The Quantum Linear Problem

And its Quantum Linear Solver (QLS) Algorithm

A PH4415 Final Year Project Presentation By

Chee Chong Hian

chee0122@ntu.edu.sg

Nanyang Technological University (NTU)

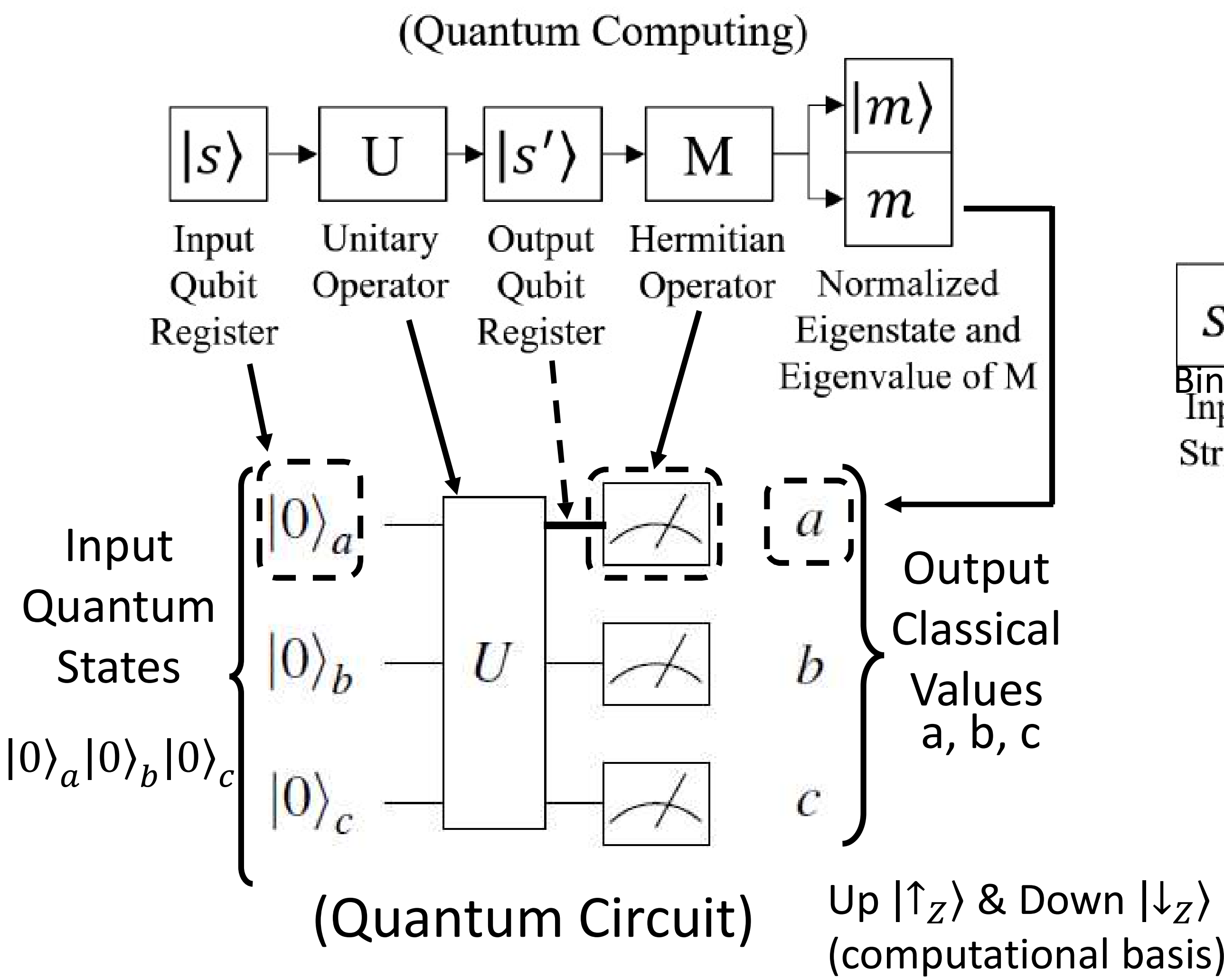
School of Physical and Mathematical Sciences (SPMS)

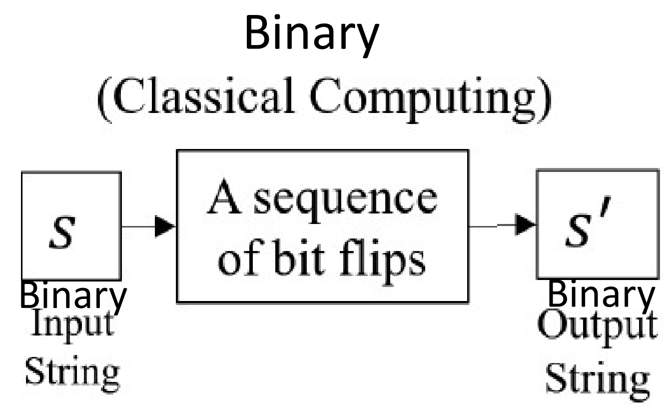
Contents

Abbreviations

- HHL : Aram Harrow, Avinatan Hassidim and Seth Lloyd
- QLS : Quantum Linear Solver

Quantum Computation





The Linear Problem

Classical Linear Problem

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

Quantum Linear Problem

$$|b\rangle = A|x\rangle$$

Vectors $\quad \vec{b} \xrightarrow{Normalised} |b\rangle \quad$ Quantum States

Matrix $\quad A \xrightarrow[Interpretation]{Quantum} \begin{cases} U & \text{Unitary Operators} \\ M & \text{Hermitian Operators (Measurement)} \end{cases}$

Solve:

Obtain Classical Vector

$$\vec{x} = A^{-1}\vec{b}$$

Solve:

Obtain Quantum State

$$|x\rangle = \begin{cases} U^{-1}|b\rangle = U^{\dagger}|b\rangle & \text{(Easy)} \\ M^{-1}|b\rangle & \text{(??? → FYP)} \end{cases}$$

Resolving $|x\rangle = M^{-1}|b\rangle$

Problem: M^{-1} is Hermitian (still a measurement ops)

- Does not preserve norm → Mathematically incorrect
- Results in a measurement collapse → Physically incorrect

Solution:

- Renormalize : $|x\rangle = \frac{1}{d} M^{-1}|b\rangle = \frac{1}{d}\sum_{j=1}^{2^n} \frac{\beta_j}{\lambda_j} \left|m_j\right\rangle$

  d : norm factor.

