Connecting Minds. Exchanging Ideas.

A Quantum-Walk-Unitary HHL Matrix Equation Solver and Its Challenges in the NISQ Era

Xinbo Li*, Christopher Phillips#, Ian Jeffrey*, and Vladimir Okhmatovski*

*University of Manitoba

*University of Waterloo

Outline

I. Motivation
II. The HHL algorithm
III. The QWU-HHL algorithm
IV. The challenge in the NISQ era

Outline

I. Motivation
II. The HHL algorithm
III. The QWU-HHL algorithm
IV. The challenge in the NISQ era

I. Motivation

- Why do we want to leverage the power of quantum computers?
  - Computational electromagnetic answer the quest of electromagnetic simulation for complex geometries
  - A main contributor to the simulation time: Solving large matrix equations with N unknowns
  - Differential-equation-based methods: Sparse matrix equations

    Conjugate Gradient Method: $O(N)$
  - Moment Method: Dense matrix equations

    Fast Algorithms: $O(N\log N)$
- The HHL quantum matrix equation solver: $O(\log N)$

Outline

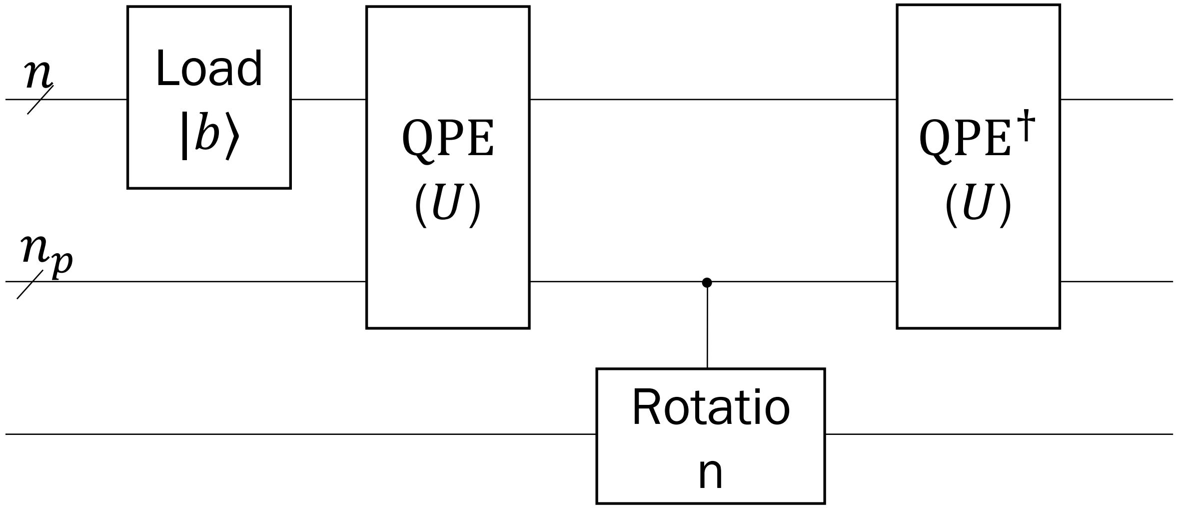
II. The HHL Algorithm

- Name comes from the acronym of its authors: [Harrow, Hassidim, Llyod, 2009, https://arxiv.org/abs/0811.3171]
- Problem Statement: Given a Hermitian matrix A and a normalized right-hand-side vector $|b\rangle$, *prepare* a quantum state $|x\rangle$ (up to a normalization factor) such that $A|x\rangle = |b\rangle$

- Complexity: $O(\log(N)\, s^2\kappa^2/\epsilon)$
  - N: matrix dimension
  - s: the number of nonzero elements per row/column
  - ϵ: desired accuracy

