Quantum Algorithm For Solving Linear Systems Of Equations

Authors: Harrow, A.W., Hassidim, A. and Lloyd, S. (2009)

Peddinti Dheeraj

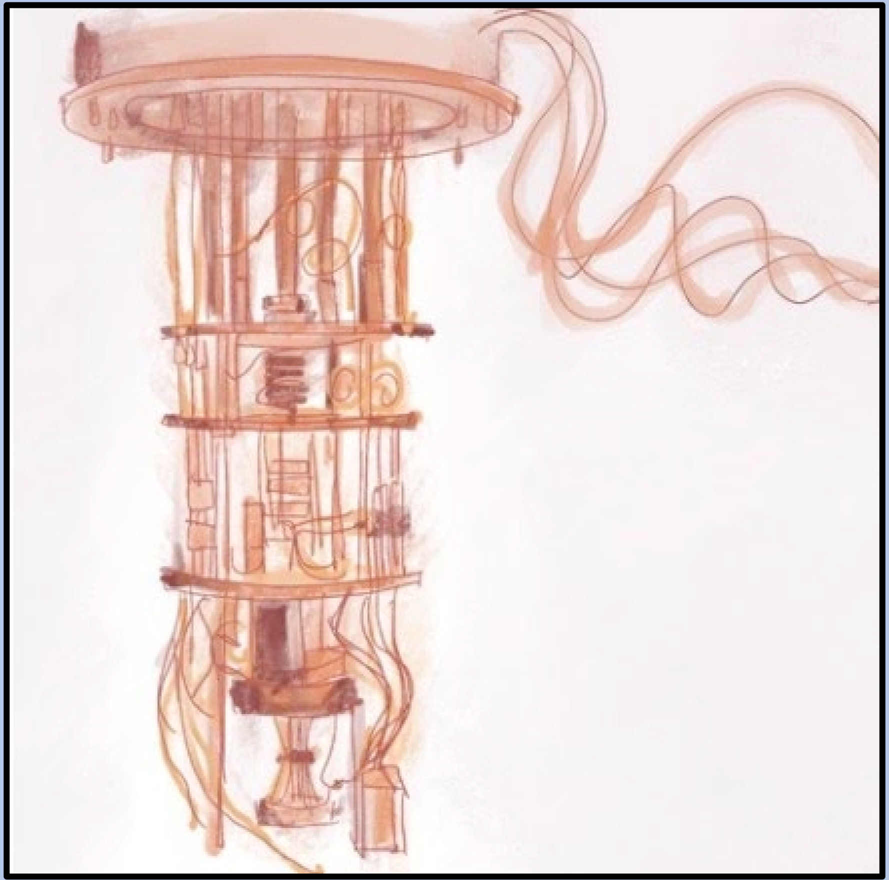
CSE Case Studies FS 22

Outline

- Motivation
- HHL Algorithm
- Results
- Caveats

$$A\vec{x} = \vec{b}$$

?



Algorithm 1 HHL algorithm.

Input: Quantum state $|b\rangle$, Unitary e^{iAt}
Output: Quantum state $|\tilde{x}\rangle \approx A^{-1}|b\rangle$
Algorithm Start:
1. Prepare the initial quantum state: $|\psi_0\rangle = |0\rangle^a|0\cdots0\rangle^C|b\rangle^B = |0\rangle^a|0\cdots0\rangle^C \sum_{j=1}^{n} \langle u_j|b\rangle|u_j\rangle^B$
2. Perform the unitary operation $U_{\text{PE}}(A)$ on the state: $|\psi_0\rangle \rightarrow |\psi_1\rangle = |0\rangle^a \sum_{j=1}^{n} \langle u_j|b\rangle|\tilde{\lambda}_j\rangle^C|u_j\rangle^B$
3. Apply a controlled rotation R to the ancilla qubit, controlled by Reg. C: $|\psi_1\rangle \rightarrow |\psi_2\rangle = \sum_{j=1}^{n}\left(\sqrt{1-\frac{\gamma^2}{\tilde{\lambda}_j^2}}|0\rangle + \frac{\gamma}{\tilde{\lambda}_j}|1\rangle\right)^a \langle u_j|b\rangle|\tilde{\lambda}_j\rangle^C|u_j\rangle^B$
4. Uncompute the Reg. C and Reg. B: $|\psi_2\rangle \rightarrow |\psi_3\rangle = \sum_{j=1}^{n}\left(\sqrt{1-\frac{\gamma^2}{\tilde{\lambda}_j^2}}|0\rangle + \frac{\gamma}{\tilde{\lambda}_j}|1\rangle\right)^a |0\cdots0\rangle^C\langle u_j|b\rangle|u_j\rangle^B$

 ▷ γ is a constant that become irrelevant after measurement, owing to the normalization of the projected wave function.