  $\left|m_j\right\rangle$: orthonormalized eigenstates of M.

  β_j : complex coefficients.

  λ_j : eigenvalues of M.

- Get $|x\rangle = \frac{1}{d}\sum_{j=1}^{2^n} \frac{\beta_j}{\lambda_j} \left|m_j\right\rangle$ by a measurement collapse of U$|b\rangle$.
  - as implicitly developed by HHL (Algorithmic Ideas)

Goal: Get $|x\rangle = \frac{1}{d}\sum_{j=1}^{2^n}\frac{\beta_j}{\lambda_j}|m_j\rangle$ by a measurement collapse of $U|b\rangle$.

Algorithmic Ideas

First Idea – a simple algorithm

1. Append an ancilla $|0\rangle_a$ to input state $|b\rangle_m \rightarrow |b\rangle_m|0\rangle_a$
2. Apply some unitary U (should contain M),

$\boxed{\text{Change the amplitudes of } |m_j\rangle \text{ in } |b\rangle \text{ to } \frac{\beta_j C}{\lambda_j}}$

$$U|b\rangle_m|0\rangle_a = \underbrace{\sum_{j=1}^{2^n}\frac{\beta_j C}{\lambda_j}|m_j\rangle_m|1\rangle_a}_{d|x\rangle_m|1\rangle_a} + \underbrace{\sum_{j=1}^{2^n}\beta_j\sqrt{1-\left(\frac{C}{\lambda_j}\right)^2}|m_j\rangle_m|0\rangle_a}_{\text{Some other orthogonal state}}$$

where C is a free real parameter (does not affect state)

3. Measure ancilla qubit in computational basis $\{|0\rangle, |1\rangle\}$.
4. Get $|x\rangle_m$ if ancilla $= |1\rangle_a$ with probability $\sum_{j=1}^{2^n}\left|\frac{\beta_j C}{\lambda_j}\right|^2$.

Algorithmic Ideas
Second Idea

To implement U (Quantum If-Else):

Requires information about M

- Apply $\overbrace{\text{controlled-}R_y(\theta_j)}^{C[R_y(\theta_j)]}$ gate on $|0\rangle_a$, conditioned on $\overbrace{\text{eigenstates } |m_j\rangle_m}^{\text{Requires information about M}}$, where $\frac{\theta_j}{2} = \sin^{-1}\left(\frac{C}{\lambda_j}\right)$ (???).

$$|b\rangle_m|0\rangle_a \xrightarrow{C[R_y(\theta_j)]} \underbrace{\sum_{j=1}^{2^n} \beta_j \sin\left(\frac{\theta_j}{2}\right)|m_j\rangle_m|1\rangle_a}_{d|x\rangle_m|1\rangle_a} + \underbrace{\sum_{j=1}^{2^n} \beta_j \cos\left(\frac{\theta_j}{2}\right)|m_j\rangle_m|0\rangle_a}_{\text{Some other orthogonal state}}$$

Controlled-$R_y(\theta_j)$ on $|b\rangle|a\rangle$:
(Do in a quantum superposition)

- If $|b\rangle = |m_1\rangle$, then $R_y(\theta_1)|a\rangle$
- Else if $|b\rangle = |m_2\rangle$, then $R_y(\theta_2)|a\rangle$
- Else if …

Note: $R_y(\theta_j) = \begin{pmatrix} \cos\left(\frac{\theta_j}{2}\right) & -\sin\left(\frac{\theta_j}{2}\right) \\ \sin\left(\frac{\theta_j}{2}\right) & \cos\left(\frac{\theta_j}{2}\right) \end{pmatrix}$

(in the computational basis)

(Easier if using Boolean conditions and not eigenvalues $|m_j\rangle$)

Algorithmic Ideas

Third Idea

eigenvalue qubits

$2 \rightarrow 10 \rightarrow |10\rangle$

binary

$|\lambda_j\rangle_e$: qubit binary representation of eigenvalues of M. (e.g. Boolean qubits)

$|1\rangle_e$: True, $|0\rangle_e$: False

Use $|\lambda_j\rangle_e$ (in $|1\rangle_e$, $|0\rangle_e$) as your Quantum If-Else conditions:

- Apply $C[R_y(\theta_j)]$ on $|0\rangle_a$ conditioned on $|\lambda_j\rangle_e$ in an appended eigenvalue register. → (Ancillary Rotation step, AR)

- Get $|\lambda_j\rangle_e$ by Quantum Phase Estimation (QPE).

Insert M

$$|m_j\rangle_m|\mathbf{0}\rangle_e \xrightarrow{QPE} |m_j\rangle_m|\lambda_j\rangle_e$$

$$|b\rangle_m|\lambda_j\rangle_e|0\rangle_a \xrightarrow{AR} \underbrace{\sum_{j=1}^{2^n}\beta_j \sin\left(\frac{\theta_j}{2}\right)|m_j\rangle_m|\lambda_j\rangle_e|1\rangle_a}_{\sum_{j=1}^{2^n}\frac{\beta_j}{\lambda_j}|m_j\rangle_m|\lambda_j\rangle_e|1\rangle_a} + \sum_{j=1}^{2^n}\beta_j \cos\left(\frac{\theta_j}{2}\right)|m_j\rangle_m|\lambda_j\rangle_e|0\rangle_a$$

$$\sum_{j=1}^{2^n}\frac{\beta_j}{\lambda_j}|m_j\rangle_m|\lambda_j\rangle_e|1\rangle_a \rightarrow \boxed{\text{uncompute } |\lambda_j\rangle_e} \rightarrow d|x\rangle_m|\mathbf{0}\rangle_e|1\rangle_a$$