Registers

IO/Input register	I
Clock/Phase register	C
Ancilla	A

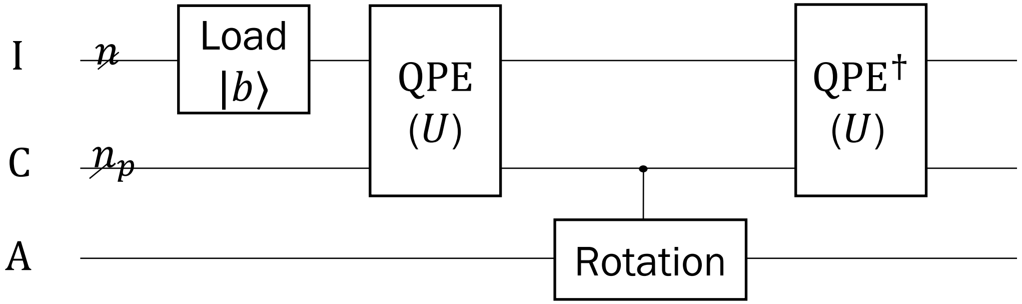


II. The HHL Algorithm: Insights

- Quantum supremacy

Quantum superposition allows n qubits to represent 2^n numbers, which needs 2^n classical bits to represent

- The idea of HHL

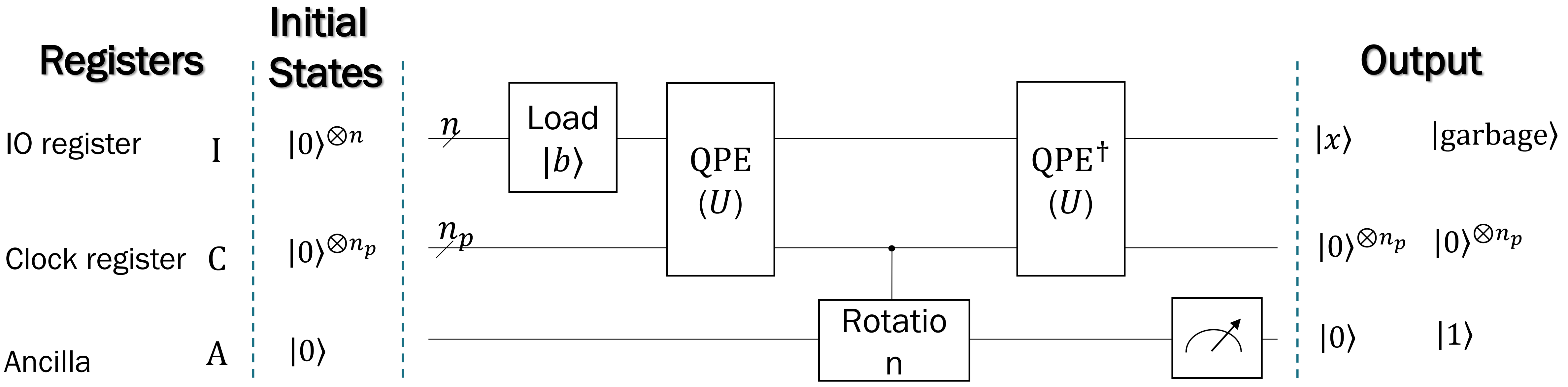


Solve the matrix equation via eigendecomposition:

There are methods to find (QPE) and invert (controlled rotation) the eigenvalues quantum mechanically

- The Quantum Phase Estimation (QPE) is the core of the HHL algorithm, and is the erroneous step in the HHL algorithm
  - The unitary U is implemented as $U = e^{iAt}$ via Hamiltonian simulation in classical HHL
  - The number of qubits n_p is chosen according to the required accuracy $\epsilon_{QPE} = 2^{-m}$ and success probability p: $n_p = m + \lceil \log(2 + 1/2p) \rceil$ [Nielsen, Chuang, 2010]
- The controlled rotation is a simplified block of "do something to the eigenvalue", i.e., $f(\lambda)$ can be implemented for other algorithm and HHL chooses $f(\lambda) = 1/\lambda$ [Harrow, Hassidim, Llyod, 2009]

II. The HHL Algorithm



- The desired output $|x\rangle$ is distinguished from the $|\text{garbage}\rangle$ via the state of the ancilla.
- The output is stored as a quantum state. If one needs classical information for post-processing, one can only retrieve that as the expectation of a measurement operator $\langle x|M|x\rangle$, where M is the measurement operator.
- The unitary U is a choice up to the user. The eigenvalues and eigenvectors of U and the Hermitian matrix A need to be closely related so that the eigendecomposition can be achieved quantum mechanically.