5. *output* ← *MeasureAncilla*()

if output is $|1\rangle$ **then**

  return $|\tilde{x}\rangle \approx \sum_{j=1}^{n} \langle u_j|b\rangle/\lambda_j|u_j\rangle$

   ▷ $|\tilde{x}\rangle$ is stored in the Reg. B

else

  goto Algorithm Start

end if

Algorithm End

State Preparation

↓

Phase Estimation

↓

Controlled Rotation

↓

Inverse Phase Estimation

↓

Measurement

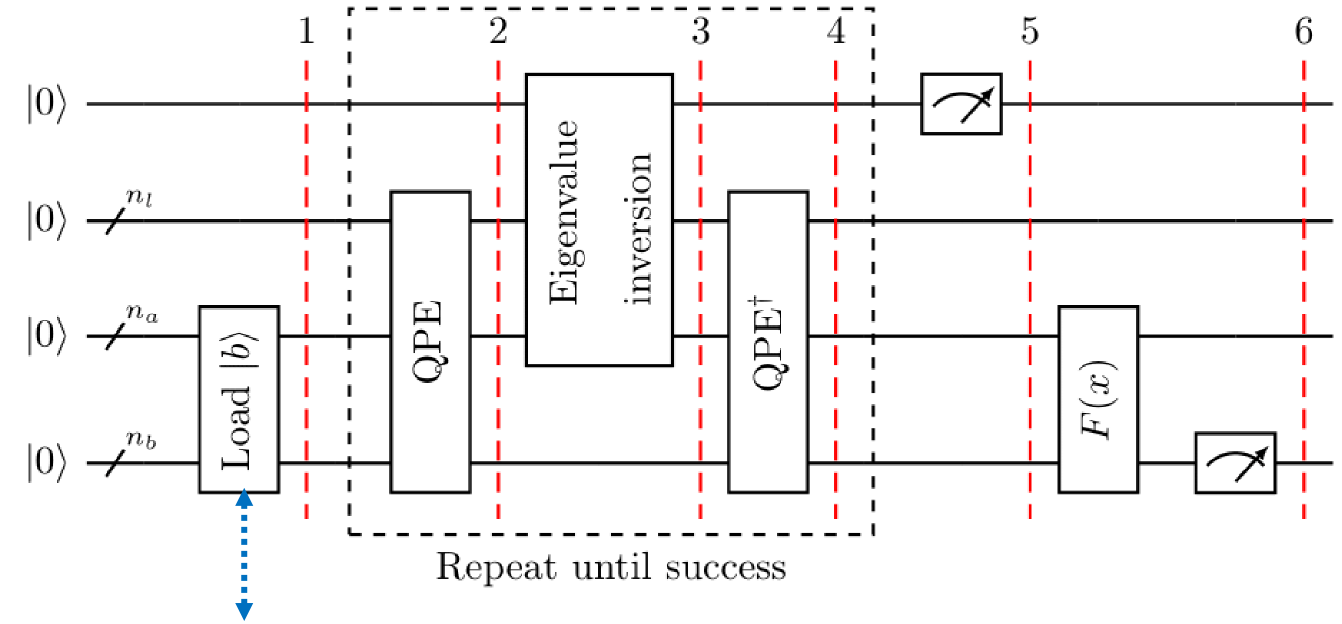
State Preparation

Phase Estimation

Controlled Rotation

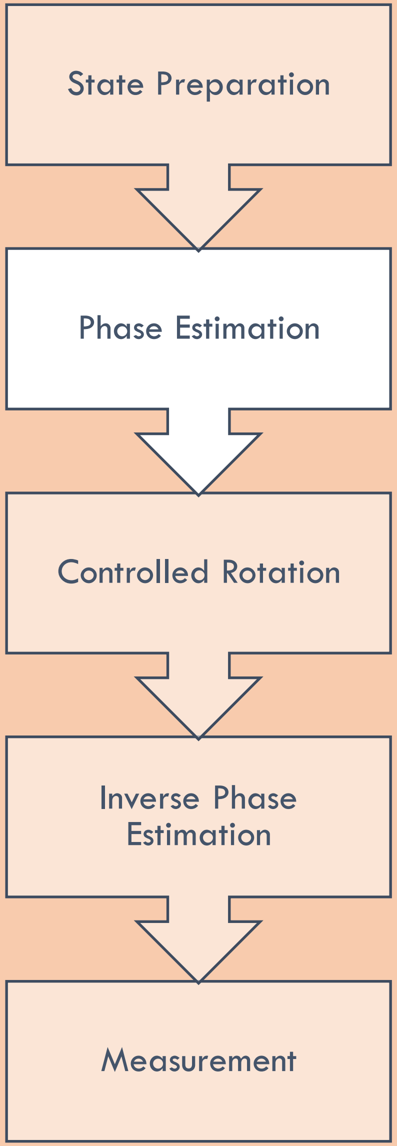
Inverse Phase Estimation

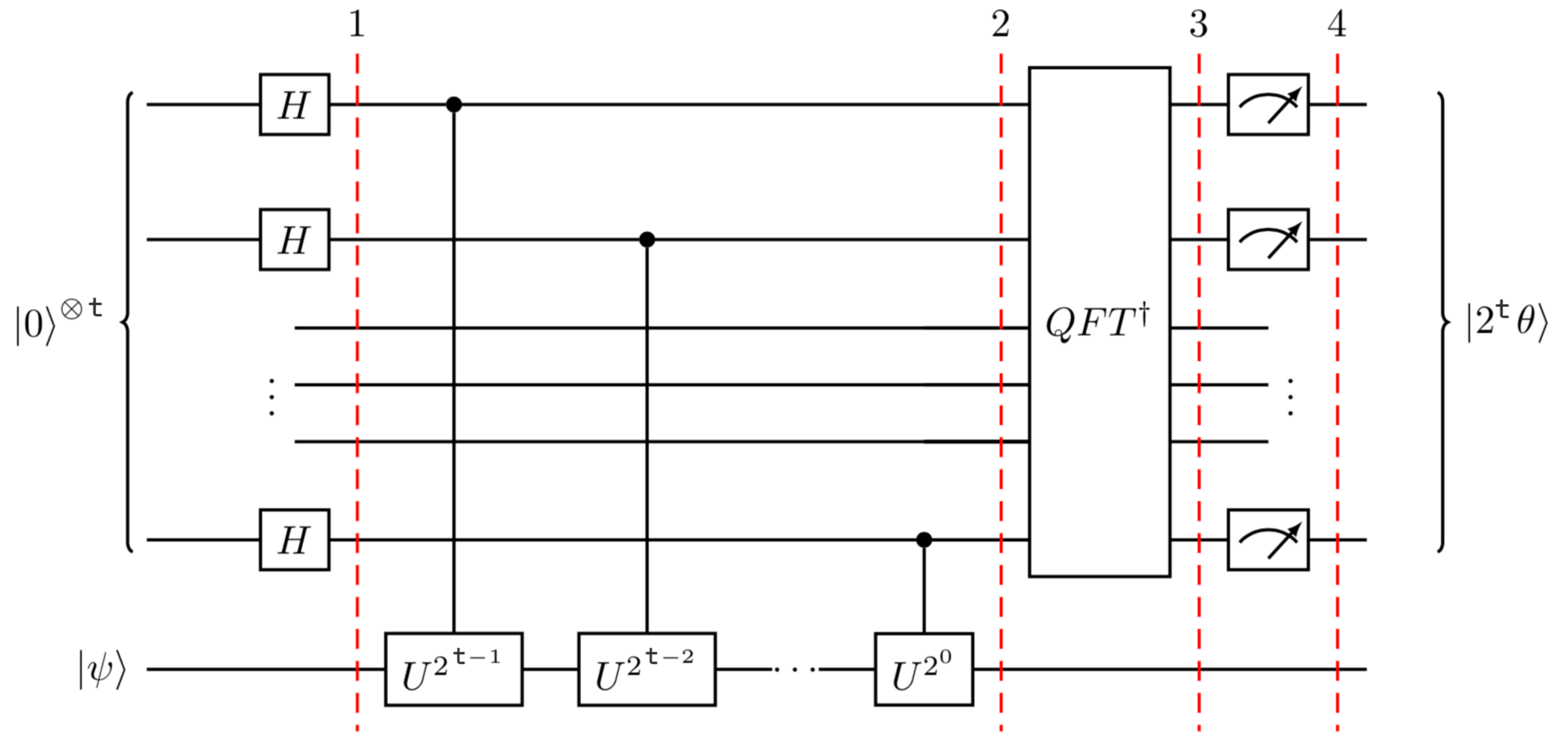
Measurement



$$A\vec{x} = \vec{b}$$