Algorithmic Ideas

Summary

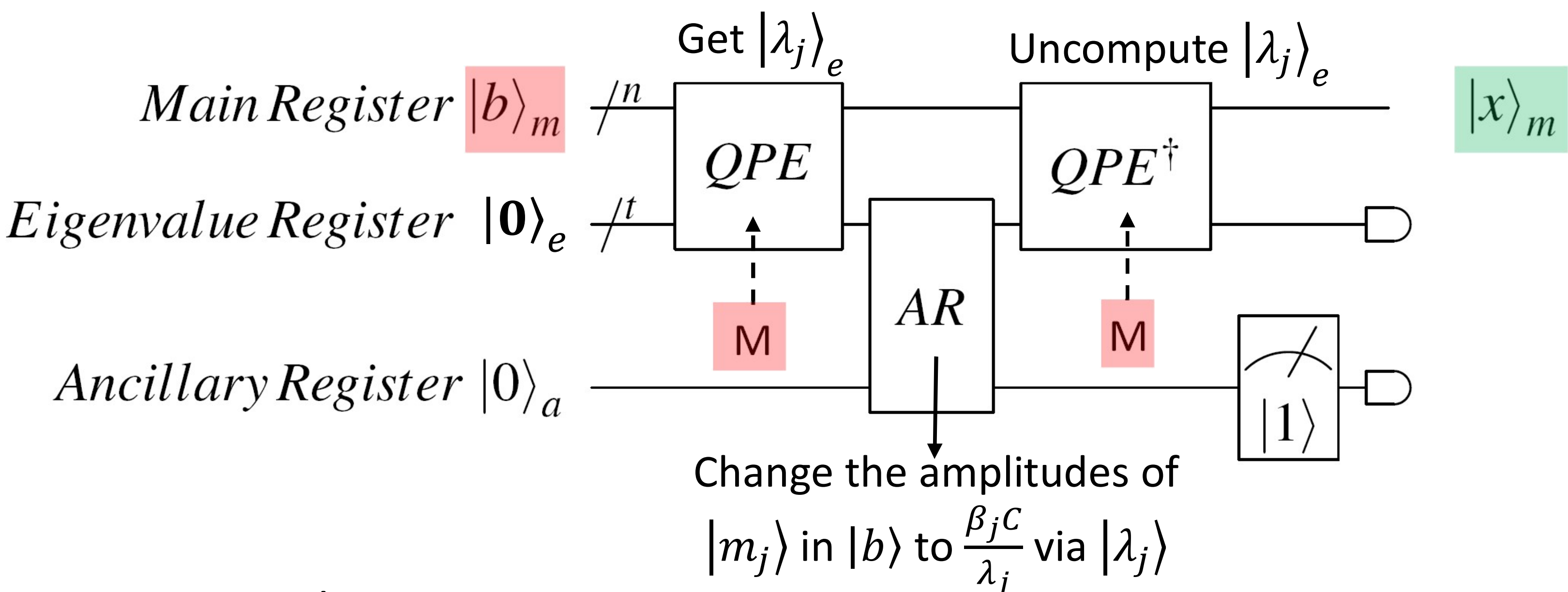
1. Get $|x\rangle = \frac{1}{d}\sum_{j=1}^{2^n}\frac{\beta_j}{\lambda_j}\left|m_j\right\rangle$ by a measurement collapse of $\mathrm{U}|b\rangle$.

   Recall: (Main Purpose of U)
   Change the amplitudes of $\left|m_j\right\rangle$ in $|b\rangle$ to $\frac{\beta_j C}{\lambda_j}$.

2. Construct U operator Using Quantum If-Else.
3. Use qubit-binary $\left|\lambda_j\right\rangle_e$, eigenvalues of M as your If-Else conditions.

Original HHL QLS Algorithm

For 2^n quantum linear problem d$|x\rangle = M^{-1}|b\rangle$, given M and $|b\rangle$, run the following quantum algorithm to obtain $|x\rangle$ with probability $\sum_{j=1}^{2^n} \left|\frac{\beta_j C}{\lambda_j}\right|^2$.



Quantum Circuit Implementation:

- QPE : (Quantum Phase Estimation) By Alexey Kitaev and Richard Cleve et al.
- AR : (Ancillary Rotation) Not given by HHL.

Original QLS Problems

- HHL's main purpose was to prove its exponential computation speedup by
  - Assuming M is sparse.
  - Assuming input $|b\rangle$ can be efficiently prepared.
  - Treating QPE and AR as black boxes oracle subroutines.
- However, the original paper did not elaborate on implementing AR subroutine.
- Naïve and Direct AR requires one to know the eigenvalues λ_j to calculate rotation angles $\frac{\theta_j}{2} = \sin^{-1}\left(\frac{C}{\lambda_j}\right)$
- QLS modifications (Rule-breaking Implementations):
  - Eigenvalue Marking Modification. → Simplified QLS
  - 1st order Taylor Approximation of AR. → Full QLS

Full QLS Algorithm

Motivation:

- λ_j Eigenvalues are not given by the problem.
- Need to find an AR method that implements controlled-$R_y(\theta_j)$ on $|0\rangle_a$ conditioned on $|\lambda_j\rangle_e$ without need of any information of M or $|b\rangle$ except C.

Solution:

- Use 1st order Taylor Approximation of AR inspired by Yudong Cao et al

$$\theta_j = 2\sin^{-1}\left(\frac{C}{\lambda_j}\right) \approx 2\left(\frac{C}{\lambda_j}\right), \qquad \left|\frac{C}{\lambda_j}\right| \ll 1$$

- Assume eigenvalues λ_j are powers of two.
  - (Practical and Easier Implementation only, otherwise not needed).

