

The HHL Algorithm

Federico Matteoni

A.A. 2021/22

1 Introduction

The HHL algorithm was designed by Aram Harrow, Avinandan Hassidim and Seth Lloyd, and published in 2008.

Linear systems of equations are very commonplace throughout all the fields of science and branches of research. The efficient manipulation of matrices has become a requirement for many different algorithms and applications, ranging from fluid simulation to machine learning. In all these fields, one of the most common problems is finding a solution to a linear system of equations.

Most classical methods compute an exact solution in polynomial time: this is a different setup from other quantum algorithms, that improve over the exponential time required from their classical counterparts. Linear systems in real contexts, however, may contain matrices with millions of parameters, and even a polynomial algorithm can take too long to compute.

HHL is an algorithm that approximates a function of the solution vector of a linear system with logarithmic time complexity. This exponential speedup has significant implications regarding applications of machine learning algorithms on quantum computers.

2 Linear Systems

Given a matrix $A \in \mathbb{C}^{N \times N}$ and a vector $b \in \mathbb{C}^N$, the solution of the linear system is $x \in \mathbb{C}^N$ such that $Ax = b$. For example, with $N = 2$:

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

The problem can be written as finding $x_1, x_2 \in \mathbb{C}$ such that

$$\begin{cases} x_1 - \frac{x_2}{3} = 1 \\ -\frac{x_1}{3} + x_2 = 0 \end{cases}$$

The HHL algorithm requires some conditions on the components of the problem:

The matrix A must be

Hermitian, meaning $A = A^H \Leftrightarrow a_{ij} = \bar{a}_{ji}$

s -sparse, meaning at most s non-zero elements per row or column

with conditioning number \mathbf{k} , meaning $\mathbf{k} = \frac{\lambda_{\max}}{\lambda_{\min}}$ where λ_i are eigenvalues of A .

The vector b must be unitary, meaning $\|b\| = 1$

Given ϵ accuracy, solving this problem on a classical computer requires

$$O\left(\frac{N s \mathbf{k}}{\epsilon}\right)$$

while the HHL algorithm can compute **a function of the solution** in

$$O\left(\frac{\log(N) s^2 \mathbf{k}^2}{\epsilon}\right)$$

By using the HHL algorithm we obtain an exponential speedup in N , dimension of the system, but we have a very important difference: the classical algorithm returns the exact solution, while **HHL can only approximate a function of the solution**. The solution can still be obtained, but reading it requires a time linear in the number of elements of the solution, which cancels out the obtained speedup. So this algorithm is very useful when we're interested in a function $x^T M x$ of the solution x rather than in the solution itself.

3 Encoding the Problem in the Quantum World

We can assume b normalized, meaning $\|b\| = 1$, and we want to map it into $|b\rangle$. We will encode b in $|b\rangle$ using the **amplitude encoding**: each b_i will be the amplitude of the i th element of the basis of the quantum state. A is an Hermitian matrix. This means that it is a square matrix that's equal to its transposed conjugate, $A = A^H$. Hence, it has a spectral decomposition: given $\lambda_j \in \mathbb{R}$ eigenvalue of $|u_j\rangle$, the j th eigenvector, we have

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle \langle u_j| \Leftrightarrow A^{-1} = \sum_{j=0}^{N-1} \lambda_j^{-1} |u_j\rangle \langle u_j|$$

The eigenvectors of A form a basis, hence we can rewrite b in that basis

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

with $b_j \in \mathbb{C}$.

$|x\rangle$ can also be expressed in this form:

$$|x\rangle = A^{-1} |b\rangle = \sum_{i=0}^{2^{n_b}-1} \lambda_i^{-1} b_i |u_i\rangle$$

The goal of HHL is to find $|x\rangle$ in this form and store it in a quantum register.

It is worth highlighting how A will be used: the matrix will not be encoded in some register but will be encoded as the Hamiltonian of the QPE, meaning A will be encoded in $U = e^{iAt}$. Hamiltonian is a term that refers to the type of evolution of a system: given $U = e^{iAt}$, A is called Hamiltonian of U and describes its evolution over time t .