II. The HHL Algorithm-Improvements

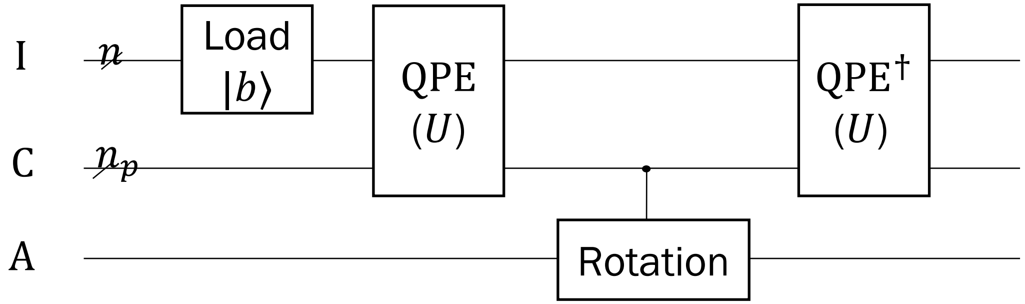
- Most HHL improvements the result (achieving better complexity) or removes the requirements [Dervovic et al, 2018, https://arxiv.org/abs/1802.08227]
  - Reduce the complexity dependence on the condition number from κ^2 to $\kappa \log^3 \kappa$ [Ambainis, https://arxiv.org/abs/1010.4458]
  - Reduce the complexity dependence on the precision from $\mathrm{ploy}(1/\epsilon)$ to $\mathrm{ploy}\log(1/\epsilon)$ [Childs et al, 2017, https://doi.org/10.1137/16M1087072]
  - Removes the requirement on the sparsity of the Hamiltonian matrix A [Wossnig et al, 2018, https://doi.org/10.1103/PhysRevLett.120.050502]
- Most improvements leaves the unitary U in the QPE untouched, with the necessity of Hamiltonian simulation
- Hamiltonian simulations are not trivial, and if we do not need it explicitly, can we remove it in the HHL process?

Outline

I. Motivation
II. The HHL algorithm
III. The QWU-HHL algorithm
IV. The challenge in the NISQ era

III. The QWU-HHL Algorithm

- The Hamiltonian simulation is not a necessary explicit step
  - The whole QPE block is for the purpose of estimating the eigenvalues of A
  - The Hamiltonian simulation (implementation of $U = e^{iAt}$) is needed because the eigenvalue and eigenvector relationship between e^{iAt} and A is straightforward

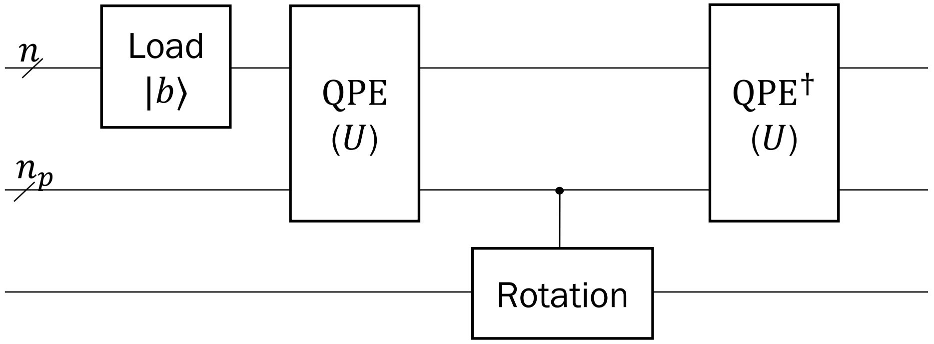


Operator	Eigenvector	Eigenvalue
A	$\|u\rangle$	λ
$U = e^{iAt}$	$\|u\rangle$	$e^{i2\pi\tilde{\theta}} = e^{i\lambda t}$