Full QLS Algorithm

1^{st} order approximation of Ancillary Rotation (AR)

- Idea: <u>Break up</u> Quantum If-Else, $C\left[R_y(\theta_j)\right]$, so that angles need not be calculated

- Apply 1^{st} order approximation: $\theta_j = 2\sin^{-1}\left(\frac{C}{\lambda_j}\right) \approx 2\left(\frac{C}{\lambda_j}\right)$.

- Then, Rotation-Y gate is approximated: $R_y(\theta_j) \approx R_y\left(\frac{2C}{\lambda}\right)$.

- Let $C = \frac{\pi}{2^r}$ and $y = \frac{1}{\lambda}$, where in binary $y = y_t 2^0 + \cdots + y_j 2^{j-t} + \cdots + y_1 2^{1-t}$.

  $y_j = \{0,1\}$

  *Introduce a new parameter r

- Substitute into $R_y\left(\frac{2C}{\lambda_j}\right)$ and <u>break up</u> using binary,

$$R_y\left(\frac{2C}{\lambda}\right) = R_y\left(\frac{2\pi}{2^r} y_t \cdot 2^0\right) \cdots R_y\left(\frac{2\pi}{2^r} y_j \cdot 2^{j-t}\right) \cdots R_y\left(\frac{2\pi}{2^r} y_1 \cdot 2^{1-t}\right)$$

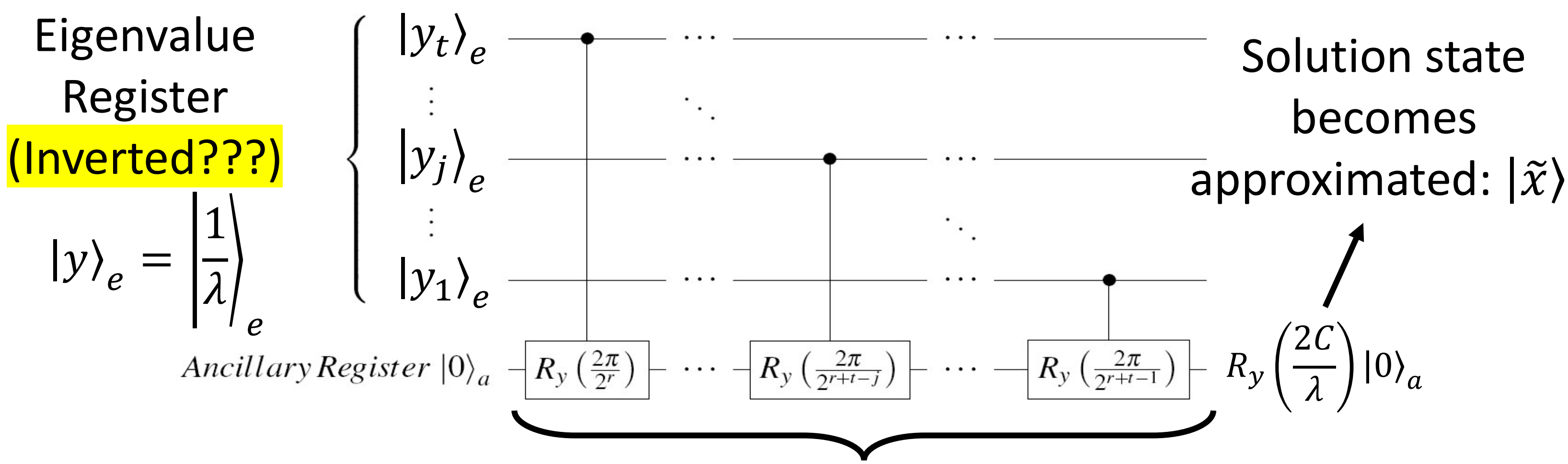
Full QLS Algorithm

1^{st} order approximation of AR

- Create a broken up and approximated Quantum If-Else,

$$C\left[R_y\left(\frac{2C}{\lambda}\right)\right] = \cdots C\left[R_y\left(\frac{2\pi}{2^r}y_j \cdot 2^{j-t}\right)\right]\cdots$$

- Then, 1^{st} order approximation of AR can be implemented as follows.



Quantum Gates are independent of eigenvalues

Full QLS Algorithm
Eigenvalue Inversion (EI)

- The 1st order AR demands inverted eigenvalues $|y\rangle_e = \left|\frac{1}{\lambda}\right\rangle_e$.
- Therefore, there is a need to use general <u>quantum inversion</u>
  - -> A collection of quantum arithmetic subroutines. (Very Difficult)

Simplification:

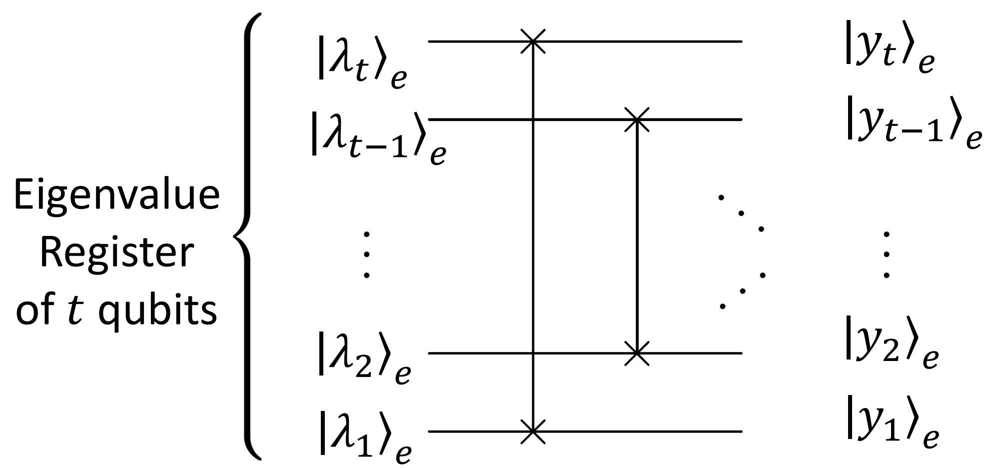
- Use the simplest quantum inversion - -> Powers of two inversion.