This encoding must guarantee the following normality conditions:

$$\sum_{j=0}^{2^{n_b}-1} |b_j|^2 = 1 \quad \sum_{i=0}^{2^{n_b}-1} |\lambda_i^{-1} b_i|^2 = 1$$

4 The Algorithm Step-by-Step

The register setup of the algorithm is as follows:

A register b of the most significative qubits, which will be used to encode the vector b .

A register c which will be used by the QPE algorithm to contain the clock qubits of the rotation. This is equivalent to containing the phases of the eigenvalues of the A matrix.

We use basis encoding: the phase is encoded in the basis vector element.

The number of qubits in this register, n , dictates the accuracy of the computation.

The least significative qubit is a register containing the ancillar qubit a . It will be discarded at the end of the computation.

The HHL algorithm follows 5 main steps: state preparation, QPE, ancillar qubit rotation, IQPE (Inverse QPE).

4.1 State Preparation

We have a total of $n_b + n + 1$ qubits which are initialized as

$$|\Psi_0\rangle = |0 \dots 0\rangle_b |0 \dots 0\rangle_c |0\rangle_a = |0\rangle^{\otimes n_b} |0\rangle^{\otimes n} |0\rangle$$

The first step is rotating the b register in the following way:

$$b = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n_b-1} \end{pmatrix} \Leftrightarrow |b\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle + \dots + \beta_{n_b-1} |n_b - 1\rangle$$

Obtaining

$$|\Psi_1\rangle = |b\rangle_b |0 \dots 0\rangle_c |0\rangle_a$$

So we prepared our initial state $|\Psi_1\rangle$ which encodes the given b vector.

4.2 Quantum Phase Estimation

The QPE algorithm is also used for estimating eigenvalues. It has three components: superposition of the clock qubits using Hadamard gates, controlled rotation, IQFT (Inverse Quantum Fourier Transform). The goal is estimating the eigenvalues of the unitary rotation matrix $U = e^{iAt}$ used by QPE. A is encoded in the Hamiltonian of the QPE, and has unitary eigenvalues in the form $e^{i\theta}$. Hence, the phase of the eigenvalue of the gate is proportional to the eigenvalue of A : as a result, after the QPE we will have the eigenvalues of A encoded through basis encoding in the c register.

4.2.1 Superposition of Clock Qubits

As first step of the QPE, we apply Hadamard gates to the clock qubits in order to create a superposition

$$|\Psi_2\rangle = (I^{\otimes n_b} \otimes H^{\otimes n} \otimes I) |\Psi_1\rangle = |b\rangle \frac{1}{\sqrt{2^n}} (|0\rangle + |1\rangle)^{\otimes n} |0\rangle$$

4.2.2 Controlled Rotation

The next step is the controlled rotation, using the clock qubits as control and $|b\rangle$ as target. The number n of qubits in the register c will control the number of controlled gates, which are in the form U^r with r index of the clock qubit. So $|c_j\rangle$ will control the gate $U^{2^j} = e^{iAt2^j}$.

To understand how this works, let's consider $|b\rangle$ as a eigenvector of U of eigenvalue $e^{2\pi i\phi}$, meaning $U|b\rangle = e^{2\pi i\phi}|b\rangle$. This way of writing the eigenvalue will encode the phase ϕ through basis encoding in the register c .

If the control bit is $|0\rangle$, then $|b\rangle$ will be unaffected.

If the control bit is $|1\rangle$, then U is applied to $|b\rangle$ which is equivalent to multiplying the $|1\rangle$ qubit of $|c_j\rangle$ times $e^{2\pi i\phi 2^j}$

Hence, after applying the U gates we end up with

$$|\Psi_3\rangle = |b\rangle \otimes \left(\frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i\phi 2^{n-1}} |1\rangle) \otimes \dots \otimes (|0\rangle + e^{2\pi i\phi 2^{n-n}} |1\rangle) \right) \otimes |0\rangle_a = |b\rangle \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i\phi k} |k\rangle |0\rangle_a$$

4.2.3 IQFT

The inverse of Quantum Fourier Transform applies a matrix U_I defined as the following

$$U_I = \frac{1}{\sqrt{N}} \begin{pmatrix} \omega_N^{-0 \cdot 0} & \dots & \omega_N^{-0 \cdot (N-1)} \\ \vdots & \ddots & \vdots \\ \omega_N^{-(N-1) \cdot 0} & \dots & \omega_N^{-(N-1) \cdot (N-1)} \end{pmatrix}$$

