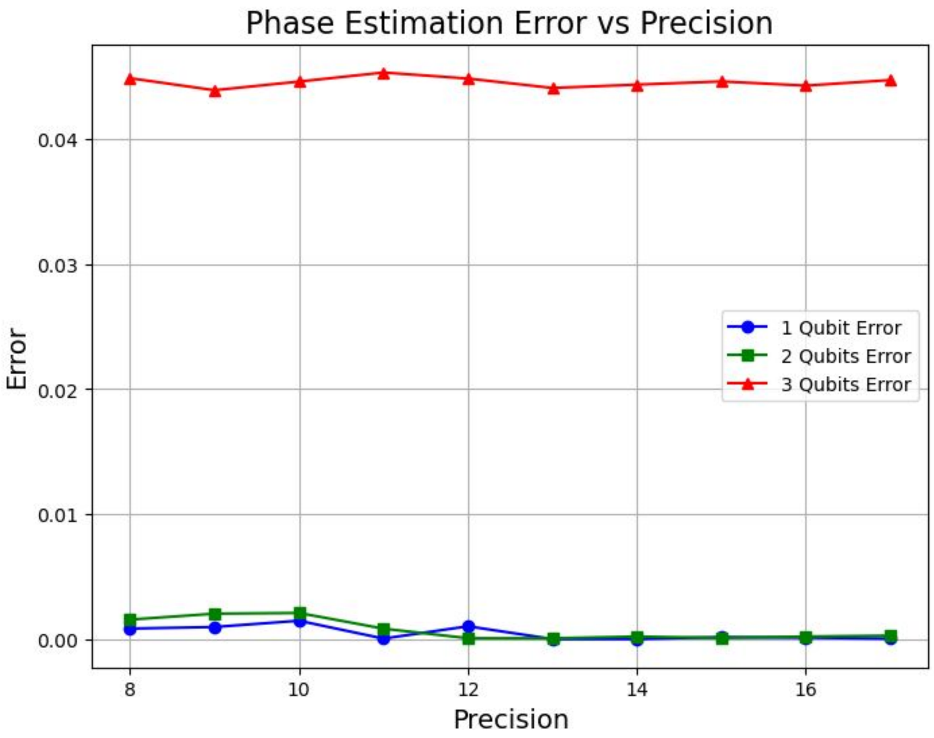


Kitaev Optimization Results



Kitaev Optimization Results

- On average, the 2-qubit setup shows a slight improvement in execution time (**6.06%**) compared to the 1-qubit setup.
    - However, at the highest precision (17), the 2-qubit setup slightly underperforms compared to the 1-qubit setup (**-3.55%**).
- 1 Qubit vs. 3 Qubits:
    - The 3-qubit setup generally performs worse than the 1-qubit setup, with an average decrease in performance of **-6.18%**.
    - At the highest precision, this decrease in performance is more pronounced (**-10.74%**).

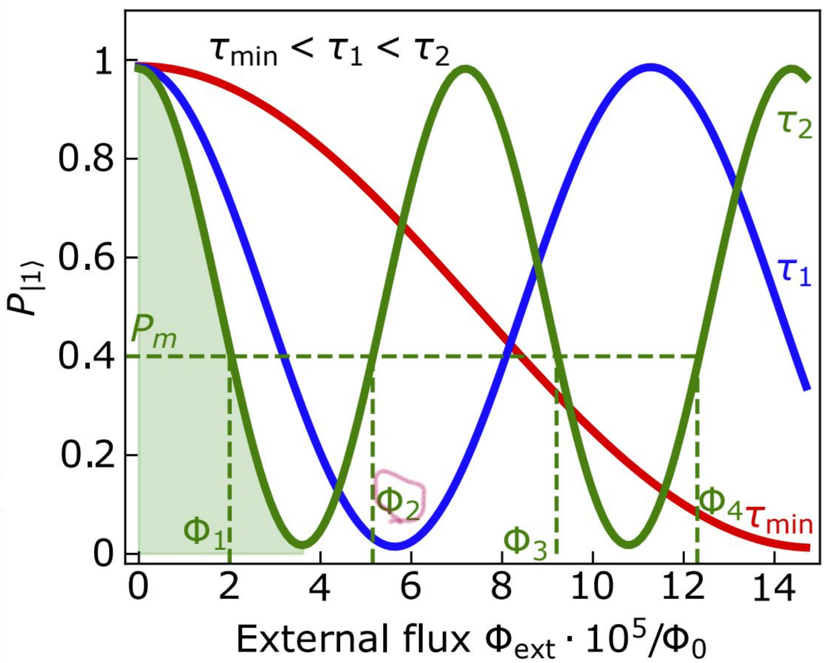


Performance and Runtime Analysis

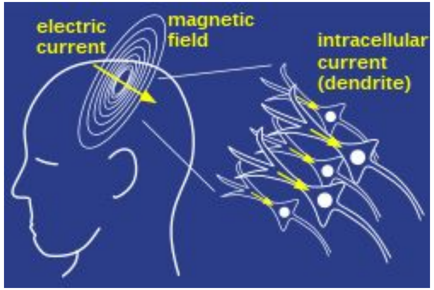
- Execution Time vs. Precision:
  - Observation: Execution time **increases** with the number of qubits and precision.
  - Explanation: More qubits and higher precision require more controlled unitary operations and a deeper quantum circuit, leading to increased computation time.
    - With more qubits (e.g., 3 qubits), the phase estimate becomes more precise, reducing ϵ but requiring more time and resources to compute.
- Accuracy vs. Precision:
  - Observation: Higher precision and more qubits result in more **accurate** phase estimation.
  - Explanation: As the number of qubits increases, the algorithm can resolve smaller phase differences, leading to a more accurate estimate.
    - With fewer qubits (e.g., 1 qubit), the phase estimate might only resolve large differences, leading to a potentially significant error ϵ.

What this optimization can improve

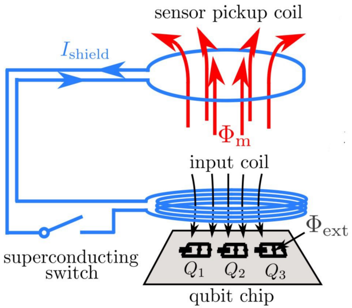
- For longer delay times, it is not possible to unambiguously determine the measured flux based on a single outcome
  - Φ1, Φ2, Φ3, Φ4?
  - Coherence time
    - Time qubit can maintain its state
      - Decoherence occur after
        - Energy relaxation
        - Dephasing
    - Depends on qubit



Application in magnetic flux sensing



- **Magnetoencephalography (MEG)**
  - Neuroimaging technique
  - used to measure magnetic fields produced by neuronal activity in the brain
- **Superconducting Quantum Interference Device (SQUIDs)** is a magnetometer that detects magnetic fields from neural activity within the brain
  - Extremely sensitive to small magnetic fields
  - Contains quantum circuits mentioned earlier



Project Improvements

- Run on IBM backend or RPI-IBM backend so we don’t have to use Aer simulated components
  - Ancilla qubits
- Run on higher precision to confirm 3-qubit accuracy improvement
  - Computational resources
- Run pipeline on actual MEG data to see practical example