Idea:

- Inverting an eigenvalue that is a power of two is the same as swapping the positions of the binary sequence about the middle.

For Example,

$$4 \to \frac{1}{4}, 100 \to 0.01 \qquad 8 \to \frac{1}{8}, 1000 \to 0.001$$



Full QLS Algorithm

4D Case – Worked Example

Obtain $|x\rangle = \frac{1}{d} M^{-1} |b\rangle$ approximately,

Given that

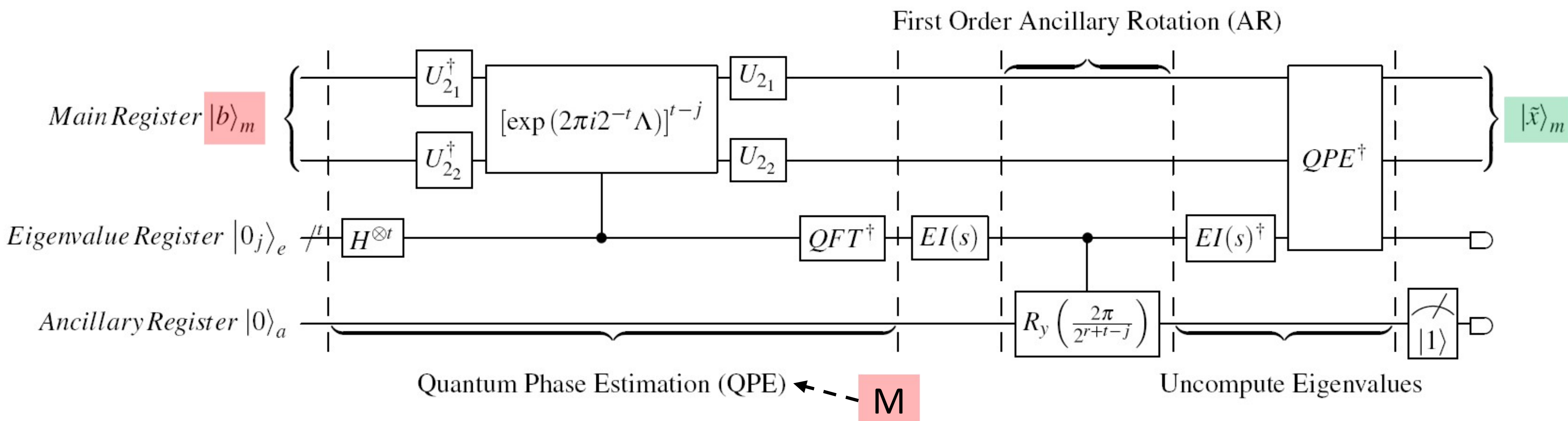
- $|b\rangle = |+\rangle \otimes |+\rangle$
- $M = \Gamma\Lambda\Gamma^\dagger$, where $\Gamma = (U_2 \otimes U_2)$, $\Lambda = diag(\lambda_j)$
- Eigenvalues $\lambda_j = 2^{j+1}$, j $= 0,1,2,3.$
- Let $C = \frac{\pi}{2^r}$

$$M = \frac{1}{2}\begin{pmatrix} 15 & -5i & -9i & -3 \\ 5i & 15 & 3 & -9i \\ 9i & 3 & 15 & -5i \\ -3 & 9i & 5i & 15 \end{pmatrix}$$

$$U_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$|x\rangle = \frac{1}{2\sqrt{85}}\begin{pmatrix} 6+7i \\ 9+2i \\ 9-2i \\ 6-7i \end{pmatrix}$$

$$|\tilde{x}\rangle = \sqrt{\frac{1}{\sum_{j=1}^{4}\sin^2\left(\frac{\pi}{2^{r+j}}\right)}}\begin{pmatrix} -\frac{i}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) + \frac{i}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \\ \frac{1}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) + \frac{i}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \\ \frac{1}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) + \frac{i}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \\ \frac{i}{2}\sin\left(\frac{\pi}{2^{r+4}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+3}}\right) + \frac{1}{2}\sin\left(\frac{\pi}{2^{r+2}}\right) - \frac{i}{2}\sin\left(\frac{\pi}{2^{r+1}}\right) \end{pmatrix}$$



