The HHL Algorithm

Pradeep Kumar
CPSC 619 Project Presentation

Course Instructor: Dr. Peter Høyer

April 17, 2024

Table of Contents

1. Abstract
2. Introduction
3. Solving Linear Systems
4. Classical Algorithms for Solving Linear Systems
5. The HHL Algorithm
6. Complexity Analysis

Abstract

The HHL algorithm, proposed by Harrow, Hassidim, and Lloyd in 2009, addresses the fundamental challenge of efficiently solving linear systems of equations on quantum computers. Unlike classical methods with polynomial time complexity, HHL leverages quantum principles to achieve exponential speedup in solving sparse linear systems. We elucidate the key components of the HHL algorithm, including quantum state preparation, phase estimation, and qubit operations, explaining how they collectively enable the quantum solution of linear systems. Furthermore, we calculate the complexity of the algorithm for bounded error scenarios.

Introduction

- Classical algorithms like Gaussian elimination and Conjugate Gradient tackle the solution of linear systems.
- These classical methods have polynomial time complexity.
- For very large systems, classical algorithms may not suffice due to computational and time constraints.

Solving Linear Systems

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

$$\vec{x} = \mathbf{A}^{-1}\vec{b} = \mathbf{A}^{-1}\sum_{i=1}^{N}\beta_i\vec{u_i} \;\; = \sum_{i=1}^{N}\beta_i\mathbf{A}^{-1}\vec{u_i} = \sum_{i=1}^{N}\frac{\beta_i}{\lambda_i}\vec{u_i}$$

where $\vec{u_i}$ and λ_i are the eigenvectors and eigenvalues of the matrix $\mathbf{A}$ respectively.

Classical Algorithms

Gaussian Elimination

- Involves row operations to reduce a matrix to its echelon form.
- Further operations transform the matrix into diagonal form.

Conjugate Gradient Descent

- Efficient for symmetric, semi-positive, well-conditioned matrices.
- Faster than Gaussian elimination under suitable conditions.

Overview of the HHL Algorithm

Input

- The input to the HHL algorithm includes an $n \times n$ Hermitian matrix $\mathbf{A}$ and a vector $\vec{b}$, provided in classical form. The vector $\vec{b}$ is encoded as a quantum state, and the matrix $\mathbf{A}$ is represented as a Hamiltonian operator for processing in the quantum algorithm.

Output

- The output of the HHL algorithm is a quantum state $|x\rangle$ that represents the solution to the linear equation $A\vec{x} = \vec{b}$. It can used to estimate $\langle x|\mathbf{M}|x\rangle$, where $\mathbf{M}$ is some operator.

Procedure of HHL Algorithm

1. Encode $\vec{b}$ into a quantum state.
2. Perform Quantum Phase Estimation (QPE).
3. Apply controlled rotations based on the estimated eigenvalues.
4. Uncompute using inverse QPE.
5. Measure to obtain the solution state.

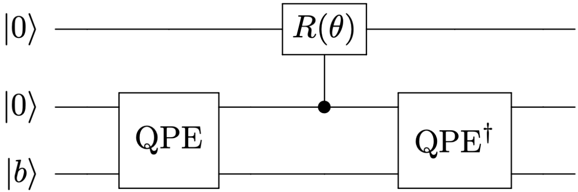

Figure: Circuit for HHL

Example of HHL Algorithm

Consider the linear system :

$$A = \begin{bmatrix} 1 & \frac{-1}{3} \\ \frac{-1}{3} & 1 \end{bmatrix}$$

and

$$b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The matrix A has eigenvectors and eigenvalues as follows:

$$\vec{u}_0 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \quad \vec{u}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

with eigenvalues $\lambda_0 = \frac{2}{3}$ and $\lambda_1 = \frac{4}{3}$ respectively.