- Popular Hamiltonian simulation methods include
  - Decomposition of the Hamiltonian [Low et al, 2023, https://arxiv.org/abs/2211.09133v2]: difficult to apply to a general Hamiltonian
  - Quantum-walk-based methods [Berry, Childs 2012, https://arxiv.org/abs/0910.4157v4]

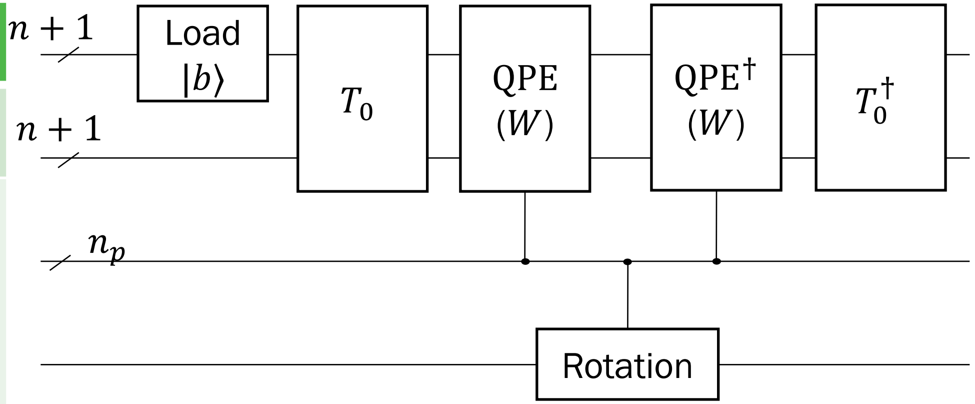
III. The QWU-HHL Algorithm

- Inspired by the quantum-walk-based Hamiltonian simulation, we use the quantum walk operator W as the unitary
  - $W \triangleq iS(2TT^\dagger - I)$
  - $T \triangleq \sum_{j=1}^{N} |j\rangle|\phi_j\rangle\langle j|$
  - $S \triangleq \sum_{j=1}^{N} \sum_{k=0}^{N-1} |k\rangle|j\rangle\langle j|\langle k|$



Classical HHL

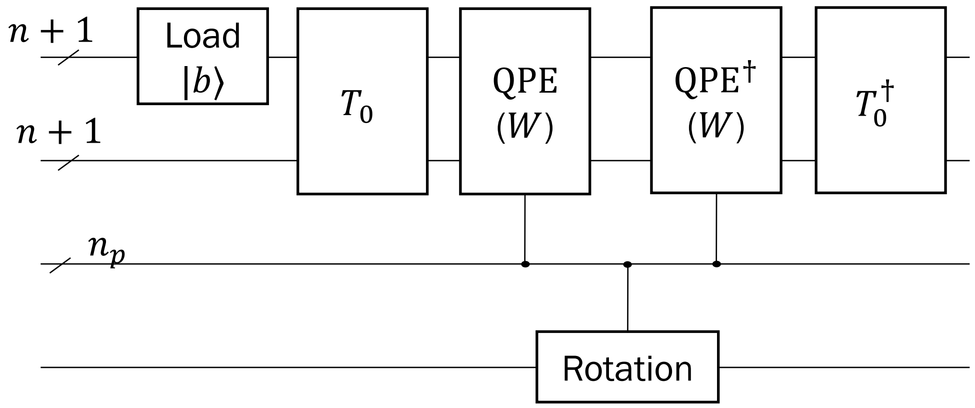
Operator	Eigenvector	Eigenvalue
A	$\|u_j\rangle$	λ_j
W	$\|v_j^{\pm}\rangle = \frac{1 + i\mu_j^{\pm} S}{\sqrt{2(1-\hat{\lambda}_j^2)}} T\|u_j\rangle$	$\mu_j^{\pm} = i\hat{\lambda}_j \pm \sqrt{1-\hat{\lambda}_j^2}$ where $\hat{\lambda}_j = \frac{\lambda_j}{X}$, $X \triangleq \max_{j,k}\|H_{jk}\|$