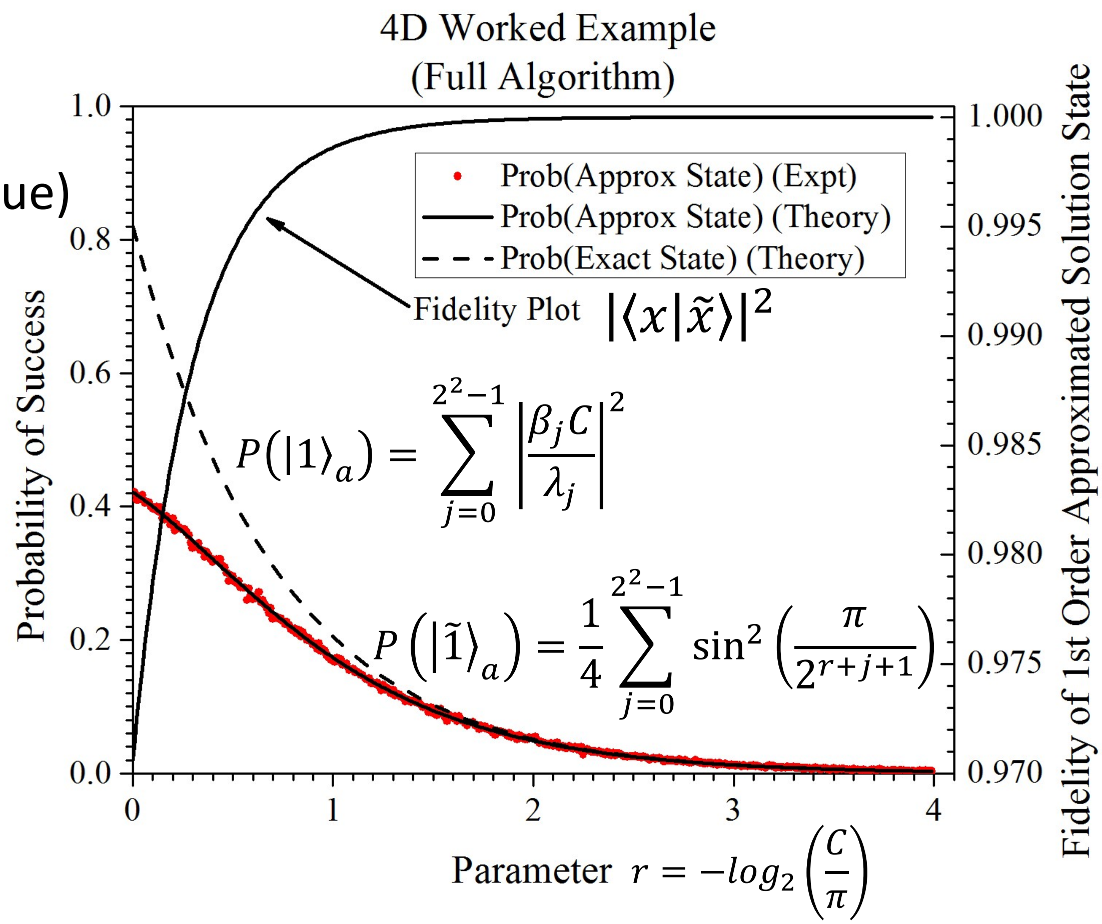
Full QLS Algorithm

4D Case – Worked Example Results

IBM Qiskit QASM Simulator

(10000 runs/ r value)

- Qubits: 9
- Quantum Operations : 812
- Average Runtime : $\sim 10^{0} s$
- Classical Runtime (Numpy Solver) : $\sim 10^{-5} s$
- Experimental value error : $\pm$ 9% (w.r.t theoretical value)



Large C → Small C

Full QLS Algorithm
Analysis

Advantages:

- No need to know of Eigenvalues and eigenstates
- It can be used to solve for any 2^nD complex M and $|b\rangle$.

Disadvantages:

- Implementing Eigenvalue Inversion (EI) subroutine is complicated and very difficult.
- Higher n^{th} order AR requires $O(t^n)$ gates.
- Predicting probability of success needs eigenvalues.

Caveat:

- Full QLS Algorithm can be implemented,
  if $\underbrace{\text{controlled-}exp(2\pi i 2^{-t}M)\text{ quantum gates}}_{\text{Crucial Gate for QPE!!!}}$ are given.

Future Research

Full QLS Applications:

- Quantum Data Fitting.
- Quantum State Tomography.

QLS Extensions:

- Replace any subroutines in the QLS with another one as long the input-output are the same.
  - E.g. QPE, AR & EI.
- Add other quantum subroutines to increase the probability of success or improve computation time.
  - E.g. Grover search algorithm

Thank You

Questions & Answers

More FYP Contents:

- Background on Quantum Computation
- Literature Review on Quantum Linear Solvers (QLS)
- Detailed descriptions and explanations on Simplified and Full QLS
  - More Worked Examples: 2D, 4D and 2^nD
- In-depth Discussions and Analysis.

Below are my experiment codes (python jupyter notebook):

- Simplified QLS Algorithm

  https://gist.github.com/cheechonghian/ea7ed914a264444b540aa57c428df497

- Full QLS Algorithm

  https://gist.github.com/cheechonghian/fc1c1a321f1e36b216eb7615d7edba65

- Anaconda Environment

  https://anaconda.org/chee0122/myQuantumCircuitSimulator2/files

Simplified HHL QLS Algorithm
Eigenvalue Marking Modification

Idea by Stefanie Barz et al:

- Modified AR: Apply $C\left[R_y\left(\theta_j\right)\right]$ on $|0\rangle_a$, conditioned on a partial sequence of the binary qubits of $\left|\lambda_j\right\rangle_e$.
- Modified QPE: Truncate QPE (Eigenvalue Marking, EM), to generate partial $\left|\lambda_j\right\rangle_e$(Eigenvalue marking qubits).

Eigenvalue Assumption:

- Modification is possible if the all eigenvalues λ_j have the same binary sequence X up to the last n bits, where the last n bits are index j in binary.

t : No. of bits needed to specify the largest eigenvalue.

$$\lambda_j \xRightarrow{\text{Assume in binary}} 0.\underbrace{X}_{t-n \text{ bits}}\underbrace{j}_{n \text{ bits}}$$

Also, assume knowing the eigenvalues.

Simplified HHL QLS Algorithm

4D Case – Truncated QPE (EM)

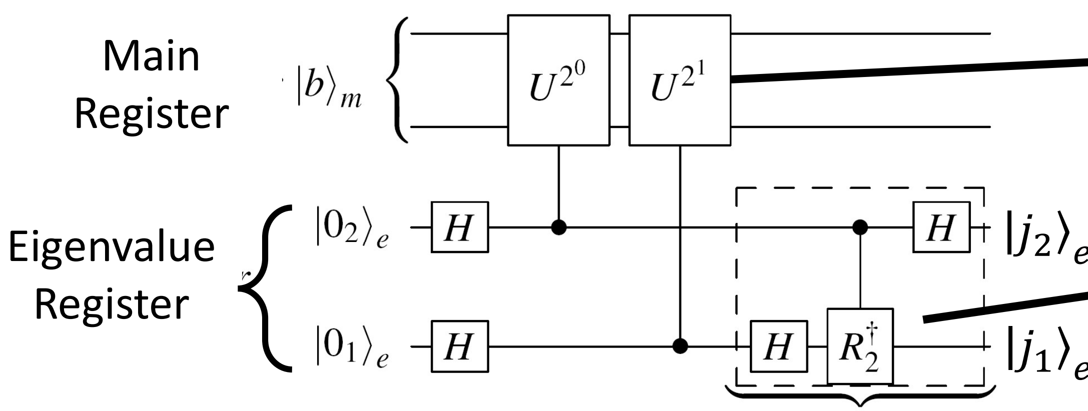
- Four eigenvalues are as follows,

$$\lambda_0 = 0.X00, \quad \lambda_1 = 0.X01$$
$$\lambda_2 = 0.X10, \quad \lambda_3 = 0.X11$$

for some binary sequence X.