Quantum State preparation

We prepare the state as :

$$|\psi_{initial}\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes m} \otimes |0\rangle$$

$$|\psi_{initial}\rangle \longrightarrow |0\rangle^{\otimes n} \otimes |b\rangle \otimes |0\rangle$$

For our example the initial state is :

$$|\psi_{initial}\rangle \longrightarrow |0\rangle^{\otimes n} \otimes |1\rangle \otimes |0\rangle$$

Quantum Phase Estimation

Consider operating $e^{i\mathbf{A}\tau}$ on an eigenvector of $\mathbf{A}$

$$e^{i\mathbf{A}\tau}|u_i\rangle = e^{i\lambda_i\tau}|u_i\rangle$$

Now we operate it on the state $|b\rangle$

$$|b\rangle = \sum_i \beta_i |u_i\rangle$$

$$e^{i\mathbf{A}\tau}|b\rangle = \sum_i \beta_i e^{i\mathbf{A}\tau}|u_i\rangle = \sum_i \beta_i e^{i\lambda_i\tau}|u_i\rangle$$

Quantum Phase Estimation

$$\left(\sum_{t=0}^{T-1} |t\rangle\langle t| \otimes e^{\frac{2\pi i \mathbf{A} \tau t}{T}}\right) \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} |t\rangle|u_j\rangle = \frac{1}{\sqrt{T}} \sum_{t=0}^{T-1} e^{\frac{2\pi i \lambda_j \tau t}{T}} |t\rangle|u_j\rangle$$

$$QFT^{-1}\left(\frac{1}{\sqrt{T}} \sum_{t=0}^{T-1} e^{\frac{2\pi i \lambda_j \tau t}{T}} |t\rangle|u_j\rangle\right) = \frac{1}{T} \sum_{k=0}^{T-1} \left(\sum_{t=0}^{T-1} e^{\frac{2\pi i (\lambda_j \tau - k) t}{T}}\right) |k\rangle|u_j\rangle$$

$$k = \lambda \tau$$

Quantum Phase Estimation

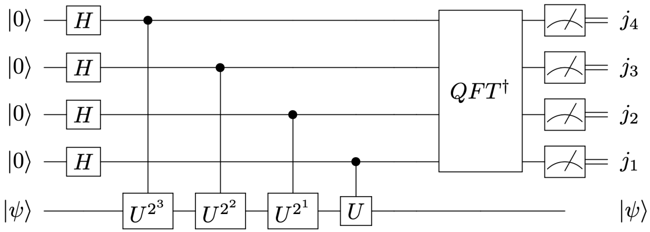

Figure: Circuit for QPE

After this step the state becomes :

$$|\psi_{QPE}\rangle = \sum_j \beta_j |\lambda_j\rangle |u_j\rangle \, |0\rangle$$

Quantum Phase Estimation

For our example :

$$|u_0\rangle = |+\rangle \quad |u_1\rangle = |-\rangle$$

$$|b\rangle = |1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$$

After QPE the state would become :

$$|\psi_{QPE}\rangle = \frac{1}{\sqrt{2}}(|01\rangle |+\rangle |0\rangle - \frac{1}{\sqrt{2}} |10\rangle |-\rangle |0\rangle)$$

We have encoded the eigenvalues in the 2-qubit register

Controlled Rotation

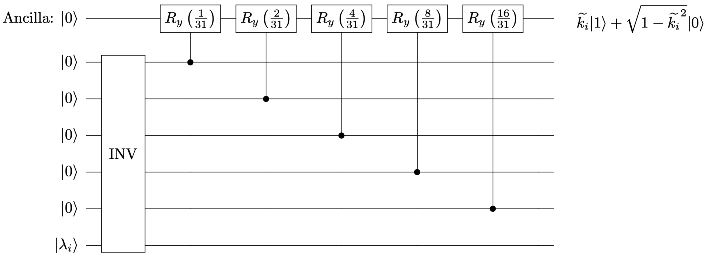

Figure: Circuit for Controlled Rotation

Controlled Rotation

Controlled rotations are performed on an ancillia qubit based on the eigenvalues encoded in the state. Each rotation angle is inversely proportional to the encoded eigenvalue $R_y(2\arcsin(C/\lambda_j))$ where,

$$R_y(2\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Therefore the state $|\psi_{QPE}\rangle$ transforms as:

$$\mathcal{R}_y\,|\psi_{QPE}\rangle = \sum_j \beta_j|\lambda_j\rangle|u_j\rangle\left(\frac{C}{\lambda_j}\,|1\rangle + \sqrt{1-\frac{C^2}{\lambda_j^2}}\,|0\rangle\right)$$

Controlled Rotation

In our example, our state was :

$$|\psi_{QPE}\rangle = \frac{1}{\sqrt{2}}(|01\rangle |+\rangle |0\rangle - \frac{1}{\sqrt{2}} |10\rangle |-\rangle |0\rangle)$$