QWU-HHL

III. The QWU-HHL Algorithm

- In the definition,
  - $W \triangleq iS\left(2TT^{\dagger} - I\right)$
  - $T \triangleq \sum_{j=1}^{N} |j\rangle|\phi_j\rangle\langle j|$
  - $S \triangleq \sum_{j=1}^{N} \sum_{k=0}^{N-1} |k\rangle|j\rangle\langle j|\langle k|$
- T is the mapping from $\mathbb{C}^N$ to $\mathbb{C}^N \otimes \mathbb{C}^N$ as the eigenvector of W lives in the latter
- S is the swapping operator



QWU-HHL

- T translates the initial state of the system (the RHS state) into the eigenbasis of W

$$T|b\rangle = T\sum_{j=1}^{N} \beta_j |u_j\rangle = \sum_{j=1}^{N} \frac{\beta_j}{\sqrt{2\left(1-\hat{\lambda}_j^2\right)}}\left[\left(1+i\hat{\lambda}_j\mu_j^-\right)|v_j^+\rangle + \left(1+i\hat{\lambda}_j\mu_j^+\right)|v_j^-\rangle\right]$$

Outline

I. Motivation
II. The HHL algorithm
III. The QWU-HHL algorithm
IV. The challenge in the NISQ era

IV. The NISQ hardware

- Logical qubits vs physical qubits
  - A logical qubit is a qubit used for programming
    - Typically made out of a collection of physical qubits
    - All qubits referred so far are logical qubits
  - A physical qubit is a quantum realization of a qubit. Physical qubits suffer from decoherence.
- Quantum errors [Devitt et al, https://arxiv.org/abs/0905.2794]
  - Coherent errors: undesired gates applied to the system
  - Environmental decoherence: qubits losing information due to interaction with the environment
  - Measurement, etc.

Proof-of-concept qubits	Potential applications
A few logical qubits	• Quantum Secret Sharing • Quantum Co-processor
~50 logical qubits	• Demonstration of Quantum supremacy
~150 logical qubits	• Quantum Chemistry • Machine Learning
$\sim 10^6$ logical qubits	A full fault-tolerant quantum computer

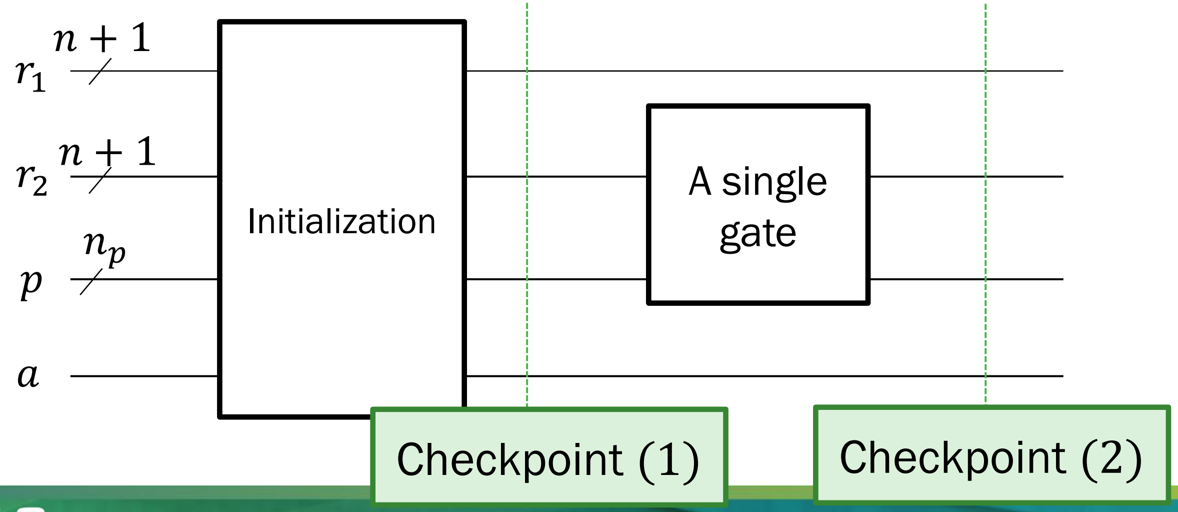
[Fruchtman, Choi, 2016, Technical Roadmap for Fault-Tolerant Quantum Computing]