Where ω_N is the N th root of the unity

$$\omega_N = e^{\frac{i2\pi}{N}}$$

So by applying IQFT to a basis vector, we get in a superposition of all the basis vectors weighted by the powers of the N th root of the unity

$$U_I |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{-jk} |j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{-i2\pi jk}{N}} |j\rangle$$

In HHL only the clock qubits are modified

$$\begin{aligned} |\Psi_4\rangle &= |b\rangle IQFT \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i\phi k} |k\rangle \right) |0\rangle_a = |b\rangle \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i\phi k} IQFT(|k\rangle) |0\rangle_a = \\ &= |b\rangle \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i\phi k} \left(\sum_{y=0}^{2^n-1} e^{-\frac{2\pi i y k}{N}} |y\rangle \right) |0\rangle_a = |b\rangle \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i k (\phi - \frac{y}{N})} |y\rangle |0\rangle_a \end{aligned}$$

Only the $|y\rangle$ for which $\phi - \frac{y}{N} = 0 \Leftrightarrow y = N\phi$ will have a non-zero amplitude of $\sum_{k=0}^{2^n-1} e^0 = 2^n$, otherwise the amplitude will be $\sum_{k=0}^{2^n-1} e^{2\pi i k (\phi - \frac{y}{N})} = 0$ due to destructive interference. So, by ignoring the zero amplitude states we will have

$$|\Psi_4\rangle = \frac{1}{2^n} |b\rangle \sum_{k=0}^{2^n-1} e^{2\pi i k \cdot 0} |N\phi\rangle |0\rangle_a = |b\rangle |N\phi\rangle |0\rangle_a$$

So we use the clock qubits to encode information about ϕ , which is the phase of U .

By using the fact that the accuracy depends on the number of qubits n , that A is the Hamiltonian of U meaning $U = e^{iAt}$, and U is diagonal in the basis formed by the eigenvectors $|u_j\rangle$ of A , we can infer that if $|b\rangle = |u_j\rangle$ then $U|b\rangle = e^{i\lambda_j t} |u_j\rangle$. With $i\lambda_j t = 2\pi i\phi$ we obtain $\phi = \frac{\lambda_j t}{2\pi}$, so

$$|\Psi_4\rangle = |u_j\rangle \left| N \frac{\lambda_j t}{2\pi} \right\rangle |0\rangle_a$$

This way the eigenvalues of A are encoded through basis encoding in the clock qubits.

In the general case $|b\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle$, and by using the superposition

$$|\Psi_4\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle \left| N \frac{\lambda_j t}{2\pi} \right\rangle |0\rangle_a$$

We also have non-integer λ_j in the general case. Hence we choose t such that $\overline{\lambda_j} = N \frac{\lambda_j t}{2\pi}$ is integer.

$$|\Psi_4\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\overline{\lambda_j}\rangle |0\rangle_a$$

4.3 Rotation and Measurement of the Ancillar Qubit

The next step is rotating $|0\rangle_a$. The goal is obtaining

$$|\Psi_5\rangle = \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\overline{\lambda_j}\rangle \left(\sqrt{1 - \frac{C^2}{\overline{\lambda_j}^2}} |0\rangle_a + \frac{C}{\overline{\lambda_j}} |1\rangle_a \right)$$

When the ancillar qubit is measured, its wavefunction will collapse either to $|0\rangle$ or to $|1\rangle$. When $|0\rangle$ is measured, the result is discarded and the computation is repeated until we measure $|1\rangle$. As such, the final interesting wave function is

$$|\Psi_6\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\overline{\lambda_j}} \right|^2}} \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle |\overline{\lambda_j}\rangle \frac{C}{\overline{\lambda_j}} |1\rangle_a$$

The factor at the beginning is needed for normalization purposes. Since $\left| \frac{b_j C}{\overline{\lambda_j}} \right|^2$ is the probability of measuring $|1\rangle$ on the ancillar qubit, C must be chosen as big as possible. The result looks like $|x\rangle$, but the correct result would be measured only using the eigenvector basis formed by $|u_j\rangle$ instead of the computational basis formed by $|0\rangle, |1\rangle$. We cannot factorize the result in a tensor product between registers b and c , because the b register is in entanglement with the clock qubits $|\overline{\lambda_j}\rangle$, hence we cannot convert the b register in the computation basis: we have to de-compute the state and disentangle the b and c registers.