We choose C $= 1$ then the angle of rotations for the two eigenvalues would be $\theta_1 = arcsin(1/1) = \frac{\pi}{2}$ and $\theta_2 = arcsin(1/2) = \frac{\pi}{6}$
The state now would become :

$$|\psi_{CR}\rangle = \frac{1}{\sqrt{2}}(|01\rangle |+\rangle (|1\rangle) - \frac{1}{\sqrt{2}} |10\rangle |-\rangle \left(\frac{1}{2} |1\rangle + \frac{\sqrt{3}}{2} |0\rangle\right)$$

Inverse QPE and Measurement

We perform inverse QPE and the state becomes :

$$\sum_j \beta_j |0\rangle^{\otimes n} |u_j\rangle \left(\frac{C}{\lambda_j} |1\rangle + \sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle \right)$$

and then we perform a measurement to get state $|1\rangle$

$$|0\rangle^{\otimes n} \otimes |x\rangle \otimes |1\rangle = |0\rangle^{\otimes n} \otimes \left(\sum_j \beta_j |u_j\rangle \frac{C}{\lambda_j} \right) \otimes |1\rangle$$

$$|x\rangle = \sum_j \frac{C}{\lambda_j} \beta_j |u_j\rangle$$

Inverse QPE and Measurement

For our example :

$$|\psi_{CR}\rangle = \frac{1}{\sqrt{2}}(|01\rangle |+\rangle (|1\rangle)) - \frac{1}{\sqrt{2}} |10\rangle |-\rangle \left(\frac{1}{2} |1\rangle + \frac{\sqrt{3}}{2} |0\rangle \right)$$

This state after inverse QPE and measurement becomes :

$$|\psi_{final}\rangle = |00\rangle \otimes \frac{1}{\sqrt{2}} |+\rangle - \frac{1}{2\sqrt{2}} |-\rangle \otimes |1\rangle$$

$$|\psi_{final}\rangle = |00\rangle \otimes \frac{1}{\sqrt{2}} \left(\frac{1}{2} |0\rangle + \frac{3}{2} |1\rangle \right) \otimes |1\rangle$$

$$|x\rangle = \frac{1}{\sqrt{10}} |0\rangle + \frac{3}{\sqrt{10}} |1\rangle$$

Solution

The solution to our example is

$$\vec{x} = \frac{3}{8}\begin{pmatrix}1\\3\end{pmatrix}$$

The solution we obtained through HHL algorithm is proportional to the actual solution $\vec{x}$

Complexity Analysis

- The computational complexity of the HHL algorithm depends primarily on the sparsity and condition number of the matrix A, as well as the desired precision ϵ of the solution.

Complexity Analysis

- **Hamiltonian Simulation:** The time complexity for simulating the evolution under A (as a Hamiltonian) is $O(\log(N)s^2\tau)$
- **Quantum Phase Estimation (QPE):** The precision of eigenvalue estimation is $\mathcal{O}(\frac{1}{\tau})$, therefore the relative error for $\frac{1}{\lambda}$ is $\frac{1}{\lambda\tau}$. If we take this error to be ϵ and also $\lambda \geq 1/\kappa$ then

$$\epsilon = \frac{1}{\lambda\tau} \text{ hence } \epsilon \leq \frac{\kappa}{\tau} \text{ which gives } \tau \leq \frac{\kappa}{\epsilon}$$

- **Controlled Rotations and Amplitude Amplification:** Amplitude amplification might be necessary to increase the probability of measuring the desired state, usually adding a multiplicative factor of $O(\kappa)$ to the complexity (this comes from $\kappa \approx \frac{1}{\sqrt{p}}$, where p is the probability of getting the state $|1\rangle$ upon measuring the ancillia qubit.)

Complexity Analysis

- Therefore, the overall complexity of HHL is $O(\log(N)\kappa^2 s^2/\epsilon)$.
- This makes HHL particularly advantageous for matrices that are sparse and well-conditioned, with large dimensions N where classical algorithms would be infeasible.

Comparison with classical algorithms

- Classical algorithms for solving linear systems, such as Gaussian elimination, have a polynomial time complexity, typically $O(N^3)$
- Methods like Conjugate Gradient have complexities that depend on the sparsity and condition number, generally requiring $O(Ns\kappa)$ operations.

Thanks