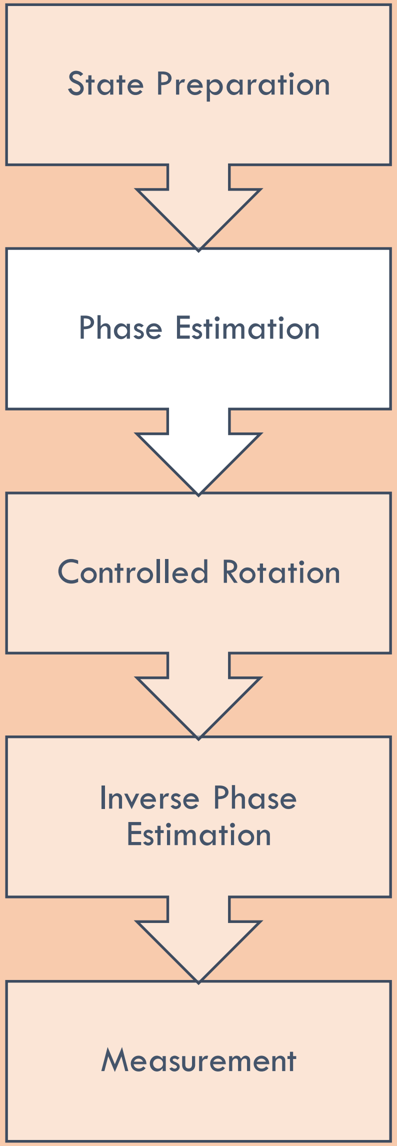
$$|b\rangle = \sum_{i=1}^{N} b_i \, |i\rangle$$





$$U|\psi\rangle = \exp(2\pi i\theta)|\psi\rangle \Rightarrow QPE(U, |0\rangle_t, |\psi\rangle) = |\tilde{\theta}\rangle|\psi\rangle$$

1. Superposition
2. Phase Kickback
3. Inverse QFT
4. Measurement



Transformation of basis:

Computational basis → Eigen basis of A

$$e^{iAt} = \sum_{j=0}^{N-1} e^{i\lambda_j t} |u_j\rangle\langle u_j|$$

$$\mathrm{QPE}\left(e^{iAt}, \sum_{j=0}^{N-1} b_j |0\rangle_{n_l} |u_j\rangle_{n_b}\right) = \sum_{j=0}^{N-1} b_j |\lambda_j\rangle_{n_l} |u_j\rangle_{n_b}$$

State Preparation → Phase Estimation → Controlled Rotation → Inverse Phase Estimation → Measurement

Conditioned rotation → Eigenvalue inversion

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle\langle u_j|, \quad \lambda_j \in \mathbb{R} \longrightarrow A^{-1} = \sum_{j=0}^{N-1} \lambda_j^{-1} |u_j\rangle\langle u_j|$$

$$|x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} \lambda_j^{-1} b_j |u_j\rangle$$

$$\sum_{j=0}^{N-1} b_j |\lambda_j\rangle_{n_l} |u_j\rangle_{n_b} \longrightarrow \sum_{j=0}^{N-1} b_j |\lambda_j\rangle_{n_l} |u_j\rangle_{n_b} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle \right)$$

Not an unitary! Possibility of failure.