IV. The NISQ hardware

- Today's available quantum hardware is Noisy Intermediate-Scale Quantum (NISQ) hardware [Preskill, 2018, https://arxiv.org/abs/1801.00862]
- The ultimate goal: Fault-tolerant quantum computer
  - low-error logical qubits: Google has demonstrated quantum error correction works in practice: increasing the number of physical qubits in a logical qubit yield a better logical qubit [Google Quantum AI, 2023, https://doi.org/10.1038/s41586-022-05434-1]
  - Many of these qubits: IBM endeavors to make large-scale quantum computers by multi-chip quantum processors with chip-to-chip couplers [IBM development roadmap, IBM Quantum Computing | Roadmap]

IV. QWU-HHL on NISQ Hardware

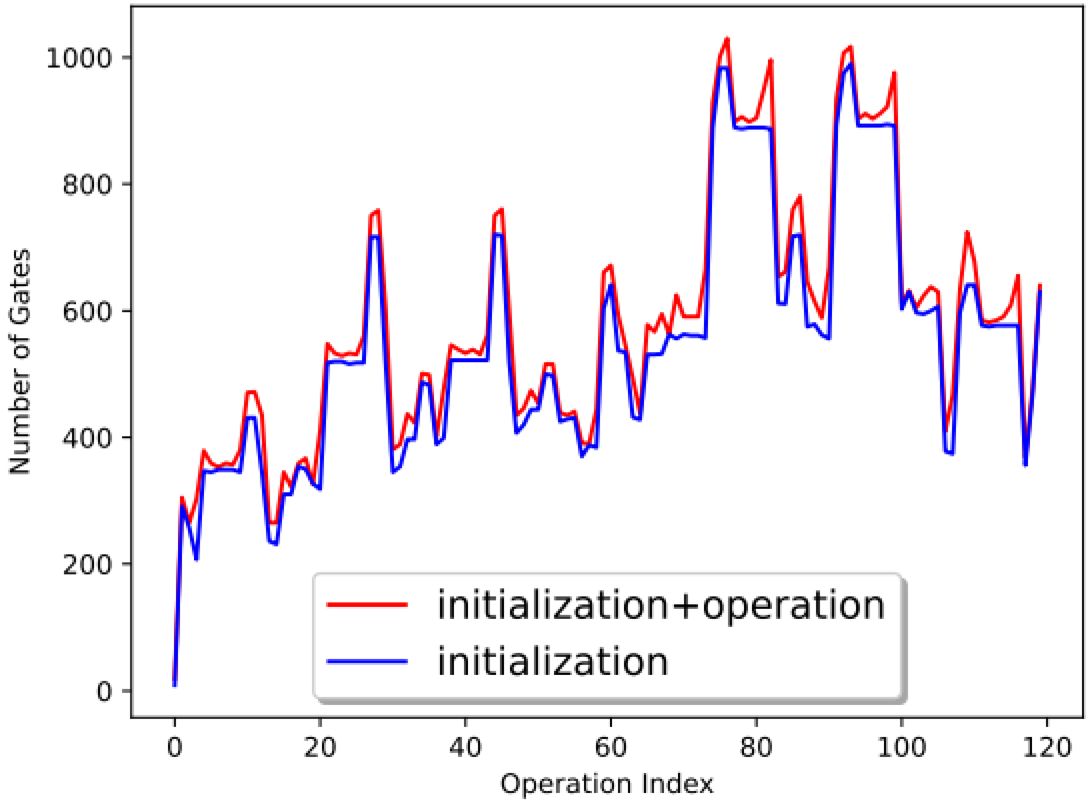
- Unfortunately, Quantum Phase Estimation is not applicable in NISQ hardware
  - IBM Jakarta processor: Median CNOT error = 8.193×10^{-3} (from IBM Lab)
  - 100 CNOT operations in sequence will cause the accuracy to drop below 50%
  - The HHL algorithm requires (tens of) thousands of gates, far beyond the capability of current hardware
- If the whole HHL circuit is too deep, what if we apply only one operation at a time?