Note that this measurement of the ancillar qubit can be, and usually is, done at the end of the computation. But since that this qubit is no longer modified, measuring now gives the same result.

4.4 Inverse Quantum Phase Estimation

The decomputation with the goal of disentangling the registers b and c is achieved via IQPE. Applying the inverse of the QPE is done by applying the three steps stated before, in reverse order.

4.4.1 QFT

The Quantum Fourier Transform in the QPE is applied in inverse order, so for the IQPE we apply the normal QFT. QFT applies a matrix U_F defined as

$$U_I = \frac{1}{\sqrt{N}} \begin{pmatrix} \omega_N^{0 \cdot 0} & \dots & \omega_N^{0 \cdot (N-1)} \\ \vdots & \ddots & \vdots \\ \omega_N^{(N-1) \cdot 0} & \dots & \omega_N^{(N-1) \cdot (N-1)} \end{pmatrix} \quad \omega_N = e^{\frac{i2\pi}{N}}$$

So we apply QFT on the clock qubits

$$\begin{aligned} |\Psi_7\rangle &= \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle \text{QFT}(|\bar{\lambda}_j\rangle) \frac{C}{\lambda_j} |1\rangle_a = \\ &= \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} b_j |u_j\rangle \left(\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i y \bar{\lambda}_j}{N}} |y\rangle \right) \frac{C}{\lambda_j} |1\rangle_a \end{aligned}$$

4.4.2 Controlled Rotations

As before, using $U^{-1} = e^{-iAt}$ applied if the r th clock qubit is $|1\rangle$. This is equivalent to multiplying by $e^{-i\lambda_j t y}$ if the c register is $|y\rangle$, since $2\pi i \phi = i\lambda_j t$

$$|\Psi_8\rangle = \frac{1}{\sqrt{2^n} \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle \left(\sum_{y=0}^{2^n-1} e^{-i\lambda_j t y} e^{\frac{2\pi i y \bar{\lambda}_j}{N}} |y\rangle \right) |1\rangle_a =$$

Since $\bar{\lambda}_k = \frac{N\lambda_j t}{2\pi}$, the two exponential terms cancel out

$$\begin{aligned} &= \frac{1}{\sqrt{2^n} \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle \sum_{y=0}^{2^n-1} |y\rangle |1\rangle_a = \\ &= \frac{1}{\sqrt{2^n} \left(\frac{Nt}{2\pi} \right) \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle \sum_{y=0}^{2^n-1} |y\rangle |1\rangle_a = \\ &= \frac{C}{\sqrt{2^n} \sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} |x\rangle \sum_{y=0}^{2^n-1} |y\rangle |1\rangle_a \end{aligned}$$

The clock qubits in the register c and the register b are now disentangled, and the register b now contains $|x\rangle$.

4.4.3 Applying the Hadamard Gates

We apply the Hadamard gates to the clock qubits

$$|\Psi_9\rangle = \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} \sum_{j=0}^{2^{n_b}-1} \frac{b_j C}{\lambda_j} |u_j\rangle |0\rangle^{\otimes n} |1\rangle_a = \frac{C}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j C}{\lambda_j} \right|^2}} |x\rangle_b |0\rangle^{\otimes n} |1\rangle_a =$$

If $C \in \mathbb{R}$ then given that $\sum_{i=0}^{2^{n_b}-1} |\lambda_i^{-1} b_i|^2 = 1$ we have

$$= \frac{1}{\sqrt{\sum_{j=0}^{2^{n_b}-1} \left| \frac{b_j}{\lambda_j} \right|^2}} |x\rangle_b |0\rangle^{\otimes n} |1\rangle_a = |x\rangle_b |0\rangle_c^{\otimes n} |1\rangle_a$$

With the register b correctly storing the solution $|x\rangle$.

References

- [1] Solving linear systems of equations using hhl. https://qiskit.org/textbook/ch-applications/hhl_tutorial.html.
- [2] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), oct 2009.
- [3] H. J. Morrell, A. Zaman, and H. Y. Wong. Step-by-step hhl algorithm walkthrough to enhance the understanding of critical quantum computing concepts, 2021.