- Apply truncated QPE (EM),

$$|m_0\rangle_m |00\rangle_e \rightarrow |m_0\rangle_m |00\rangle_e$$
$$|m_1\rangle_m |00\rangle_e \rightarrow |m_1\rangle_m |01\rangle_e$$
$$|m_2\rangle_m |00\rangle_e \rightarrow |m_2\rangle_m |10\rangle_e$$
$$|m_3\rangle_m |00\rangle_e \rightarrow |m_3\rangle_m |11\rangle_e$$



$$U = exp(2\pi i 2^{t-2} M)$$

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & exp(2\pi i 2^{-k}) \end{bmatrix}$$

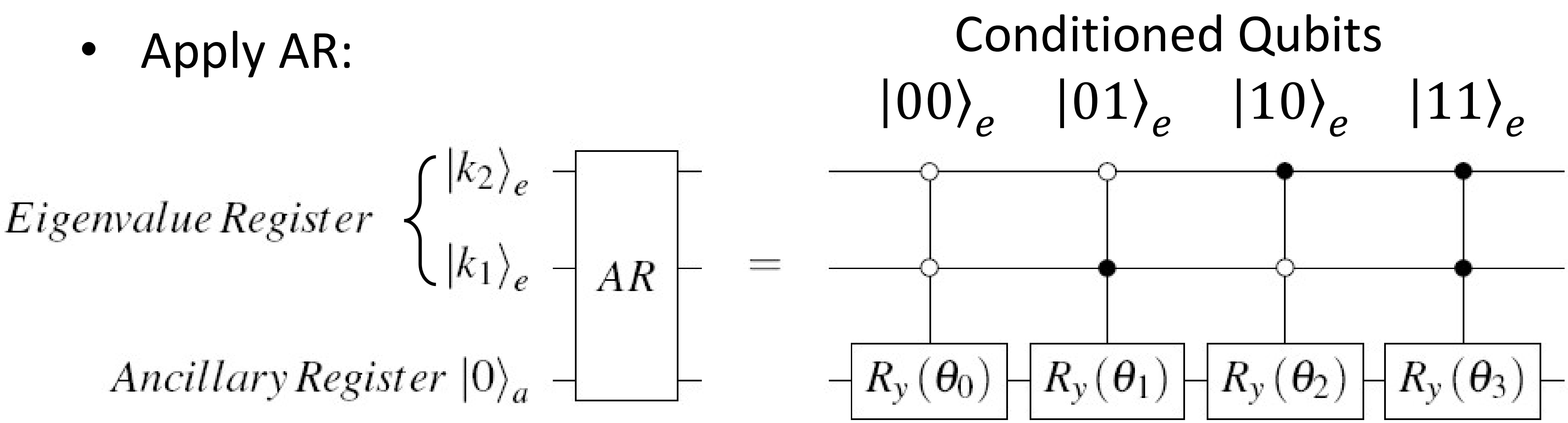
(Phase Gate)

$QFT^\dagger$: Inverse Quantum Fourier Transform

Simplified HHL QLS Algorithm

4D Case – AR

- Apply AR:



Calculate angles $\theta_j = 2\sin^{-1}\left(\frac{C}{\lambda_j}\right)$

- After that, apply inverse EM to un-compute eigenvalue marking qubits.
- Finally, measure and select $|1\rangle_a$ quantum state to obtain $|x\rangle_m = \frac{1}{d}\sum_{j=0}^{2^2-1}\frac{\beta_j}{\lambda_j}|m_j\rangle$ with probability $\sum_{j=0}^{2^n-1}\left|\frac{\beta_j C}{\lambda_j}\right|^2$.

Simplified HHL QLS Algorithm

4D Case – Worked Example

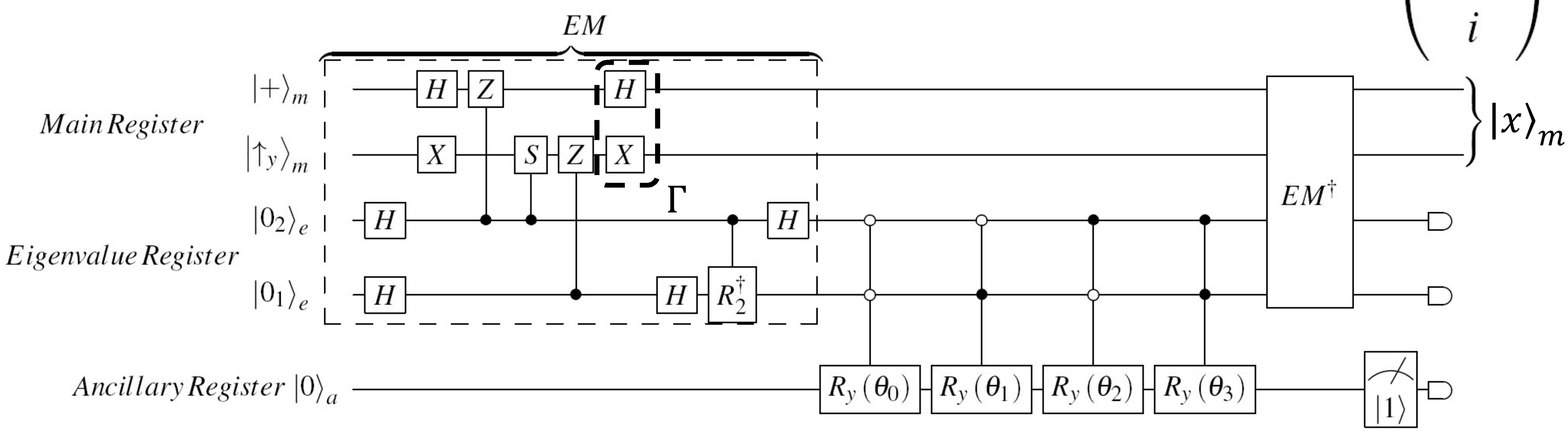
$$M = \frac{1}{16}\begin{pmatrix} 12 & 0 & -2 & 0 \\ 0 & 10 & 10 & -2 \\ -2 & 0 & 12 & 0 \\ 0 & -2 & 0 & 10 \end{pmatrix}$$