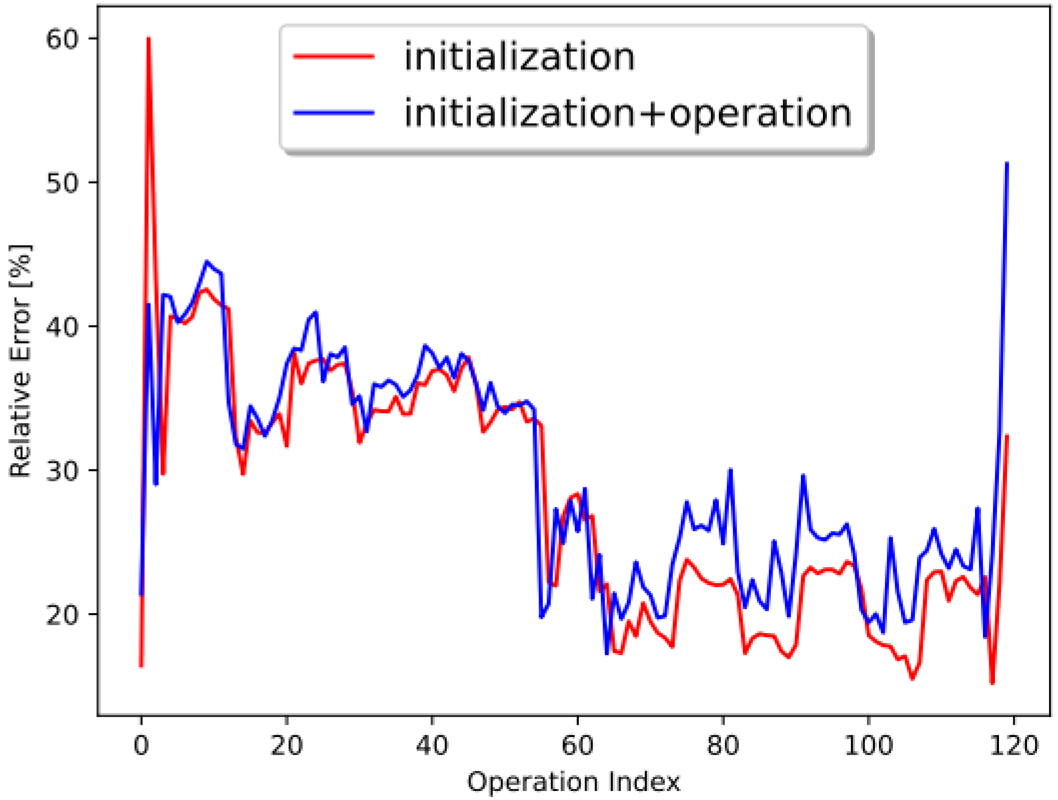


IV. QWU-HHL on NISQ Hardware

- We use an elementary 2×2 matrix equation as an example
  - $A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}, |b\rangle = |1\rangle$ as the equal superposition of the eigenvector
  - 7-qubit IBM Jakarta, 1 qubit for all registers except the clock register, which have 2 qubits
  - Eigenphases of W: 0.00, 0.01(repeated twice), 0.10. Exactly representable using 2 qubits
  - 120 fundamental gates in the entire QWU-HHL circuit
- We execute 120 circuits and record the results at the two checkpoints using both an ideal simulator and Jakarta
- On Jakarta, the subcircuit is decomposed into 6 basis gates CX, I, Rz, $\sqrt{X}$, X, and if-else

IV. QWU-HHL on NISQ Hardware





- The final vector is less than 50% accurate
- The initialization is the main error source in the subcircuit
- The re-initialization and division routine is not effective with Qiskit's default initialization functionality

Summary and Future Work

- The QWU-HHL is an improvement to the classical HHL which removes the necessity of Hamiltonian simulation by choosing the quantum walk unitary
- The implementation of HHL is not meaningful in current noisy hardware
- The re-initialization and division routine is not effective with Qiskit's default initialization functionality, more sophisticated initialization schemes need to be investigated
- NISQ-specific HHL [Yalovetzky et al, 2021, https://arxiv.org/abs/2110.15958] is worth studying for deploying HHL to hybrid quantum-classical systems