State Preparation

Phase Estimation

Controlled Rotation

Inverse Phase Estimation

Measurement

Transformation of basis:
Eigen basis of A → Computational basis

$$\sum_{j=0}^{N-1} b_j |\lambda_j\rangle_{n_l} |u_j\rangle_{n_b} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle \right)$$

$$\Big\downarrow QPE^\dagger$$

$$\sum_{j=0}^{N-1} b_j |0\rangle_{n_l} |u_j\rangle_{n_b} \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle \right)$$

State Preparation

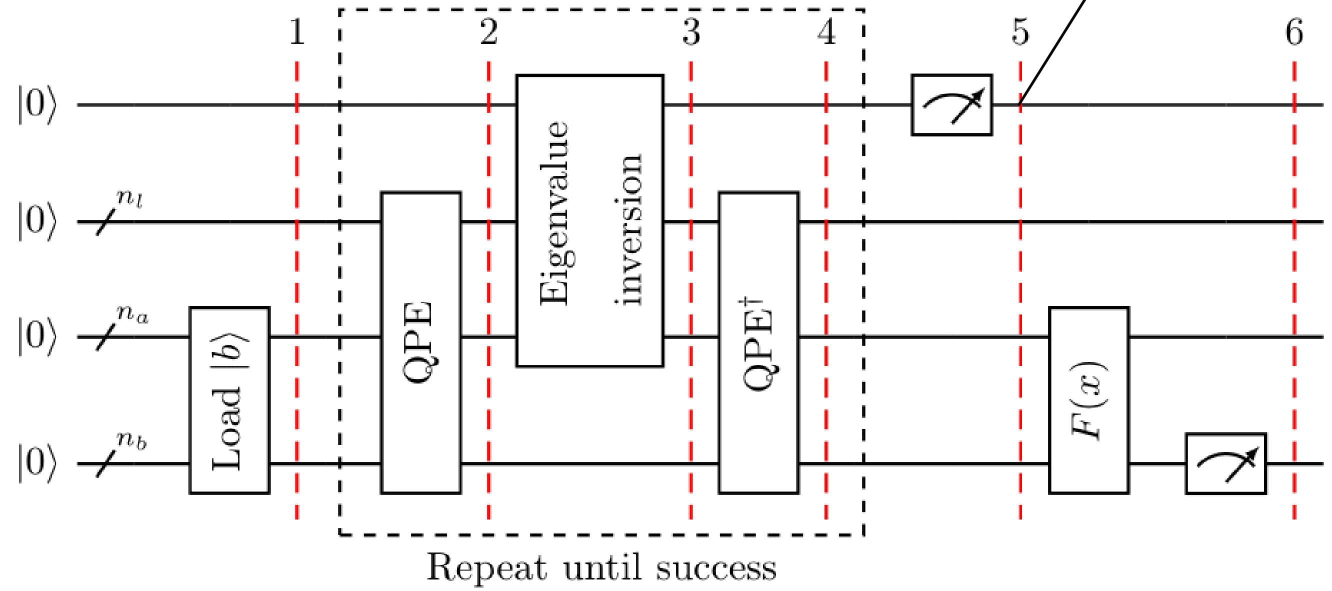
Phase Estimation

Controlled Rotation

Inverse Phase Estimation

Measurement

If measured state is 1 $\Rightarrow$ Success.



$$F(x) := \langle x|M|x\rangle$$

#1. Runtime improvement over best classical algorithm for s-sparse matrices with condition number κ.

Conjugate Gradient (w/ a p.d. matrix)	HHL Algorithm (w/ a Hermitian matrix)
$\mathcal{O}(Ns\sqrt{\kappa}\log\left(\frac{1}{\epsilon}\right))$	$\mathcal{O}(\log(N)\, s^2\kappa^2/\epsilon)$

#2. Improving probability of success using Amplitude amplification.

#3. Extension of the algorithm to ill-conditioned and non-sparse matrices.

- Exponentially better in runtime, but exponentially worse in error of the outcome.
- Speedups hold only for the measurement of a summary statistic of the solution.
- Relies on other subroutines and existence of oracles for:
  - Proving optimality of error bounds
  - Preparation of states and unitary for Hamiltonian simulation

Further Reading

Thank you!