Let use a simplified case,

- $M = \Gamma\Lambda\Gamma^{\dagger}$, where $\Gamma = (H \otimes X)$.
- Eigenvalues $\lambda_j = 0.1j$, where $j = 00,01,10,11$, and $X = 1$.
- Input $|b\rangle = |+\rangle \otimes |\uparrow_y\rangle$.
- $C = \frac{1}{2^r}$ (Re-parameterized)

$$\theta_j = 2\sin^{-1}\left(\frac{1}{2^r \cdot 0.1j}\right)$$

$$|x\rangle_m = \sqrt{\frac{25}{82}}\begin{pmatrix} \frac{8}{10} \\ i \\ \frac{8}{10} \\ i \end{pmatrix}$$



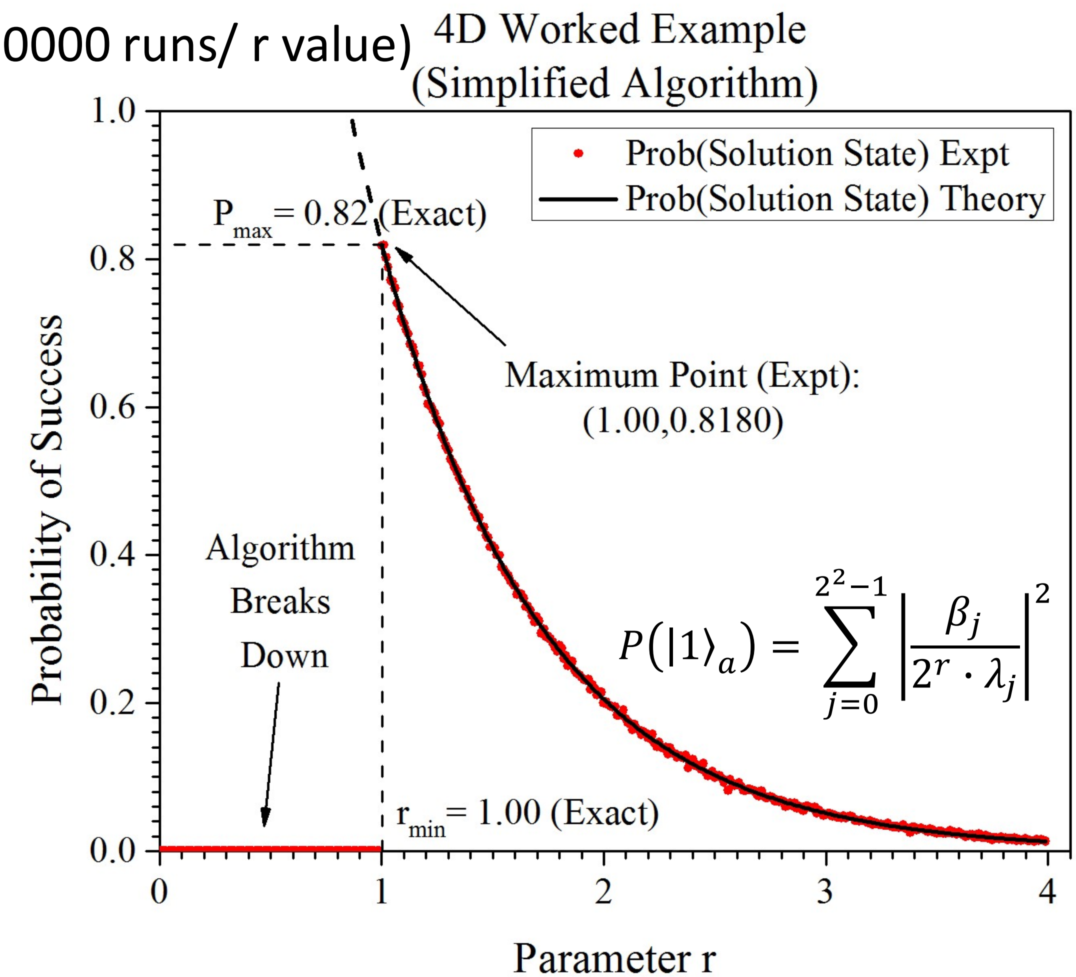
Simplified HHL QLS Algorithm
4D Case – Worked Example Results

IBM Qiskit QASM Simulator (10000 runs/ r value)

- Qubits: 6
- Quantum Operations : 51
- Average Runtime : $\sim 10^{-3}s$
- Classical Runtime (Numpy Solver) : $\sim 10^{-5}s$
- Experimental value error : $\pm$ 6% (w.r.t theoretical value)

Theoretical min parameter r :

$$r_{min} = -\log_2(C_{max}) \\ = -\log_2\left(min(\lambda_j)\right)$$



Simplified HHL QLS Algorithm

Analysis

Advantages:

- Possible to maximize the probability by finding the minimum parameter r.
- Any parameter r below the minimum is unattainable as the corresponding quantum gate does not exist.

Disadvantages:

- Needs to know all eigenvalues so that angles can be calculated for AR.
- Physical examples of eigenvalues $0.Xj$ are unknown, applications are limited.